



**TBS1000C Series  
Digital Oscilloscopes  
Programmer Manual**







**TBS1000C Series  
Digital Oscilloscopes  
Programmer Manual**

Copyright © Tektronix. All rights reserved. Licensed software products are owned by Tektronix or its subsidiaries or suppliers, and are protected by national copyright laws and international treaty provisions. Tektronix products are covered by U.S. and foreign patents, issued and pending. Information in this publication supersedes that in all previously published material. Specifications and price change privileges reserved.

TEKTRONIX and TEK are registered trademarks of Tektronix, Inc.

### **Contacting Tektronix**

Tektronix, Inc.  
14150 SW Karl Braun Drive  
P.O. Box 500  
Beaverton, OR 97077  
USA

For product information, sales, service, and technical support:

- In North America, call 1-800-833-9200.
- Worldwide, visit [www.tek.com](http://www.tek.com) to find contacts in your area.

---

# Table of Contents

TBS1000C Series Programmer Manual .....	0
---	---

## Getting Started

## Command Syntax

Command Syntax .....	3
Command and Query Structure .....	4
Commands .....	5
Queries .....	5
Headers in Query Responses .....	6
Clearing the Output Queue .....	6
Command Entry .....	6
Abbreviating Commands .....	7
Concatenating Commands .....	7
Message Terminators .....	8
Constructed Mnemonics .....	9
Reference Waveform Mnemonics .....	9
Waveform Mnemonics .....	9
Cursor Position Mnemonic .....	9
Measurement Specifier Mnemonics .....	10
Argument Types .....	10
Numeric Arguments .....	10
Quoted String Arguments .....	11
Block Arguments .....	11

## Command groups

Alias command group .....	13
Acquisition command group .....	14
Calibration and Diagnostic command group .....	14
Cursor command group .....	15
Display command group .....	16
FFT command group .....	17
File system command group .....	17
Help everywhere command group .....	18
Horizontal command group .....	19

Math command group .....	20
Measurement command group .....	21
Miscellaneous command group .....	23
Save and Recall command group .....	25
Status and Error command group .....	26
Trigger command group .....	27
Vertical command group .....	28
Waveform command group .....	30
Zoom command group .....	35

## A commands

ACQuire? .....	37
ACQuire:MAXSamplerate? .....	38
ACQuire:MODE .....	38
ACQuire:NUMAcq? .....	40
ACQuire:NUMAVg .....	41
ACQuire:STATE .....	41
ACQuire:STOPAfter .....	42
ALias .....	43
ALias:CATalog? .....	44
ALias:DEFine .....	45
ALias:DELEte .....	46
ALias:DELEte:ALL .....	46
ALias:DELEte[:NAME] .....	47
ALias[:STATE] .....	47
ALLEv? .....	48
AUTOSet .....	49
AUTOSet:ENABLE .....	49

## B commands

BUSY? .....	51
-------------	----

## C commands

*CAL? .....	53
CALibrate:INTERNAL .....	54
CALibrate:INTERNAL:STARt .....	54
CALibrate:INTERNAL:STATus? .....	55
CALibrate:RESults? .....	56

CALibrate:REsults:SPC?	56
CH<x>?	57
CH<x>:AMPSVIAVOLTs:ENABLe	58
CH<x>:AMPSVIAVOLTs :Factor	59
CH<x>:BANdwidth	59
CH<x>:COUPling	60
CH<x>:DESKew	61
CH<x>:INVert	62
CH<x>:LABel	62
CH<x>:OFFSet	63
CH<x>:POSition	64
CH<x>:PRObe	65
CH<x>:PRObe:GAIN	66
CH<x>:PRObe:ID?	67
CH<x>:PRObe:ID:SERnumber?	67
CH<x>:PRObe:ID:TYPE?	68
CH<x>:PRObe:SIGnal	68
CH<x>:PRObe:UNIts?	69
CH<x>:SCALE	69
CH<x>:VOLts	70
CH<x>:YUNit	71
CLEARMenu	72
*CLS	72
CURSor?	73
CURSor:ENABLe	74
CURSor:FUNCTion	75
CURSor:HBArs?	76
CURSor:HBArs:DELTA?	76
CURSor:HBArs:POSITION<x>	77
CURSor:HBArs:UNIts	78
CURSor:HBArs:USE	78
CURSor:MODE	79
CURSor:VBArS?	80
CURSor:VBArS:ALTERNATE<x>?	80
CURSor:VBArS:DELTA?	81
CURSor:VBArS:HPOS<x>?	82
CURSor:VBArS:POSITION<x>	82
CURSor:VBArS:UNIts	83
CURSor:VBArS:VDELTA?	84

CURVe .....	85
-------------	----

## D commands

DATA .....	87
DATA:DESTination .....	88
DATA:SOUrce .....	89
DATA:STARt .....	90
DATA:STOP .....	91
DATA:WIDth .....	92
DATE .....	93
DESE .....	94
DIAG:FAN .....	95
DIAG:LOOP:OPTion .....	95
DIAG:LOOP:OPTion:NTIMes .....	96
DIAG:LOOP:STOP .....	96
DIAG:RESUlt:FLAg? .....	97
DIAG:RESUlt:LOG? .....	97
DIAG:SElect .....	98
DIAG:SElect:<function> .....	99
DIAG:STATE .....	99
DIAG:TEMPVAL .....	100
DISplay:GRAticule .....	100
DISplay:INTENSITy:BACKLight .....	101
DISplay:PERSistence:STATe .....	102
DISplay:PERSistence:VALUe .....	102

## E commands

ERRLOG:FIRST? .....	105
ERRLOG:NEXT? .....	105
*ESE .....	106
*ESR? .....	107
EVENT? .....	108
EVMsg? .....	108
EVQty? .....	109

## F commands

FACTory .....	111
FEAEN:PASSWORD .....	112



FFT?	113
FFT:HORizontal:POSition	113
FFT:HORizontal:SCAle	114
FFT:SOURce	114
FFT:SRCWFM	115
FFT:VERTical:POSition	116
FFT:VERTical:SCAle	116
FFT:VERTical:UNIts	117
FFT:VType	117
FFT:WINdow	118
FILESystem?	118
FILESystem:CWD	119
FILESystem:DELEte	120
FILESystem:DIR?	121
FILESystem:FORMat	122
FILESystem:FREESpace?	122
FILESystem:MKDir	123
FILESystem:READFile	124
FILESystem:REName	124
FILESystem:RMDir	125
FILESystem:WRITEFile	126
FPAnel:PRESS	127
FPAnel:TURN	128
FWUpdate:Update	129

## H commands

HDR	131
HEADer	131
HELPevery:ACQuire	132
HELPevery:ALL	133
HELPevery:CURsor	133
HELPevery:FFT	134
HELPevery:MATH	134
HELPevery:MEASUrement	135
HELPevery:REFeRence	136
HELPevery:TRIGger	136
HELPevery:UTIlity	137
HELPevery:VERTical	138
HORizontal?	138

HORizontal:ACQLENGTH .....	139
HORizontal:DIVisions .....	140
HORizontal[:MAIn][:DELay]:POSition .....	140
HORizontal:MAIn:DELay:MODe .....	141
HORizontal:MAIn:DELay:STATe .....	142
HORizontal[:MAIn]:DELay:TIME .....	143
HORizontal[:MAIn]:SAMPLERate .....	144
HORizontal[:MAIn]:SCAlE .....	144
HORizontal[:MAIn]:SECdiv .....	145
HORizontal:MAIn:UNIts[:STRing] .....	146
HORizontal:PREViewstate .....	146
HORizontal:RECOrdlength .....	147
HORizontal:RECOrdlength:Auto .....	147
HORizontal:RESOLution .....	148
HORizontal:ROLL .....	149
HORizontal:TRIGger:POSition .....	149

## I commands

ID? .....	151
*IDN? .....	152

## L commands

LANGUage .....	153
LOCK .....	154
*LRN? .....	154

## M commands

MATH? .....	155
MATH:DEFINE .....	156
MATH:HORizontal:POSition .....	157
MATH:HORizontal:SCAlE .....	157
MATH:HORizontal:UNIts .....	158
MATH:LABel .....	159
MATH:VERTical:POSition .....	159
MATH:VERTical:SCAlE .....	160
MATH:VERTical:UNIts .....	161
MEASUrement? .....	161
MEASUrement:CLEARSNapshot .....	163

MEASUrement:ENABLE .....	163
MEASUrement:GATing .....	164
MEASUrement:IMMed? .....	165
MEASUrement:IMMed:DElay? .....	165
MEASUrement:IMMed:DElay:EDGE<x> .....	166
MEASUrement:IMMed:SOUrce1 .....	167
MEASUrement:IMMed:SOUrce2 .....	168
MEASUrement:IMMed:TYPe .....	169
MEASUrement:IMMed:UNIts? .....	171
MEASUrement:IMMed:VALue? .....	172
MEASUrement:MEAS<x>? .....	173
MEASUrement:MEAS<x>:DElay? .....	173
MEASUrement:MEAS<x>:DElay:EDGE<x> .....	174
MEASUrement:MEAS<x>:SOUrce1 .....	174
MEASUrement:MEAS<x>:SOUrce2 .....	175
MEASUrement:MEAS<x>:STATE .....	176
MEASUrement:MEAS<x>:TYPe .....	177
MEASUrement:MEAS<x>:UNIts? .....	180
MEASUrement:MEAS<x>:VALue? .....	181
MEASUrement:REFLevel? .....	182
MEASUrement:REFLevel:ABSolute:LOW .....	182
MEASUrement:REFLevel:ABSolute:MID1 .....	183
MEASUrement:REFLevel:ABSolute:MID2 .....	184
MEASUrement:REFLevel:METHod .....	185
MEASUrement:REFLevel:PERCent:HIGH .....	186
MEASUrement:REFLevel:PERCent:LOW .....	187
MEASUrement:REFLevel:PERCent:MID1 .....	188
MEASUrement:REFLevel:PERCent:MID2 .....	189
MEASUrement:SNAPSHOT .....	190
MEASUrement:SOURCESNAPShot .....	190

## O commands

*OPC .....	191
------------	-----

## P commands

*PSC .....	193
------------	-----

## R commands

*RCL .....	195
RECALL:SETUp .....	196
RECALL:WAVEForm .....	197
REF<x>? .....	198
REF<x>:DATE? .....	198
REF<x>:TIME? .....	199
REF<x>:HORizontal:DELay:TIME? .....	199
REF<x>:HORizontal:SCAle? .....	200
REF<x>:POSition? .....	200
REF<x>:VERTical:POSition? .....	201
REF<x>:VERTical:SCAle? .....	201
*RST .....	202

## S commands

*SAV .....	203
SAVe:ASSIgn:TYPe .....	204
SAVe:IMAge .....	204
SAVe:IMAge:FILEFormat .....	205
SAVe:IMAge:LAYout .....	206
SAVe:SETUp .....	207
SAVe:WAVEform .....	208
SAVe:WAVEform:FILEFormat .....	209
SElect:CH<x> .....	210
SElect:CONTRol .....	211
SElect:FFT .....	212
SElect:MATH .....	213
SElect:REF<x> .....	214
SET? .....	215
SETUP<x>:DATE? .....	216
SETUP<x>:TIME? (Query Only) .....	216
*SRE .....	217
*STB? .....	218

## T commands

TEKSecure .....	219
TIME .....	219
TRIGger .....	220
TRIGger:A .....	221

TRIGger:A:EDGE?	222
TRIGger:A:EDGE:COUPling	222
TRIGger:A:EDGE:SLOpe	223
TRIGger:A:EDGE:SOUrce	224
TRIGger:A:HOLDOff?	225
TRIGger:A:HOLDOff:TIME	225
TRIGger:A:LEVel	226
TRIGger:A:LEVel:CH<x>	227
TRIGger:A:LOWerthreshold:CH<x>	227
TRIGger:A:MODE	228
TRIGger:A:PULse?	229
TRIGger:A:PULse:CLAss	230
TRIGger:A:PULSE:Width?	231
TRIGger:A:PULse:WIDTH:POLarity	231
TRIGger:A:PULSEWidth:SOUrce	232
TRIGger:A:PULse:WIDTH:WHEN	232
TRIGger:A:PULse:WIDTH:WIDTH	234
TRIGger:A:RUNT?	235
TRIGger:A:RUNT:POLarity	235
TRIGger:A:RUNT:SOUrce	236
TRIGger:A:RUNT:WHEN	237
TRIGger:A:RUNT:WIDTH	238
TRIGger:A:TYPE	238
TRIGger:A:UPPerthreshold:CH<x>	239
TRIGger:FREQuency?	240
TRIGger:STATE?	240

## U commands

UNLock	243
--------	-----

## V commands

VERBose	245
---------	-----

## W commands

*WAI	247
WAVFrm?	248
WFMInpre?	248
WFMInpre:BIT_Nr	249

WFMInpre:BYT_Nr .....	250
WFMInpre:ENCdg .....	250
WFMInpre:NR_Pt? .....	251
WFMInpre:XINcr .....	252
WFMInpre:XUNit .....	252
WFMInpre:XZero .....	253
WFMInpre:YMUlt .....	254
WFMInpre:YOff .....	255
WFMInpre:YUNit .....	256
WFMInpre:YZero .....	257
WFMOutpre? .....	258
WFMOutpre:BIT_Nr .....	258
WFMOutpre:BN_Fmt .....	259
WFMOutpre:BYT_Nr .....	260
WFMOutpre:ENCdg .....	261
WFMOutpre:NR_Pt? .....	262
WFMOutpre:RECOrdlength? .....	262
WFMOutpre:WFId? .....	263
WFMOutpre:XINcr? .....	264
WFMOutpre:XUNit? .....	264
WFMOutpre:XZero? .....	265
WFMOutpre:YMUlt? .....	266
WFMOutpre:YOff? .....	266
WFMOutpre:YUNit? .....	267
WFMOutpre:YZero? .....	268

## Z commands

ZOOM? .....	269
ZOoM{:MODe :STATE} .....	269
ZOoM:ZOOM1? .....	270
ZOOM:ZOOM1:FACtor .....	271
ZOOM:ZOOM1:HORizontal:POSition .....	271
ZOOM:ZOOM1:HORizontal:SCAle .....	272
ZOoM:ZOOM1:POSition .....	273
ZOoM:ZOOM1:SCAle .....	273
ZOOM:ZOOM1:STATE .....	274

## Status and Events

Registers .....	275
Overview .....	275
Status Registers .....	275
Enable Registers .....	278
*PSC Command .....	279
Queues .....	279
Output Queue .....	279
Event Queue .....	280
Event Handling Sequence .....	280
Synchronization Methods .....	282
Overview .....	282
Using the *WAI Command .....	284
Using the BUSY Query .....	284
Using the *OPC Command .....	285
Using the *OPC? Query .....	287
Messages .....	288
No Event .....	288
Command Error .....	288
Execution Error .....	289
Device Error .....	292
System Event .....	292
Execution Warning .....	293
Internal Warning .....	293

## Programming Examples

## ASCII Code Chart

## Factory setup

TBS1000C Series Oscilloscopes .....	299
-------------------------------------	-----

## Reserved words

## Glossary

Glossary terms .....	303
----------------------	-----





---

# Getting Started

This manual contains information on how to remotely control and operate your instrument through communications protocol and commands.

Refer to the instrument user manual for information on how to configure and test your instrument remote connectivity (USB or Ethernet).

Download the latest version of the programmer manual from [www.tek.com/downloads](http://www.tek.com/downloads) for up-to-date command syntax information.



---

# Command Syntax

You can control the instrument through the Ethernet or USB interface using a large group of commands and queries.

This section describes the syntax these commands and queries use and the conventions the instrument uses to process them. The commands and queries themselves are listed in the *Command Descriptions* section.

## Command Syntax

**Table 1: Instrument communication protocol**

Model or option	GPIO	RS-232	USB
TBS1000C	Yes	No	Yes

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

The Backus Naur Form (BNF) notation is used in this manual to describe commands and queries.

**Table 2: BNF notation**

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

## Command and Query Structure

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

Most commands have both a set form and a query form. The query form of the command is the same as the set form except that it ends with a question mark. For example, the set command ACQUIRE:MODE has a query form ACQUIRE:MODE?. Not all commands have both a set and a query form; some commands are set only and some are query only.

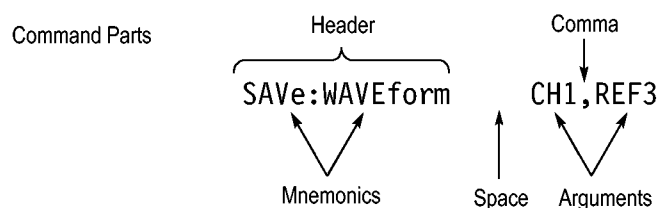
A few commands do both a set and query action. For example, the \*CAL? command runs a self-calibration program on the instrument, then returns the result of the calibration.

A command message is a command or query name, followed by any information the instrument needs to execute the command or query. Command messages consist of five different element types.

**Table 3: Command message elements**

Symbol	Meaning
<Header>	The basic command name. If the header ends with a question mark, the command is a query. The header may begin with a colon (:) character; if the command is concatenated with other commands the beginning colon is required. The beginning colon can never be used with command headers beginning with a star (*).
<Mnemonic>	A header subfunction. Some command headers have only one mnemonic. If a command header has multiple mnemonics, they are always separated from each other by a colon (:) character.
<Argument>	A quantity, quality, restriction, or limit associated with the header. Not all commands have an argument, while other commands have multiple arguments. Arguments are separated from the header by a <Space>. Arguments are separated from each other by a <Comma>.
<Comma>	A single comma between arguments of multiple-argument commands. It may optionally have white space characters before and after the comma.
<Space>	A white space character between command header and argument. It may optionally consist of multiple white space characters.

The following figure shows the five command message elements.



**Figure 1: Command message elements**

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

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

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

**Queries** Queries cause the instrument to return information about its status or settings. Queries have the structure:

[:]<Header>

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

You can specify a query command at any level within the command tree unless otherwise noted. These branch queries return information about all the mnemonics below the specified branch or level.

For example, MEASUrement:MEAS<x>:UNIts? returns the measurement units, while MEASUrement:MEAS<x>:TYPe? returns the measurement type selected for the measurement, and MEASUrement:MEAS<x>? returns all the measurement parameters for the specified measurement.

## Headers in Query Responses

You can control whether the instrument returns headers as part of the query response. Use the **HEADER** command to control this feature. If header is on, the instrument returns command headers as part of the query and formats the query response as a valid set command. When header is off, the instrument sends back only the values in the response. This format can make it easier to parse and extract the information from the response.

**Table 4: Comparison of Header Off and Header On responses**

Query	Header Off response	Header On response
ACQuire:NUMAVg	64	ACQUIRE:NUMAVG 64
CHx1:COUPling	DC	CH1:COUPLING DC

## Clearing the Output Queue

To clear the output queue and reset the instrument to accept a new command or query, send a Device Clear (DCL) from a GPIB host.

From an RS-232 host, send a break signal. The RS-232 interface responds by returning the ASCII string "DCL."

From a USB host, send an **INITIATE\_CLEAR** followed by a **CHECK\_CLEAR\_STATUS**. The USB interface responds to **CHECK\_CLEAR\_STATUS** with **STATUS\_SUCCESS** when it is finished clearing the output queue.

## Command Entry

Follow these general rules when entering commands:

- Enter commands in upper or lower case.
- You can precede any command with white space characters. White space characters include any combination of the ASCII control characters 00 through 09 and 0B through 20 hexadecimal (0 through 9 and 11 through 32 decimal).
- The instrument ignores commands that consists of just a combination of white space characters and line feeds.

**Abbreviating Commands**

You can abbreviate many instrument commands. These abbreviations are shown in capital letters in the command listing in the *Command Groups* section and *Command Descriptions* section. For example, the command ACQUIRE:NUMAVg can be entered simply as ACQ:NUMA or acq:numa.

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

**Concatenating Commands**

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

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

```
TRIGGER:MODE NORMAL;:ACQUIRE:NUMAVg 16
```

- If concatenated commands have headers that differ by only the last mnemonic, you can abbreviate the second command and eliminate the beginning colon. For example, the commands ACQUIRE:MODE AVERAGE and ACQUIRE:NUMAVg 16 could be concatenated into a single command:

```
ACQUIRE:MODE AVERAGE; NUMAVg 16
```

The longer version works equally well:

```
ACQUIRE:MODE AVERAGE;:ACQUIRE:NUMAVg 16
```

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

```
ACQUIRE:MODE AVERAGE;*TRG
```

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

```
ACQUIRE:MODE AVERAGE;*TRG;NUMAVg 16
```

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

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

```
CH1:COUPLING;BANDWIDTH
```

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

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

```
ACQUIRE:MODE AVERAGE;NUMAVg;STATE
```

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

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

Here are some INVALID concatenation examples:

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

### Message Terminators

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

**GPIB End of Message (EOM) Terminators.** GPIB EOM terminators can be the END message (EOI asserted concurrently with the last data byte), the ASCII code for line feed (LF) sent as the last data byte, or both. The instrument always terminates messages with LF and EOI. White space is allowed before the terminator; for example, CR LF is acceptable.

**USB End of Message (EOM) Terminators.** The EOM bit must be set in the USB header of the last transfer of a command message

See the USB Test and Measurement Class Specification (USBTMC) section 3.2.1 for details. The instrument terminates messages by setting the EOM bit in the USB header of the last transfer of a message to the host (USBTMC Specification section 3.3.1), and by terminating messages with a LF. White space is allowed before the terminator; for example, CR LF is acceptable.



## Constructed Mnemonics

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

### Channel mnemonics

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

Symbol	Meaning
CH<x>	A channel specifier; <x> is 1 or 2.

### Reference Waveform Mnemonics

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

Symbol	Meaning
REF<x>	A reference waveform specifier; <x> is 1 or 2.

### Waveform Mnemonics

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

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

### Cursor Position Mnemonic

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

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

## Measurement Specifier Mnemonics

Commands can specify which measurement to set or query as a mnemonic in the header. The instrument can display up to six automated measurements.

Symbol	Meaning
MEAS<x>	A measurement specifier; <x> is 1-6.

## Argument Types

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

### Numeric Arguments

Many instrument commands require numeric arguments.

**Table 5: Types of numeric arguments**

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

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

When you enter an incorrect numeric argument, the instrument automatically forces the numeric argument to a correct value.

**Table 6: Instrument handling of incorrect numeric arguments**

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

## Quoted String Arguments

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

"this is a quoted string"

Symbol	Meaning
<QString>	Quoted string of ASCII text

Follow these rules when you use quoted strings:

1. A quoted string can include any character defined in the 7-bit ASCII character set. [ASCII Code Chart](#) on page 297.
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 if you follow the previous rule:  
"this is an 'acceptable' string"
4. You can include a quote character within a string simply by repeating the quote. For example,  
"here is a "" mark"
5. Strings can have upper or lower case characters.
6. If you use a GPIB network, you cannot terminate a quoted string with the END message before the closing delimiter.
7. A carriage return or line feed embedded in a quoted string does not terminate the string, but is treated as just another character in the string.
8. The maximum length of a quoted string returned from a query is 1000 characters.

Here are some examples of invalid strings:

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

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

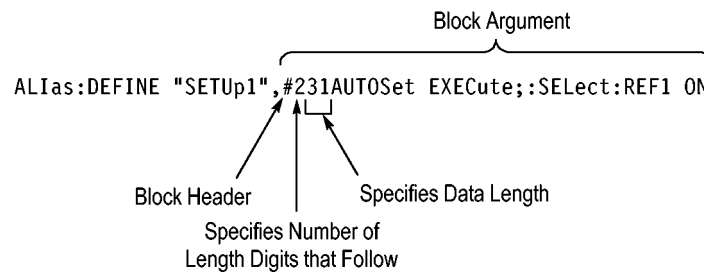
## Block Arguments

Several instrument commands use a block argument form.

**Table 7: Parts of a block argument**

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

The following figure shows an example of a block argument.

**Figure 2: Block argument example**

<NZDig> specifies the number of <Dig> elements that follow. Taken together, the <Dig> elements form a decimal integer that specifies how many <DChar> elements follow.

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

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

# Command groups

This section lists the commands organized by functional group. The following sections lists all commands alphabetically.

The instrument GPIB and USB interfaces conform to Tektronix standard codes and formats except where noted. The GPIB interface also conforms to IEEE Std 488.2–1987 except where noted. The USB interface also conforms to USB Test and Measurement Class, Subclass USB488 Specification, except where noted.

## Alias command group

Use the Alias commands to define new commands as a sequence of standard commands. You may find this useful when repeatedly using the same commands to perform certain tasks like setting up measurements.

Aliases are similar to macros but do not include the capability to substitute parameters into alias bodies. The alias mechanism obeys the following rules:

- The alias name must consist of a valid IEEE488.2 message unit, which may not appear in a message preceded by a colon, comma, or a command or query program header.
- The alias name may not appear in a message followed by a colon, comma, or question mark.
- An alias name must be distinct from any keyword or keyword short form.
- An alias name cannot be redefined without first being deleted using one of the alias deletion functions.
- Alias names do not appear in response messages.

**Table 8: Alias commands**

Command	Command
<a href="#">ALias</a> on page 43	Sets or returns the alias state.
<a href="#">ALias:CATalog?</a> on page 44	Returns a list of the currently defined alias.
<a href="#">ALias:DEFine</a> on page 45	Assigns a sequence of program messages.
<a href="#">ALias:DELEte</a> on page 46	Removes a specified alias.
<a href="#">ALias:DELEte:ALL</a> on page 46	Deletes all existing aliases.
<a href="#">ALias:DELEte[:NAME]</a> on page 47	Removes a specified alias.
<a href="#">ALias[:STATE]</a> on page 47	Sets or returns the alias state.

## Acquisition command group

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

**Table 9: Acquisition commands**

Command	Command
<a href="#">ACQUIRE?</a> on page 37	Returns current acquisition settings.
<a href="#">ACQUIRE:MAXSAMPLERATE?</a> on page 38	Returns the maximum real-time sample rate.
<a href="#">ACQUIRE:MODE</a> on page 38	Sets or queries the instrument acquisition mode.
<a href="#">ACQUIRE:NUMACQ?</a> on page 40	Indicates the number of acquisitions that have taken place since starting instrument acquisition.
<a href="#">ACQUIRE:NUMAVG</a> on page 41	Sets the number of instrument waveform acquisitions that make up an averaged waveform.
<a href="#">ACQUIRE:STATE</a> on page 41	Starts or stops instrument acquisitions.
<a href="#">ACQUIRE:STOPAFTER</a> on page 42	Tells the instrument when to stop taking acquisitions.

## Calibration and Diagnostic command group

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

**Table 10: Calibration and Diagnostic commands**

Command	Description
<a href="#">*CAL?</a> on page 53	Performs an internal self-calibration and returns its status.
<a href="#">CALIBRATE:INTERNAL</a> on page 54	Performs an internal self-calibration but does not return any status.
<a href="#">CALIBRATE:INTERNAL:START</a> on page 54	Starts the internal signal path calibration.
<a href="#">CALIBRATE:INTERNAL:STATUS?</a> on page 55	Returns the current status of the internal signal path calibration.
<a href="#">CALIBRATE:RESULTS?</a> on page 56	Returns the status of all calibration subsystems without performing an SPC operation.
<a href="#">CALIBRATE:RESULTS:SPC?</a> on page 56	Returns the results of the last SPC operation
<a href="#">DIAG:LOOP:OPTION</a> on page 95	Sets the self-test loop option
<a href="#">DIAG:LOOP:OPTION:NTIMES</a> on page 96	Sets the self-test loop option to run N times
<a href="#">DIAG:LOOP:STOP</a> on page 96	Stops the self-test at the end of the current loop
<a href="#">DIAG:RESULT:FLAG?</a> on page 97	Returns the Pass/Fail status from the last diagnostic test sequence execution (those run automatically at power on, or those requested through the Service Menu).

Command	Description
<a href="#">DIAG:RESUlt:LOG?</a> on page 97	Returns the internal results log from the last diagnostic test sequence execution (those run automatically at power on, or those requested through the Service Menu).
<a href="#">DIAG:SElect</a> on page 98	Sets the type of diagnostics grouping.
<a href="#">DIAG:SElect:&lt;function&gt;</a> on page 99	Runs self-tests on the specified system subsystem.
<a href="#">DIAG:STATE</a> on page 99	Starts or stops the instrument self-test.
<a href="#">DIAG:FAN</a> on page 95	Read out the currently set PWM fan value
<a href="#">DIAG:TEMPVAL</a> on page 100	Read out the currently FPGA chip and ambient temperature

## Cursor command group

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

Use the commands in the cursor command subsystem to control the cursor display and readout. You can use these commands to control the setups for cursor 1 and cursor 2, such as cursor position. You can also use the commands to select one of the following cursor functions:

Off. Turns off the display of all cursors.

Waveform Cursors. Consists of two cursors. Waveform cursors enable you to conveniently measure waveform amplitude and time.

Screen Cursors. Consists of two pairs of independent horizontal and vertical cursors.

You can use these cursors to indicate an arbitrary position within the waveform display area.

**Table 11: Cursor commands**

Command	Description
<a href="#">CURSor?</a> on page 73	Returns current cursor settings.
<a href="#">CURSor:ENABLE</a> on page 74	Allows educators to disable or enable the Cursor functions. The function can be manually set from the Utility menu. To access the menu, refer to the product user manual.
<a href="#">CURSor:FUNction</a> on page 75	Selects and displays the instrument cursor type.
<a href="#">CURSor:HBArs?</a> on page 76	Returns the settings for the instrument horizontal bar cursors.
<a href="#">CURSor:HBArs:DELta?</a> on page 76	Returns the difference (in vertical units) between the two horizontal bar cursors in the instrument display.

Command	Description
<a href="#">CURSor:HBArS:POSITION&lt;x&gt;</a> on page 77	Positions a horizontal bar cursor.
<a href="#">CURSor:HBArS:UNIts</a> on page 78	Returns the vertical scale units for the selected cursor source waveform.
<a href="#">CURSor:HBArS:USE</a> on page 78	Sets the horizontal bar cursor measurement scale.
<a href="#">CURSor:MODe</a> on page 79	Sets or returns whether cursors move in unison or separately.
<a href="#">CURSor:VBArS?</a> on page 80	Returns the current vertical bar cursor horizontal position and units settings.
<a href="#">CURSor:VBArS:ALTERNATE&lt;x&gt;?</a> on page 80	Returns the alternate readout for the waveform (Vbar) cursors.
<a href="#">CURSor:VBArS:DELTA?</a> on page 81	Returns the time or frequency difference between the two vertical bar cursors.
<a href="#">CURSor:VBArS:HPOS&lt;x&gt;?</a> on page 82	Returns the horizontal value of the specified vertical bar ticks for cursor <x>
<a href="#">CURSor:VBArS:POSITION&lt;x&gt;</a> on page 82	Sets or returns the vbar cursor<x> horizontal position
<a href="#">CURSor:VBArS:UNIts</a> on page 83	Sets or queries the units for the vertical bar cursors.
<a href="#">CURSor:VBArS:VDELTA?</a> on page 84	Returns the vertical (amplitude) difference between the two vertical bar cursors.

## Display command group

These commands provide control of the Display feature.

**Table 12: Display commands**

Command	Description
<a href="#">DISplay:GRATicule</a> on page 100	Sets or queries the Graticule state.
<a href="#">DISplay:INTENSITy:BACKLight</a> on page 101	Sets or queries the display Backlight.
<a href="#">DISplay:PERSiStence:STATe</a> on page 102	Sets or returns the display persistence to ON or OFF. The query form returns the persistence state. This affects the display only.
<a href="#">DISplay:PERSiStence:VALUe</a> on page 102	Sets the value of persistence if the persistence state is set to on. This affects the display only.



## FFT command group

These commands provide control over the instrument FFT feature.

**Table 13: FFT commands**

Command	Description
<a href="#">FFT?</a> on page 113	Returns all FFT parameters.
<a href="#">FFT:HORizontal:POSition</a> on page 113	Sets or queries the FFT horizontal display position.
<a href="#">FFT:HORizontal:SCAle</a> on page 114	Sets or queries the FFT zoom factor.
<a href="#">FFT:SOURce</a> on page 114	Sets or queries the FFT source.
<a href="#">FFT:SRCWFM</a> on page 115	Sets or queries the FFT source waveform display state.
<a href="#">FFT:VERTical:POSition</a> on page 116	Sets or queries the FFT vertical display position.
<a href="#">FFT:VERTical:SCAle</a> on page 116	Sets or queries the FFT vertical zoom factor.
<a href="#">FFT:VERTical:UNIts</a> on page 117	Sets or returns the FFT vertical measurement units label.
<a href="#">FFT:VType</a> on page 117	Sets or queries the FFT waveform vertical units.
<a href="#">FFT:WINDow</a> on page 118	Sets or queries the FFT window state.
<a href="#">SElect:FFT</a> on page 212	Sets or queries the FFT display state.

## File system command group

File system commands perform file management tasks.

**Table 14: File system commands**

Command	Description
<a href="#">FILESystem?</a> on page 118	Returns the directory listing of the current working directory and the number of bytes of free space available.
<a href="#">FILESystem:CWD</a> on page 119	Sets or queries returns the current working directory (CWD) for FILESystem commands.
<a href="#">FILESystem:DELEte</a> on page 120	Deletes the specified file name.
<a href="#">FILESystem:DIR?</a> on page 121	Returns a list of strings.
<a href="#">FILESystem:FORMat</a> on page 122	Formats a mass storage device.
<a href="#">FILESystem:FREEspace?</a> on page 122	Returns the number of bytes of free space on the current drive.
<a href="#">FILESystem:MKDir</a> on page 123	Creates a folder at the specified location.
<a href="#">FILESystem:READFile</a> on page 124	Writes the contents of the specified file to the specified interface
<a href="#">FILESystem:REName</a> on page 124	Assigns a new name to a file or folder.

Command	Description
<a href="#">FILESystem:RMDir</a> on page 125	Deletes a folder at the specified location.
<a href="#">FILESystem:WRITEFile</a> on page 126	Writes the specified block data to the instrument current working directory

## File System Conventions

Use the following conventions when specifying file paths and file names:

- File and folder names have a maximum of 11 characters; eight characters, followed by a period, followed by up to three characters. This format is referred to as 8.3 naming.
- Wild card characters (\*, %, ) are not valid characters in file or path names.
- Lists the Windows-generated short file and folder names for long file or folder names created on PC Windows operating systems.

## Help everywhere command group

Help everywhere commands provide helpful user information.

**Table 15: Help everywhere commands**

Command	Description
<a href="#">HELPevery:ACQuire</a> on page 132	Enables or disables the display of help everywhere information for the acquire module.
<a href="#">HELPevery:ALL</a> on page 133	Enables or disables the display of help everywhere.
<a href="#">HELPevery:FFT</a> on page 134	Enables or disables the display of help everywhere information for the fft module.
<a href="#">HELPevery:CURSor</a> on page 133	Enables or disables the display of help everywhere information for the cursor module.
<a href="#">HELPevery:MATH</a> on page 134	Enables or disables the display of help everywhere information for the math module.
<a href="#">HELPevery:MEASUrement</a> on page 135	Enables or disables the display of help everywhere information for the measurement module.
<a href="#">HELPevery:REFerence</a> on page 136	Enables or disables the display of help everywhere information for the reference module.
<a href="#">HELPevery:TRIGger</a> on page 136	Enables or disables the display of help everywhere information for the trigger module.
<a href="#">HELPevery:UTIlity</a> on page 137	Enables or disables the display of help everywhere information for the utility module.
<a href="#">HELPevery:VERTical</a> on page 138	Enables or disables the display of help everywhere information for the vertical module.

## Horizontal command group

Horizontal commands control the time bases of the instrument. You can set the position and time per division of both the main and window time bases. You can substitute SECdiv for SCALE in all appropriate horizontal commands. This provides program compatibility with previous Tektronix digitizing instruments.

**Table 16: Horizontal commands**

Command	Description
<a href="#">HORIZontal?</a> on page 138	Returns all settings for the horizontal commands.
<a href="#">HORIZontal[:MAIN]:SCALE</a> on page 144	Sets or queries the time base horizontal scale.
<a href="#">HORIZontal[:MAIN]:SECdiv</a> on page 145	Specifies the horizontal time/div.
<a href="#">HORIZontal:TRIGger:POSition</a> on page 149	Same as :HORIZontal:POSition.
<a href="#">HORIZontal:TRIGger:POSition</a> on page 149	Set or queries the trigger position.
<a href="#">HORIZontal[:MAIN]:DElay:POSition</a> on page 140	Sets or returns the horizontal position, as percent of record, that is used when HORIZontal:DElay:MODE is set to OFF.
<a href="#">HORIZontal:MAIn:DElay:MODE</a> on page 141	The boolean argument type sets delay mode to on or off.
<a href="#">HORIZontal:MAIn:DElay:STATe</a> on page 142	Same as HORIZontal:DElay:MODE.
<a href="#">HORIZontal[:MAIN]:DElay:TIME</a> on page 143	Specifies the delay time in time units.
<a href="#">HORIZontal:RESOLution</a> on page 148	Same as HORIZontal:RECOrdlength.
<a href="#">HORIZontal:RECOrdlength</a> on page 147	Sets or queries the horizontal record length.
<a href="#">HORIZontal:RECOrdlength:Auto</a> on page 147	Sets or queries the record length mode.
<a href="#">HORIZontal:ACQLENGTH</a> on page 139	Queries the record length.
<a href="#">HORIZontal:PREViewstate</a> on page 146	Returns a boolean to indicate whether the acquisition system is in the preview state
<a href="#">HORIZontal[:MAIN]:SAMPLERate</a> on page 144	Returns the sample rate
<a href="#">HORIZontal:MAIn:UNIts[:STRing]</a> on page 146	Queries the horizontal units.
<a href="#">HORIZontal:DIVisions</a> on page 140	Queries the number of horizontal divisions.
<a href="#">HORIZontal:ROLL</a> on page 149	Query the state of roll mode.

## Math command group

Math commands provide math function definition.

**Table 17: Math commands**

Command	Description
<a href="#">MATH?</a> on page 155	Returns the definition for the math waveform.
<a href="#">MATH:DEFINE</a> on page 156	Performs the specified mathematical operation on the input signal or signals.
<a href="#">MATH:HORizontal:POSition</a> on page 157	Sets or returns the horizontal position of the math waveform.
<a href="#">MATH:HORizontal:SCALe</a> on page 157	Sets or returns the horizontal scale of the math waveform.
<a href="#">MATH:HORizontal:UNIts</a> on page 158	Sets or returns the math horizontal measurement units label.
<a href="#">MATH:LABel</a> on page 159	Sets or queries the waveform label for the math waveform.
<a href="#">MATH:VERTical:POSition</a> on page 159	Sets or returns the math waveform display position.
<a href="#">MATH:VERTical:SCALe</a> on page 160	Sets or returns the math waveform display scale in units per division.
<a href="#">MATH:VERTical:UNIts</a> on page 161	Sets or returns the math vertical measurement units label.

## Measurement command group

Measurement commands control the automated measurement system. The instrument can display up to six automated measurements. In the commands, these measurement readouts are named MEAS<x>, where <x> can be 1, 2, 3, 4, 5, or 6.

The best method for taking measurements over the computer interface is to use the MEASUREMENT:IMMED commands and queries. The immediate measurement has no front-panel equivalent, and the instrument never displays immediate measurements.

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

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

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

**Table 18: Measurement commands**

Command	Description
<a href="#">MEASUrement?</a> on page 161	Returns the current MEASUrement settings.
<a href="#">MEASUrement:CLEARSNapshot</a> on page 163	Clears the existing snapshot results and removes the snapshot window.
<a href="#">MEASUrement:GATing</a> on page 164	Sets or returns the measurement gating.
<a href="#">MEASUrement:IMMed?</a> on page 165	Returns all immediate measurement setup parameters.
<a href="#">MEASUrement:IMMed:DELAy?</a> on page 165	Returns information about the immediate delay measurement.
<a href="#">MEASUrement:IMMed:DELAy:EDGE&lt;x&gt;</a> on page 166	Sets or returns the slope of the edge used for immediate delay from and to waveform measurements
<a href="#">MEASUrement:IMMed:SOUrce1</a> on page 167	Sets or queries the source for all single source immediate measurements and specifies the source to measure from when taking an immediate delay or phase measurement.
<a href="#">MEASUrement:IMMed:SOUrce2</a> on page 168	Sets or queries the secondary source for dual-source immediate measurements.
<a href="#">MEASUrement:IMMed:TYPe</a> on page 169	Sets or queries the immediate measurement type.
<a href="#">MEASUrement:IMMed:UNIts?</a> on page 171	Returns the units for the immediate instrument measurement.
<a href="#">MEASUrement:IMMed:VALue?</a> on page 172	Executes the immediate instrument measurement specified by the MEASUrement:IMMed:TYPe command.

Command	Description
<a href="#">MEASUrement:MEAS&lt;x&gt;?</a> on page 173	Returns all measurement parameters for the displayed instrument periodic measurement specified by <x>.
<a href="#">MEASUrement:MEAS&lt;x&gt;:DElay?</a> on page 173	Returns the delay measurement parameters for the specified measurement.
<a href="#">MEASUrement:MEAS&lt;x&gt;:DElay:EDGE&lt;x&gt;</a> on page 174	Sets or returns the slope of the edge to use for delay “from” and “to” waveform measurements.
<a href="#">MEASUrement:MEAS&lt;x&gt;:SOUrce1</a> on page 174	Sets or queries the source for all single source measurements and specifies the source to measure from when taking a delay measurement or phase measurement.
<a href="#">MEASUrement:MEAS&lt;x&gt;:SOUrce2</a> on page 175	Sets or queries the reference source to measure to when taking a delay or phase measurement.
<a href="#">MEASUrement:MEAS&lt;x&gt;:STATE</a> on page 176	Sets or returns whether the specified measurement slot is computed and displayed.
<a href="#">MEASUrement:MEAS&lt;x&gt;:TYPe</a> on page 177	Sets or queries the on-screen periodic instrument measurement type for the measurement specified by <x>.
<a href="#">MEASUrement:MEAS&lt;x&gt;:UNIts?</a> on page 180	Returns the units for the instrument measurement specified by MEASUrement:MEAS<x>:TYPe.
<a href="#">MEASUrement:MEAS&lt;x&gt;:VALue?</a> on page 181	Returns the value that was calculated for the instrument on-screen periodic measurement specified by <x>.
<a href="#">MEASUrement:REFLevel?</a> on page 182	Returns the current reference level parameters. Query only.
<a href="#">MEASUrement:REFLevel:ABSolute:LOW</a> on page 182	Sets or returns the low reference level, and is the lower reference level when MEASUrement:REFLevel:METHod is set to Absolute.
<a href="#">MEASUrement:REFLevel:ABSolute:MID1</a> on page 183	Sets or returns the mid reference level, and is the 50% reference level when MEASUrement:REFLevel:METHod is set to Absolute.
<a href="#">MEASUrement:REFLevel:ABSolute:MID2</a> on page 184	Sets or returns 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.
<a href="#">MEASUrement:REFLevel:METHod</a> on page 185	Specifies or returns the reference level units used for measurement calculations.
<a href="#">MEASUrement:REFLevel:PERCent:HIGH</a> on page 186	Sets or returns the percent (where 100% is equal to HIGH) used to calculate the high reference level when MEASUrement:REFLevel:METHod is set to Percent.

Command	Description
<a href="#">MEASUrement:REFLevel:PERCent:LOW</a> on page 187	Sets or returns the percent (where 100% is equal to HIGH) used to calculate the low reference level when MEASUrement:REFLevel:METHod is set to Percent.
<a href="#">MEASUrement:REFLevel:PERCent:MID1</a> on page 188	Sets or returns 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.
<a href="#">MEASUrement:REFLevel:PERCent:MID2</a> on page 189	Sets or returns the percent (where 100% is equal to HIGH) that is used to calculate the mid reference level for the second waveform specified when MEASUrement:REFLevel:METHod is set to Percent.
<a href="#">MEASUrement:SNAPSHOT</a> on page 190	Sets the measurement snapshot feature.
<a href="#">MEASUrement:SOURCESNAPShot</a> on page 190	Sets or returns the snapshot source.

## Miscellaneous command group

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

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

**Table 19: Miscellaneous commands**

Command	Description
<a href="#">AUTOSet</a> on page 49	Causes the instrument to adjust its vertical, horizontal, and trigger controls to display a stable waveform.
<a href="#">AUTOSet:ENABLE</a> on page 49	Allows educators to disable or enable the Autorange and Autoset functions. The function can be manually set from the Utility menu. To access the menu, refer to the product user manual.
<a href="#">CLEARMenu</a> on page 72	Clears the current menu from the display
<a href="#">DATE</a> on page 93	Sets or queries the instrument date value.

Command	Description
<a href="#">FEAEN:PASSWORD</a> on page 112	Checks if the password entered is equal to the preset password for Feature Enable which allows the educators to enable or disable features such as Autoset, Cursors or Measurements. The same can be set manually from the Utility menu when we try to change the settings of Autoset Enable, Cursors Enable or Measurement Enable from Feature Enable.
<a href="#">FPAnel:PRESS</a> on page 127	Simulates the action of pressing a specified front-panel button.
<a href="#">FPAnel:TURN</a> on page 128	Simulates the action of turning a specified front-panel control knob.
<a href="#">FWUpdate:Update</a> on page 129	Update firmware from u-disk.
<a href="#">HDR</a> on page 131	This command is identical to the HEADer query and is included for compatibility with other Tektronix instruments.
<a href="#">HEADer</a> on page 131	Sets and queries the Response Header Enable State that causes the instrument to either include or omit headers on query responses.
<a href="#">ID?</a> on page 151	Returns identifying information about the instrument and its firmware in Tektronix Codes and Formats notation.
<a href="#">*IDN?</a> on page 152	Returns the instrument identification code in IEEE 488.
<a href="#">LANGuage</a> on page 153	Sets or queries the languages that the instrument uses to display information on the screen.
<a href="#">LOCK</a> on page 154	Enables and disables all front-panel buttons and knobs.
<a href="#">*LRN?</a> on page 154	This is identical to the query.
<a href="#">MEASUrement:ENABLE</a> on page 163	Allows educators to disable or enable the Measurement functions. The function can be manually set from the Utility menu. To access the menu, refer to the product user manual.
<a href="#">*RST</a> on page 202	(Reset) Returns the instrument to a known set of instrument settings, but does not purge any stored settings.
<a href="#">SET?</a> on page 215	Returns most instrument settings.
<a href="#">TEKSecure</a> on page 219	Equivalent to invoking Teksecure from the Utility->Config->TekSecure Erase Memory menu.
<a href="#">TIme</a> on page 219	Sets or queries the instrument time value.
<a href="#">UNLock</a> on page 243	Unlocks the front panel.
<a href="#">VERBose</a> on page 245	Sets and queries the Verbose state that controls the length of keywords on query responses.



## Save and Recall command group

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

To display a saved waveform, use the SElect:<wfm> command.

**Table 20: Save and Recall commands**

Command	Description
<a href="#">FACTory</a> on page 111	Resets the instrument to factory default settings.
<a href="#">*RCL</a> on page 195	Restores the instrument to factory default settings stored in internal nonvolatile memory.
<a href="#">RECAll:SETUp</a> on page 196	Restores the factory-default instrument settings, user-saved settings from internal nonvolatile memory, or user-saved settings from a file on a USB flash drive.
<a href="#">RECAll:WAVEForm</a> on page 197	Recalls a stored waveform from the USB flash drive into a reference location.
<a href="#">*SAV</a> on page 203	Saves the state of the instrument into a specified nonvolatile memory location.
<a href="#">SAVe:ASSign:TYPe</a> on page 204	Sets or queries the assignment of the data to be saved.
<a href="#">SAVe:IMAge</a> on page 204	Saves the screen image to a file on the USB flash drive.
<a href="#">SAVe:IMAge:FILEFormat</a> on page 205	Sets the screen image file format used by the SAvE:IMAge command and by the SAvE > Action > Save Image and SAvE > Action > Save All front-panel operations.
<a href="#">SAVe:IMAge:LAYout</a> on page 206	Sets or queries the layout to use for saved screen images.
<a href="#">SAVe:SETUp</a> on page 207	Saves the current state of the instrument into the specified nonvolatile memory location, or to a file on the USB flash drive.
<a href="#">SAVe:WAVEform</a> on page 208	Stores a waveform in one of the nonvolatile reference memory locations, or to a file on the USB flash drive.
<a href="#">SAVe:WAVEform:FILEFormat</a> on page 209	Sets or returns the format for saved waveforms.
<a href="#">SETUP&lt;x&gt;:DATE?</a> on page 216	Returns the date when the specified instrument setup was saved.
<a href="#">SETUP&lt;x&gt;:TIME? (Query Only)</a> on page 216	Returns the time when the specified instrument setup was saved

## Status and Error command group

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

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

**Table 21: Status and Error commands**

Command	Description
<a href="#">ALLEv?</a> on page 48	Causes the instrument to return all events and their messages, and removes the returned events from the Event Queue.
<a href="#">BUSY?</a> on page 51	Returns the status of the instrument.
<a href="#">*CLS</a> on page 72	Clears the Event Queue, Standard Event Status Register, and Status Byte Register (except the MAV bit)..
<a href="#">DESE</a> on page 94	Sets and queries the bits in the Device Event Status Enable Register (DESER).
<a href="#">*ESE</a> on page 106	Sets and queries the bits in the Event Status Enable Register (ESER).
<a href="#">*ESR?</a> on page 107	Returns the contents of the Standard Event Status Register (SESR).
<a href="#">EVENT?</a> on page 108	Returns from the Event Queue an event code that provides information about the results of the last *ESR? read.
<a href="#">EVMsg?</a> on page 108	Removes from the Event Queue a single event code associated with the results of the last *ESR? read, and returns the event code with an explanatory message.
<a href="#">EVQty?</a> on page 109	Return number of events in the event queue.
<a href="#">*OPC</a> on page 191	Generates the operation complete message in the Standard Event Status Register (SESR) when all pending commands that generate an OPC message are complete.
<a href="#">*PSC</a> on page 193	Sets and queries the power-on status flag that controls the automatic power-on handling of the DESER, SRER, and ESER registers.
<a href="#">*RST</a> on page 202	Resets the instrument to factory default settings.
<a href="#">*SRE</a> on page 217	(Service Request Enable) sets and queries the bits in the Service Request Enable Register (SRER).

Command	Description
<a href="#">*STB?</a> on page 218	(Read Status Byte) query returns the contents of the Status Byte Register (SBR) using the Master Summary Status (MSS) bit.
<a href="#">*WAI</a> on page 247	Prevents the instrument from executing further commands or queries until all pending commands that generate an OPC message are complete.

## Trigger command group

Trigger commands control all aspects of instrument triggering.

The three types of triggers are edge, pulse width, and video. Edge triggering is the default type. Edge triggering lets you acquire a waveform when the signal passes through a voltage level of your choosing. Pulse width triggering lets you trigger on normal or aberrant pulses. Video triggering adds the capability of triggering on video fields and lines.

**Table 22: Trigger commands**

Command	Description
<a href="#">TRIGger</a> on page 220	Forces a trigger event to occur.
<a href="#">TRIGger:A</a> on page 221	Sets the instrument trigger level to 50% of the minimum and maximum values of the signal.
<a href="#">TRIGger:A:EDGE?</a> on page 222	Returns the trigger coupling, source, and slope settings for the edge trigger.
<a href="#">TRIGger:A:EDGE:COUPling</a> on page 222	Sets or queries the type of coupling for the edge trigger.
<a href="#">TRIGger:A:EDGE:SLOpe</a> on page 223	Selects a rising or falling slope for the edge trigger.
<a href="#">TRIGger:A:EDGE:SOUrce</a> on page 224	Sets or queries the source for the edge trigger.
<a href="#">TRIGger:A:HOLDOff?</a> on page 225	Returns the A trigger holdoff settings.
<a href="#">TRIGger:A:HOLDOff:TIME</a> on page 225	Sets or queries the instrument trigger holdoff time.
<a href="#">TRIGger:A:LEVel</a> on page 226	Sets or returns the trigger level for the A trigger.
<a href="#">TRIGger:A:LEVel:CH&lt;x&gt;</a> on page 227	Sets or returns the trigger level for the specified channel. Each channel can have an independent level.
<a href="#">TRIGger:A:LOWerthreshold:CH&lt;x&gt;</a> on page 227	Sets or returns the lower threshold for the channel selected.
<a href="#">TRIGger:A:MODE</a> on page 228	Sets or queries the trigger mode for the Edge (all models) and Pulse width trigger types.
<a href="#">TRIGger:A:PULse?</a> on page 229	Returns the current Pulse Trigger settings.
<a href="#">TRIGger:A:PULSE:Width?</a> on page 231	Returns the pulse trigger width settings.

Command	Description
<a href="#">TRIGger:A:PULse:WIDTH:POLarity</a> on page 231	Sets or queries the polarity for the pulse trigger.
<a href="#">TRIGger:A:PULSEWidth:SOURce</a> on page 232	Sets or queries the source for the pulse trigger.
<a href="#">TRIGger:A:PULse:WIDTH:WHEN</a> on page 232	Sets or queries the trigger conditions for the pulse trigger.
<a href="#">TRIGger:A:PULse:WIDTH:WIDTH</a> on page 234	Sets or queries the width for the pulse trigger.
<a href="#">TRIGger:A:RUNT?</a> on page 235	Returns the current A runt trigger parameters.
<a href="#">TRIGger:A:RUNT:POLarity</a> on page 235	Sets or returns the polarity for the runt trigger.
<a href="#">TRIGger:A:RUNT:SOURce</a> on page 236	Sets or returns the source for the A runt trigger.
<a href="#">TRIGger:A:RUNT:WHEN</a> on page 237	Sets or returns the type of pulse width the trigger checks for when it detects a runt.
<a href="#">TRIGger:A:RUNT:WIDTH</a> on page 238	Sets or returns the width for a runt trigger.
<a href="#">TRIGger:A:TYPE</a> on page 238	Sets or queries the type of instrument trigger.
<a href="#">TRIGger:A:UPPerthreshold:CH&lt;x&gt;</a> on page 239	Sets the upper threshold for channel <x>, where x is the channel number. Each channel can have an independent level. Used only for runt trigger type.
<a href="#">TRIGger:FREQuency?</a> on page 240	Returns the edge or pulse width trigger frequency.
<a href="#">TRIGger:STATE?</a> on page 240	Returns the current state of the triggering system.

## Vertical command group

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

**Table 23: Vertical commands**

Command	Description
<a href="#">CH&lt;x&gt;?</a> on page 57	Returns vertical parameters for the specified channel.
<a href="#">CH&lt;x&gt;:BANdwidth</a> on page 59	Sets or queries the bandwidth setting of the specified instrument channel.
<a href="#">CH&lt;x&gt;:COUPLing</a> on page 60	Sets or queries the input coupling setting of the specified instrument channel.
<a href="#">CH&lt;x&gt;:DESKew</a> on page 61	Sets or queries the deskew time for the specified channel.
<a href="#">CH&lt;x&gt;:INVert</a> on page 62	Sets or returns the invert function for the specified channel.
<a href="#">CH&lt;x&gt;:LABel</a> on page 62	This commands sets or queries the waveform label for the specified channel.
<a href="#">CH&lt;x&gt;:OFFSet</a> on page 63	Sets or queries the channel offset.

Command	Description
<a href="#">CH&lt;x&gt;:POSition</a> on page 64	Sets or queries the vertical position of the specified instrument channel.
<a href="#">CH&lt;x&gt;:PRObe</a> on page 65?	Returns the gain, resistance, units, and ID of the probe that is attached to the specified channel.
<a href="#">CH&lt;x&gt;:PRObe:GAIN</a> on page 66	Sets or queries the gain factor of the probe that is attached to the specified channel.
<a href="#">CH&lt;x&gt;:PRObe:ID?</a> on page 67	Returns the type and serial number of the probe that is attached to the specified channel.
<a href="#">CH&lt;x&gt;:PRObe:ID:SERnumber?</a> on page 67	Returns the serial number of the probe that is attached to the specified channel.
<a href="#">CH&lt;x&gt;:PRObe:ID:TYPE?</a> on page 68	Returns the type of probe that is attached to the specified channel.
<a href="#">CH&lt;x&gt;:PRObe:SIGNAL</a> on page 68	Sets or queries the input bypass setting of the TekVPI probe attached to the specified channel.
<a href="#">CH&lt;x&gt;:PRObe:UNIts?</a> on page 69	Returns the units of measure of the probe that is attached to the specified channel.
<a href="#">CH&lt;x&gt;:SCALE</a> on page 69	Sets or queries the vertical scale of the specified instrument channel.
<a href="#">CH&lt;x&gt;:VOLts</a> on page 70	Sets or queries the vertical sensitivity of the specified channel.
<a href="#">CH&lt;x&gt;:YUNit</a> on page 71	Sets or queries the units of the specified channel.
<a href="#">REF&lt;x&gt;?</a> on page 198	Returns reference waveform data for the specified channel.
<a href="#">REF&lt;x&gt;:DATE?</a> on page 198	Returns the date that the reference waveform was stored.
<a href="#">REF&lt;x&gt;:TIME?</a> on page 199	Returns the time that the reference waveform was stored.
<a href="#">REF&lt;x&gt;:HORizontal:DELay:TIME?</a> on page 199	Returns the horizontal position of the specified reference waveform in percent of the waveform that is displayed to the right of the center vertical graticule.
<a href="#">REF&lt;x&gt;:HORizontal:SCALE?</a> on page 200	Returns the horizontal scale for the reference waveform.
<a href="#">REF&lt;x&gt;:POSition?</a> on page 200	Returns the vertical position for channel <x>, where x is the reference channel number. Query only.
<a href="#">REF&lt;x&gt;:VERTical:POSition?</a> on page 201	Returns the vertical position of the specified reference waveform.
<a href="#">REF&lt;x&gt;:VERTical:SCALE?</a> on page 201	Returns the reference waveform vertical scale in vertical units/div.
<a href="#">SElect:CH&lt;x&gt;</a> on page 210	Turns the display of the channel <x> waveform on or off.

Command	Description
<a href="#">SElect:CONTROL</a> on page 211	Sets or returns the waveform that is selected as the implied recipient of channel-related commands
<a href="#">SElect:FFT</a> on page 212	Turns on or off the FFT waveform or queries whether the FFT waveform is on or off.
<a href="#">SElect:MATH</a> on page 213	Turns on or off the math waveform or queries whether the math waveform is on or off.
<a href="#">SElect:REF&lt;x&gt;</a> on page 214	Turns on or off the specified reference waveform or queries whether the specified reference waveform is on or off.

## Waveform command group

Waveform commands let you transfer waveform data points to and from the instrument. Waveform data points are a collection of values that define a waveform. One data value usually represents one data point in the waveform record. When working with peak-detect waveforms, each data value is either the min or max of a min/max pair. Before you can transfer waveform data, you must specify the data format and waveform locations.

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

**Table 24: Waveform commands**

Command	Description
<a href="#">CURVe</a> on page 85	Transfers instrument waveform data to and from the instrument in binary or ASCII format.
<a href="#">DATa</a> on page 87	Sets or queries the format and location of the waveform data that is transferred with the CURVe command.
<a href="#">DATa:DESTination</a> on page 88	Sets or queries the reference memory location for storing instrument waveform data that is transferred into the instrument by the CURVe command.
<a href="#">DATa:SOUrce</a> on page 89	Sets or queries which waveform will be transferred from the instrument by the CURVe, WFMPre, or WAVFrm? queries.
<a href="#">DATa:STARt</a> on page 90	Sets or queries the starting data point for waveform data transfers.
<a href="#">DATa:STOP</a> on page 91	Sets or queries the last data point in the waveform that will be transferred when executing the CURVe? command.
<a href="#">DATa:WIDth</a> on page 92	Sets the number of bytes per waveform data point to be transferred when executing the CURVe command.

Command	Description
<a href="#">WAVFrm?</a> on page 248	Returns WFMPre? and CURVe? data for the waveform specified by the DATA:SOURce command.
<a href="#">WFMinpre:BIT_Nr</a> on page 249	Sets or queries the number of bits per waveform point for the waveform to be transferred.
<a href="#">WFMinpre:BYT_Nr</a> on page 250	Sets or queries the data width for the waveform to be transferred.
<a href="#">WFMinpre:ENCdg</a> on page 250	Sets or queries the type of encoding for waveform data transferred with the CURVe command.
<a href="#">WFMinpre:NR_Pt?</a> on page 251	Returns the number of points that are in the transmitted waveform record, as specified by DATA:SOURce.
<a href="#">WFMinpre:XINcr</a> on page 252	The set form of this command specifies the interval (seconds per point for nonFFT, Hertz per point for FFT) between samples of the reference waveform specified by the DATA:DESTination command.
<a href="#">WFMinpre:XUNit</a> on page 252	Sets the horizontal units ("s" for seconds and "Hz" for Hertz) for the reference waveform specified by the DATA:DESTination command.
<a href="#">WFMinpre:XZEro</a> on page 253	The set form of this command specifies the position, in XUNits, of the first sample of the reference waveform specified by the DATA:DESTination command, relative to the trigger.
<a href="#">WFMinpre:YMUlt</a> on page 254	Sets or queries the vertical scale factor of the incoming waveform, expressed in YUNits per waveform data point level.
<a href="#">WFMinpre:YOFf</a> on page 255	Sets or queries the vertical position of the incoming waveform in digitizing levels.
<a href="#">WFMinpre:YUNit</a> on page 256	Sets the vertical units for the reference waveform specified by DATA:DESTination.
<a href="#">WFMinpre:YZEro</a> on page 257	Sets or returns the vertical offset of the incoming waveform in units specified by WFMinpre:YUNit.
<a href="#">WFMOupre?</a> on page 258	Returns waveform transmission and formatting settings for the waveform specified by the DATA:SOURce command.
<a href="#">WFMOupre:BIT_Nr</a> on page 258	Sets and queries the number of bits per waveform point that outgoing waveforms contain, as specified by the DATA:SOURce command.
<a href="#">WFMOupre:BYT_Nr</a> on page 260	Sets or queries the data width for the outgoing waveform specified by the DATA:SOURce command.
<a href="#">WFMOupre:ENCdg</a> on page 261	Sets and queries the type of encoding for outgoing waveforms.

Command	Description
<a href="#">WFMOutpre:NR_Pt?</a> on page 262	Returns the number of points for the DATA:SOURce waveform that will be transmitted in response to a CURVe? query.
<a href="#">WFMOutpre:RECOrdlength?</a> on page 262	Returns the record length for the source waveform as specified by the DATA:SOURce command.
<a href="#">WFMOutpre:WFId?</a> on page 263	Returns a descriptive string from the waveform specified in the DATA:SOURce command, if that waveform is active or displayed.
<a href="#">WFMOutpre:XINcr?</a> on page 264	Returns the horizontal point spacing in units of WFMOutpre:XUNit for the waveform specified by the DATA:SOURce command.
<a href="#">WFMOutpre:XUNit?</a> on page 264	Returns the horizontal units for the waveform specified by the DATA:SOURce command.
<a href="#">WFMOutpre:XUNit?</a> on page 264	Returns the time coordinate of the first point in the outgoing waveform.
<a href="#">WFMOutpre:YMUlt?</a> on page 266	Returns the vertical scale factor per digitizing level in units specified by WFMOutpre:YUNit for the waveform specified by the DATA:SOURce command.
<a href="#">WFMOutpre:YOFf?</a> on page 266	Returns the vertical position in digitizing levels for the waveform specified by the DATA:SOURce command.
<a href="#">WFMOutpre:YUNit?</a> on page 267	Returns the vertical units for the waveform specified by the DATA:SOURce command.
<a href="#">WFMOutpre:YZEro?</a> on page 268	Returns the vertical offset in units specified by WFMOutpre:YUNit? for the waveform specified by the DATA:SOURce command.

## Waveform data formats

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

The DATA:WIDth command lets you specify the number of bytes per data point when transferring data to and from an instrument. This provides compatibility with other digitizing instruments.

When DATA:WIDth is set to two:

- If sending data, the instrument multiplies each point by 256; the most significant byte then has meaningful data and the least significant byte is 0.
- If receiving data, the instrument truncates the data (divides by 256) and saves the most significant byte.

---

**NOTE.** *The instrument uses these methods to handle waveforms transmitted in ASCII or binary format.*

---

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



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

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

An example of an ASCII waveform data string follows:

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

- Binary data can be represented by signed integer or positive integer values. The range of the values depends on the byte width specified.

**Table 25: Binary data ranges**

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

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

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

## Waveform data record

You can transfer multiple points for each waveform record. You can transfer a part 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 instrument you must specify the location of the first data point within the waveform record. For example, when DATA:START is set to 1, data points will be stored starting with the first point in the record, and when DATA:START is set to 500, data will be stored starting at the 500th point in the record. The instrument ignores DATA:STOP when reading in data as the instrument will stop reading data when there is no more data to read or when it has reached 2500 data points.

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

## Waveform data locations and memory allocation

The DATA:SOURce command specifies the location of the data when transferring waveforms from the instrument. You can transfer one waveform into the instrument at a time. Each waveform is stored in one of two stored waveform locations. You specify the stored waveform location with the DATA:DESTination command.

---

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

---

## Waveform preamble

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

## Scaling waveform data

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

## Transferring waveform data

Data transfer times depend on data format, data width, and the speed of the controller. [Programming Examples](#) on page 295

From the instrument. To transfer waveforms from the instrument to an external controller, follow these steps:

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

To the instrument. To transfer waveform data to an instrument waveform storage location, follow these steps:

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

## Zoom command group

These commands support the zoom feature.

**Table 26: Zoom commands**

Command	Description
<a href="#">ZOOM?</a> on page 269	Returns the current horizontal positioning and scaling of the display.
<a href="#">ZOOM{:MODE[:STATE]}</a> on page 269	Sets or queries the zoom on/off state.
<a href="#">ZOOM:ZOOM1?</a> on page 270	Returns the current horizontal positioning and scaling of the display
<a href="#">ZOOM:ZOOM1:FACTOR</a> on page 271	Sets or queries the zoom factor of a particular zoom box.
<a href="#">ZOOM:ZOOM1:HORIZONTAL:POSITION</a> on page 271	Sets or queries the horizontal position of a particular zoom box.
<a href="#">ZOOM:ZOOM1:HORIZONTAL:SCALE</a> on page 272	Sets or queries the horizontal zoom scale of the specified waveform in the specified zoom.
<a href="#">ZOOM:ZOOM1:POSITION</a> on page 273	Sets or returns the horizontal zoom position for the specified waveform in the specified zoom
<a href="#">ZOOM:ZOOM1:SCALE</a> on page 273	Sets or returns the horizontal zoom scale of the specified waveform in the specified zoom
<a href="#">ZOOM:ZOOM1:STATE</a> on page 274	Specifies or returns a trace as zoomed, on or off.



---

## A commands

This section lists commands and queries that begin with the letter A.

### ACQuire?

Returns the current acquisition settings. Query only.

**Group** Acquisition

**Syntax** ACQuire?

**Related Commands** [ACQuire:MODE](#) on page 38, [ACQuire:NUMACq?](#) on page 40, [ACQuire:NUMAVg](#) on page 41, [ACQuire:NUMAVg](#) on page 41

**Returns** Returns current acquisition settings: Stop after, Acquisition state, Mode, Number of averages.

**Examples** ACQuire? might return the following string for the current acquisition:  
ACQUIRE:STOPAFTER RUNSTOP;STATE 1; MODE SAMPLE;NUMAVG  
16

## ACQuire:MAXSamplerate?

Returns the maximum real-time sample rate, which varies from model to model. Query only.

**Group** Acquisition

**Syntax** ACQuire:MAXSamplerate?

**Examples** ACQUIRE:MAXSAMPLERATE? might return 1.0000E+9 indicating the maximum real-time sample rate is 1.0 GS/s.

## ACQuire:MODe

Sets or queries the acquisition mode of the instrument for all live waveforms.

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

The instrument sampling system always samples at the maximum rate, so the acquisition interval may include more than one sample. The acquisition mode, which you set using this ACQuire:MODe command, determines how the final value of the acquisition interval is generated from the many data samples.

**Group** Acquisition

**Syntax** ACQuire:MODe {SAMple|PEAKdetect|HIRes|AVERage}  
ACQuire:MODe?

**Related commands**    *ACQuire:NUMAVg* on page 41, *CURVe* on page 85

### Arguments

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

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

**HIRes** specifies Hi Res mode where the displayed data point value is the average of all the samples taken during the acquisition interval. This is a form of averaging, where the average comes from a single waveform acquisition. The number of samples taken during the acquisition interval determines the number of data values that compose the average.

**AVErage** specifies averaging mode, in which the resulting waveform shows an average of **SAMple** data points from several separate waveform acquisitions. The instrument processes the number of waveforms you specify into the acquired waveform, creating a running exponential average of the input signal. The number of waveform acquisitions that go into making up the average waveform is set or queried using the *ACQuire:NUMAVg* command.

### Examples

*ACQuire:MODE AVEr* sets average acquisition mode so that the resulting waveform is the average of the specified number of waveform acquisitions.

*ACQuire:MODE?* might return *ACQUIRE:MODE AVERAGE* indicating that the displayed waveform is the average of the specified number of waveform acquisitions.

## ACQuire:NUMACq?

Indicates the number of acquisitions that have taken place since starting instrument acquisition. The acquisition number will continue to increase while acquisitions are running until there is a reset.

Starting and stopping acquisitions do not cause this number to reset. For example, if acquisitions are running, the acquisition count will increase (assuming the instrument is triggering). If you stop the acquisitions, the acquisition number will freeze at a given number (For example: 5000). If you start acquisitions again, it will continue from 5000. The number will reset to 0 only if you change the horizontal scale while acquisitions are running.

**Group**      Acquisition

**Syntax**     ACQuire:NUMACq?

**Related commands**    [ACQuire:STATE](#) on page 41

**Returns**      <NR1>

**Examples**      ACQuire:NUMACq? might return ACQUIRE:NUMACQ 350 indicating that 350 acquisitions have occurred.



## ACQuire:NUMAVg

Sets or queries the number of waveform acquisitions that make up an averaged waveform. Use the ACQuire:MODE command to enable Average mode. Sending this command is equivalent to turning a multipurpose knob to enter the number of waveform acquisitions to average.

**Group** Acquisition

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

**Arguments** <NR1> is the number of waveform acquisitions to average. The range of values is from 2 to 512 in powers of two.

**Examples** ACQuire:NUMAVg 16 specifies that 16 waveform averages are performed before exponential averaging starts.  
ACQuire:NUMAVg? might return ACQUIRE:NUMAVG 64 indicating that there are 64 acquisitions specified for averaging.

## ACQuire:STATE

Starts or stops acquisitions.

When State is set to ON or RUN, a new acquisition is started. If the last acquisition was a single acquisition sequence, a new single sequence acquisition is started. If the last acquisition was continuous, a new continuous acquisition is started.

If RUN is issued in the middle of completing a single sequence acquisition (for example, averaging or enveloping), the acquisition sequence is restarted, and any accumulated data is discarded. Also, the instrument resets the number of acquisitions. If the RUN argument is issued while in continuous mode, acquisition continues.

<b>Group</b>	Acquisition
<b>Syntax</b>	ACQuire:STATE {OFF ON RUN STOP <NR1>} ACQuire:STATE?
<b>Related Commands</b>	<i>*OPC</i> on page 191, <i>ACQuire:STOPAfter</i> on page 42
<b>Arguments</b>	OFF STOP <NR1> = 0 stops acquisitions; any other value starts acquisitions.. ON RUN <NR1> ≠ 0 starts acquisition and display of waveforms.
<b>Examples</b>	ACQuire:STATE RUN starts acquisition of waveform data and resets the number of acquisitions count (NUMACq) to zero.  ACQuire:STATE? might return: ACQUIRE:STATE 0 indicating that the acquisition is stopped.

## ACQuire:STOPAfter

Sets or returns whether the instrument continually acquires acquisitions or acquires a single sequence.

<b>Group</b>	Acquisition
<b>Syntax</b>	ACQuire:STOPAfter {RUNSTop SEQuence} ACQuire:STOPAfter?

**Related commands**    *ACquire:STATE* on page 41

**Arguments**    RUNSTop specifies that the instrument will continually acquire data, if ACquire:STATE is turned on.

SEquence specifies that the next acquisition will be a single-sequence acquisition.

**Examples**    ACquire:STOPAfter RUNSTOP sets the instrument to continually acquire data.  
ACquire:STOPAfter? might return: ACQUIRE:STOPAFTER SEQUENCE  
indicating that the next acquisition the instrument makes will be of the single-sequence type.

## ALias

Sets or queries the state of alias functionality.

Use Alias commands to define new commands as a sequence of standard commands. You may find this useful when repeatedly using the same commands to perform certain tasks like setting up measurements. Aliases are similar to macros but do not include the capability to substitute parameters into alias bodies.

To use Alias commands, first define the alias, then turn on the alias state.

**Group**    Alias

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

**Related commands**    *ALias:DEFine* on page 45, *ALias[:STATE]* on page 47

**Arguments**    OFF turns alias expansion off. If a defined alias is sent when ALias is off, a command error (110) will be generated.

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

<NR1> = 0 disables alias mode; any other value enables alias mode.

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

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

## ALias:CATalog?

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

**Group**    Alias

**Syntax**    ALias:CATalog?

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

## ALias:DEFine

Assigns a sequence of program messages to an alias label. These messages are then substituted for the alias whenever it is received as a command or query, provided that ALias:STATE has been turned on. The query form of this command returns the definitions of a selected alias.

---

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

---

**Group** Alias

**Syntax** ALias:DEFine <QString><,>{<QString>|<Block>}  
ALias:DEFine? <QString>

**Related commands** [ALias\[:STATE\]](#) on page 47

**Arguments** The first <QString> is the alias label. This label cannot be a command name. Labels must start with a letter and can contain only letters, numbers, and underscores; other characters are not allowed. The label must be less than or equal to 12 characters. The second <QString> or <Block> is a complete sequence of program messages. The messages can contain only valid commands that must be separated by semicolons and must follow all rules for concatenating commands. The sequence must be less than or equal to 256 characters.

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

## ALias:DELEte

Removes a specified alias and is identical to ALIas:DELEte:NAME. An error message is generated if the named alias does not exist. (No query form.

**Group** Alias

**Syntax** ALIas:DELEte <QString>

**Related commands** [\\*ESR?](#) on page 107, [ALIas:DELEte:ALL](#) on page 46

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

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

## ALias:DELEte:ALL

Deletes all existing aliases. No query form.

**Group** Alias

**Syntax** ALIas:DELEte:ALL

**Related commands** [ALIas:DELEte](#) on page 46, [ALIas:DELEte\[:NAME\]](#) on page 47

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

## ALias:DELEte[:NAME]

Removes a specified alias. This command is identical to ALias:DELEte. No query form.

**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 an existing 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] {<NR1>|OFF|ON}  
ALias[:STATE]?

**Arguments** OFF or  $\langle \text{NR1} \rangle = 0$  turns alias expansion off. If a defined alias is sent when ALIAS:STATE is OFF, a command error (102) is generated.

ON or  $\langle \text{NR1} \rangle \neq 0$  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 the alias feature is off.

## ALLEv?

Causes the instrument 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. Refer to the Status and Events section for a complete description of how to use these registers. This command is similar to repeatedly sending \*EVMsg? queries to the instrument. Query only.

**Group** Status and Error

**Syntax** ALLEv?

**Related Commands** [\\*CLS](#) on page 72, [DESE](#) on page 94, [\\*ESE](#) on page 106, [\\*ESR?](#) on page 107, [EVENT?](#) on page 108, [EVMsg?](#) on page 108, [EVQty?](#) on page 109, [\\*SRE](#) on page 217, [\\*STB?](#) on page 218

**Returns** The event code and message in the following format:

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

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

$\langle \text{Command} \rangle$  is the command that caused the error and may be returned when a command error is detected by the instrument. As much of the command is 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** ALLEv? might return the following string: ALLEV 2225,"MEASUREMENT ERROR, NO WAVEFORM TO MEASURE; ",420,"QUERY UNTERMINATED; "

## AUTOSet

Causes the instrument to adjust its vertical, horizontal, and trigger controls to display a stable waveform. This command is equivalent to pushing the front-panel AUTOSET button. For a detailed description of the Autoset function, refer to the user manual for your instrument. Command only, no query form.

**Group** Miscellaneous

**Syntax** AUTOSet {EXECute | UNDo}

**Arguments** EXECute runs Autoset on the selected waveform.  
UNDo restores the oscilloscope settings to those prior to running Autoset.

## AUTOSet:ENABLE

Allows educators to disable or enable the Autorange and Autoset functions. The function can be manually set from the Utility menu. To access the menu, refer to the product user manual.

**Group** Miscellaneous

**Syntax** AUTOSet:ENABLE {ON | OFF}  
AUTOSet:ENABLE?

**Related commands**    *AUTOSet* on page 49  
                              *FEAEN:PASSWORD* on page 112

**Arguments**        ON enables the autoset feature.  
                          OFF disables the autoset feature.

**Examples**        AUTOSET:ENABLE OFF disables autoset.  
                          AUTOSET:ENABLE? might return 1 indicating that autoset is enabled.

---

## B commands

This section lists commands and queries that begin with the letter B.

### BUSY?

Returns the status of the instrument. This command allows you to synchronize the operation of the instrument with your application program. Query only.

Certain instrument operations can affect the BUSY? response. [Table 30: Instrument operations that can generate OPC](#) on page 282

**Group** Status and Error

**Syntax** BUSY?

**Related Commands** [\\*OPC](#) on page 191, [\\*WAI](#) on page 247

**Returns** <NR1> = 0 means the instrument is not busy processing a command whose execution time is extensive.

<NR1> = 1 means the instrument is busy processing a command whose execution time is extensive. [Table 30: Instrument operations that can generate OPC](#) on page 282

**Examples** BUSY? might return :BUSY 1 indicating that the instrument is now busy. See [Using the BUSY Query](#) on page 284 for an example of how to use this query.



## C commands

This section lists commands and queries that begin with the letter C.

### \*CAL?

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

---

**NOTE.** *The self-calibration can take several minutes to complete. During this time, the instrument does not execute any commands.*

*Disconnect all signals from the instrument before performing an internal self-calibration.*

---

**Group** Calibration and Diagnostic

**Syntax** \*CAL?

**Related Commands** [CALibrate:INTERNAL](#) on page 54

**Returns** 0 indicates that the self-calibration completed without any errors detected.  
Any value other than zero indicates that the self-calibration did not complete successfully or completed with errors.

**Examples** \*CAL? performs a self-calibration and might return 0 to indicate that it completed successfully.

## CALibrate:INTERNAL

This command starts a signal path compensation. Command only, no query form.

---

**NOTE.** *The self-calibration can take several minutes to complete. During this time, the instrument does not execute any commands.*

---

*Disconnect all signals from the instrument before performing an internal self-calibration.*

**Group** Calibration and Diagnostic

**Syntax** CALibrate:INTERNAL

**Examples** CALibrate:INTERNAL starts a signal path compensation cycle.

## CALibrate:INTERNAL:START

Starts the internal signal path calibration (SPC) of the instrument. You can use the CALibrate:INTERNAL:STATus? query to return the current status of the internal signal path calibration of the instrument. No query form.

**Group** Calibration and Diagnostic

**Syntax** CALibrate:INTERNAL:START

**Related commands** [CALibrate:RESults:SPC?](#) on page 56

**Examples**     CALIBRATE:INTERNAL:START initiates the internal signal path compensation of the instrument.

## CALibrate:INTERNaL:STATus?

Returns the current status of the instrument internal signal path compensation for the last SPC operation. Query only.

**Group**     Calibration and Diagnostic

**Syntax**     CALibrate:INTERNaL:STATus?

**Related commands**     [\\*CAL?](#) on page 53

**Returns**     INIT indicates the instrument has not had internal signal path calibration run.  
PASS indicates the signal path calibration completed successfully.  
FAIL indicates the signal path calibration did not complete successfully.  
RUNNING indicates the signal path calibration is currently running.

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

## CALibrate:RESults?

Returns the status of internal and factory calibrations, without performing any calibration operations. Query only.

The results returned do not include the calibration status of attached probes. The CALibrate:RESults? query is intended to support GO/NoGO testing of the instrument calibration readiness: all returned results should indicate PASS status if the instrument is fit for duty. It is quite common, however, to use uncalibrated probes (particularly when the instrument inputs are connected into a test system with coaxial cables).

**Group** Calibration and Diagnostic

**Syntax** CALibrate:RESults?

**Related commands** [\\*CAL?](#) on page 53

**Examples** CALibrate:RESults? might return :CALibrate:RESults INIT indicating the instrument has not be calibrated.

## CALibrate:RESults:SPC?

Returns the status of the SPC operation. This query does not initiate a SPC. Query only.

**Group** Calibration and Diagnostic



**Syntax** CALibrate:RESults:SPC?

**Related commands** [\\*CAL?](#) on page 53

**Returns** INIT indicates that SPC has never successfully completed.  
PASS indicates that the last SPC operation passed.  
FAIL indicates that the last SPC operation failed.  
RUNNING indicates that the SPC operation is running.

**Examples** CALibrate:RESults:SPC? might return :CALibrate:RESults:SPC INIT indicating SPC has not been run successfully.

## CH<x>?

Returns the vertical parameters for the specified channel. The value of <x> can vary from 1 through 4 depending on instrument model. Query only.

Because CH<x>:SCALE and CH<x>:VOLts are identical, only CH<x>:SCALE is returned.

**Group** Vertical

**Syntax** CH<x>?

**Related Commands** [SElect:REF<x>](#) on page 214

**Returns** instrument vertical settings for the specified channel.

**Examples** CH1? might return :CH1:SCALE 1.0E0;POSITION 0.0E0; COUPLING DC;BANDWIDTH FULL;PROBE 1.0E0.

## CH<x>:AMPSVIAVOLTS:ENABLE

Sets or queries measure current status as ON or Off. The value <x> can vary from 1 through 2 depending upon the channel.

**Group** Vertical

**Syntax** CH<x>:AMPSVIAVOLTS:ENABLE {ON|OFF|<NR>}  
CH<x>:AMPSVIAVOLTS:ENABLE?

**Arguments** OFF turns current status off.  
ON turns current status on.  
<NR1> = 0 turns current status off; any other value turns current status on.

**Examples** Ch1:AMPSVIAVOLTS:ENABLE ON will change the Ch1 measure current status as Yes.

## CH<x>:AMPSVIAVOLTs:Factor

Sets or queries current factor . The value <x> can vary from 1 through 2 depending upon the channel .

<b>Group</b>	Vertical
<b>Syntax</b>	CH<x>:AMPSVIAVOLTs:FACTOR <NR3> CH<x>:AMPSVIAVOLTs:FACTOR?
<b>Arguments</b>	<NR3> is the factor value.
<b>Examples</b>	CH<x>:AMPSVIAVOLTs:FACTOR 1 will set it as 1 A per volt measurement. Ch1:AMPSVIAVOLTs:ENABLE ON will change the Ch1 measure current status as Yes.

## CH<x>:BANdwidth

Sets or queries the selectable low-pass bandwidth limit filter setting of the specified instrument channel. The value of <x> can vary from 1 through 2 depending on instrument model.

This command is equivalent to setting the BW Limit option in the Vertical menu.

<b>Group</b>	Vertical
<b>Syntax</b>	CH<x>:BANdwidth {TWEnty FULl <NR3>} CH<x>:BANdwidth?

<b>Arguments</b>	<p>TWEnty sets the upper bandwidth limit of channel &lt;x&gt; to 20 MHz.</p> <p>FULl disables any optional bandwidth limiting. The specified channel operates at its maximum attainable bandwidth.</p> <p>&lt;NR3&gt; is a double-precision ASCII string. The instrument rounds this value to an available bandwidth using geometric rounding, and then uses this value to set the upper bandwidth limit.</p> <p><i><b>NOTE.</b> Other values may be possible depending on the attached probes.</i></p>
<b>Examples</b>	<p>CH1:BANDWIDTH TWENTY sets the bandwidth of channel 1 to 20 MHz.</p> <p>CH1:BANDWIDTH? might return FULl. This indicates there is no bandwidth limiting on channel 1.</p>

## CH<x>:COUPling

Sets or queries the input attenuator coupling setting of the specified instrument channel. The value of <x> can vary from 1 through 2 depending on the instrument model.

This command is equivalent to setting the Coupling option in the Vertical menu.

<b>Group</b>	Vertical
<b>Syntax</b>	<p>CH&lt;x&gt;:COUPling {AC   DC}</p> <p>CH&lt;x&gt;:COUPling?</p>
<b>Arguments</b>	<p>AC sets the specified instrument channel to AC coupling.</p> <p>DC sets the specified instrument channel to DC coupling.</p>

**Examples** CH1:COUPLING AC establishes AC coupling on channel 1.  
 CH2:COUPLING? might return :CH2:COUPLing DC indicating that channel 2 is set to DC coupling.

## CH<x>:DESKew

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

**Group** Vertical

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

**Arguments** <NR3> is the deskew time for channel <x>, ranging from -100 ns to +100 ns with a resolution of 1 ns.

**Examples** CH1 :DESKew 5.0E-9 sets the deskew time for channel 1 to 5 ns.  
 CH2:DESKew? might return :CH2:DESKEW 2.0000E-09 indicating that the deskew time for channel 2 is set to 2 ns.

## CH<x>:INVert

Sets or queries the inversion state of the specified instrument channel. The value of <x> can vary from 1 through 2 depending on the instrument model.

This command is equivalent to setting the Invert option in the Vertical channel menus.

**Group** Vertical

**Syntax** CH<x>:INVert {ON|OFF}  
CH<x>:INVert?

**Arguments** ON inverts the specified instrument channel.  
OFF sets the specified instrument channel to noninverted.

**Examples** CH1:INVERT ON inverts the signal on channel 1.  
CH2:INVERT? might return :CH2:INVERT 0, indicating that channel 2 is not inverted.

## CH<x>:LABel

This command sets or queries the waveform label for channel<x>, where x is the channel number (1- 2).

**Group** Vertical

<b>Syntax</b>	CH<x>:LABel <Qstring> CH<x>:LABel?
<b>Arguments</b>	<Qstring> is an alphanumeric string of text, enclosed in quotes, that contains the text level information for the channel<x>waveform. The text string is limited to 30 characters.
<b>Examples</b>	CH1:LABEL "ICCDATA" sets the label name of Channel 1 waveform output to ICCDATA.  CH1:LABEL? might return "ICCDATA", if the channel label was already set, else would return "" if not set.

## CH<x>:OFFSet

Sets or queries the vertical offset for channel <x>, where x is the channel number.

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

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

The channel offset range depends on the vertical scale factor. The valid ranges for the instrument are (when the probe and external attenuation factor is X1):

For V/Div settings from 2 mV/div to 200 mV/div, the offset range is +/- 0.8 V

For V/Div settings from 202 mV/div to 5 V/div, the offset range is +/- 20 V

<b>Group</b>	Vertical
<b>Syntax</b>	CH<x>:OFFSet <NR3> CH<x>:OFFSet?
<b>Related commands</b>	<a href="#">CH&lt;x&gt;:POStion</a> on page 64
<b>Arguments</b>	<NR3> is the offset value for the specified channel <x>.
<b>Examples</b>	CH1:OFFSet 2.0E-3 sets the offset for channel 1 to 2 mV. CH2:OFFSet? might return : CH2:OFFSET 1.0000E-03 indicating that the offset for channel 2 is set to 1 mV.

## CH<x>:POStion

Sets or queries the vertical position of the specified instrument channel. The value of <x> can vary from 1 through 2 depending on the instrument model.

The position voltage value is applied to the signal before digitization. Increasing the position value of a waveform causes the waveform to move up. Decreasing the position value causes the waveform to move down. The position value determines the vertical graticule coordinate at which input signal values, minus the present offset setting for that channel, are displayed. For example, if the position for Channel 1 is set to 2.0 and the offset is set to 3.0, then input signals equal to 3.0 units are displayed 2.0 divisions above the center of the screen (at 1 V/div).

<b>Group</b>	Vertical
--------------	----------



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

**Related commands** [CH<x>:OFFSet](#) on page 63 [REF<x>:VERTical:POSition?](#) on page 201 [MATH:VERTical:POSition](#) on page 159

**Arguments** <NR3> is the position in divisions from the center graticule for the specified channel. The range is 5 to -5 divisions.

**Examples** CH2:POSITION 1.3E0 positions the channel 2 input signal 1.3 divisions above the center of the display.  
CH1:POSITION? might return :CH1:POSITION -1.3000 indicating that the vertical position of Channel 1 is 1.3 divisions below the center graticule.

## CH<x>:PRObe

Returns all information concerning the probe attached to channel <x>, where x is the channel number. The value of <x> can vary from 1 through 2 depending on the instrument model.

**Group** Vertical

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

**Examples** CH1:PROBE? might return No probe.

## CH<x>:PRObe:GAIN

Sets or queries the gain factor for the probe attached to the channel specified by <x>, where x is the channel number. The gain of a probe is the output divided by the input transfer ratio. For example, a common 10x probe has a gain of 0.1.

**Group** Vertical

**Syntax** CH<x>:PRObe:GAIN <NR3>  
CH<x>:PRObe:GAIN?

**Related commands** [CH<x>:SCale](#) on page 69

**Arguments** <NR3> is the probe gain. Allowed values depend on the specific probe.

**Examples** CH1:PRObe:GAIN 0.1 sets the channel 1 probe gain to 0.1.  
CH2:PROBE:GAIN? might return :CH2:PROBE:GAIN 0.1000E+00 indicating that the attached 10x probe delivers 1 V to the channel 2 BNC for every 10 V applied to the probe input.

## CH<x>:PRObe:ID?

Returns the type and serial number of the probe attached to channel <x>, where x is the channel number. Query only.

**Group** Vertical

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

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

## CH<x>:PRObe:ID:SERNuMBER?

Returns the serial number of the probe attached to channel <x>, where x is the channel number. Query Only.

---

**NOTE.** For Level 0 and 1 probes, the serial number will be "".

---

**Group** Vertical

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

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

## CH<x>:PRObe:ID:TYPE?

Returns the type of probe attached to the channel specified by <x>, where x is the channel number. Level 2 (or higher) probes supply their exact product nomenclature; for Level 0 or 1 probes, a generic “No Probe Detected message is returned. Query Only.

**Group** Vertical

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

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

## CH<x>:PRObe:SIGnal

Sets or queries the input bypass setting of a TekVPI probe attached to channel <x>, where x is the channel number. The probe must support input bypass, for example TCP0001. This command is ignored if sent to an unsupported probe.

**Group** Vertical

**Syntax** CH<x>:PRObe:SIGnal {BYPass|PASS}  
CH<x>:PRObe:SIGnal?

**Arguments** BYPass sets the probe to Bypass mode.  
PASS sets the probe to Pass mode.

**Examples** CH1:PROBe:SIGnal PASS set the probe attached to channel 1 to Pass mode.  
 CH1:PROBe:SIGnal? might return :CH1:PROBe:SIGnal PASS indicating that the probe attached to channel 1 is in Pass mode

## CH<x>:PROBe:UNIts?

Returns a string describing the units of measure for the probe attached to channel <x>, where x is the channel number. Query Only.

**Group** Vertical

**Syntax** CH<x>:PROBe:UNIts?

**Examples** CH2:PROBE:UNITS? might return :CH4:PROBE:UNITS "V" indicating that the units of measure for the probe attached to channel 2 are volts.

## CH<x>:SCAlE

Sets or queries the vertical scale of the specified instrument channel. The value of <x> can vary from 1 through 2 depending on the instrument model.

Each waveform has a vertical scale parameter. For a signal with constant amplitude, increasing the Scale causes the waveform to be displayed smaller. Decreasing the scale causes the waveform to be displayed larger.

Scale affects all waveforms, but affects channel waveforms differently from other waveforms:

For channel waveforms, this setting controls the vertical size of the acquisition window as well as the display scale. The range and resolution of scale values depends on the probe attached and any other external factors you have specified.

For reference and math waveforms, this setting controls the display only, graphically scaling these waveforms and having no affect on the acquisition hardware.

This command is equivalent to adjusting the front-panel VOLTS/DIV knob.

<b>Group</b>	Vertical
<b>Syntax</b>	CH<x>:SCALE <NR3> CH<x>:SCALE?
<b>Related Commands</b>	<a href="#">CH&lt;x&gt;:OFFSet</a> on page 63, <a href="#">CH&lt;x&gt;:POSition</a> on page 64, <a href="#">REF&lt;x&gt;:VERTical:SCALE?</a> on page 201, <a href="#">MATH:VERTical:SCALE</a> on page 160
<b>Arguments</b>	<NR3> is the scale, in units-per-division. The value entered here is truncated to three significant digits.
<b>Examples</b>	CH1:SCALE 100E-3 sets the channel 1 gain to 100 mV/div. CH2:SCALE? might return :CH2:SCALE 1.0000, indicating that the current V/div setting of channel 2 is 1 V/div.

## CH<x>:VOLts

Sets or queries the vertical sensitivity of the specified channel. The value of <x> can vary from 1 through 2 depending on the instrument model.

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.

<b>Group</b>	Vertical
<b>Syntax</b>	CH<x>:VOLts <NR3> CH<x>:VOLts?

**Arguments** <NR3> is the vertical sensitivity, in volts.

**Examples** CH1:VOLts 1.0 sets channel 1 to 1 Volt per division.  
CH1:VOLts? Might return CH1:VOLts 1.0 indication that the ch1 volts per division is 1 Volt per division.

## CH<x>:YUNit

Sets or queries the units of the specified channel. The value of <x> can vary from 1 through 2.

String arguments are case insensitive and any unsupported units will generate an error. Supported units are:

%, /Hz, A, A/A, A/V, A/W, A/dB, A/s, AA, AW, AdB, As, B, Hz, IRE, S/s, V, V/A, V/V, V/W, V/dB, V/s, VV, VW, VdB, Volts, Vs, W, W/A, W/V, W/W, W/dB, W/s, WA, WV, WW, WdB, Ws, dB, dB/A, dB/V, dB/W, dB/dB, dBA, dBV, dBW, dBdB, day, degrees, div, hr, min, ohms, percent, s.

**Group** Vertical

**Syntax** CH<x>:YUNit <QString>  
CH<x>:YUNit?

**Arguments** <QString> is a string of text surrounded by quotes, specifying the supported units. This command is case insensitive.

**Examples** CH2:YUNit "V" sets the units for channel 2 to Volts.  
CH2:YUNIT might return CH2:YUNIT "V", indicating that the channel 2 units are volts.

## CLEARMenu

Clears the current menu from the display. This command is equivalent to pressing the front panel Menu off. No query form.

**Group**      Miscellaneous

**Syntax**     CLEARMenu

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

## \*CLS

Command only, no query form. The \*CLS command clears the following instrument status data structures:

- The Event Queue
- The Standard Event Status Register (SESR)
- The Status Byte Register (except the MAV bit)

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

\*CLS does not clear the output queue or MAV. \*CLS can suppress a service request that is to be generated by an \*OPC command. This will happen if a hard copy output or single sequence acquisition operation is still being processed when the \*CLS command is executed. See [Registers](#) on page 275

**Group**      Status and Error



**Syntax**     \*CLS

**Related Commands**     [DESE](#) on page 94, [\\*ESE](#) on page 106, [\\*ESR?](#) on page 107, [EVENT?](#) on page 108, [EVMsg?](#) on page 108, [\\*SRE](#) on page 217, [\\*STB?](#) on page 218

**Examples**     \*CLS clears the instrument status data structures.

## CURSor?

Returns current cursor settings. Query only.

**Group**     Cursor

**Syntax**     CURSor?

**Returns**     instrument cursor settings.

**Examples**     CURSor? might return the following string as the current cursor settings: :CURSOR:FUNCTION SCREEN;HBARS:POSITION1 0.0000;POSITION2 0.0000;UNITS BASE;:CURSOR:MODE INDEPENDENT;VBARS:POSITION1 -19.0006E-6;POSITION2 -18.9994E-6;UNITS SECONDS.

## CURSor:ENABLE

Allows educators to disable or enable the Cursor functions. The function can be manually set from the Utility menu. To access the menu, refer to the product user manual.

**Group**      Miscellaneous

**Syntax**      CURSor:ENABLE {ON | OFF}  
CURSor:ENABLE?

**Related commands**      [CURSor](#)  
[FEAEN:PASSWORD](#) on page 112

**Arguments**      ON enables the cursor feature.  
OFF disables the cursor feature.

**Examples**      CURSOR:ENABLE OFF disables cursor.  
CURSOR:ENABLE? might return 1 indicating that cursor is enabled.

## CURSor:FUNcTion

Sets or queries the instrument cursor type. Cursors are attached to the selected waveform in Waveform mode and are attached to the display area in Screen mode.

**Group** Cursor

**Syntax** CURSor:FUNcTion{OFF|SCREEN|WAVEform|VBArS|HBArS}  
CURSor:FUNcTion?

**Arguments** OFF removes the cursors from the display but does not change the cursor type.

SCREEN specifies both horizontal and vertical bar cursors, which measure the selected waveform in horizontal and vertical units. Use these cursors to measure anywhere in the waveform display area.

WAVEform specifies paired cursors in YT display format for measuring waveform amplitude and time. In XY and XYZ format, these cursors indicate the amplitude positions of an XY pair (Ch1 vs Ch2 voltage, where Ch1 is the X axis and Ch2 is the Y axis) relative to the trigger.

VBArS specifies the vertical bar cursor to measure the selected waveform in vertical units.

HBArS specifies the horizontal bar cursor to measure the selected waveform in horizontal units.

**Examples** CURSOR:FUNCTION WAVEFORM selects the paired cursors for measuring waveform amplitude and time.

CURSOR:FUNCTION? might return :CURSor:FUNcTion SCREEN indicating that the screen cursors are currently selected.

## CURSor:HBArS?

Returns the settings for the instrument horizontal bar cursors. Query only.

**Group** Cursor

**Syntax** CURSor:HBArS?

**Returns** Current horizontal bar cursor settings.

**Examples** CURSor:HBArS? might return the horizontal bar setting as return the horizontal bar setting as CURSOR:HBARS:DELTA 0.0E+0;POSITION1 320.0000E+0;POSITION2 320.0000E+0;UNITS BASE.

## CURSor:HBArS:DELTA?

Returns the difference (in vertical units) between the two horizontal bar cursors in the instrument display. Query only.

**Group** Cursor

**Syntax** CURSor:HBArS:DELTA?

**Related commands** [\*CURSor:HBArS:UNITS\*](#) on page 78

**Returns** <NR3> is the difference between the horizontal bar cursors.

**Examples**     `CURSOR:HBARS:DELTA?` might return `:CURSOR:HBARS:DELTA 5.0800E+00` indicating that the difference between the two cursors is 5.08.

## **CURSor:HBArS:POSITION<x>**

Sets or returns the horizontal bar cursor position relative to ground, which is expressed in vertical units (usually volts). The cursor is specified by x, which can be 1 or 2.

**Group**     Cursor

**Syntax**     `CURSor:HBArS:POSITION<x> <NR3>`  
`CURSor:HBArS:POSITION<x>?`

**Related commands**     [\*CURSor:FUNction\*](#) on page 75

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

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

---

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

---

**Examples**     `CURSOR:HBARS:POSITION1 25.0E-3` positions Cursor 1 of the horizontal cursors at 25 mV.

`CURSOR:HBARS:POSITION2?` might return `:CURSOR:HBARS:POSITION2 -64.0000E-03` indicating that Cursor 2 of the horizontal bar cursors is at -64 mV.

## CURSor:HBArS:UNIts

Sets or queries the vertical scale units for the selected cursor source waveform.

**Group** Cursor

**Syntax** CURSor:HBArS:UNIts {BASe|PERcent}  
CURSor:HBArS:UNIts?

**Arguments** BASe selects the vertical units for the selected waveform.  
PERcent selects ratio cursors.

**Examples** CURSor:HBArS:UNIts? might return :CURSOR:HBARS:UNITS BASE indicating that the units for the horizontal bar cursors are base.

## CURSor:HBArS:USE

Sets the horizontal bar cursor measurement scale. This command is only applicable when ratio cursors are on. No query form.

**Group** Cursor

**Syntax** CURSor:HBArS:USE {CURrent | HALFgrat | FIVEdivs}

**Related commands** [\*CURSor:HBArS:UNIts\*](#) on page 78

<b>Arguments</b>	<p>CURrent sets the H Bar measurement scale so that 0% is the current position of the lowest H Bar cursor and 100% is the current position of the highest H Bar cursor.</p> <p>HALFgrat resets the H bar measurement scale to half the number of divisions (five for some models and four for others) so that 25% is the current position of the lowest H Bar cursor and 75% is the current position of the highest H Bar.</p> <p>FIVEdivs sets H Bar measurement scale so that five screen major divisions is 100%, where 0% is -2.5 divisions and 100% is +2.5 divisions from the center horizontal graticule.</p>
<b>Examples</b>	<p>CURSOR:HBARS:USE FIVEDIVS sets the H Bar measurement scale so that 5 screen major divisions equals 100%.</p>

## CURSor:MODE

Sets or returns whether the two cursors move linked together in unison or separately. This applies to the Waveform cursors display mode.

<b>Conditions</b>	This command is only applicable when waveform cursors are displayed.
<b>Group</b>	Cursor
<b>Syntax</b>	<p>CURSor:MODE {TRACk INDependent}</p> <p>CURSor:MODE?</p>
<b>Arguments</b>	<p>TRACk ties the navigational functionality of the two cursors together. For cursor 1 adjustments, this ties the movement of the two cursors together; however, cursor 2 continues to move independently of cursor 1.</p> <p>INDependent allows independent adjustment of the two cursors.</p>

**Examples**     `CURSOR:MODE TRACK` specifies that the cursor positions move in unison.  
`CURSOR:MODE?` might return `:CURSOR:MODE TRACK` indicating that the two cursors move in unison.

## **CURSor:VBArS?**

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

**Group**     Cursor

**Syntax**     `CURSor:VBArS?`

**Examples**     `CURSor:VBArS?` might return `CURSOR:VBARS:POSITION1 -7.680E-6;POSITION2 7.680E-6;DELTA 15.3600E-6;HPOS1 0.0E+0;HPOS2 0.0E+0;UNITS SECONDS;VDELTA 0.0E+0`.

## **CURSor:VBArS:ALTERNATE<x>?**

Returns the alternate readout for the waveform (Vbar) cursors specified by <x>.  
This alternate readout is in effect for a bus waveform. Query only.

**Group**     Cursor

**Syntax**     `CURSor:VBArS:ALTERNATE<x>?`



<b>Arguments</b>	X = 1 specifies vertical bar cursor 1. X = 2 specifies vertical bar cursor 2.
<b>Examples</b>	CURSor:VBArS:ALTErNATE1? might return 1.001 indicating the vertical bar cursor 1 readout is 1.001.

## CURSor:VBArS:DELTA?

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

---

**NOTE.** *If Trigger View is active, this query returns 9.9E37 and generates event 221 (Settings conflict).*

---

<b>Group</b>	Cursor
<b>Syntax</b>	CURSor:VBArS:DELTA?
<b>Returns</b>	<NR3>
<b>Examples</b>	CURSor:VBArS:DELTA? might return 8.92E-1, indicating that the time difference between the vertical bar cursors is 0.892 seconds.

## CURSor:VBArS:HPOS<x>?

Returns the horizontal value of the specified vertical bar ticks for cursor <x>. The units are specified by the CURSor:HBArS:UNIts query. <x> specifies the cursor. Valid values are 1 and 2. Query only.

**Group** Cursor

**Syntax** CURSor:VBArS:HPOS<x>?

**Related Commands** [CURSor:HBArS:UNIts](#) on page 78

**Returns** <x> indicates the cursor. Valid values are 1 and 2.

**Examples** CURSOR:VBARS:HPOS1? might return CURSOR:VBARS:HPOS2 100E-3, indicating the value of one vertical bar tick.

## CURSor:VBArS:POSITION<x>

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

---

**NOTE.** *If Trigger View is active, the query form returns 9.9E37 and generates event 221 (Settings conflict).*

---

<b>Group</b>	Cursor
<b>Syntax</b>	CURSor:VBArS:POSITION<x> CURSor:VBArS:POSITION<x>?
<b>Arguments</b>	<p>&lt;x&gt; specifies which cursor to position. Correct values are 1 and 2.</p> <p>&lt;NR3&gt; specifies the cursor position in the units specified by the CURSor:VBArS:UNIts command. The position is relative to the trigger except when the cursor source is a math FFT waveform. The cursor position is limited to the graticule whenever an attempt is made to move it outside the graticule.</p>
<b>Examples</b>	<p>CURSOR:VBARS:POSITION2 9.00E-6 positions the second vertical bar cursor at 9ms.</p> <p>CURSOR:VBARS:POSITION1? might return 1.00E-6, indicating the first vertical bar cursor is at 1 <math>\mu</math>s.</p>

## CURSor:VBArS:UNIts

Sets or queries the units for the vertical bar cursors.

---

**NOTE.** When Trigger View is active, CURSor:VBArS:UNIts? generates event 221 (Settings conflict).

---

<b>Group</b>	Cursor
<b>Syntax</b>	CURSor:VBArS:UNIts CURSor:VBArS:UNIts?

<b>Arguments</b>	SECOnds specifies units of time. HERtz specifies units of frequency (reciprocal of time).
<b>Examples</b>	CURSor:VBArS:UNItsSECONDS sets the units for the vertical bar cursors to seconds. CURSor:VBArS:UNIts? returns HERTZ when the vertical bar cursor units are Hertz.

## CURSor:VBArS:VDELTA?

Returns the vertical (amplitude) difference between the two vertical bar cursors. The units are specified by the CURSor:HBArS:UNits query. Query only.

<b>Group</b>	Cursor
<b>Syntax</b>	CURSor:VBArS:VDELTA?
<b>Related commands</b>	<a href="#"><i>CURSor:HBArS:UNits</i></a> on page 78
<b>Returns</b>	<NR3> indicates the vertical difference between the two vertical bar cursors.
<b>Examples</b>	CURSor:VBArS:VDELTA? might return :CURSOR:VBARS:VDELTA 1.064E+0, indicating that the vertical difference between the vertical bar cursors is 1.064 units.

## CURVe

Transfers instrument waveform data to and from the instrument in binary or ASCII format. Each waveform that is transferred has an associated waveform preamble that contains information such as data format, scale, and associated information.

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

---

**NOTE.** *If the waveform specified by the DATA:SOURce command is not displayed, the CURVe? query returns nothing, and generates events 2244 (Waveform requested is not activated) and 420 (Query UNTERMINATED).*

---

The instrument returns data from the last acquisition if the source is a channel waveform that is being previewed. The data does not reflect the acquisition preview parameters. You should always follow acquisition parameter changes with a single sequence OPC command prior to CURVe? to ensure the return data reflects the new acquisition parameters.

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

Refer to *Waveform Commands* for a description of the waveform transfer process. [Waveform command group](#) on page 30

**Group**     Waveform

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

**Related Commands**    *DATA* on page 87, *DATA:START* on page 90, *DATA:STOP* on page 91, *WFMInpre?* on page 248, *WFMInpre:BYT\_Nr* on page 250, *WFMOutpre?* on page 258, *HEADer* on page 131

**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 samples are one or two bytes wide. Use the WFMInpre:BYT\_Nr command to set the width for waveforms transferred into the instrument. Use WFMOutpre:BYT\_Nr to set the width for waveforms transferred out from the instrument.

<data> is the curve data.

<newline> is a single byte new line character at the end of the data.

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

**Examples**    CURVe? with ASCII encoding, start and stop of 1 and 10 respectively, and a width set to 1 might return the following ASCII data:

:CURVE 61,62,61,60,60,-59,-59,-58,-58,-59.

---

## D commands

This section lists commands and queries that begin with the letter D.

### DATa

Sets or queries the format and location of the waveform data that is transferred with the CURVe command.

**Group**      Waveform

**Syntax**     DATa {INIT|SNAp}  
DATa?

**Related Commands**    [CURVe](#) on page 85, [DATa:START](#) on page 90, [DATa:STOP](#) on page 91, [WFMImpre:NR\\_Pt?](#) on page 251, [WFMOupre:NR\\_Pt?](#) on page 262

**Arguments**          INIT reinitializes the waveform data settings to their factory defaults except for DATa:STOP, which is set to the current acquisition record length.  
  
SNAp sets DATa:START and DATa:STOP to match the current waveform cursor positions.

**Examples**            DATaINIT initializes the waveform data settings to their factory defaults.  
  
DATa? might return :DATA:DESTINATION REF1:ENCDG  
RIBINARY;SOURCE CH1;START 1;STOP 500;WIDTH 1.

## DATA:DESTination

Sets or queries the reference memory location for storing waveform data that is transferred into the instrument by the CURVe command.

**Group**      Waveform

**Syntax**     DATA:DESTination REF<x>  
DATA:DESTination?

**Related Commands**    [CURVe](#) on page 85

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

**Examples**      DATA:DESTination REF1 stores incoming waveform data into reference memory 1.  
  
DATA:DESTination? might return :DATA:DESTINATION REF2 indicating that reference 2 is the currently selected reference memory location for incoming waveform data.



## DATA:SOURce

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

**Group**     Waveform

**Syntax**     DATA:SOURce{CH1|CH2| MATH|REF1|REF2}  
DATA:SOURce?

**Related Commands**     [CURVe](#) on page 85

**Arguments**     CH1–CH2 specifies which analog channel data will be transferred from the instrument to the controller, channels 1 through 2.

MATH specifies that the math waveform data will be transferred from the instrument to the controller.

REF1–REF2 specifies which reference waveform data will be transferred from the instrument to the controller, waveforms, 1 or 2.

**Examples**     DATA:SOURCE CH1 specifies that the channel 1 waveform will be transferred in the next CURVe? query.

DATA:SOURceREF1 specifies that reference waveform REF1 will be transferred in the next CURVe? query.

DATA:SOURce? might return :DATA:SOURCE REF2 indicating that the source for the waveform data which is transferred using a CURVe? query is reference 2.

## DATA:START

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

**Group**      Waveform

**Syntax**     DATA:START <NR1>  
DATA:START?

**Related Commands**    [CURVe](#) on page 85, [DATA](#) on page 87, [DATA:STOP](#) on page 91, [WFMInpre:NR\\_Pt?](#) on page 251, [WFMOutpre:NR\\_Pt?](#) on page 262

**Arguments**    <NR1> is the first data point that will be transferred, which ranges from 1 to the record length. Data will be transferred from <NR1> to DATA:STOP or the record length, whichever is less. If <NR1> is greater than the record length, the last data point in the record is transferred. DATA:START and DATA:STOP are order independent. When DATA:STOP is greater than DATA:START, the values will be swapped internally for the CURVE? query.

**Examples**      DATA:START10 specifies that the waveform transfer will begin with data point 10.  
  
DATA:START? might return :DATA:START 214 indicating that data point 214 is the first waveform data point that will be transferred.

## DATA:STOP

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

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

When using the CURVe command, the instrument stops reading data when there is no more data to read.

**Group**      Waveform

**Syntax**     DATA:STOP <NR1>  
DATA:STOP?

**Related Commands**    [CURVe](#) on page 85, [DATA](#) on page 87, [DATA:STOP](#) on page 91, [WFMinpre:NR\\_Pt?](#) on page 251, [WFMOupre:NR\\_Pt?](#) on page 262

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

DATA:START and DATA:STOP are order independent. When DATA:STOP is less than DATA:START, the values will be swapped internally for the CURVe? query.

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

- Examples**     DATA:STOP15000 specifies that the waveform transfer will stop at data point 15000.
- DATA:STOP? might return :DATA:STOP 14900 indicating that 14900 is the last waveform data point that will be transferred.

## DATA:WIDTH

Sets or queries the number of bytes per data point in the waveform transferred using the CURVe command.

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

**Group**     Waveform

**Syntax**     DATA:WIDTH <NR1>  
DATA:WIDTH?

**Related Commands**     [CURVe](#) on page 85

**Arguments**     <NR1> = 1 sets the number of bytes per waveform data point to 1 byte (8 bits).  
<NR1> = 2 sets the number of bytes per waveform data point to 2 bytes (16 bits).  
If DATA:WIDTH is set to 2, the least significant byte is always zero. This format is useful for AVErage waveforms.

**Examples**     DATA:WIDTH1 sets the data width to 1 byte per data point for CURVe data.

## DATE

Sets or queries the instrument date value. The instrument uses these values to time stamp files saved to the USB flash drive, as well as show the time and date on the instrument display.

**Group**     Miscellaneous

**Syntax**     DATE  
DATE?

**Related Commands**     [TIme](#) on page 219

**Arguments**     <QString> is a date in the form "yyyy-mm-dd".

**Examples**     DATE"2010-05-06" sets the date to May 6th, 2010.  
DATE? might return :DATE 2015-10-29 indicating that the current date is set to Oct. 29, 2015.

## DESE

Sets or 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 detailed discussion of the use of these registers, see Registers.

**Group** Status and Error

**Syntax** DESE <NR1>  
DESE?

**Related Commands** [\\*CLS](#) on page 72, [\\*ESE](#) on page 106, [\\*ESR?](#) on page 107, [EVENT?](#) on page 108, [EVMsg?](#) on page 108, [\\*SRE](#) on page 217, [\\*STB?](#) on page 218

**Arguments** <NR1> is an integer value in the range from 0 to 255. The binary bits of DESER are set according to this value. For example, DESE 209 sets the DESER to the binary value 11010001 (that is, the most significant bit in the register is set to 1, the next most significant bit to 1, the next bit to 0, and so on).

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

---

**NOTE.** Setting DESER and ESER to the same value allows only those codes to be entered into the Event Queue and summarized on the ESB bit (bit 5) of the Status Byte Register. Use the \*ESE command to set ESER. For more information on event handling, refer to the Status and Events section.

---

**Examples** DESE209 sets the DESER to binary 11010001, which enables the PON, URQ, EXE, and OPC bits.

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

## DIAg:FAN

Returns the currently set PWM fan value. Only query.

**Group** Calibration and diagnostic

**Syntax** DIAg:FAN?

**Examples** DIAg:FAN? might return PWM=10, VOL=7.4V.

## DIAg:LOOP:OPTion

Sets the self-test loop option.

**Group** Calibration and diagnostic

**Syntax** DIAg:LOOP:OPTion {ALWAYS|FAIL|ONFAIL|ONCE|NTIMES}

**Arguments** ALWAYS continues looping until the self tests (diagnostics) are stopped via the front panel or by an instrument command.

FAIL causes looping until the first self test (diagnostic) failure or until self tests (diagnostics) are stopped.

ONFAIL causes looping on a specific test group as long as a FAIL status is returned from the test.

ONCE executes self test (diagnostics test) sequence once.

NTIMES runs “n” number of loops.

**Examples**     `DIAG:LOOP:OPTion ONCE` sets diagnostics to run one loop of self tests.

## **DIAG:LOOP:OPTion:NTIMes**

Sets the self-test loop option to run N times.

**Group**     Calibration and diagnostic

**Syntax**     `DIAG:LOOP:OPTion:NTIMes <NR1>`  
`DIAG:LOOP:OPTion:NTIMes?`

**Arguments**     <NR1> is the number of self-test loops.

**Examples**     `DIAG:LOOP:OPTION:NTIMES 3` sets the self-test loop to run three times.  
`DIAG:LOOP:OPTION:NTIMES?` might  
return `:DIAG:LOOP:OPTION:NTIMES 5`, indicating that the self-test loop is set  
to run five times.

## **DIAG:LOOP:STOP**

Stops the self-test at the end of the current loop. No query form.

**Group**     Calibration and diagnostic

**Syntax**     `DIAG:LOOP:STOP`



**Examples**     `DIAG:LOOP:STOP` stops the self test at the end of the current loop.

## **DIAG:RESULT:FLAG?**

Returns the Pass/Fail status from the last diagnostic test sequence execution (those run automatically at power on, or those requested through the Service Menu). Use the `DIAG:RESULT:LOG?` query to determine which test(s) has failed. Query only.

**Group**     Calibration and Diagnostic

**Syntax**     `DIAG:RESULT:FLAG?`

**Returns**     `PASS` means that the instrument passes all selected diagnostic tests.  
`FAIL` means that the instrument has failed at least one of the diagnostic tests.

**Examples**     `DIAG:RESULT:FLAG`  
Returns either `PASS` or `FAIL`.

## **DIAG:RESULT:LOG?**

Returns the internal results log from the last diagnostic test sequence execution (those run automatically at power on, or those requested through the Service Menu). The list contains all modules and module interfaces that were tested with the pass or fail status of each. Query only.

**Group**     Calibration and Diagnostic

**Syntax**     `DIAG:RESUlt:LOG?`

**Returns**     `<QString>` in the following format:  
`<Status>,<Module name>[,<Status>,<Module name>...]`

**Examples**     `DIAG:RESUlt:LOG?` might return `:DIAG:RESULT:LOG "NOT RUN--CPU,NOT RUN--DISPLAY,NOT RUN--FPANEL,NOT RUN--IO,NOT RUN--ACQ,NOT RUN--ROM,NOT RUN--APPKEY"` for power-up diagnostics.

## DIAG:SElect

Sets the type of diagnostics grouping. No query form.

**Group**     Calibration and diagnostic

**Syntax**     `DIAG:SElect {ALL|APPKey|CPU|DISplay|FPAnel|IO|ROM|ACQ}`

**Arguments**     ALL runs all diagnostic groups.  
CPU runs just the CPU diagnostic group.  
DISplay runs just the display circuit diagnostic group.  
FPAnel runs just the front panel diagnostic group.  
IO runs just the IO board diagnostic group.  
ROM runs just the IO board diagnostic group.  
ACQ runs just the acquisition system diagnostic group.

**Examples**     DIAg:SElect ALL runs all diagnostic groups.

## DIAg:SElect:<function>

Runs self-tests on the specified system subsystem. No query form.

**Group**     Calibration and diagnostic

**Syntax**     DIAg:SElect:<function>

**Arguments**     <function> specifies a single instrument subsystem on which to run self tests (diagnostics). Valid values are:

- ACQ tests the acquisition system.
- CPU tests the CPU.
- DISplay tests the display.
- FPAnel tests the front panel controls.
- IO tests the IO ports.
- ROM tests the system read only memory.

**Examples**     DIAg:SElect:ACQ specifies to run self tests on the acquisition system.

## DIAg:STATE

This command starts or stops the instrument self-test. Depending on the argument, self-test capabilities are either turned on or off. No query form.

**Group**     Calibration and diagnostic

**Syntax**     `DIAG:STATE {EXECute|ABORt}`

**Arguments**     EXECute starts diagnostics.  
                    ABORt stops diagnostics at the end of the current loop.

**Examples**     `DIAG:STATE EXECute` starts diagnostics.

## DIAG:TEMPVAL

Read out the currently FPGA chip and ambient temperature. Only query.

**Group**     Calibration and diagnostic

**Syntax**     `DIAG:TEMPVAL?`

**Examples**     `DIAG:TEMPVAL?` might return: VDC Temp=-256, Ambient Temp=32

## DISplay:GRAticule

Sets and returns the display graticule intensity settings.

**Group**     Miscellaneous

**Syntax**     `DISplay:GRAticule {<NR1>|ON|OFF}`  
                    `DISplay:GRAticule?`

- Arguments** ON or <NR1>  $\neq 0$  turns on the graticule in the screen display.  
OFF or <NR1> = 0 turns off the graticule in the screen display.
- Examples** DISPLAY:GRATICULE 0 sets NO graticule to display.  
DISPLAY:GRATICULE? might return :DISPLAY:GRATICULE 1 indicating that the graticule is on.

## DISplay:INTENSITY:BACKLight

Sets and returns the waveform backlight intensity settings.

- Group** Miscellaneous
- Syntax** DISplay:INTENSITY:BACKLight <NR1>  
DISplay:INTENSITY:BACKLight ? <NR1>
- Arguments** <NR1> specifies the range from 10 to 100.
- Examples** DISplay:INTENSITY:BACKLight <NR1>  
DISPLAY:INTENSITY:BACKLIGHT? might  
return :DISPLAY:INTENSITY:BACKLIGHT 60

## DISplay:PERStence:STATe

Sets or returns the display persistence to ON or OFF. The query form returns the persistence state. This affects the display only.

**Group** Display

**Syntax** DISplay:PERStence:STATe {ON | OFF | <NR1>}  
DISplay:PERStence:STATe?

**Arguments** ON turns on the persistence on for the display.  
OFF turns the persistence off for the display.  
<NR1>=0 turns off the persistence for the display; any other value turns on the persistence for the display.

**Examples** DISplay:PERStence:STATe ON turns on the persistence on the display for all waveforms.  
DISplay:PERStence:STATe? might return 1 indicating that the persistence is ON for the display.

## DISplay:PERStence:VALUe

Sets the value of persistence if the persistence state is set to on. This affects the display only.

**Group** Display

**Syntax**     DISplay:PERStence:VALUe {<NR3> | AUTO | INFInite}  
              DISplay:PERStence:VALUe?

**Arguments**     <NR3> specifies the time of the persistence.

INFInite displays waveform points until a control change resets the acquisition system. When persistence is set to infinite, it does not mean that the brightness of any pixel should never decrease. The brightness of a pixel is proportionally dependent on the ratio between the intensity, which does NOT decrease at infinite persistence, and the maximum value of intensity of any pixel on the screen. Thus, if a particular pixel gets hit less often than others, its brightness will decrease over time. It will become less bright relative to the pixels that get hit often. AUTO specifies that the oscilloscope automatically determines the best waveform persistence based on the value of waveform intensity (DISPLAY:INTENSITY:WAVEFORM)

**Examples**     DISplay:PERStence:VALUe 3 specifies that the waveform points are displayed fading for 3 seconds before they completely disappear.





---

## E commands

This section lists commands and queries that begin with the letter E.

### ERRLOG:FIRST?

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

**Group** Calibration and Diagnostic

**Syntax** ERRLOG:FIRST?

**Returns** Refer to the service manual for your instrument for information about error log message format.

### ERRLOG:NEXT?

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

**Group** Calibration and Diagnostic

**Syntax** ERRLOG:NEXT?

**Returns** Refer to the service manual for your instrument for information about error log message format.

## \*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 detailed discussion on how to use registers, see Registers. Command only, no query form.

**Group** Status and Error

**Syntax** \*ESE <NR1>  
\*ESE?

**Related Commands** [\\*CLS](#) on page 72, [DESE](#) on page 94, [\\*ESR?](#) on page 107, [EVENT?](#) on page 108, [EVMsg?](#) on page 108, [\\*SRE](#) on page 217, [\\*STB?](#) on page 218

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

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

---

**NOTE.** Setting the DESER and the ESER to the same value allows only those codes to be entered into the Event Queue and summarized on the ESB bit (bit 5) of the Status Byte Register. Use the DESE command to set the DESER. See [Event Handling Sequence](#) on page 280.

---

**Examples**     \*ESE209 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?

Returns the contents of the Standard Event Status Register (SESR). \*ESR? also clears the SESR (since reading the SESR clears it). For a detailed discussion on how to use registers, see Registers. Query only.

**Group**        Status and Error

**Syntax**       \*ESR?

**Related Commands**     [ALLEv?](#) on page 48, [\\*CLS](#) on page 72, [DESE](#) on page 94, [\\*ESE](#) on page 106, [EVENT?](#) on page 108, [EVMsg?](#) on page 108, [\\*OPC](#) on page 191, [\\*SRE](#) on page 217, [\\*STB?](#) on page 218

**Returns**       Contents of the Standard Event Status Register.

**Examples**       \*ESR? might return the value 213, showing that the SESR contains binary 11010101.

## EVENT?

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. Query only.

**Group**      Status and Error

**Syntax**     *EVENT?*

**Related Commands**    *ALLEv?* on page 48, *\*CLS* on page 72, *DESE* on page 94, *\*ESE* on page 106, *\*ESR?* on page 107, *EVMsg?* on page 108, *\*SRE* on page 217, *\*STB?* on page 218

**Returns**      <NR1> the last *\*ESR*.

**Examples**      *EVENT?* might return *EVENT 110*, indicating there was an error in a command header.

## EVMsg?

Removes from the Event Queue a single event code associated with the results of the last *\*ESR?* read, and returns the event code with an explanatory message. Query only.

**Group**      Status and Error

**Syntax** EVMsg?

**Related Commands** [ALLEv?](#) on page 48, [\\*CLS](#) on page 72, [DESE](#) on page 94, [\\*ESE](#) on page 106, [\\*ESR?](#) on page 107, [EVENT?](#) on page 108, [\\*SRE](#) on page 217, [\\*STB?](#) on page 218

**Returns** The event code and message in the following format:  
 <Event Code><Comma><QString>[<Event Code><Comma> <QString>...]  
 <QString>::= <Message>;[<Command>]  
 where <Command> is the command that caused the error and may be returned when a command error is detected by the instrument. As much of the command as possible is returned without exceeding the 60 character limit of the <Message> and <Command> strings combined. The command string is right-justified.

**Examples** EVMsg? might return the message EVMSG 110, "Command header error"

## EVQty?

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

**Group** Status and Error

**Syntax** EVQty?

**Related Commands** [ALLEv?](#) on page 48, [EVENT?](#) on page 108, [EVMsg?](#) on page 108

**Returns** <NR1> is the number of event codes in the Event Queue.

**Examples** EVQty? might return :EVQTY 3 indicating the number of event codes in the Event Queue is 3.



---

## F commands

This section lists commands and queries that begin with the letter F.

### FACTORy

Resets the instrument to its factory default settings. Refer to Appendix B: Factory Setup for a list of the factory default settings. No query.

This command does the following:

- Clears the Event Status Enable Register
- Clears the Service Request Enable Register
- Sets the Device Event Status Enable Register to 255
- Purges all defined aliases
- Enables all Command Headers
- Sets the macro defined by \*DDT to a "zero-length field
- Clears the pending operation flag and associated operations

This command does not reset the following:

- Communication settings
- State of the VXI-11 (Ethernet IEEE Std 488.2) interface
- Calibration data that affects device specifications
- Protected user data
- Stored settings
- Power On Status Clear Flag
- instrument password

**Group** Save and Recall

**Syntax** FACTORY

**Related Commands**    [\\*PSC](#) on page 193, [\\*RCL](#) on page 195, [RECall:SETUp](#) on page 196, [\\*RST](#) on page 202, [\\*SAV](#) on page 203, [SAVe:SETUp](#) on page 207, [SAVe:IMAge:FILEFormat](#) on page 205

**Examples**    FACTORY resets the instrument to its factory default settings. Refer to *Factory Setup*.

## FEAEN:PASSWORD

Checks if the password entered is equal to the preset password for Feature Enable which allows the educators to enable or disable features such as Autoset, Cursors or Measurements. The same can be set manually from the Utility menu when we try to change the settings of Autoset Enable, Cursors Enable or Measurement Enable from Feature Enable. No query form. To access the menu, refer product user manual.

**Group**    Miscellaneous

**Syntax**    FEAEN:PASSWORD <password>

**Related Commands**    [AUTOSet:ENABLE](#) on page 49  
[CURSor:ENABLE](#) on page 74  
[MEASUrement:ENABLE](#)

**Arguments**    <password> is the feature enable password, enclosed in quotes.

**Examples**    FEAEN:PASSWORD "111111" allows the user to change the settings of Feature Enable if the feature enable password matches the string "111111".



## FFT?

Returns all FFT parameters. Query only.

**Group** FFT

**Syntax** FFT?

**Related commands** [FFT:VERTical:SCale](#) on page 116, [FFT:VERTical:POSition](#) on page 116, [FFT:VERTical:UNIts](#) on page 117, [FFT:HORizontal:SCale](#) on page 114, [FFT:HORizontal:POSition](#) on page 113, [FFT:SOURce](#) on page 114, [FFT:SRCWFM](#) on page 115, [FFT:WINDow](#) on page 118, [SElect:FFT](#) on page 212

**Examples** FFT? might return 1; CH1; 20; 0.000; "dB"; 250.000E+3; 750.000E+3; "Hz"; 1; HANNING; X1; DB

## FFT:HORizontal:POSition

Sets or queries the FFT horizontal display position.

**Group** FFT

**Syntax** FFT:HORizontal:POSition <NR3>  
FFT:HORizontal:POSition?

**Arguments** <NR3> is the FFT horizontal display position.

**Examples**    FFT:HORizontal:POSition 750.0E+3 sets the FFT horizontal position to 750.0E+3.  
FFT:HORizontal:POSition? might return 750.000E+3.

## FFT:HORizontal:SCAle

Sets or queries the horizontal scale of the FFT waveform.

**Group**    FFT

**Syntax**    FFT:HORizontal:SCAle <NR3>  
FFT:HORizontal:SCAle?

**Arguments**    <NR3> is the FFT horizontal scale.

**Examples**    FFT:HORizontal:SCAle 500.00E+6 sets the FFT horizontal scale to 500 MHz.  
FFT:HORizontal:SCAle? might return 500.00E+6 indicating the FFT horizontal scale is set to 500 MHz.

## FFT:SOURce

Sets or queries the source of the FFT waveform.

**Group**    FFT

<b>Syntax</b>	FFT:SOURce {CH1 CH2} FFT:SOURce?
<b>Arguments</b>	{CH1 CH2} the FFT source channel.
<b>Examples</b>	FFT:SOURce ch2 sets the FFT source waveform to CH2. FFT:SOURce? might return "CH2" if CH2 is the FFT source waveform.

## FFT:SRCWFM

Sets or queries the FFT source waveform display state.

<b>Group</b>	FFT
<b>Syntax</b>	FFT:SRCWFM <ON OFF NR1> FFT:SRCWFM?
<b>Arguments</b>	<NR1> = 0 does not display the FFT source waveform, any other value displays the FFT source waveform.
<b>Examples</b>	FFT:SRCWFM 0 turns off the display of the FFT source waveform. FFT:SRCWFM? might return 1 indicating the FFT source waveform is displayed.

## FFT:VERTical:POSition

Sets or queries the FFT vertical display position.

**Group**     FFT

**Syntax**     FFT:VERTical:POSition <NR2>  
FFT:VERTical:POSition?

**Arguments**     <NR2> is the FFT vertical position.

**Examples**     FFT:VERTical:POSition 2 sets the FFT vertical position to 2 divisions above center screen.  
FFT:VERTical:POSition? might return 2.000.

## FFT:VERTical:SCAle

Sets or queries the FFT vertical zoom factor.

**Group**     FFT

**Syntax**     FFT:VERTical:SCAle <NR2>  
FFT:VERTical:SCAle?

**Arguments**     <NR2> is the FFT vertical scale.

**Examples**    FFT:VERTical:SCAle 20 sets the FFT waveform vertical scale to 20.  
FFT:VERTical:SCAle? might return 20.00 indicating the FFT waveform vertical scale is 20 dB.

## FFT:VERTical:UNIts

Queries the FFT vertical measurement units label.

**Group**    FFT

**Syntax**    FFT:VERTical:UNIts?

**Examples**    FFT:VERTical:UNIts? might return dB indicating the FFT vertical units are set to dB.

## FFT:VType

Sets or queries the FFT waveform vertical units.

**Group**    FFT

**Syntax**    FFT:VType {DB|LINEAr}  
FFT:VType?

**Examples**    FFT:VType DB sets the FFT waveform vertical units to dB.  
FFT:VType? might return DB.

## FFT:WINDow

Sets or queries the FFT window type.

**Group**     FFT

**Syntax**     FFT:WINDow {HAMming|HANning|RECTangular|BLAckmanharris}  
FFT:WINDow?

**Arguments**     RECTangular window function is equivalent to multiplying all gate data by one.  
HAMming window function is based on a cosine series.  
HANning window function is based on a cosine series.  
BLAckmanharris window function is based on a cosine series.

**Examples**     FFT:WINDow HAMMING sets the FFT window to Hamming.  
FFT:WINDow? might return HAMMING.

## FILESystem?

Returns the current working directory and amount of free space. This query is the same as the FILESystem:DIR? query and the FILESystem:FREESpace? query.  
Query only.

.

**Group**     File system

**Syntax**     FILESystem  
              FILESystem?

**Related commands**     *FILESystem:CWD* on page 119, *FILESystem:DELEte* on page 120,  
                             *FILESystem:DIR?* on page 121, *FILESystem:REName* on page 124

**Examples**     FILESYSTEM? might return :FILESYSTEM:DIR  
                     "TEK00000.BMP","GLITCH1.PNG","TEMP.TMP",  
                     "FILE1.WFM","FILE2.WFM","MATH1.WFM","REF1.WFM","REF2.WFM".

## FILESystem:CWD

Sets or queries the current working directory (CWD) for FILESystem commands.

The default working directory is USB0. Anytime you use this command to change the directory, the directory that you specify is retained as the current working directory until you either change the directory or you delete the directory. If you delete the current working directory, the instrument resets current working directory to the default directory (USB0) the next time the instrument is powered on or the next time you execute a file system command.

This command supports the permutations of file and directory names supported by Microsoft Windows:

Relative path names; for example, ".temp"

Absolute path names; for example, " USB0/Wfms"

Implied relative path names; for example "NEWFILE.TXT" becomes " USB0/TEKSCOPE/NEWFILE.TXT" if the current working directory is " USB0/TEKSCOPE"

**Group**     File system

**Syntax**     FILESystem:CWD {<new working directory path>}  
FILESystem:CWD?

**Arguments**     <new working directory path> is a quoted string that defines the current working; a directory name can have up to 8 characters with an extension of up to 3 characters.

**Examples**     FILESYSTEM:CWD " USB0/TEKSCOPE/IMAGES" sets the current working directory to images.  
  
FILESYSTEM:CWD? might return :FILESYSTEM:CWD " USB0/TEKSCOPE/WAVEFORMS" indicating that the current working directory is set to waveforms.

## FILESystem:DELEte

This command deletes a named file. If you specify a directory name, it will delete the directory and all of its contents, the same as the RMDir command. You can also specify the filename as \*.\* to delete all of the files in the current or specified directory. Command only, no query form.

**Group**     File system

**Syntax**     FILESystem:DELEte <file path>

**Related commands**     [FILESystem:CWD](#) on page 119, [FILESystem:RMDir](#) on page 125

**Arguments**     <file path> is a quoted string that defines the folder path and file name of the file to delete. If the file path is within the current working directory, you need only specify the file name. The argument \*.\* will delete all files and subdirectories within the current working directory.



**Examples**     FILESYSTEM:DELETE "NOT\_MINE.SET" deletes the file named NOT\_MINE.SET from the current working directory.

## FILESystem:DIR?

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

**Group**     File system

**Syntax**     FILESystem:DIR?

**Related commands**     [FILESystem:CWD](#) on page 119, [FILESystem:RMDir](#) on page 125

**Returns**     FILESystem:DIR? returns a list of files and directories in the current working directory.

**Examples**     FILESystem:DIR? might return :FILESYSTEM:DIR  
"TEK00000.PNG", "CANSETUP.SET", "WFM1.ISF", "MYIMAGES".

## FILESystem:FORMat

Formats a mass storage device. This command should be used with extreme caution as it causes all data on the specified mass storage device to be lost. Drive letters (such as USB0) are case sensitive and must be upper case. For all other FILESYSTEM commands, drives letters are not case sensitive. Example: FILESystem:FORMAT " USB0/" Formats the USB flash drive installed in the instrument's front panel USB port. Command only, no query form.

**Group** File system

**Syntax** FILESystem:FORMat <drive>

**Arguments** <drive> is a quoted string that sets the drive to format.

**Examples** FILESystem:FORMat"/usb0/" formats the USB flash drive installed in the instrument front panel USB port.

## FILESystem:FREESpace?

Returns a numeric value, in bytes, of the memory space available on the current drive. Query only.

**Group** File system

**Syntax** FILESystem:FREESpace?

**Related commands**    [FILESystem:CWD](#) on page 119

**Examples**    FILESystem:FREEspace? might return 6242501.

## FILESystem:MKDir

Creates a folder at the specified location. Command only, no query form.

**Group**    File system

**Syntax**    FILESystem:MKDir <directory path>

**Related commands**    [FILESystem:CWD](#) on page 119, [FILESystem:DIR?](#) on page 121

**Arguments**    <directory path> is a quoted string that defines the location and name of the directory to create. If you do not specify a path to the directory, the instrument creates the directory in the current working directory. The current directory refers to the name of a directory as returned by the FILESystem:CWD query.

Directory names must follow the same rules as file names. [File System Conventions](#) on page 18

**Examples**    FILESYSTEM:MKDIR " USB0/NewDirectory" creates the directory named NEWDIRECTORY at the root of the E drive.

The following two commands create the directory MYNEWSUBDIRECTORY within the existing directory mydirectory at the root of the USB0 drive:  
FILESYSTEM:CWD " USB0/MyDirectory";FILESYSTEM:MKDIR "MyNewSubDirectory" This assumes that USB0/MYDIRECTORY already existed and was not a read-only directory.

## FILESystem:READFile

Writes the contents of the specified file to the specified interface. If the file does not exist or is not readable, an appropriate error event is posted. No query form.

**Group** File System

**Syntax** FILESystem:READFile <QString>

**Related commands** [FILESystem:CWD](#) on page 119

**Arguments** <QString> is a quoted string that defines the file name and path. If the file path is within the current working directory, specify only the file name.

**Examples** FILESYSTEM:READFILE "USB0/TEST\_DATA/TEK00016CH1.CSV" reads the content of the specified file, if the file exists and is readable, and sends the content of the file to the current interface.

## FILESystem:REName

Assigns a new name to an existing file or folder. You can also move a file or folder by specifying the new name in a different folder. Command only, no query form.

For file and folder name rules, see *File System Conventions*. [File System Conventions](#) on page 18

**Group** File system

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

**Related commands**     [FILESystem:CWD](#) on page 119

**Arguments**     <old filepath> is a quoted string that defines the path and name of the file to rename. If you do not specify a path to the file, the instrument looks for the file in the current working folder. The current directory refers to the name of a folder as returned by the FILESystem:CWD query.

                     <new filepath> is a quoted string that defines the path and new name of the file. If you do not specify a path to a folder, the instrument places the renamed file into the current working folder. [File System Conventions](#) on page 18

**Examples**     FILESYSTEM:RENAME " USB0/TEK00000.SET","D:/MYSETTING.SET"  
gives the file named TEK00000.SET the new name of MYSETTING.SET. The file remains in the root directory on the D drive.

## FILESystem:RMDir

Deletes a named directory. This command deletes the specified directory and all of its contents. The directory must not be a read-only directory. Command only, no query form.

**Group**     File system

**Syntax**     FILESystem:RMDir <directory path>

**Arguments**     <directory path> is a quoted string that defines the location and name of the directory to delete. If you do not specify a path to the folder, the instrument deletes the specified folder in the current working folder. The current folder refers to the name of a folder as returned by the FILESystem:CWD query.

---

**NOTE.** *A folder must be empty before you can delete it.*

---

**Examples**     FILESYSTEM:RMDIR " USB0/OldDirectory" removes the directory named olddirectory from the root of the E drive.

## FILESystem:WRITEFile

Writes the specified block data to a file in the instrument current working directory. If the specified file does not exist or is not readable, an appropriate error event is posted. The maximum length of the block data is 262144 bytes. No query form.

**Group**     File System

**Syntax**     FILESystem:WRITEFile <file path>, <data>

**Related commands**     [FILESystem:CWD](#) on page 119

**Arguments**     <file path> is the quoted string that defines the file name and path. If the path is within the current working directory, specify the file name only.  
                     <data> can be either DEFINITE LENGTH encoding or INDEFINITE LENGTH ARBITRARY BLOCK PROGRAM DATA encoding as described in IEEE488.2.

## FPAnel:PRESS

Simulates the action of pressing a specified front-panel button. No query form.

When the front panel is locked, the front-panel buttons and multipurpose knob operations are suspended. The FPAnel:PRESS and the FPAnel:TURN commands will also not work. You can work around this by using the appropriate programmatic interface commands, instead of the front-panel commands.

**Group** Miscellaneous

**Syntax** FPAnel:PRESS <button>

**Arguments** <button> is the name of a front-panel button. Most of the argument names associate directly with their front panel buttons. For example, AUTOSet is for the Autoset button. The <button> enumeration arguments and their associations with the front panel buttons are listed below.

Argument	Button
ACQuire	Acquire button
SAVE Recall	Save/Recall Menu button
MEASurement	Measure button
UTILity	Utility button
MATh	M button
REF	R button
FFT	F button
TRIGger	Trigger Menu button
FORCetrig	Force Trig button
CH1	Channel1 select button
CH2	Channel2 select button
DEFaultsetup	Default Setup button
FUNCTion	Function button
Help	Help button
ZOOM	Zoom button
FINE	Fine button
CURsor	Cursors button
RUNstop	Run/Stop button
SINGLEseq	Single button
AUTOset	Autoset button
SETTO50	Trigger level knob can be pressed to Set Trigger to 50%

Argument	Button
Save	Save button
RMENU1	Screen top-most side menu button
RMENU2	Screen side menu button
RMENU3	Screen side menu button
RMENU4	Screen side menu button
RMENU5	Screen bottom-most side menu button
MENUOff	Menu On/Off button
GPKNOB	Multipurpose knob can be pressed for selection.
HORZPos	Horizontal Position knob can be pressed to set horizontal position to center.

**Examples**     FPANEL:PRESS AUTOSET executes the instrument Autoset function.

## FPAnel:TURN

Simulates the action of turning a specified front-panel control knob. No query form.

When the front panel is locked, the front-panel button and multipurpose knob operations are suspended. The FPAnel:PRESS and FPAnel:TURN commands will also not work, and they will not generate an error. You can work around this by using the appropriate programmatic interface commands, instead of the front-panel commands. For example, to set the trigger level to 50%, you could use TRIGger:A SETLevel. To force a trigger, you could use TRIGger FORCe.

**Group**     Miscellaneous

**Syntax**     FPAnel:TURN <knob>,[<n>]



**Arguments** <knob> is the name of a rotating control. A comma (,) separates the control knob argument from the numeric optional rotation value argument. In the absence of the numeric rotation value argument, the default is 1 (clockwise). You do not need a white space between the arguments and the comma. <n> represents the rotation direction and magnitude of rotation. Negative values represent a counterclockwise knob rotation, and positive values represent a clockwise rotation. The magnitude of <n> specifies the amount of the turn, where <n> = 1 represents turning the knob one unit, <n> = 2 represents turning the knob two units, <n> = 4 represents turning the knob four units, and so on. The range of units depends on which front panel knob is specified.

**Table 27: FPanel:TURN arguments**

Argument	Knob
GPKNOB	Multipurpose knob
HORZPos	Horizontal Position knob
HORZScale	Horizontal Scale knob
TRIGLevel	Trigger Level knob
VERTPOS<n>	Vertical Position knob
VERTSCALE<n>	Vertical Scale knob

**Examples** FPanel:TURN TRIGLEVEL,10 duplicates turning the front panel Trigger Level knob clockwise by 10 units.

## FWUpdate:Update

Updates the oscilloscope firmware from a file on a USB flash drive. Before executing this command, make sure the USB flash drive is plugged into the instrument, and contains the firmware update file **TBS1000C.TEK** at the root (top) directory. If the update file is not in the root directory, the oscilloscope shows a warning message and the firmware is not updated.

**Group** Miscellaneous

**Syntax** FWUpdate:Update



---

## H commands

This section lists commands and queries that begin with the letter H.

### HDR

This command is identical to the HEADer query and is included for compatibility with other Tektronix scopes.

**Group**     Miscellaneous

**Syntax**     HDR  
                 HDR?

### HEADer

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

**Group**     Miscellaneous

**Syntax**     HEADer  
                 HEADer?

<b>Arguments</b>	ON or <NR1> ≠ 0 sets the Response Header Enable State to true. This causes the instrument to include header on applicable query responses. You can then use the query response as a command.  OFF or <NR1> = 0 sets the Response Header Enable State to false. This causes the instrument to omit headers on query responses so that only the argument is returned.
<b>Examples</b>	HEADer OFF causes the instrument to omit headers from query responses.  HEADer? might return 1, showing that the Response Header Enable State is true. Query only.

## HELPevery:ACQuire

Enables or disables the display of Help Everywhere information for the acquire measurements.

<b>Group</b>	Help everywhere
<b>Syntax</b>	HELPevery:ACQuire {ON OFF} HELPevery:ACQuire?
<b>Arguments</b>	ON enables displaying Help Everywhere for the acquire measurements. OFF disables displaying Help Everywhere for the acquire measurements.
<b>Examples</b>	HELPevery:ACQuire ON enables Help Everywhere for the acquire settings.

## HELPevery:ALL

Enables or disables the display of Help Everywhere information for all measurement settings (acquire, trigger, vertical, math, fft ,cursor, reference, measurement, and utility modules).

**Group** Help everywhere

**Syntax** HELPevery:ALL {ON|OFF}

**Arguments** ON enables Help Everywhere.  
OFF disables Help Everywhere.

**Examples** HELPevery:ALL ON enables Help Everywhere.

## HELPevery:CURsor

Enables or disables the display of Help Everywhere information for the cursor module.

**Group** Help everywhere

**Syntax** HELPevery:CURsor {ON|OFF}  
HELPevery:CURsor?

**Arguments** ON enables Help Everywhere for the cursor settings.  
OFF disables Help Everywhere for the cursor settings.

**Examples**    HELPevery:CURSor ON enables Help Everywhere for the cursor settings.

## HELPevery:FFT

Enables or disables the display of Help Everywhere information for the fft settings.

**Group**    Help everywhere

**Syntax**    HELPevery:FFT {ON|OFF}  
HELPevery:FFT?

**Arguments**    ON enables Help Everywhere for the FFT module.  
OFF disables Help Everywhere for the FFT module.

**Examples**    HELPevery:FFT ON enables Help Everywhere for the FFT module.

## HELPevery:MATH

Enables or disables the display of Help Everywhere information for the math module.

**Group**    Help everywhere

**Syntax**    HELPevery:MATH {ON|OFF}  
HELPevery:MATH?

**Arguments**    ON enables Help Everywhere for the math module.  
                  OFF disables Help Everywhere for the math module..

**Examples**    HELPevery:MATH ON enables Help Everywhere for the math module.

## HELPevery:MEASUrement

Enables or disables the display of Help Everywhere information for the measurement module.

**Group**        Help everywhere

**Syntax**       HELPevery:MEASUrement {ON|OFF}  
                  HELPevery:MEASUrement?

**Arguments**    ON enables Help Everywhere for the measurement module.  
                  OFF disables Help Everywhere for the measurement module..

**Examples**    HELPevery:MEASUrement ON enables Help Everywhere for the measurement module.

## HELPevery:REFerence

Enables or disables the display of Help Everywhere information for the reference module.

**Group** Help everywhere

**Syntax** HELPevery:REFerence {ON|OFF}  
HELPevery:REFerence?

**Arguments** ON enables Help Everywhere for the reference module.  
OFF disables Help Everywhere for the reference module.

**Examples** HELPevery:REFerence ON enables Help Everywhere for the reference module.

## HELPevery:TRIGger

Enables or disables the display of Help Everywhere information for the trigger module.

**Group** Help everywhere

**Syntax** HELPevery:TRIGger {ON | OFF}  
HELPevery:TRIGger?



**Arguments**    ON enables Help Everywhere for the trigger module.  
                  OFF disables Help Everywhere for the trigger module.

**Examples**    HELPevery:TRIGger ON enables Help Everywhere for the trigger module.

## HELPevery:UTility

Enables or disables the display of Help Everywhere information for the utility module.

**Group**        Help everywhere

**Syntax**       HELPevery:UTility {ON|OFF}  
                  HELPevery:UTility?

**Arguments**    ON enables Help Everywhere for the utility module.  
                  OFF disables Help Everywhere for the utility module.

**Examples**    HELPevery:UTility ON enables Help Everywhere for the utility module.

## HELPevery:VERTical

Enables or disables the display of Help Everywhere information for the vertical module.

**Group** Help everywhere

**Syntax** HELPevery:VERTical {ON|OFF}  
HELPevery:VERTical?

**Arguments** ON enables Help Everywhere for the vertical module.  
OFF disables Help Everywhere for the vertical module.

**Examples** HELPevery:VERTical ON enables Help Everywhere for the vertical module.

## HORizontal?

Returns all settings for the horizontal commands. Query only.

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

**Group** Horizontal

<b>Syntax</b>	HORizontal?
<b>Returns</b>	Returns all horizontal settings.
<b>Examples</b>	HORIZONTAL? might return the following horizontal settings :HORIZONTAL:POSITION 50.0000; SAMPLERATE 500.0000E+6; SCALE 200.0000E-9; RECORDLENGTH 2000; RECORDLENGTH : AUTO 0; DELAY:MODE 1; TIME 0.0E+0.

## HORizontal:ACQLENGTH

Queries the record length. Query only.

<b>Group</b>	Horizontal
--------------	------------

<b>Syntax</b>	HORizontal:ACQLENGTH?
---------------	-----------------------

<b>Related commands</b>	<a href="#"><i>HORizontal:RECOrdlength</i></a> on page 147
-------------------------	--

<b>Examples</b>	HORIZONTAL:ACQLENGTH? might return 20000 indicating that the record length is 20k points.
-----------------	---

## HORizontal:DIVisions

Returns the current horizontal divisions: 15.0000. Query only.

**Group** Horizontal

**Syntax** HORizontal:DIVisions?

**Examples** HORizontal:DIVisions? might return 15.

## HORizontal[:MAIn][:DELay]:POSition

Sets or queries the horizontal position. If Horizontal Delay Mode is turned off, this command is equivalent to adjusting the HORIZONTAL POSITION knob on the front panel. When Horizontal Delay Mode is on, this command stores a new horizontal position that is used when Horizontal Delay Mode is turned off.

**Group** Horizontal

**Syntax** HORizontal:MAIn:DELay:POSition <NR1>  
HORizontal:MAIn:DELay:POSition?  
HORizontal:MAIn:POSition <NR1>  
HORizontal:MAIn:POSition?  
HORizontal:DELay:POSition <NR1>  
HORizontal:DELay:POSition?  
HORizontal:POSition <NR1>  
HORizontal:POSition?

<b>Arguments</b>	<NR1> is the horizontal position expressed as the percentage of the waveform displayed left of the center of the graticule.
<b>Examples</b>	<p>HORIZONTAL:DElay:POSITION 50 sets the horizontal position to 50%.</p> <p>HORIZONTAL:MAIn:DElay:POSITION? might return 100 indicating that the horizontal position is set to 100%.</p>

## HORizontal:MAIn:DElay:MODE

Sets or returns the horizontal delay mode.

<b>Group</b>	Horizontal
<b>Syntax</b>	<p>HORizontal:MAIn:DElay:MODE {OFF ON &lt;NR1&gt;}</p> <p>HORizontal:MAIn:DElay:MODE?</p> <p>HORizontal:DElay:MODE {OFF ON &lt;NR1&gt;}</p> <p>HORizontal:DElay:MODE?</p>

**Related Commands**    [HORizontal\[:MAIn\]\[:DElay\]:POSition](#) on page 140

<b>Arguments</b>	<p>OFF sets the Horizontal Delay Mode to off. This causes the HORizontal:POSition command to horizontally position the waveform.</p> <p>ON sets the Horizontal Delay Mode to on. This causes the HORizontal:DElay:TIME command to horizontally position the waveform.</p> <p>&lt;NR1&gt; = 0 sets the Horizontal Delay Mode to off; any other value sets this mode to on.</p>
------------------	---

- Examples**     `HORIZONTAL:DELAY:MODE OFF` sets the Horizontal Delay Mode to off, allowing the `HORizontal:POSition` command to horizontally position the waveform.
- `HORIZONTAL:MAIn:DELAY:MODE?` might return OFF indicating that the `HORizontal:POSition` command horizontally positions the waveform.

## **HORizontal:MAIn:DELay:STATe**

Sets or returns the horizontal delay state . The same as `HORizontal[:MAIn]:DELay:MODE`.

**Group**     Horizontal

**Syntax**     `HORizontal:MAIn:DELay:STATe {OFF|ON|<NR1>}`  
`HORizontal:MAIn:DELay:STATe?`  
`HORizontal:DELay:STATe {OFF|ON|<NR1>}`  
`HORizontal:DELay:STATe?`

**Related Commands**     [\*HORizontal\[:MAIn\]\[:DELay\]:POSition\*](#) on page 140

**Arguments**     OFF sets the Horizontal Delay State to off. This causes the `HORizontal:POSition` command to horizontally position the waveform.

ON sets the Horizontal Delay State to on. This causes the command to horizontally position the waveform.

<NR1> = 0 sets the Horizontal Delay State to off; any other value sets this mode to on.

- Examples**     `HORIZONTAL:DELAY:STATE OFF` sets the Horizontal Delay State to off, allowing the `HORizontal:POSition` command to horizontally position the waveform.
- `HORIZONTAL:MAIN:DELAY:STATE?` might return `OFF` indicating that the `HORizontal:POSition` command horizontally positions the waveform.

## **HORizontal[:MAIn]:DELay:TIME**

Sets or queries the horizontal delay time. The amount of time the acquisition is delayed depends on sample rate and record length.

**Group**     Horizontal

**Syntax**     `HORizontal:MAIn:DELay:TIME <NR3>`  
`HORizontal:MAIn:DELay:TIME?`  
`HORizontal:DELay:TIME <NR3>`  
`HORizontal:DELay:TIME?`

**Arguments**     `<NR3>` is the delay in seconds.

**Examples**     `HORizontal:DELay:TIME 0.3` sets the delay of acquisition data so that the resulting waveform is centered 300 ms after the trigger occurs.

## HORizontal[:MAIn]:SAMPLERate

Returns the current horizontal sample rate. Query only.

**Group** Horizontal

**Syntax** HORizontal:SAMPLERate?  
HORizontal [:MAIn] :SAMPLERate?

**Examples** HORizontal:SAMPLERate? might return 2.0000E+9.

## HORizontal[:MAIn]:SCAle

Sets or queries the time base horizontal scale. Query only.

**Group** Horizontal

**Syntax** HORizontal:SCAle <NR3>  
HORizontal:SCAle?  
HORizontal:MAIn:SCAle <NR3>  
HORizontal:MAIn:SCAle?

**Arguments** <NR3> specifies the range from 2 ns to 100 s, depending on the model.

**Returns** All settings for the time base



**Examples**     HORizontal:MAIn:SCAle 2E-6 sets the main scale to 2  $\mu$ s per division.  
HORizontal:MAIn:SCAle? might return 2.0000E-06, indicating that the main scale is currently set to 2  $\mu$ s 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 scopes.

**Group**     Horizontal

**Syntax**     HORizontal: SECdiv <NR3>  
HORizontal: SECdiv ?  
HORizontal:MAIn:SECdiv <NR3>  
HORizontal:MAIn:SECdiv?

**Arguments**     <NR3> specifies the range from 2 ns to 100 s, depending on the model.

**Examples**     HORizontal:MAIn:SECdiv 2E-6 sets the main scale to 2  $\mu$ s per division.  
HORizontal:MAIn:SECdiv? might return 2.0000E-06 indicating that the main scale is currently set to 2  $\mu$ s per division.

## HORizontal:MAIn:UNIts[:STRing]

Returns the current horizontal unit "s". Query only.

**Group** Horizontal

**Syntax** HORizontal:MAIn:UNIts?  
HORizontal:MAIn:UNIts:STRing?

**Examples** HORizontal:MAIn:UNIts? might return SECONDS.

## HORizontal:PREViewstate

Returns a boolean value to indicate whether the acquisition system is in the preview state. Query only.

**Group** Horizontal

**Syntax** HORizontal:PREViewstate?

**Returns** <NR1> = 1 if the acquisition system is in the preview state.  
<NR1> = 0 if the acquisition system is not in the preview state.

**Examples** HORizontal:PREViewstate? might return 1 indicating the acquisition system is in the preview state.

## HORizontal:RECOrdlength

Sets the horizontal record length of acquired waveforms. The query form of this command returns the current horizontal record length.

**Group** Horizontal

**Syntax** HORizontal:RECOrdlength <NR1>  
HORizontal:RECOrdlength?

**Arguments** <NR1> represents the supported values for horizontal record lengths, which are:  
1000, 2000, 20000

**Examples** HORIZONTAL:RECORDLENGTH 2000 specifies that 2000 data points will be acquired for each record.  
HORIZONTAL:RECORDLENGTH? might return 2000 indicating that the horizontal record length is equal to 2000 data points.

## HORizontal:RECOrdlength:Auto

Sets or queries the horizontal record length mode.

**Group** Horizontal

**Syntax** HORizontal:RECOrdlength:Auto <NR1>  
HORizontal:RECOrdlength:Auto?

<b>Arguments</b>	<NR1> sets the record length mode. 1 enables auto record length mode, 0 disables auto record length mode.
<b>Examples</b>	<p>HORizontal:RECOrdlength:Auto 1 enables auto record length mode of the analog channels.</p> <p>HORizontal:RECOrdlength:Auto? might return 0 indicating that auto record length mode of the analog channels is off.</p>

## HORizontal:RESOlution

Sets or returns the horizontal record length of acquired waveforms. The sample rate is automatically adjusted at the same time to maintain a constant time per division. The query form of this command returns the current horizontal record length.

<b>Group</b>	Horizontal
<b>Syntax</b>	<p>HORizontal:RESOlution &lt;NR1&gt;</p> <p>HORizontal:RESOlution?</p>
<b>Arguments</b>	<NR1> represents the supported values for horizontal record lengths.
<b>Examples</b>	HORizontal:RESOlution 20000 set the record length to 20000 points.

## HORizontal:ROLL

Returns the current horizontal roll mode state: on/off. Query only.

**Group** Horizontal

**Syntax** HORizontal:ROLL?

**Examples** HORizontal:ROLL? might return HORizontal:ROLL ON indicating that roll mode is on.

## HORizontal:TRIGger:POSition

Sets or queries the horizontal position when delay mode is OFF. It is similar to HORizontal:POSition.

Sets the

**Group** Horizontal

**Syntax** HORizontal:TRIGger:POSition <NR3>  
HORizontal:TRIGger:POSition?

**Arguments** <NR3> is the horizontal position expressed as the percentage of the waveform displayed left of the center of the graticule.

**Examples** HORizontal:MAIn:POSition50 sets the horizontal position to 50%.  
HORizontal:MAIn:POSition? might return 100, indicating that the horizontal position is set to 100%.



# I commands

This section lists commands and queries that begin with the letter I.

## ID?

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

---

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

---

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

**Group**    Miscellaneous

**Syntax**    ID?

**Returns**    Returns the instrument identification in the following format for TBS1000C instruments:  
ID TEK/<model number>,CF:91.1CT FV:v<instrument firmware version number>

**Examples**    ID? might return the following response ID TEK/TBS2104,CF:91.1CT,FV:v2015-12-10\_01-00-59rootfs; FPGA:v1.21;

**\*IDN?**

Returns the instrument identification code in IEEE 488.2 notation. Query only.

---

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

---

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

**Group**      Miscellaneous

**Syntax**     \*IDN?

**Returns**     Returns the instrument identification in the following format for TBS1000C instruments:  
  
TEKTRONIX,<model number>,CF:91.1CT FV:v<instrument firmware version number> TBS1XXXC:v<module firmware version number>

**Examples**    \*IDN? might return the following response for a TBS1072C instrument with the serial number CU10100: TEKTRONIX,TBS1072C,CU10100,CF:91.1CT FV:v2015-12-10\_01-00-59rootfs; FPGA:v1.21;



---

## L commands

This section lists commands and queries that begin with the letter L.

### LANGuage

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

**Group**      Miscellaneous

**Syntax**      LANGuage  
                 LANGuage?

**Arguments**      Specifies the language used to display instrument information on the screen.

**Examples**      LANGuageFRENch specifies that the instrument displays information in French.  
                 LANGuage? might return SPANISH.

## LOCK

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

When the front panel is locked, neither the FPanel:PRESS nor the FPanel:TURN commands work. They will not generate an error event either. You can work around this by using the appropriate programmatic interface commands, instead of the front-panel commands. For example, to set the trigger level to 50%, you could use TRIGger:A SETLevel. To force a trigger, you could use TRIGger FORCe.

**Group** Miscellaneous

**Syntax** LOCK {ALL|NONE}  
LOCK?

**Related commands** [UNLock](#) on page 243

**Arguments** ALL disables all front-panel controls.  
NONE enables all front-panel controls. This is equivalent to the UNLock ALL command.

**Examples** LOCK ALL locks the front-panel controls.  
LOCK? might return :LOCK NONE indicating the front-panel controls are enabled by this command.

## \*LRN?

This is identical to the query. Query only.

Miscellaneous

**Group** Miscellaneous

**Syntax** \*LRN?

---

## M commands

This section lists commands and queries that begin with the letter M.

### MATH?

Returns the current math parameters. Query only.

**Group** Math

**Syntax** MATH?

**Related commands** [MATH:DEFINE](#) on page 156, [MATH:VERTical:SCAle](#) on page 160, [MATH:VERTical:POSition](#) on page 159, [MATH:VERTical:UNIts](#) on page 161, [MATH:HORizontal:SCAle](#) on page 157, [MATH:HORizontal:POSition](#) on page 157, [MATH:HORizontal:UNIts](#) on page 158

**Returns** Returns the current math parameters:

- The definition of the math waveform:
- Source1 operation source2
- Vertical scale
- Vertical position
- Vertical units
- Horizontal scale
- Horizontal position
- Horizontal units

**Examples** MATH? might return “CH1+CH2”;2.000;0.0E+0”V”,20.0000E-6;0.0E0;”s”.

## MATH:DEFINE

Sets or returns the math waveform definition for the active math operation.

---

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

---

**Group** Math

**Syntax** MATH:DEFINE <QString>  
MATH:DEFINE?

**Arguments** <QString> specifies a math waveform, and can be one of the following,  
CH1+CH2, CH1-CH2, CH2-CH1,  
CH1\*CH2

**Examples** MATH:DEFINE"CH1-CH2" sets the math waveform so that it displays the difference of channel 1 and channel 2.  
MATH:DEFine? Might return "CH1-CH2".

## MATH:HORizontal:POSition

Sets or queries the math horizontal display position for math waveforms. The horizontal position of a dual math waveform with a channel waveform source is set through the commands described in the horizontal section.

<b>Group</b>	Math
<b>Syntax</b>	MATH:HORizontal:POSition <NR2> MATH:HORizontal:POSition?
<b>Arguments</b>	<NR2> is the math horizontal position in percent of record.
<b>Examples</b>	MATH:HORizontal:POSition 20 sets the math horizontal position to 20% of the record.  MATH:HORizontal:POSition? might return 20.000 indicating that the math horizontal position is at 20% of the record.

## MATH:HORizontal:SCALe

Sets or queries the math horizontal display scale for dual math waveforms that only have source waveforms. The horizontal scale of a dual math waveform with a channel source waveform is set through the HORIZONTAL:SCALE command.

<b>Group</b>	Math
<b>Syntax</b>	MATH:HORizontal:SCALe <NR3> MATH:HORizontal:SCALe?

**Arguments** <NR3> is the math display scale.

**Examples** MATH:HORizontal:SCALe 20.000E-6 sets the math horizontal display scale to 20  $\mu$ s per division.  
MATH:HORizontal:SCALe? might return 20.000E-6 indicating the math horizontal display scale is 20  $\mu$ s per division.

## MATH:HORizontal:UNIts

Queries the math horizontal measurement units label.

**Group** Math

**Syntax** MATH:HORizontal:UNIts?

**Arguments** <Qstring> is a quoted string representing the math horizontal units.

**Examples** MATH:HORizontal:UNIts? might return “us” indicating the math horizontal units are  $\mu$ s.

## MATH:LABel

This command sets or queries the waveform label for the math waveform.

**Group** Math

**Syntax** MATH:LABel <Qstring>  
MATH:LABel?

**Arguments** <Qstring> is the quoted string used as the label for the math waveform.

**Examples** MATH:LABEL Output sets the label for the math waveform to "Output."  
MATH:LABEL? might return MATH:LABEL "Sum of channel 1 and channel 2"  
indicating the current label for the math waveform.

## MATH:VERTical:POSition

Sets or queries the math waveform display position.

**Group** Math

**Syntax** MATH:VERTical:POSition <NR3>  
MATH:VERTical:POSition?

**Arguments** <NR3> specifies the math vertical position in divisions from center screen.

- Examples** MATH:VERTical:POSition 4 sets the math vertical position to 4 divisions above center screen.
- MATH:VERTical:POSition? might return -3.000, indicating that the math waveform is 3 divisions below center screen.

## MATH:VERTical:SCAle

Sets or queries the vertical display scale, which should not be confused with the math waveform vertical scale returned in the math waveform pre-amble (MATH?). The display scale is the same as that adjusted through the instrument vertical scale knob that controls waveform zoom factors. The math waveform scale is not affected by this control, rather the math calculation software automatically determines the optimum vertical scale through examination of input waveform data.

---

**NOTE.** *The vertical display scale is reset to the waveform pre-amble scale whenever a vertical scale change to a math source waveform results in a new math autoscale operation. The vertical display scale should be changed only after math source waveform adjustments are complete.*

---

- Group** Math
- Syntax** MATH:VERTical:SCAle <NR3>  
MATH:VERTical:SCAle?
- Arguments** <NR3> specifies the math vertical scale in units per division.
- Examples** MATH:VERTical:SCAle5.0E0 sets the math vertical scale to five math waveform units per division.
- MATH:VERTical:SCAle? might return 5.000.



## MATH:VERTical:UNIts

Queries the math vertical measurement units.

**Group** Math

**Syntax** MATH:VERTical:UNIts?

**Examples** MATH:VERTical:UNIts? might return “V”.

## MEASUrement?

Returns the current MEASUrement settings. Query only.

**Group** Measurement

**Syntax** MEASUrement?

**Returns** Instrument measurement settings.

**Examples** MEASUrement? might return the following:

```
:MEASUREMENT:IMMED:DELAY:DIRECTION FORWARDS;EDGE1  
RISE;EDGE2 RISE;;MEASUREMENT:IMMED:TYPE PERIOD;UNITS  
"s";SOURCE1 CH1;SOURCE2  
CH2;;MEASUREMENT:MEAS1:DELAY:DIRECTION FORWARDS;EDGE1  
RISE;EDGE2 RISE;;MEASUREMENT:MEAS1:STATE 1;TYPE  
FREQUENCY;UNITS "Hz";SOURCE1 CH1;SOURCE2 CH2;COUNT  
0;MAXIMUM 0.0000;MEAN 0.0000;MINIMUM 0.0 000;STDDEV  
0.0000;;MEASUREMENT:MEAS2:DELAY:DIRECTION  
FORWARDS;EDGE1 RISE;EDGE2 RISE;;MEASUREMENT:MEAS2:STATE  
1;TYPE PERIOD;UNITS "s";SOURCE1 CH1;SOURCE2 CH2;COUNT  
0;MAXIMUM 0.0000;MEAN 0.0000;MINIMUM 0.0000;STDDEV  
0.0000;;MEASUREMENT:MEAS3: DELAY:DIRECTION  
FORWARDS;EDGE1 RISE;EDGE2 RISE;;MEASUREMENT:MEAS3:STATE  
1;TYPE PK2PK;UNITS "V";SOURCE1 CH1;SOURCE2 CH2;COUNT  
0;MAXIMUM 0.0000;MEAN 0.0000;MINIMUM 0.0000;STDDEV  
0.0000;;MEASUREMENT:MEAS4:DELAY:DIRECTION  
FORWARDS;EDGE1 RISE;EDGE2 RISE;;MEASUREMENT:MEAS4:STATE  
0;TYPE PERIOD;UNITS "s";SOURCE1 CH1;SOURCE2 CH2;COUNT  
0;MAXIMUM 0.0000;MEAN 0.0000;MINIMUM 0.0000;STDDEV  
0.0000;;MEASUREMENT:METHOD AUTO;REFLEVEL:METHOD  
PERCENT;ABSOLUTE:HIGH 0.0000;LOW 0.0000;MID1 0.0000;MID2  
0.0000;;MEASUREMENT:REFLEVEL:PERCENT:HIGH 90.0000;LOW  
10.0000;MID1 50.0000;MID2  
50.0000;;MEASUREMENT:INDICATORS:STATE OFF;NUMHORZ  
0;NUMVERT 0;HORZ1 99.0000E +36;HORZ2 99.0000E+36;HORZ3 99.0000E  
+36;HORZ4 99.0000E+36;VERT1 99.0000E+36;VERT2 99.0000E+36;VERT3  
99.0000E+36;VERT4 99.0000E+36;;MEASUREMENT:STATISTICS:MODE  
OFF;WEIGHTING 32;;MEASUREMENT:GATING SCREEN.
```

## MEASUrement:CLEARSNapshot

Clears the existing measurement snapshot results and removes the snapshot window. Command only, no query form.

**Group** Measurement

**Syntax** MEASUrement:CLEARSNapshot

**Examples** MEASUrement:CLEARSNapshot clears the existing snapshot results and removes the snapshot window.

## MEASUrement:ENABLE

Allows educators to disable or enable the Measurement functions. The function can be manually set from the Utility menu. To access the menu, refer to the product user manual.

**Group** Miscellaneous

**Syntax** MEASUrement:ENABLE {ON|OFF}  
MEASUrement:ENABLE?

**Related Commands** [MEASUrement?](#) on page 161  
[FEAEN:PASSWORD](#) on page 112

<b>Arguments</b>	ON enables the measurement feature. OFF disables the measurement feature.
<b>Examples</b>	MEASUREMENT:ENABLE OFF disable measurement. MEASUREMENT:ENABLE? might return 1 indicating that measurements is enabled.

## MEASUREMENT:GATING

Sets or queries the measurement gating setting.

<b>Group</b>	Measurement
<b>Syntax</b>	MEASUREMENT:GATING {OFF SCREEN CURSOR} MEASUREMENT:GATING?
<b>Arguments</b>	OFF turns off measurement gating (full record). SCREEN turns on gating, using the left and right edges of the screen. CURSOR limits measurements to the portion of the waveform between the vertical bar cursors, even if they are off screen.
<b>Examples</b>	MEASUREMENT:GATING CURSOR turns on measurement gating using the cursors as limits. MEASUREMENT:GATING? might return :MEASUREMENT:GATING CURSOR indicating that measurements are limited to the portion of the waveform between the vertical bar cursors.

## MEASUrement:IMMed?

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

**Group** Measurement

**Syntax** MEASUrement:IMMed?

**Returns** Immediate measurement setup parameters

**Examples** MEASUrement:IMMed? might  
return :MEASUREMENT:IMMED:DELAY:DIRECTION  
FORWARDS;EDGE1 RISE;EDGE2 RISE;;MEASUREMENT:IMMED:TYPE  
PERIOD;UNITS "s";SOURCE1 CH1;SOURCE2 CH2.

## MEASUrement:IMMed:DELaY?

Returns information about the immediate delay measurement. This command is equivalent to viewing the delay measurement settings on the measurement readout. Query only.

**Group** Measurement

**Syntax** MEASUrement:IMMed:DELaY?

**Examples** MEASUREMENT:IMMED:DELAY? might  
return :MEASUREMENT:IMMED:DELAY:DIRECTION FORWARDS;  
EDGE1 RISE; EDGE2 RISE.

## MEASUrement:IMMed:DElAy:EDGE<x>

Sets or queries the slope of the edge the instrument uses for the delay from or to waveform when taking an immediate delay measurement.

**Group** Measurement

**Syntax** MEASUrement:IMMed:DElAy:EDGE<x> {FALL|RISe}  
MEASUrement:IMMed:DElAy:EDGE<x>?

**Arguments** <x> specifies which waveform to use, where <x> = 1 is the from waveform, and <x> = 2 is the to waveform.  
FALL specifies the falling edge.  
RISe specifies the rising edge.

**Examples** MEASUREMENT:IMMED:DELAY:EDGE1 RISE specifies that the from waveform rising edge be used for the immediate delay measurement.  
MEASUREMENT:IMMED:DELAY:EDGE1? returns either RISE or FALL.

## MEASUrement:IMMed:SOUrce1

Sets or queries the source for all single source immediate measurements and specifies the source to measure from when taking an immediate delay or phase measurement.

---

**NOTE.** *If you do not specify a numerical suffix, the source is assumed to be SOURCE 1.*

---

**Group** Measurement

**Syntax** MEASUrement:IMMed:SOUrce1 {CH<x> | MATH}  
MEASUrement:IMMed:SOUrce1?

**Related commands** [MEASUrement:IMMed:SOUrce2](#) on page 168

**Arguments** CH<x> specifies the measurement source channel as one of the input channels. The value of <x> can vary from 1 through 2.  
MATH specifies the measurement source channel as the math waveform.

**Examples** MEASUREMENT:IMMED:SOURCE1 CH1 specifies channel 1 as the immediate measurement source.  
MEASUREMENT:IMMED:SOURCE1? might return CH2 indicating that channel 2 is the immediate measurement source.

## MEASUrement:IMMed:SOUrce2

Sets or queries the secondary source for dual-source immediate measurements.

---

**NOTE.** *Source2 measurements only apply to phase and delay measurement types, which require both a target (Source1) and reference (Source2) source.*

---

**Group** Measurement

**Syntax** MEASUrement:IMMed:SOUrce2 {CH<x> | MATH}  
MEASUrement:IMMed:SOUrce2?

**Related commands** [MEASUrement:IMMed:SOUrce1](#) on page 167

**Arguments** CH<x> specifies the measurement source channel as one of the input channels. The value of <x> can vary from 1 through 2.  
MATH specifies the measurement source channel as the math waveform.

**Examples** MEASUrement:IMMed:SOUrce2 CH2 sets the immediate measurement source 2 to channel 2.  
MEASUrement:IMMed:SOUrce2? might return MATH indicating that Math is the immediate measurement source.



## MEASUrement:IMMed:TYPe

Sets or queries the immediate measurement type.

<b>Group</b>	Measurement
<b>Syntax</b>	<p>MEASUrement:IMMed:TYPe {AMPlitude AREa BURst CAREa CMEan CRMs DELay FALL FREQuency  HIGH LOW MAXimum MEAN MINImum NDUty NEDGECount NOVershoot  NPULSECount NWidth PEDGECount PDUty  PERIOD PHase PK2Pk POVershoot PPULSECount PWidth RISe RMS}</p> <p>MEASUrement:IMMed:TYPe?</p>
<b>Arguments</b>	<p>AMPlitude measures the amplitude of the selected waveform. It measures the high value less the low value measured over the entire waveform or gated region. Amplitude = High – Low</p> <p>AREa measures the voltage over time. The area is over the entire waveform or gated region and is measured in volt-seconds. The area measured above the ground is positive, while the area below ground is negative.</p> <p>BURst measures the duration of a burst. The measurement is made over the entire waveform or gated region.</p> <p>CAREa (cycle area) measures the voltage over time. It measures, in volt-seconds, the area over the first cycle in the waveform or the first cycle in the gated region. The area measured above the common reference point is positive, while the area below the common reference point is negative.</p> <p>CMEan (cycle mean) measures the arithmetic mean over the first cycle in the waveform or the first cycle in the gated region.</p> <p>CRMs (cycle rms) measures the true Root Mean Square voltage over the first cycle in the waveform or the first cycle in the gated region.</p> <p>DELay measures the time between the middle reference (default = 50%) amplitude point of the source waveform and the destination waveform.</p> <p>FALL measures the time taken for the falling edge of the first pulse in the waveform or gated region to fall from a high reference value (default is 90%) to a low reference value (default is 10%).</p> <p>FREQuency measures the first cycle in the waveform or gated region. Frequency is the reciprocal of the period and is measured in hertz (Hz), where 1 Hz = 1 cycle per second.</p> <p>HIGH measures the High reference (100% level, sometimes called Topline) of a waveform.</p> <p>LOW measures the Low reference (0% level, sometimes called Baseline) of a waveform.</p> <p>MAXimum finds the maximum amplitude. This value is the most positive peak voltage found. It is measured over the entire waveform or gated region.</p>

MEAN amplitude measurement finds the arithmetic mean over the entire waveform or gated region.

MINimum finds the minimum amplitude. This value is typically the most negative peak voltage. It is measured over the entire waveform or gated region.

NDUty (negative duty cycle) is the ratio of the negative pulse width to the signal period, expressed as a percentage. The duty cycle is measured on the first cycle in the waveform or gated region.  $\text{Negative Duty Cycle} = ((\text{Negative Width}) / \text{Period}) \times 100\%$

NEDGECount is the count of falling edges.

NOVershoot (negative overshoot) finds the negative overshoot value over the entire waveform or gated region.  $\text{Negative Overshoot} = ((\text{Low} - \text{Minimum}) / \text{Amplitude}) \times 100\%$

NPULSECount is the count of negative pulses.

NWidth (negative width) measurement is the distance (time) between the middle reference (default = 50%) amplitude points of a negative pulse.

PDUty (positive duty cycle) is the ratio of the positive pulse width to the signal period, expressed as a percentage. It is measured on the first cycle in the waveform or gated region.  $\text{Positive Duty Cycle} = ((\text{Positive Width} / \text{Period}) \times 100\%$

PEDGECount is the count of rising edges.

PERIOD is the time required to complete the first cycle in a waveform or gated region. Period is the reciprocal of frequency and is measured in seconds.

PHase measures the phase difference (amount of time a waveform leads or lags the reference waveform) between two waveforms. The measurement is made between the middle reference points of the two waveforms and is expressed in degrees, where  $360^\circ$  represents one waveform cycle.

PK2Pk (peak-to-peak) finds the absolute difference between the maximum and minimum amplitude in the entire waveform or gated region.

POVershoot is the positive overshoot value over the entire waveform or gated region.  $\text{Positive Overshoot} = ((\text{Maximum} - \text{High}) / \text{Amplitude}) \times 100\%$

PPULSECount is the count of positive pulses.

PWidth (positive width) is the distance (time) between the middle reference (default = 50%) amplitude points of a positive pulse. The measurement is made on the first pulse in the waveform or gated region.

RISe timing measurement finds the rise time of the waveform. The rise time is the time it takes for the leading edge of the first pulse encountered to rise from a low reference value (default is 10%) to a high reference value (default is 90%).

RMS amplitude measurement finds the true Root Mean Square voltage in the entire waveform.

- Examples**    MEASUrement:IMMed:TYPe FREQUENCY defines the immediate measurement to be a frequency measurement.
- MEASUREMENT:IMMED:TYPE? might return RMS indicating that the immediate measurement is the true Root Mean Square voltage.

## MEASUrement:IMMed:UNIts?

Returns the units for the immediate instrument measurement. Query only.

**Group**      Measurement

**Syntax**     MEASUrement:IMMed:UNIts?

**Returns**     <QString> where the string is one of the following: VOLTS, VOLTS SQUARED, SEC, HERTZ, PERCENT, DIVS, SAMPLES, OHMS, AMPS, WATTS, MINUTES, DEGREES, UNKNOWN, AMPS SQUARED, HOURS, DAYS, DB, BYTES, INVERSE HERTZ, IRE, V OVER V, V OVER A, VOLTS WATTS, V OVER W, VOLTS DB, V OVER DB, A OVER V, A OVER A, AMPS WATTS, A OVER W, AMPS DB, A OVER DB, WATTS VOLTS, W OVER V, WATTS AMPS, W OVER A, WATTS SQUARED, W OVER W, WATTS DB, W OVER DB, DB VOLTS, DB OVER V, DB AMPS, DB OVER A, DB WATTS, DB OVER W, DB SQUARED, DB OVER DB, VOLTS SEC, AMPS SEC, WATTS SEC, V OVER S, A OVER S, W OVER S.

**Examples**    MEASUrement:IMMed:UNIts? might return :MEASUREMENT:IMMED:UNIts "s", indicating that the unit for the immediate measurement is seconds.

## MEASUrement:IMMed:VALue?

Returns the value of the measurement specified by the MEASUrement:IMMed:TYPE command. The measurement is immediately taken on the source(s) specified by a MEASUrement:IMMed:SOURce1 command. Query only.

---

**NOTE.** *A change to HORizontal:MAIn:SCALe or CH<x>:SCALe will not necessarily have taken affect if immediately followed by this command.*

---

**Group** Measurement

**Syntax** MEASUrement:IMMed:VALue?

**Related Commands** [MEASUrement:IMMed:TYPE](#) on page 169,  
[\\*ESR?](#) on page 107, [EVENT?](#) on page 108

**Returns** <NR3>

**Examples** MEASUrement:IMMed:VALue? might return 28.75E6 if you are measuring the frequency of a 28.75 MHz signal.

MEASUrement:IMMed:VALue? might return 9.9000E+37. If the measurement has an error or warning associated with it, then an item is added to the error queue. The error can be checked for with the \*ESR? and ALLEv? commands..

## MEASUrement:MEAS<x>?

Returns all measurement parameters for the displayed instrument periodic measurement specified by <x>. Where <x> identifies the measurement, 1 through 6 depending on instrument model. Query only.

**Group** Measurement

**Syntax** MEASUrement:MEAS<x>?

**Returns** Settings for the specified measurement source.

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

## MEASUrement:MEAS<x>:DELay?

Returns the delay measurement parameters for the measurement specified by <x>, which ranges from 1 through 6. Query only.

**Group** Measurement

**Syntax** MEASUrement:MEAS<x>:DELay?

**Examples** MEASUrement:MEAS1:DELay? might return :MEASUREMENT:MEAS1:DELAY:DIRECTION FORWARDS;EDGE1 RISE;EDGE2 RISE.

## MEASUrement:MEAS<x>:DELay:EDGE<x>

Sets or queries the slope of the edge used for the delay from or to waveform when taking an immediate delay measurement. The waveform is specified by MEASUrement:MEAS<x>:SOURCE[1].

**Group** Measurement

**Syntax** MEASUrement:MEAS<x>:DELay:EDGE<x> {FALL|RISe}  
MEASUrement:MEAS<x>:DELay:EDGE<x>?

**Arguments** <x> specifies which waveform to use, where <x> = 1 is the "from" waveform, and <x> = 2 is the "to" waveform.  
FALL specifies the falling edge.  
RISe specifies the rising edge.

**Examples** MEASUREMENT:MEAS1:DELAY:EDGE1 RISE specifies that the "from" waveform rising edge be used for the immediate delay measurement.  
MEASUREMENT:MEAS1:DELAY:EDGE1? returns either RISE or FALL

## MEASUrement:MEAS<x>:SOUrce1

Sets or queries the source for all single source measurements and specifies the source to measure from when taking a delay measurement or phase measurement.

Where <x> identifies the measurement, 1 through 6. This is equivalent to selecting the measurement source in the MEASURE menu.

<b>Group</b>	Measurement
<b>Syntax</b>	MEASUrement:MEAS<x>:SOUrce1 {CH<x> MATH} MEASUrement:MEAS<x>:SOUrce1?
<b>Arguments</b>	CH<x> specifies the input channel source for the measurement. MATH specifies the measurement source channel as the math waveform.
<b>Examples</b>	MEASUREMENT:MEAS2:SOURCE1 CH1 sets source1 for Measurement 2 to channel 1. MEASUrement:MEAS1:SOUrce1? might return MATH indicating the source for measurement 1 is the math waveform.

## MEASUrement:MEAS<x>:SOUrce2

For SOURce1: Sets or queries the source for all single channel measurements.  
For delay or phase measurements, sets or queries the waveform to measure from.  
For SOURce2: Sets or queries the waveform to measure to when taking a delay measurement or phase measurement (two-source waveforms measurements).

<b>Group</b>	Measurement
<b>Syntax</b>	MEASUrement:MEAS<x>:SOUrce2 {CH<x> MATH} MEASUrement:MEAS<x>:SOUrce2?

**Related commands**    [\*MEASUrement:MEAS<x>:TYPE\*](#) on page 177

**Arguments**    CH<x> specifies the input channel source for the measurement, where x is the channel number.

MATH specifies the measurement source channel as the math waveform.

**Examples**    MEASUrement:MEAS1:SOUrce2 CH1 sets source2 for Measurement 2 to channel 1.

MEASUrement:MEAS1:SOUrce2? might return MATH indicating the to source for measurement 1 is the math waveform.

## MEASUrement:MEAS<x>:STATE

Sets or queries whether the specified measurement slot is computed and displayed. The measurement slot is specified by <x>, which ranges from 1 through 6. For a measurement to display, you must have selected a source waveform and defined the measurement you want to take and display. You select the measurement using the MEASUrement:MEAS<x>:SOURCE[1] command. You define the measurement type using the MEASUrement:MEAS<x>:TYPE command.

**Group**    Measurement

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

**Related commands**    [\*MEASUrement:MEAS<x>:TYPE\*](#) on page 177

[\*MEASUrement:MEAS<x>:SOUrce1\*](#) on page 174



<b>Arguments</b>	OFF disables calculation and display of the specified measurement slot.
	ON enables calculation and display of the specified measurement slot.
	<NR1> = 0 disables calculation and display of the specified measurement slot; any other value enables calculation and display of the specified measurement slot.
<b>Examples</b>	MEASUrement:MEAS2:STATE ON computes and displays the measurement defined as measurement 2.
	MEASUrement:MEAS1:STATE? might return :MEASUREMENT:MEAS1:STATE 0 indicating that measurement defined for measurement slot 1 is disabled.

## MEASUrement:MEAS<x>:TYPE

Sets or queries the on-screen periodic instrument measurement type for the measurement specified by <x>. Where <x> identifies the measurement, 1 through 6 depending on instrument model.

This is equivalent to selecting the measurement type in the MEASURE menu. Setting the type to anything other than NONE displays the MEASURE menu on the screen.

---

**NOTE.** You should use the MEASUrement:IMMed command with programming to take measurements, as this is preferred to the MEASUrement:MEAS<x> command.

---

<b>Group</b>	Measurement
<b>Syntax</b>	<p>MEASUrement:MEAS&lt;x&gt;:TYPE {AMPLitude   AREa   BURst   CAREa   CMEan   CRM s   DELay   FALL   FREQuency   HIGH   LOW   MAXimum   MEAN   MINImum   NDUty   NEDGECount   NOVershoot   NPULSECount NWIdth   PDUty   PEDGECount   PERIod   PHAse   PK2Pk   POVershoot   PPULSECount   PWIdth   RIs e   RMS}</p> <p>MEASUrement:MEAS&lt;x&gt;:TYPE?</p>

**Arguments**

AMplitude is the high value less the low value measured over the entire waveform or gated region.

Amplitude = High – Low.

AREA is a voltage over time measurement. It returns the area over the entire waveform or gated region in volt-seconds. Area measured above ground is positive. Area measured below ground is negative.

BURst measures the duration of a burst. The measurement is made over the entire waveform or gated region.

CAREA (cycle area) is a voltage over time measurement. The measurement is the area over the first cycle in the waveform or the first cycle in the gated region expressed in volt-seconds. The area above the common reference point is positive. The area below the common reference point is negative.

CMEAN (cycle mean) is the arithmetic mean over the first cycle in the waveform or the first cycle in the gated region.

CRMs is the true Root Mean Square voltage of the first complete cycle in the waveform.

DElay measures the time between the middle reference (default = 50%) amplitude point of the source waveform and the destination waveform.

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

FREQuency measures the first cycle in the waveform or gated region. Frequency is the reciprocal of the period and is measured in hertz (Hz), where 1 Hz = 1 cycle per second.

HIGH is the value used as 100% whenever high reference, mid reference, or low reference values are needed, such as in fall time or rise time measurements. Calculate using either the min-max or histogram method. The min-max method uses the maximum value found. The histogram method uses the most common value found above the midpoint. This value is measured over the entire waveform or gated region.

LOW is the value used as 0% whenever high reference, mid reference, or low reference values are needed, such as in fall time or rise time measurements. Calculate using either the min-max or histogram method. The min-max method uses the minimum value found. The histogram method uses the most common value found below the midpoint. This value is measured over the entire waveform or gated region.

MAXImum finds the maximum amplitude. This value is the most positive peak voltage found. It is measured over the entire waveform or gated region.

MEAN is the arithmetic mean over the entire waveform or gated region.

MINImum finds the minimum amplitude. This value is typically the most negative peak voltage. It is measured over the entire waveform or gated region.

NDUty is the ratio of the negative pulse width to the signal period expressed as a percentage. The duty cycle is measured on the first cycle in the waveform or gated region.

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

NEDGECount is the number of negative transitions from the high reference value to the low reference value in the waveform or gated region.

NOVERshoot is measured over the entire waveform or gated region and is expressed as:

Negative Overshoot =  $(\text{Low} - \text{Minimum}) / \text{Amplitude} * 100\%$ .

NPULSECount is the number of negative pulses that fall below the mid reference crossing in the waveform or gated region.

NWidth (negative width) measurement is the distance (time) between the middle reference (default = 50%) amplitude points of a negative pulse.

PDuty (positive duty cycle) is the ratio of the pulse width to the signal period, expressed as a percentage. It is measured on the first cycle in the waveform or gated region.

Positive Duty Cycle =  $((\text{Positive Width}) / \text{Period}) \times 100\%$

PEDGECount is the number of positive transitions from the low reference value to the high reference value in the waveform or gated region.

PERiod is the duration, in seconds, of the first complete cycle in the waveform or gated region. Period is the reciprocal of frequency and is measured in seconds.

PHase measures the phase difference (amount of time a waveform leads or lags the reference waveform) between two waveforms. The measurement is made between the middle reference points of the two waveforms and is expressed in degrees, where 360° represents one waveform cycle.

PK2pk (peak-to-peak) finds the absolute difference between the maximum and minimum amplitude in the entire waveform or gated region.

POVERshoot is the positive overshoot value over the entire waveform or gated region. The measurement is expressed as:

Positive Overshoot =  $(\text{Maximum} - \text{High}) / \text{Amplitude} * 100\%$ .

PPULSECount is the number of positive pulses that rise above the mid reference crossing in the waveform or gated region.

PWidth (positive width) is the distance (time) between the first rising edge and the next falling edge at the waveform 50% level (default). Rising and falling edges must be displayed to measure. The measurement is made on the first pulse in the waveform or gated region. The instrument automatically calculates the 50% measurement point.

RISe is the rise time between 10% and 90% of the first rising edge of the waveform or gated region. Rising edge must be displayed to measure. The instrument automatically calculates the 10% and 90% measurement points.

RMS amplitude measurement finds the true Root Mean Square voltage in the entire waveform or gated region.

**Examples** MEASUREMENT:MEAS2:TYPE FREQUENCY specifies MEAS2 to measure the frequency of a waveform.

MEASUREMENT:MEAS2:TYPE? might return :MEASUREMENT:MEAS1:TYPE RMS indicating that measurement 1 is defined to measure the RMS value of a waveform.

## MEASUrement:MEAS<x>:UNIts?

Returns the units for the instrument measurement specified by MEASUrement:MEAS<x>:TYPE. Where <x> identifies the measurement, 1 through 6. Query only.

**Group** Measurement

**Syntax** MEASUrement:MEAS<x>:UNIts?

**Related commands** [MEASUrement:MEAS<x>:TYPE](#) on page 177,  
[MEASUrement:IMMed:UNIts?](#) on page 171

**Returns** <QString> returns the units for the measurement.

**Examples** MEASUREMENT:MEAS3:UNITS? might return :MEASUREMENT:MEAS1:UNIts % indicating units for measurement 1 are set to percent.

## MEASUrement:MEAS<x>:VALue?

Returns the value that was calculated for the instrument on-screen periodic measurement specified by <x>. Where <x> identifies the measurement, 1 through 6. Query only.

This is the same value as displayed on-screen. If measurement statistics are enabled, a new value is calculated with every waveform. In addition, this value is updated approximately every 1/3 second. If you are acquiring a long acquisition record, the instrument may take longer to update.

**Group** Measurement

**Syntax** MEASUrement:MEAS<x>:VALue?

**Related commands** [MEASUrement:MEAS<x>:UNIts?](#) on page 180  
[\\*ESR?](#) on page 107  
[ALLEv?](#) on page 48

**Returns** <NR3> is the measurement value.

**Examples** MEASUREMENT:MEAS3:VALUE? might return 28.75E6 if measurement number three is frequency. If the measurement has an error or warning associated with it, then an item is added to the error queue. The error can be checked for with the \*ESR? and ALLEv? commands.

## MEASUrement:REFLevel?

Returns the current reference level parameters. Query only.

---

**NOTE.** *This command affects the associated reference level parameter for all MEASurements:IMMed and the four periodic measurements.*

---

**Group** Measurement

**Syntax** MEASUrement:REFLevel?

**Example** MEASUREMENT:REFLEVEL? might return these reference level settings  
PERCENT;ABSOLUTE:HIGH 0.0000;LOW 0.0000;MID1 0.0000;MID2  
0.0000;:MEASUREMENT:REFLEVEL:PERCENT:HIGH 90.0000;LOW  
10.0000;MID1 50.0000; MID2 50.0000

## MEASUrement:REFLevel:ABSolute:LOW

Sets or returns the low reference level, and is the lower reference level when MEASUrement:REFLevel:METHod is set to Absolute.

---

**NOTE.** *This command affects the associated reference level parameter for all MEASurements:IMMed and the four periodic measurements.*

---

**Group** Measurement

**Syntax** MEASUrement:REFLevel:ABSolute:LOW <NR3>  
MEASUrement:REFLevel:ABSolute:LOW?

**Related commands**    [MEASUrement:REFLevel:METHod](#) on page 185  
                               [MEASUrement:IMMed:TYPE](#) on page 169  
                               [MEASUrement:MEAS<x>:TYPE](#) on page 177

**Arguments**        <NR3> is the low reference level, in volts. The default is 0.0 V.

**Example**            MEASUREMENT:REFLEVEL? might return these reference level settings:  
 MEASUREMENT:REFLEVEL:METHOD  
 PERCENT;ABSOLUTE:HIGH 0.0000;LOW 0.0000;MID1 0.0000;MID2 0.0000;  
 MEASUREMENT:REFLEVEL:PERCENT:HIGH 90.0000;LOW 10.0000;  
 MID1 50.0000; MID2 50.0000

## MEASUrement:REFLevel:ABSolute:MID1

Sets or returns the mid reference level, and is the 50% reference level when MEASUrement:REFLevel:METHod is set to Absolute. This command affects the results of period, frequency, delay, and all cyclic measurements.

---

**NOTE.** *This command affects the associated reference level parameter for all MEASurements:IMMed and the four periodic measurements.*

---

**Group**            Measurement

**Syntax**            MEASUrement:REFLevel:ABSolute:MID[1] <NR3>  
 MEASUrement:REFLevel:ABSolute:MID[1]?

**Related commands**    [MEASUrement:REFLevel:METHod](#) on page 185

**Arguments** <NR3> is the mid reference level, in volts. The default is 0.0 V.

**Example** MEASUREMENT:REFLEVEL:ABSOLUTE:MID 1.71 sets the mid reference level to 0.71 V.

MEASUREMENT:REFLEVEL:ABSOLUTE:MID? might return 0.7100E+00 indicating that the absolute mid1 reference level is set to 0.71 V.

## MEASUREMENT:REFLEVEL:ABSOLUTE:MID2

Sets or returns 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 affects the results of delay measurements.

---

**NOTE.** *This command affects the associated reference level parameter for all MEASUREMENTS:IMMed and the four periodic measurements.*

---

**Group** Measurement

**Syntax** MEASUREMENT:REFLEVEL:ABSOLUTE:MID2 <NR3>  
MEASUREMENT:REFLEVEL:ABSOLUTE:MID2?

**Related commands** [MEASUREMENT:REFLEVEL:METHOd](#) on page 185

**Arguments** <NR3> is the mid reference level, in volts. The default is 0.0 V.

**Example** MEASUREMENT:REFLEVEL:ABSOLUTE:MID2 0.5 sets the mid reference level for the delay waveform to 0.5 V.

MEASUREMENT:REFLEVEL:ABSOLUTE:MID2? might return 0.5000E+00 indicating that the absolute mid2 reference level is set to 0.5 V.



## MEASUrement:REFLevel:METHod

Sets or returns the reference level units used for measurement calculations.

---

**NOTE.** *This command affects the associated reference level parameter for all MEASurements:IMMed and the eight periodic measurements. To change the parameter for individual measurements, use the MEASUrement:MEAS<x>:REFLevel commands.*

---

<b>Group</b>	Measurement
<b>Syntax</b>	MEASUrement:REFLevel:METHod {ABSolute   PERCent} MEASUrement:REFLevel:METHod?
<b>Arguments</b>	<p>ABSolute specifies that the reference levels are set explicitly using this commands. This method is useful when precise values are required (for example, when designing to published interface specifications, such as RS-232-C).</p> <p>PERCent specifies that the reference levels are calculated as a percent relative to HIGH and LOW. The percentages are defined using this commands.</p>
<b>Example</b>	<p>MEASUREMENT:REFLEVEL:METHOD ABSOLUTE specifies that explicit user-defined values are used for the reference levels.</p> <p>MEASUREMENT:REFLEVEL:METHOD? might return PERCENT indicating that the reference level units used are calculated as a percent relative to HIGH and LOW.</p>

## MEASUrement:REFLevel:PERCent:HIGH

Sets or returns the percent (where 100% is equal to HIGH) used to calculate the high reference level when MEASUrement:REFLevel:METHod is set to Percent. This command affects the results of rise and fall measurements.

---

**NOTE.** *This command affects the associated reference level parameter for all MEASurements:IMMed and the four periodic measurements.*

---

**Group** Measurement

**Syntax** MEASUrement:REFLevel:PERCent:HIGH <NR3>  
MEASUrement:REFLevel:PERCent:HIGH?

**Related commands** [MEASUrement:REFLevel:METHod](#) on page 185  
[MEASUrement:IMMed:TYPe](#) on page 169  
[MEASUrement:MEAS<x>:TYPe](#) on page 177

**Arguments** <NR3> is the high reference level, ranging from 0 to 100%. The default high reference level is 90%.

**Example** MEASUrement:REFLevel:PERCent:HIGH 95 sets the high reference level to 95% of HIGH.  
MEASUrement:REFLevel:PERCent:HIGH? might return 90 indicating that the percentage high reference level is set to 90% of HIGH.

## MEASUrement:REFLevel:PERCent:LOW

Sets or returns the percent (where 100% is equal to HIGH) used to calculate the low reference level when MEASUrement:REFLevel:METHod is set to Percent. This command affects the results of rise and fall measurements.

---

**NOTE.** *This command affects the associated reference level parameter for all MEASurements:IMMed and the four periodic measurements.*

---

**Group** Measurement

**Syntax** MEASUrement:REFLevel:PERCent:LOW <NR3>  
MEASUrement:REFLevel:PERCent:LOW?

**Related commands** [MEASUrement:REFLevel:METHod](#) on page 185  
[MEASUrement:IMMed:TYPE](#) on page 169  
[MEASUrement:MEAS<x>:TYPE](#) on page 177

**Arguments** <NR3> is the low reference level, ranging from 0 to 100%. The default low reference level is 10%.

**Example** MEASUrement:REFLevel:PERCent:LOW 15 sets the high reference level to 15% of HIGH.  
MEASUrement:REFLevel:PERCent:LOW? might return 10 indicating that the percentage high reference level is set to 10% of HIGH.

## MEASUrement:REFLevel:PERCent:MID1

Sets or returns 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 affects the results of period, frequency, delay, and all cyclic measurements.

---

**NOTE.** *This command affects the associated reference level parameter for all MEASurements:IMMed and the four periodic measurements.*

---

**Group** Measurement

**Syntax** MEASUrement:REFLevel:PERCent:MID[1] <NR3>  
MEASUrement:REFLevel:PERCent:MID[1]?

**Related commands** [MEASUrement:REFLevel:METHod](#) on page 185

**Arguments** <NR3> is the mid reference level, ranging from 0 to 100%. The default mid reference level is 50%.

**Example** MEASUrement:REFLevel:PERCent:MID[1] 60 sets the mid reference level to 60% of HIGH.  
MEASUrement:REFLevel:PERCent:MID[1]? might return 65 indicating that the percentage mid reference level is set to 65% of HIGH.

## MEASUrement:REFLevel:PERCent:MID2

Sets or returns the percent (where 100% is equal to HIGH) that is used to calculate the mid reference level for the second waveform specified when MEASUrement:REFLevel:METHod is set to Percent. This command affects the results of delay measurements.

---

**NOTE.** *This command affects the associated reference level parameter for all MEASurements:IMMed and the four periodic measurements.*

---

**Group** Measurement

**Syntax** MEASUrement:REFLevel:PERCent:MID2 <NR3>  
MEASUrement:REFLevel:PERCent:MID2?

**Related commands** [MEASUrement:REFLevel:METHod](#) on page 185

**Arguments** <NR3> is the mid reference level, ranging from 0 to 100%. The default mid reference level is 50%.

**Example** MEASUrement:REFLevel:PERCent:MID2 40 sets the mid reference level to 40% of HIGH.  
MEASUrement:REFLevel:PERCent:MID2? might return 45 indicating that the percentage mid reference level is set to 45% of HIGH.

# MEASUrement:SNAPSHOT

Displays the measurement snapshot list on the instrument screen. Command only, no query form.

**Group**      Measurement

**Syntax**     MEASUrement:SNAPSHOT

# MEASUrement:SOURCESNAPShot

Sets or returns the snapshot source.

**Group**      Measurement

**Syntax**     MEASUrement:SOURCESNAPShot {CH1|CH2|CH3|CH4|MATH}  
MEASUrement:SOURCESNAPShot?

**Examples**    MEASUrement:SOURCESNAPShot CH1 sets the snapshot source to channel 1.  
MEASUrement:SOURCESNAPShot? might return CH1 indicating the snapshot source is channel 1.

---

## O commands

This section lists commands and queries that begin with the letter O.

### \*OPC

Generates the operation complete message in the Standard Event Status Register (SESR) when all pending commands that generate an OPC message are complete. The \*OPC? query places the ASCII character "1" into the output queue when all such OPC commands are complete. The \*OPC? response is not available to read until all pending operations finish.

The \*OPC command allows you to synchronize the operation of the instrument with your application program. See [Synchronization Methods](#) on page 282.

Certain instrument operations can affect the \*OPC response. See [Table 30: Instrument operations that can generate OPC](#) on page 282.

**Group**      Status and Error

**Syntax**      \*OPC  
                 \*OPC?

**Related Commands**      [BUSY?](#) on page 51, [\\*WAI](#) on page 247

**Examples**      \*OPC generates the operation complete message in the SESR at the completion of all pending OPC operations.  
                 \*OPC? might return 1 to indicate that all pending OPC operations are finished.





---

## P commands

This section lists commands and queries that begin with the letter P.

### \*PSC

Sets or 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. Refer to the Status and Events section for more information. Command only, no query form.

**Group** Status and Error

**Syntax** \*PSC {OFF|ON|NR1>}

**Related Commands** [DESE](#) on page 94, [\\*ESE](#) on page 106, [FACtory](#) on page 111, [\\*RST](#) on page 202, [\\*SRE](#) on page 217

**Arguments** OFF sets the power-on status clear flag to false.  
ON sets the power-on status clear flag to true.  
<NR1> = 0 sets the power-on status clear flag to false, disables the power on clear, and allows the instrument to possibly assert SRQ after power on.  
<NR1> ≠ 0 sets the power-on status clear flag true. Sending \*PSC 1, therefore, enables the power-on status clear and prevents any SRQ assertion after power on.

**Examples** \*PSC0 sets the power-on status clear flag to false.  
\*PSC? might return the value 1, showing that the power-on status clear flag is set to true.



---

## R commands

This section lists commands and queries that begin with the letter R.

### \*RCL

Restores the state of the instrument 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. Command only, no query form.

**Group** Save and Recall

**Syntax** \*RCL <NR1>

**Related Commands** [FACTory](#) on page 111, [\\*LRN?](#) on page 154, [RECALL:SETUp](#) on page 196, [\\*RST](#) on page 202, [\\*SAV](#) on page 203, [SAVe:SETUp](#) on page 207

**Arguments** <NR1> is an integer value in the range from 1 to 10, and specifies a setup storage location.

**Examples** \*RCL3 restores the instrument from a copy of the settings stored in memory location 3.

## RECALL:SETUp

Restores the factory-default instrument settings, user-saved settings from internal nonvolatile memory, or user-saved settings from a file on a USB flash drive. Using the FACTORY argument is equivalent to pushing the DEFAULT SETUP front-panel button. Command only, no query form.

**Group** Save and Recall

**Syntax** RECALL:SETUp {FACTory|<NR1>|<file path>}

**Related Commands** [FACTory](#) on page 111, [\\*RCL](#) on page 195, [\\*RST](#) on page 202, [\\*SAV](#) on page 203, [SAVe:SETUp](#) on page 207, [FILESystem:CWD](#) on page 119

**Arguments** FACTory selects the factory setup.

<NR1> is a value in the range from 1 to 10, and specifies a setup memory storage location.

<file path> is a quoted string that defines the path and name of the .SET setup file to recall from a USB drive. Input the file path using the form <drive>:/<dir>/<filename>.<extension> and one or more <dir>s are optional. If you do not specify them, the instrument will read the file from the default directory (see FILESystem:CWD). <filename> stands for a filename; the use of wildcard characters in filenames is not supported. Filename extensions are not required, but highly recommended.

**Examples** RECALL:SETUp FACTORY recalls the instrument setup to its factory defaults.

RECALL:SETUP 2 recalls the instrument setup from setup storage location 2.

RECALL:SETUP "TEK00000.SET" recalls the setup from the file TEK00000.SET in the current directory (such as "usb0/").

## RECALL:WAVEForm

Recalls a saved waveform file to a reference location. Command only, no query form.

**Group** Save and Recall

**Syntax** RECALL:WAVEForm <file path>,REF<x>

**Related Commands** [SAVE:WAVEform](#) on page 208, [FILESystem:CWD](#) on page 119, [FILESystem?](#) on page 118

**Arguments** <file path> specifies a location of the instrument setup file. <file path> is a quoted string that defines the file name and path. Input the file path using the form <drive>:/<dir>/<filename>.<extension> and one or more <dir>s are optional. If you do not specify them, the instrument will read the file from the default directory (see FILESystem:CWD). <filename> stands for a filename; the use of wildcard characters in filenames is not supported. Filename extensions are not required, but highly recommended.

REF<x> is the instrument reference memory location in which to load the waveform. You must load a saved waveform into a reference memory location before displaying the waveform. Reference memory location values are 1 or 2.

**Examples** RECALL:WAVEFORM "tek00000.isf",REF1 recalls the waveform stored in the file named tek00000.isf from the current directory for waveforms to the reference location 1.

**REF<x>?**

Returns reference waveform data for the channel specified by <x>, where x is the reference channel number. Query only.

**Group** Vertical

**Syntax** REF<x>?

**Examples** REF1? might return the reference waveform data for reference channel 1.

**REF<x>:DATE?**

Returns the date that reference waveform data for channel <x> was copied into the internal reference memory, where x is the reference channel number. Query only.

**Group** Vertical

**Syntax** REF<x>:DATE?

**Examples** REF1:DATE? might return the date the reference waveform data for reference channel 1 was created.

## REF<x>:TIme?

Returns the time that reference waveform data was copied into the internal reference memory for reference channel <x>, where x is the reference channel number. Query only.

**Group** Vertical

**Syntax** REF<x>:TIme?

**Examples** REF2:TIme? might return “16:54:05”.

## REF<x>:HORizontal:DELay:TIme?

Returns the horizontal delay time for reference waveform <x>, where x is the reference channel number. Query only.

**Group** Vertical

**Syntax** REF<x>:HORizontal:DELay:TIme?

**Examples** REF1:HORizontal:DELay:TIme? might return the horizontal delay time for reference waveform 1.

## REF<x>:HORizontal:SCAle?

Returns the horizontal scale for reference waveform <x>, where x is the reference channel number. Query only.

**Group** Vertical

**Syntax** REF<x>:HORizontal:SCAle?

**Examples** REF<x>:HORizontal:SCAle? might return :REF1:HORIZONTAL:SCALE 4.0E-4.

## REF<x>:POSition?

Returns the vertical position for channel <x>, where x is the reference channel number. Query only.

**Group** Vertical

**Syntax** REF<x>:POSition?

**Examples** REF2:POSition? might return the vertical position for reference 2.



## REF<x>:VERTical:POSition?

Returns the vertical position of the reference waveform specified by <x>, where x is the reference channel number. Query only.

**Group** Vertical

**Syntax** REF<x>:VERTical:POSition?

**Examples** REF1:VERTical:POSition? might return :REF1:VERTICAL:POSITION -1.3000E+00 indicating that the current position of Reference 1 is 1.3 divisions below the center graticule.

## REF<x>:VERTical:SCAle?

Returns the vertical scale for the reference waveform specified by <x>, where x is the reference channel number. Query only.

**Group** Vertical

**Syntax** REF<x>:VERTical:SCAle?

**Examples** REF2:VERTICAL:SCALE? might return :REF2:VERTICAL:SCALE 1.0000e+00 indicating that the current vertical scale setting for reference 2 is 1 V per division.

## \*RST

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

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

- Calibration data that affects device specifications
- Output queue
- Service Request Enable Register settings
- Power-On Status Clear flag setting
- Alias definitions
- Stored settings or waveforms
- The \*PUD? response
- Any of the values associated with the DATA command
- Instrument password

**Group**      Status and Error

**Syntax**     \*RST

**Related Commands**    [FACTory](#) on page 111, [\\*PSC](#) on page 193, [\\*RCL](#) on page 195, [RECALL:SETUp](#) on page 196, [\\*SAV](#) on page 203, [SAVe:SETUp](#) on page 207

**Arguments**    None

**Examples**      \*RST resets the instrument settings to factory defaults.

---

## S commands

This section lists commands and queries that begin with the letter S.

### \*SAV

Saves the state of the instrument into a specified nonvolatile memory location. You can later use the \*RCL command to restore the instrument to this saved state. This is equivalent to selecting the Save Setup option in the Save/Recall Setup menu. Command only, no query form.

**Group**      Save and Recall

**Syntax**     \*SAV <NR1>

**Related Commands**    [FACtory](#) on page 111, [\\*RCL](#) on page 195, [RECALL:SETUp](#) on page 196

**Arguments**    <NR1> is an integer value in the range from 1 to 10 and specifies a memory location. Any settings that have been stored previously at this location are overwritten.

**Examples**      \*SAV2 saves the current instrument settings in memory location 2.

## SAVe:ASSIgn:TYPe

Sets or queries the assignment of the data to be saved when the front-panel Save button is pressed.

**Group** Save and Recall

**Syntax** SAvE:ASSIgn:TYPe {IMAGe|WAVEform|SETUp|}  
SAVe:ASSIgn:TYPe?

**Arguments** IMAGe assigns the Save button to save screen images.  
WAVEform assigns the Save button to save waveforms.  
SETUp assigns the Save button to save setups.

**Examples** SAvE:ASSIgn:TYPe WAVEform sets the data to be saved to waveform.  
SAVe:ASSIgn:TYPe? might return :SAVe:ASSIgn:TYPe WAVEform indicating that a waveform will be saved when the Save button is pressed.

## SAVe:IMAge

Saves the screen image to a file. Command only, no query form.

Supported image formats are png, windows bitmap, and jpg. The format to use is determined by the value obtained from the :SAVe:IMAge:FILEFormat? query.

**Group** Save and Recall

<b>Syntax</b>	SAVe:IMAge <file path>
<b>Arguments</b>	<p>&lt;file path&gt; is a quoted string that defines the path and name of the screen image file to save. Use file name extensions that are appropriate for image format. If you do not specify a path to a folder, the instrument saves the screen image file in the current working folder, using the current save image file format. The current folder refers to the name of a folder as returned by the FILESystem:CWD query.</p> <p>Use the SAvE:IMAge:FILEFormat command to set the screen image graphical file format.</p>
<b>Examples</b>	SAVe:IMAge" usb0\PROD-TST\VID-EVAL.BMP" saves the screen image to the file VID-EVAL.BMP in the folder usb0\PROD-TST .

## SAVe:IMAge:FILEFormat

Sets or queries the file format to use for saving screen images when the file type cannot be determined from the given file name or when screen images are captured by using the front panel.

<b>Group</b>	Save and Recall
<b>Syntax</b>	SAVe:IMAge:FILEFormat {PNG BMP JPG} SAVe:IMAge:FILEFormat?
<b>Arguments</b>	<p>BMP sets the screen image file format to Microsoft Windows Bitmap format.</p> <p>PNG saves the file in Portable Network Graphics format.</p> <p>JPG saves the file in Joint Picture Group format.</p>

**Examples**     `SAVe:IMAGe:FILEFormatPNG` sets the screen image graphical file format to PNG.

## **SAVe:IMAGe:LAYout**

Sets or returns the layout to use for saved screen images.

**Group**     Save and Recall

**Syntax**     `SAVe:IMAGe:LAYout {LANdscape|PORTRait}`  
`SAVe:IMAGe:LAYout?`

**Arguments**     LANDscape specifies that screen images are saved in landscape format.  
PORTRait specifies that screen images are saved in portrait format.

**Examples**     `SAVe:IMAGe:LAYout LANDscape` specifies that images are saved in landscape format.  
`SAVe:IMAGe:LAYout?` might return `:SAVe:IMAGe:LAYout LANDscape` indicating that images are saved in landscape format.

## SAVe:SETUp

Saves the current state of the instrument into the specified memory location. This is equivalent to selecting the Save Setup option in the Save/Recall Setup menu. You can later use the \*RCL command to restore the instrument to this saved state. Command only, no query form.

**Group** Save and Recall

**Syntax** SAvE:SETUp {<file path>|<NR1>}

**Related Commands** [RECALL:SETUp](#) on page 196, [\\*RCL](#) on page 195, [\\*SAV](#) on page 203

**Arguments** <NR1> specifies a location for saving the current front-panel setup. The front-panel setup value ranges from 1 to 10. Using an out-of-range value causes an execution error. Any settings that have been stored previously at this location will be overwritten.

<file path> is the target location for storing the setup file. <file path> is a quoted string that defines the file name and path. Input the file path using the form <drive>:<dir>/<filename>. <extension> and one or more <dir>s are optional. If you do not specify them, the instrument will store the file in the current working directory. <filename> stands for a filename. (Use of wildcard characters in filenames is not supported.) Filename extensions are not required but are highly recommended. For setups, use the extension .SET.

**Examples** SAvE:SETUp5 saves the current front-panel setup to memory location 5.  
SAVE:SETUP "TEK00000.SET" saves the current instrument setup in the file TEK00000.SET in the current working directory.

## SAVe:WAVEform

Stores a waveform in one of the nonvolatile reference memory locations. This command is equivalent to selecting the Save Waveform option in the Save/Recall Waveform menu. Only individual analog waveforms (CH<x>, MATH, FFT ) can be saved to reference memory locations. Command only, no query form.

You can save a specified waveform or waveforms to a single CSV file when the SAVE:WAVEFORM:FILEFORMAT is set to SPREADSHEET.

You can save a specified waveform or waveforms to consecutive ISF (internal save format) files when the SAVE:WAVEFORM:FILEFORMAT is set to INTERNAL.

**Group** Save and Recall

**Syntax** SAvE:WAVEform[<wfm>,{REF<x>}] | [REF<x>,<QString>]

**Related commands** [RECall:WAVEForm](#) on page 197, [SAVe:WAVEform:FILEFormat](#) on page 209

**Arguments** <wfm>, <REF<x> saves the specified waveform to the specified reference memory location. <wfm> can be any live analog channel (where <x> is the channel number), the MATH waveform, or FFT waveform. <wfm>, <QString> saves the specified waveform to the file specified in the quoted string argument. Any live channel (such as CH1), the MATH waveform, the FFT waveform, or any reference waveform can be saved to a file.

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

<file path> is a quoted string that defines the path and name of the waveform file to save. Use the extension .CSV for saved waveform files. Waveform data is saved as self-documented comma-separated ASCII values.



**Examples**     `SAVE:WAVEFORM CH1,REF1` saves the CH1 waveform in reference memory location 1.

`:SAVE:WAVEFORM:FILEFORMAT SPREADSHEET; :SAVE:WAVEFORM Ch1, "usb0/test1_ch1.csv"` saves channel 1 waveforms to usb0/test1\_ch1.csv.

`:SAVe:WAVEform:FILEFormat INTERNal; :SAVe:WAVEform Ch1, "usb0/test1_ch1.isf"` saves channel 1 waveforms usb0/test1\_ch1.isf

## SAVe:WAVEform:FILEFormat

Specifies or queries the file format for saved waveforms. Waveform header and timing information is included in the resulting file of non-internal formats.

**Group**     Save and Recall

**Syntax**     `SAVe:WAVEform:FILEFormat {INTERNal|SPREADSheet}`  
`SAVe:WAVEform:FILEFormat?`

**Related commands**     [CURVe](#) on page 85, [DATa](#) on page 87, [DATa:STARt](#) on page 90, [DATa:STOP](#) on page 91, [SAVe:WAVEform](#) on page 208, [WFMinpre:NR\\_Pt?](#) on page 251, [WFMinpre:NR\\_Pt?](#) on page 251

**Arguments**     `INTERNal` specifies that waveforms are saved in an internal format, using aisf filename extension. These files can be recalled as reference waveforms.

`SPREADSheet` specifies that waveform data is saved in a format that contains comma delimited values. These waveform data files are named using the csv filename extension. Saving waveforms in CSV format enables spreadsheet programs to import the data.

**Examples** SAVE:WAVEFORM:FILEFORMAT INTERNAL specifies that the internal file format is the format used for saving waveforms.

SAVE:WAVEFORM:FILEFORMAT? might return :SAVE:WAVEFORM:FILEFORMAT INTERNAL indicating that waveforms are saved using the internal format.

## SElect:CH<x>

Turns the display of the channel <x> waveform on or off, where <x > is the channel number. This command also resets the acquisition. The query returns whether the channel is on or off but does not indicate whether it is the selected waveform.

**Group** Vertical

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

**Arguments** ON turns on the display of the specified waveform. This waveform also becomes the selected waveform.

OFF turns off the display of the specified waveform.

<NR1> = 0 turns off the display of the specified waveform; any other value turns on the display of the specified waveform.

**Examples** SELECT:CH2 ON turns the channel 2 waveform display on, and selects channel 2.

SELECT:CH1? might return :SELECT:CH1 1 indicating that channel 1 is being displayed.

## SElect:CONTROL

Sets or queries the waveform that is the recipient (focus) of future channel-related commands, for example, the cursor commands. The command form also performs the equivalent of a SElect:CH<x> ON command, as well as the Math, FFT and Reference of that command.

**Group** Vertical

**Syntax** SElect:CONTROL {CH<x> | MATH | FFT | REF<x>}  
SElect:CONTROL?

**Arguments** CH<x> specifies a channel waveform as the waveform affected by the front-panel controls. <x> is the channel number.

MATH specifies the math waveform as the waveform that is affected by the front-panel controls.

FFT specifies the FFT waveform as the waveform that is affected by the front-panel controls.

REF<x> specifies a reference waveform as the waveform affected by the front-panel controls. <x> specifies the reference waveform number (1 or 2).

**Returns** NONE if all the channels are turned off. NONE is ignored on input.

**Examples** SElect:CONTROL CH1 sets channel 1 as the recipient of future channel related commands.

SElect:CONTROL? might return :SElect:CONTROL CH1 indicating that channel 1 is the recipient of future channel related commands.

## SElect:FFT

Turns on and off the display of the FFT waveform. The query returns whether the FFT waveform is on or off, but does not indicate whether it is the selected waveform.

**Group** Vertical

**Syntax** SElect:FFT {ON|OFF|<NR1>}  
SElect:FFT?

**Arguments** ON turns on the display of the specified waveform. This waveform also becomes the selected waveform.  
OFF turns off the display of the specified waveform.  
<NR1> = 0 turns off the display of the specified waveform; any other value turns on the display of the specified waveform.

**Examples** SELECT:FFT ON turns the math waveform display on, and selects it.  
SELECT:FFT? might return :SELECT:FFT 1 indicating that the math waveform is being displayed.

## SElect:MATH

Turns on and off the display of the math waveform. The query returns whether the math waveform is on or off but does not indicate whether it is the selected waveform.

**Group** Vertical

**Syntax** SElect:MATH {ON|OFF|<NR1>}  
SElect:MATH?

**Arguments** ON turns on the display of the specified waveform. This waveform also becomes the selected waveform.  
OFF turns off the display of the specified waveform.  
<NR1> = 0 turns off the display of the specified waveform; any other value turns on the display of the specified waveform.

**Examples** SELECT:MATH ON turns the math waveform display on, and selects it.  
SELECT:MATH? might return :SELECT:MATH 1 indicating that the math waveform is being displayed.

**SElect:REF<x>**

Turns on and off the display of reference waveform <x>. The <x> variable represents the reference channel number. The query returns whether the channel is on or off.

**Group** Vertical

**Syntax** SElect:REF<x> {ON|OFF|<NR1>}  
SElect:REF<x>?

**Arguments** ON turns on the display of the specified waveform. This waveform also becomes the selected waveform.  
OFF turns off the display of the specified waveform.  
<NR1> = 0 turns off the display of the specified waveform; any other value turns on the display of the specified waveform.

**Examples** SELECT:REF2 ON displays reference 2 and makes reference 2 the selected waveform.  
SELECT:REF2? might return :SELECT:REF2 1 indicating that reference waveform 2 is being displayed.

## SET?

Returns most instrument settings. You can send these responses back to the instrument to return the instrument to the state it was in when you sent SET. This query is identical to the \*LRN? query. Query only.

---

**NOTE.** *The SET? query always returns command headers, regardless of the setting of the HEADER command. This is because the returned data is intended to be able to be sent back to the instrument as concatenated commands. The VERbose command can still be used to specify whether the returned headers should be abbreviated or full length.*

---

**Group**      Miscellaneous

**Syntax**     SET?

**Related Commands**    [HEADER](#) on page 131, [\\*LRN?](#) on page 154

**Returns**       Most instrument settings. See *Appendix B: Factory Setup*.

**Examples**       SET? might return a partial string like the following: ACQUIRE:STOPAFTER  
RUNSTOP;STATE 1;MODE SAMPLE; NUMAVG 16;;HEADER  
1;;VERBOSE 1;;DISPLAY:FORMAT YT;STYLE VECTORS;PERSISTENCE  
500.0E-3;CONTRAST 50;;LOCK NONE;;HARDCOPY:FORMAT  
EPSON;PORT RS232;LAYOUT PORTRAIT;

## SETUP<x>:DATE?

Returns the date when the instrument setup was saved for the specified setup<x>. Query only.

**Group** Save and Recall

**Syntax** SETUP<x>:DATE?

**Examples** SETUP4:DATE? might return :SETUP4:DATE: 04-18-06 which is the setup date for channel 4.

## SETUP<x>:TIME? (Query Only)

Returns the time when the instrument setup was saved for the specified setup<x>.

**Group** Save and Recall

**Syntax** SETUP<x>:TIME?

**Examples** SETUP2:TIME? might return :SETUP2:TIME: 15:24:07 which is the setup time for channel 2. The default port is 4000.



## \*SRE

(Service Request Enable) sets or queries the bits in the Service Request Enable Register (SRER). Refer to the Status and Events section for more information.

**Group**      Status and Error

**Syntax**      \*SRE <NR1>  
                 \*SRE?

**Related Commands**      [\\*CLS](#) on page 72, [DESE](#) on page 94, [\\*ESE](#) on page 106, [\\*ESR?](#) on page 107, [EVENT?](#) on page 108, [EVMsg?](#) on page 108, [FACTory](#) on page 111, [\\*PSC](#) on page 193

**Arguments**      <NR1> is an integer value in the range from 0 to 255. The binary bits of the SRER are set according to this value. Using an out-of-range value causes an execution error. The power-on default for SRER is 0 if \*PSC is 1. If \*PSC is 0, the SRER maintains its value through a power cycle.

**Examples**      \*SRE48 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?**

(Read Status Byte) query returns the contents of the Status Byte Register (SBR) using the Master Summary Status (MSS) bit. Refer to the Status and Events section for more information. Query only.

**Group**      Status and Error

**Syntax**     \*STB?

**Related Commands**    [\\*CLS](#) on page 72, [DESE](#) on page 94, [\\*ESE](#) on page 106, [\\*ESR?](#) on page 107, [EVENT?](#) on page 108, [EVMsg?](#) on page 108, [FACtory](#) on page 111, [\\*SRE](#) on page 217

**Returns**      <NR1> is the contents of the Status Byte Register (SBR)

**Examples**      \*STB? might return the value 96, showing that the SBR contains the binary value 01100000.

---

## T commands

This section lists commands and queries that begin with the letter T.

### TEKSecure

Sets the Teksecure Erase Memory option to erase user data, which may be settings or confidential data files. This is equivalent to invoking Teksecure from the Utility->Config->TekSecure Erase Memory menu. This is a time-consuming operation (3 to 5 minutes) and the instrument is inoperable until the TekSecure operation is complete.

**Group**     Miscellaneous

**Syntax**     TEKSecure

### TIME

Sets or queries the time the instrument displays. The instrument uses the time and date values to time stamp files and show the time and date on the instrument display.

#### Conditions

**Group**     Miscellaneous

**Syntax**    TIME <QString>  
TIME?

**Related commands**    [DATE](#) on page 93

**Arguments**    <QString> is a time in the form "hh:mm:ss" where hh refers to a two-digit hour number, mm refers to a two-digit minute number from 00 to 59, and ss refers to a two-digit second number from 00 to 59.

**Examples**    TIME "14:00:00" sets the time to exactly 2:00 p.m.  
TIME? might return :TIME "11:25:03" indicating that the current time is set to 11:25 a.m. and 03 seconds.

## TRIGger

Forces a trigger event to occur. No query form.

**Group**    Trigger

**Syntax**    TRIGger FORCE

**Arguments**    FORCE creates a trigger event. If TRIGger:STATE is READy, the acquisition will complete; otherwise this command is ignored.

**Examples**    TRIGgerFORCE forces a trigger event to occur.

## TRIGger:A

Sets the instrument trigger level to 50% of the minimum and maximum values of the signal. Returns the current A trigger settings when used as a query.

The trigger level is the voltage threshold through which the trigger source signal must pass to generate a trigger event. This command works for the following cases: Edge Trigger (when source is not Line), Logic Trigger (when Clock Source is not Off or Logic Pattern is Don't Care), and Pulse Width Trigger.

**Group** Trigger

**Syntax** TRIGger:A SETLevel  
TRIGger:A?

**Related commands** [TRIGger:A:EDGE?](#) on page 222, [TRIGger:A:PULse?](#) on page 229

**Arguments** SETLevel sets the A trigger level to 50% of the minimum and maximum values of the trigger source input signal. This is equivalent to pressing the front-panel PUSH to SET to 50% button.

**Examples** TRIGger:ASETLEVEL sets the A trigger level to 50% of the range of the minimum and maximum values of the trigger input signal.

TRIGGER:A? might return a long response with A trigger parameters, some of which could be as follows: :TRIGGER:A:MODE AUTO;TYPE EDGE;LEVEL 20.0000E-3;LEVEL:CH1 20.0000E-3;CH2 0.0000; :TRIGGER:A:UPPERTHRESHOLD:CH1 1.4000;CH2 800.0000E-3;:TRIGGER:A:LOWERTHRESHOLD:CH1 20.0000E-3;CH2 0.0000;:TRIGGER:A:HOLDOFF:TIME 20.0000E-9;:TRIGGER:A:EDGE:SOURCE CH1;COUPLING DC;SLOPE RISE;:TRIGGER:A:LOGIC:CLASS SETHOLD;FUNCTION AND;THRESHOLD: CH1 20.0000E-3;CH2 0.0000; :TRIGGER:A:LOGIC:INPUT:CH1 X;CH2 X; CLOCK:SOURCE NONE;EDGE.

## TRIGger:A:EDGE?

Returns the trigger coupling, source, and slope settings for the A edge trigger. Query only.

**Group** Trigger

**Syntax** TRIGger:A:EDGE?

**Related commands** [TRIGger:A:PULse?](#) on page 229

**Returns** Trigger coupling, source, and slope settings for the A edge trigger.

**Examples** TRIGger:A:EDGE? might return :TRIGGER:A:EDGE:SOURCE CH1;COUPLING DC; SLOPE RISE indicating the trigger source, coupling, and slope for the A edge trigger.

## TRIGger:A:EDGE:COUPling

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

**Group** Trigger

**Syntax** TRIGger:A:EDGE:COUPling {DC|HFRej|LFRej|NOISErej}  
TRIGger:A:EDGE:COUPling?

**Related commands**    [TRIGger:A:EDGE:SOUrce](#) on page 224, [TRIGger:A:EDGE:SLOpe](#) on page 223

**Arguments**    DC selects DC coupling, which passes all input signals to the trigger circuitry.  
 HFRej coupling attenuates the high-frequency components, above 50 kHz, of the trigger signal.  
 LFRej coupling removes the low-frequency components, below 50 kHz, of the trigger signal.  
 NOISErej selects DC low sensitivity, which provides stable triggering by increasing the trigger hysteresis. Increased hysteresis reduces the trigger sensitivity to noise but may require greater trigger signal amplitude.

**Examples**    TRIGger:A:EDGE:COUPlingDC sets the A edge trigger coupling to DC.  
 TRIGGER:A:EDGE:COUPLING? might  
 return :TRIGGER:A:EDGE:COUPLING DC indicating that the A edge trigger coupling is set to DC.

## TRIGger:A:EDGE:SLOpe

Sets or queries the slope for the A edge trigger. This is equivalent to setting the Slope option in the Trigger menu.

**Group**    Trigger

**Syntax**    TRIGger:A:EDGE:SLOpe{RISe|FALL}  
 TRIGger:A:EDGE:SLOpe?

**Related commands**    [TRIGger:A:EDGE:SOUrce](#) on page 224, [TRIGger:A:EDGE:COUPling](#) on page 222

<b>Arguments</b>	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.
<b>Examples</b>	TRIGger:A:EDGE:SLOPeRISE sets the A edge trigger to occur on the rising edge of the signal. TRIGGER:A:EDGE:SLOPE? might return :TRIGGER:A:EDGE:SLOPE FALL indicating that the A edge trigger slope is negative.

## TRIGger:A:EDGE:SOUrce

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

<b>Group</b>	Trigger
<b>Syntax</b>	TRIGger:A:EDGE:SOUrce {{CH1   CH2   LINE}} TRIGger:A:EDGE:SOUrce?
<b>Related commands</b>	<a href="#">TRIGger:A:EDGE:SLOPe</a> on page 223, <a href="#">TRIGger:A:EDGE:COUPling</a> on page 222
<b>Arguments</b>	CH<x> specifies one of the analog input channels. The value of <x> can vary from 1 through 4 for 4-channel instruments or 1 through 2 for 2-channel instruments. AC LINE specifies the AC line as a trigger source.
<b>Examples</b>	TRIGger:A:EDGE:SOUrceCH1 specifies channel 1 as the A edge trigger source. TRIGger:A:EDGE:SOUrce? might return :TRIGGER:A:EDGE:SOURCE CH1 indicating that channel 1 is the A edge trigger source.



## TRIGger:A:HOLDOff?

Returns the A trigger holdoff parameters. These parameters specify the time period during which the trigger circuitry is not looking to generate a trigger event. Query only.

**Group** Trigger

**Syntax** TRIGger:A:HOLDOff?

**Related commands** [TRIGger:A:HOLDOff:TIME](#) on page 225

**Returns** A trigger holdoff value.

**Examples** TRIGger:A:HOLDOff? might return :TRIGGER:A:HOLDOff:TIME 900.0000E-09, indicating that the A edge trigger holdoff time (by default) is 900 ns.

## TRIGger:A:HOLDOff:TIME

Sets or queries the A trigger holdoff time.

**Group** Trigger

**Syntax** TRIGger:A:HOLDOff:TIME <NR3>  
TRIGger:A:HOLDOff:TIME?

<b>Arguments</b>	<NR3> is the A trigger holdoff time. The range is 20 ns to 8.0 s.
<b>Examples</b>	TRIGger:A:HOLDOff:TIME10 sets the holdoff time to 10 s. TRIGGER:A:HOLDOFF:TIME? might return :TRIGGER:A:HOLDOFF:TIME 1.2000E-06 indicating that the A trigger holdoff time is set to 1.2 $\mu$ s.

## TRIGger:A:LEVel

Sets or queries the trigger level for the A trigger. This command is equivalent to adjusting the front-panel TRIGGER LEVEL knob.

---

**NOTE.** When the edge trigger source is set to AC LINE, the instrument ignores the set form of the command.

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

---

<b>Group</b>	Trigger
<b>Syntax</b>	TRIGger:A:LEVel{ECL TTL <NR3>} TRIGger:A:LEVel?
<b>Arguments</b>	<NR3> specifies the trigger level in user units (usually volts). ECL specifies a preset ECL high level of -1.3V. TTL specifies a preset TTL high level of 1.4V.
<b>Examples</b>	TRIGGER:A:LEVEL TTL sets the A edge trigger to TTL high level, which is 1.4 V. TRIGger:A:LEVel? might return :TRIGGER:A:LEVel 1.3000E+00 indicating that the A edge trigger is set to 1.3 V.

## TRIGger:A:LEVel:CH<x>

Sets or queries the trigger level for the specified channel. Each channel can have an independent level.

**Group** Trigger

**Syntax** TRIGger:A:LEVel:CH<x> {<NR3>|TTL|ECL}  
TRIGger:A:LEVel:CH<x>?

**Arguments** ECL specifies a preset ECL high level of -1.3V.  
TTL specifies a preset TTL high level of 1.4V.  
<NR3> specifies the trigger level in user units (usually volts).

**Examples** TRIGGER:A:LEVEL:CH1 TTL sets the A edge trigger to TTL high level for channel 1.  
TRIGGER:A:LEVEL:CH2? might return :TRIGGER:A:LEVEL:CH2 1.3000E+00 indicating that the A edge trigger is set to 1.3 V for channel 2.

## TRIGger:A:LOWerthreshold:CH<x>

Sets or queries the lower threshold for the channel selected. Each channel can have an independent level. Used in Runt trigger as the lower threshold. Used for all other trigger types as the single level/threshold.

**Group** Trigger

**Syntax** TRIGger:A:LOWerthreshold:CH<x> {ECL|TTL|<NR3>}  
TRIGger:A:LOWerthreshold:CH<x>?

**Related commands** [TRIGger:A:LEVel:CH<x>](#) on page 227

**Arguments** ECL specifies a preset ECL high level of -1.3 V.  
TTL specifies a preset TTL high level of 1.4 V.  
<NR3> is the clock level, in volts.

**Examples** TRIGGER:A:LOWERTHRESHOLD:CH2 50E-3 sets the lower limit threshold for CH2 of the pulse runt trigger to 50 mV.  
TRIGGER:A:LOWERTHRESHOLD:CH2? might return :TRIGGER:A:LOWERTHRESHOLD:CH2 1.2000E-01 indicating that the lower limit threshold for CH2 of the pulse runt trigger is set to 120 mV.

## TRIGger:A:MODE

Sets or queries the trigger mode.

**Group** Trigger

**Syntax** TRIGger:A:MODE {AUTO|NORMal}  
TRIGger:A:MODE?

**Related Commands** [TRIGger:A:LEVel](#) on page 226

<b>Arguments</b>	AUTO generates a trigger if a trigger is not detected within a specific time period. NORMal waits for a valid trigger event.
<b>Examples</b>	TRIGger:A:MODENORMAL specifies that a valid trigger event must occur before a trigger is generated. TRIGGER:A:MODE ? might return :TRIGGER:A:MODE NORMAL indicating that a valid trigger event must occur before a trigger is generated.

## TRIGger:A:PULse?

Returns the current Pulse Trigger settings. Query only.

**Group** Trigger

**Syntax** TRIGger:A:PULse?

**Related commands** [TRIGger:A:EDGE?](#) on page 222

**Examples** TRIGger:A:PULse? might return :TRIGGER:A:PULSE:CLASS TRAnsITION.

## TRIGger:A:PULse:CLAss

Sets or queries the type of pulse on which to trigger.

**Group** Trigger

**Syntax** TRIGger:A:PULse:CLAss {RUNt|WIDth}  
TRIGger:A:PULse:CLAss?

**Related commands** [TRIGger:A:RUNT?](#) on page 235, [TRIGger:A:PULSE:Width?](#) on page 231, [TRIGger:A:TYPE](#) on page 238

**Arguments** RUNt triggers when a pulse crosses the first preset voltage threshold but does not cross the second preset threshold before recrossing the first.  
WIDth triggers when a pulse is found that has the specified polarity and is either inside or outside the specified time limits.

**Examples** TRIGGER:A:PULSE:CLASS WIDTH specifies a width pulse for the A trigger.  
TRIGGER:A:PULSE:CLASS? might return :TRIGGER:A:PULSE:CLASS  
WIDTH indicating that a pulse was found that is of the specified polarity and width.

## TRIGger:A:PULSE:Width?

Queries the width for the pulse-width trigger. Query only.

<b>Group</b>	Trigger
<b>Syntax</b>	TRIGger:A:PULSEWidth?
<b>Examples</b>	TRIGger:A:PULSEWidth? might return :TRIGGER:A:PULSEWIDTH:POLARITY POSITIVE;WHEN LESSTHAN;WIDTH 8.0E-9

## TRIGger:A:PULse:WIDth:POLarity

Sets or queries the polarity for the pulse width trigger. This is equivalent to setting the Polarity option in the Pulse Trigger menu.

<b>Group</b>	Trigger
<b>Syntax</b>	TRIGger:A:PULse:WIDth:POLarity {NEGative POSitive} TRIGger:A:PULse:WIDth:POLarity?
<b>Arguments</b>	POSITIVE polarity specifies pulses with a rising leading edge. NEGAtive polarity specifies pulses with a falling leading edge.
<b>Examples</b>	TRIGGER:A:PULSEWIDTH:POLARITY NEGATIVE sets the pulse polarity to negative. TRIGGER:A:PULSEWIDTH:POLARITY? might return :TRIGGER:A:WIDTH:POLARITY POSITIVE indicating a positive pulse.

## TRIGger:A:PULSEWidth:SOUrce

Sets or queries the source for the pulse width trigger. This is equivalent to setting the Source option in the Trigger menu.

**Group** Trigger

**Syntax** TRIGger:A:PULSe:SOUrce {CH1|CH2| LINE}  
TRIGger:A:PULSe:SOUrce?

**Arguments** CH<x> specifies one of the analog input channels. The value of <x> can be 1 through 4 on four channel instruments, or 1 or 2 on two channel instruments.  
LINE specifies AC line voltage.

**Examples** TRIGGER:A:PULSEWIDTH:SOURCE CH1 sets channel 1 as the pulse width source.  
TRIGGER:A:PULSEWIDTH:SOURCE? might  
return :TRIGGER:A:PULSEWIDTH:SOURCE CH1 indicating that channel 1 is the pulse width source.

## TRIGger:A:PULSe:WIDth:WHEN

Sets or queries whether to trigger on a pulse that meets, falls outside, or within the specified range of limits. This is equivalent to setting the When option in the Pulse Trigger menu.

**Group** Trigger



**Syntax** TRIGger:A:PULse:WIDth:WHEN {LESSthan|MOREthan|EQUAL|UNEQUAL}  
 TRIGger:A:PULse:WIDth:WHEN?

**Related commands** [TRIGger:A:PULse:WIDth:WIDth](#) on page 234

**Arguments** LESSthan sets the instrument to trigger if a pulse is detected with width less than the time set by the TRIGger:A:PULSEWidth:WIDth command.

MOREthan sets the instrument to trigger if a pulse is detected with width more than the time set by the TRIGger:A:PULSEWidth:WIDth command.

EQUAL sets the instrument to trigger if a pulse is detected with width equal to the time period specified in TRIGger:A:PULSEWidth:WIDth within a  $\pm 5\%$  tolerance.

UNEQUAL sets the instrument to trigger if a pulse is detected with width greater than or less than (but not equal) the time period specified in TRIGger:A:PULSEWidth:WIDth within a  $\pm 5\%$  tolerance.

**Examples** TRIGGER:A:PULSEWIDTH:WHEN LESSTHAN specifies that the duration of the A pulse will fall within defined high and low limits.

TRIGGER:A:PULSEWIDTH:WHEN? might return :TRIGGER:A:PULSEWIDTH:WHEN LESSTHAN indicating that the conditions for generating a width trigger.

## TRIGger:A:PULse:WIDth:WIDth

Sets or queries the width setting for the pulse width trigger. This is equivalent to setting the Width option by using the Pulse Trigger menu and the TRIGGER knob.

**Group** Trigger

**Syntax** TRIGger:A:PULse:WIDth:WIDth <NR3>  
TRIGger:A:PULse:WIDth:WIDth?

**Related commands** [\*TRIGger:A:PULse:WIDth:WHEN\*](#) on page 232

**Arguments** <NR3> specifies the pulse width, in seconds.

**Examples** TRIGGER:A:PULSEWIDTH:WIDTH 5.0E-6 sets the pulse width to 5  $\mu$ s.  
TRIGGER:A:PULSEWIDTH:WIDTH? might  
return :TRIGGER:A:PULSEWIDTH:WIDTH 2.0000E-9 indicating that the  
pulse width is set to 2 ns.

## TRIGger:A:RUNT?

Returns the current A runt trigger parameters. Query only.

**Group** Trigger

**Syntax** TRIGger:A:RUNT?

**Examples** TRIGGER:A:RUNT? might return :TRIGGER:A:RUNT:SOURCE  
CH1;POLARITY POSITIVE;WHEN OCCURS;WIDTH 4.0000E-9.

## TRIGger:A:RUNT:POLarity

Sets or queries the polarity for the runt trigger.

**Group** Trigger

**Syntax** TRIGger:A:RUNT:POLarity {NEGative|POSitive}  
TRIGger:A:RUNT:POLarity?

**Arguments** POSitive indicates that the rising edge crosses the low threshold and the falling edge recrosses the low threshold without either edge ever crossing the high threshold.

NEGative indicates that the falling edge crosses the high threshold and the rising edge recrosses the high threshold without either edge ever crossing the low threshold.

**Examples** TRIGGER:A:RUNT:POLARITY NEGATIVE specifies that the polarity of the A pulse runt trigger is negative.  
TRIGGER:A:RUNT:POLARITY? might return :TRIGGER:A:RUNT:POLARITY POSITIVE indicating that the polarity of the A pulse runt trigger is positive.

## TRIGger:A:RUNT:SOUrce

Sets or queries the source for the A runt trigger.

**Group** Trigger

**Syntax** TRIGger:A:RUNT:SOUrce {CH1|CH2}

**Arguments** CH1-CH2 specifies an analog input channel as the trigger source.

**Examples** TRIGGER:A:RUNT:SOURCE CH1 sets channel 2 as the source for the A pulse trigger.  
TRIGGER:A:RUNT:SOURCE? might return :TRIGGER:A:RUNT:SOURCE CH2 indicating that channel 2 is the source for the A pulse trigger.

## TRIGger:A:RUNT:WHEn

Sets or queries the type of pulse width the trigger checks for when it detects a runt.

**Group** Trigger

**Syntax** TRIGger:A:RUNT:WHEn {LESSthan|MOREthan|EQUAL|UNEQUAL|OCCURS}  
TRIGger:A:RUNT:WHEn?

**Related commands** [TRIGger:A:RUNT:WIDth](#) on page 238

**Arguments** OCCURS specifies a trigger event if a runt of any detectable width occurs.

LESSthan sets the instrument to trigger if a runt pulse is detected with a width less than the time set by the TRIGger:A:RUNT:WIDth command.

MOREthan sets the instrument to trigger if a runt pulse is detected with a width more than the time set by the TRIGger:A:RUNT:WIDth command.

EQUAL sets the instrument to trigger if a runt pulse is detected with a width equal to the time period specified in TRIGger:A:RUNT:WIDth within a  $\pm 5\%$  tolerance.

UNEQUAL sets the instrument to trigger if a runt pulse is detected with a width greater than or less than (but not equal to) the time period specified in TRIGger:A:RUNT:WIDth within a  $\pm 5\%$  tolerance.

**Examples** TRIGGER:A:RUNT:WHEn THAN sets the runt trigger to occur when the instrument detects a runt in a pulse wider than the specified width.

TRIGGER:A:RUNT:WHEn? might return :TRIGGER:A:PULSE:RUNT:WHEn OCCURS indicating that a runt trigger will occur if the instrument detects a runt of any detectable width.

## TRIGger:A:RUNT:WIDth

Sets or queries the width for a runt trigger.

**Group** Trigger

**Syntax** TRIGger:A:RUNT:WIDth <NR3>  
TRIGger:A:RUNT:WIDth?

**Related commands** [\*TRIGger:A:RUNT:WHEn\*](#) on page 237

**Arguments** <NR3> specifies the minimum width, in seconds.

**Examples** TRIGGER:A:RUNT:WIDTH 15E-6 sets the minimum pulse width of the runt trigger to 15  $\mu$ s.  
TRIGGER:A:RUNT:WIDTH? might  
return :TRIGGER:A:PULSE:RUNT:WIDTH 2.0000E-09 indicating that the minimum pulse width of a runt trigger is 2 ns.

## TRIGger:A:TYPe

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

**Group** Trigger

**Syntax** TRIGger:A:TYPE{EDGE|PULSe}  
TRIGger:A:TYPE?

**Related commands** [TRIGger:A:EDGE?](#) on page 222, [TRIGger:A:PULSe:CLass](#) on page 230

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

PULSe specifies that a trigger occurs when the specified signal meets the pulse width criteria that is controlled by the TRIGger: A:PULSe commands.

**Examples** TRIGGER:A:TYPE EDGE sets the A trigger type to EDGE.

TRIGGER:A:TYPE? might return :TRIGGER:A:TYPE PULSE indicating that the A trigger type is a pulse trigger.

## TRIGger:A:UPPerthreshold:CH<x>

Sets or queries the upper threshold for channel <x>, where x is the channel number. Each channel can have an independent level. Used only for runt trigger type.

**Group** Trigger

**Syntax** TRIGger:A:UPPerthreshold:CH<x> {<NR3>|ECL|TTL}  
TRIGger:A:UPPerthreshold:CH<x>?

**Arguments** <NR3> is the threshold level in volts.

ECL specifies a preset ECL high level of -1.3 V.

TTL specifies a preset TTL high level of 1.4 V.

**Examples**    `TRIGGER:A:UPPERTHRESHOLD:CH2 50E-3` sets the upper limit of the pulse runt trigger to 50 mV for channel 2.

`TRIGGER:A:UPPERTHRESHOLD:CH2?` might return `:TRIGGER:A:UPPERTHRESHOLD:CH2 1.2000E-01` indicating that the upper limit of the pulse runt trigger is set to 120 mV.

## TRIGger:FREQuency?

Returns the edge or pulse width trigger frequency. This is the same as the readout in the lower right corner of the screen. Query only.

**Group**    Trigger

**Syntax**    `TRIGger:FREQuency?`

**Returns**    Edge or pulse width trigger frequency.

**Examples**    `TRIGger:FREQuency?` might return `TRIGGER:FREQUENCY 1.0E3`.

## TRIGger:STATE?

Returns the current state of the triggering system. Query only.

**Group**    Trigger

**Syntax**    `TRIGger:STATE?`



**Related commands**    *TRIGger:A:MODe* on page 228

**Returns**    ARMED indicates that the instrument is acquiring pretrigger information. All triggers are ignored when TRIGger:STATE is ARMED.

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

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

SAVE indicates that the instrument is in save mode and is not acquiring data.

TRIGGER indicates that the instrument triggered and is acquiring the post trigger information.

**Examples**    TRIGGER:STATE? might return :TRIGGER:STATE ARMED indicating that the pretrigger data is being acquired.



---

## U commands

This section lists commands and queries that begin with the letter U.

### UNLock

Unlocks the front panel. This command is equivalent to LOCK NONE. Command only, no query form.

**Group**      Miscellaneous

**Syntax**     UNLock ALL

**Related commands**    [LOCK](#) on page 154

**Arguments**    ALL specifies all front-panel buttons.

**Examples**      UNLock ALL unlocks all front-panel buttons and knobs so they can be used.



---

## V commands

This section lists commands and queries that begin with the letter V.

### VERBose

Sets and queries the Verbose state that controls the length of keywords on query responses. Keywords can be both headers and arguments. This command does not affect IEEE Std 488.2-1987 Common Commands (those starting with an asterisk).

**Group**      Miscellaneous

**Syntax**      VERBose  
VERBose?

**Related Commands**      [HEADer](#) on page 131, [\\*LRN?](#) on page 154

**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**      VERBoseON sets the Verbose state true.  
VERBose? might return the value 1, showing that the Verbose state is true.



---

## W commands

This section lists commands and queries that begin with the letter W.

### \*WAI

Prevents the instrument from executing further commands or queries until all pending commands that generate an OPC message are complete. This command allows you to synchronize the operation of the instrument with your application program. Command only, no query form.

The \*WAI command will stop execution until certain instrument operations are complete. See [Table 30: Instrument operations that can generate OPC](#) on page 282.

**Group**      Status and Error

**Syntax**     \*WAI

**Related Commands**    [BUSY?](#) on page 51, [\\*OPC](#) on page 191

**Examples**          \*WAI prevents the instrument from executing any further commands or queries until all pending commands that generate an OPC message are complete.

## WAVFrm?

Returns WFMOutpre? and CURVe? data for the waveform as specified by the DATA:SOURce command. This command is equivalent to sending both WFMOutpre? and CURVe?, with the additional provision that the response to WAVFrm? is guaranteed to provide a synchronized preamble and curve. The source waveform, as specified by :DATA:SOURCE, must be active or the query will not return any data and will generate an error indicator. Query only.

**Group**      Waveform

**Syntax**     WAVFrm?

**Related Commands**    [CURVe](#) on page 85, [DATA:SOURce](#) on page 89, [WFMOutpre?](#) on page 258

**Returns**      See WFMPre? and CURVe? commands.

## WFMinpre?

Returns the waveform formatting and scaling specifications to be applied to the next incoming CURVe command data. Query only.

**Group**      Waveform

**Syntax**     WFMinpre?

**Related commands**    [CURVe](#) on page 85, [DATA:SOURce](#) on page 89, [WFMOutpre?](#) on page 258



**Returns** Returns the response in the following format:  
 WFMPre:<wfm>:WFID <Qstring>;PT\_FMT { ENV | Y }; XINcr  
 <NR3>;PT\_Off <NR1>;XZZero <NR3>;XUNit <QString>; YMult  
 <NR3>;YZZero <NR3>;YOFF <NR3>;YUNit <QString>; NR\_Pt <NR1>

**Examples** WFMINPRE? might return the waveform formatting  
 as :WFMINPRE:BIT\_NR8;BN\_FMT RI;BYT\_NR 1; BYT\_OR MSB;ENCDG  
 BIN;NR\_PT 500;PT\_FMTY; PT\_OFF 0;XINCR 2.0000E-6;XZERO 1.7536E-6;  
 XUNIT "s";YMULT 1.0000E-3;YOFF 0.0000; YZERO 0.0000;YUNIT "V".

## WFMInpre:BIT\_Nr

Sets or returns the number of bits per binary waveform point for the incoming waveform. Changing the value of WFMInpre:BIT\_Nr also changes the value of WFMInpre:BYT\_Nr.

**Group** Waveform

**Syntax** WFMInpre:BIT\_Nr  
 WFMInpre:BIT\_Nr?

**Arguments** <NR1> is either 8 or 16.

**Examples** WFMINPRE:BIT\_NR 16 sets the number of bits per waveform point to 16, for incoming data.  
 WFMINPRE:BIT\_NR? might return :WFMINPRE:BIT\_NR 8 indicating that incoming waveform data uses 8 bits per waveform point.

## WFMinpre:BYT\_Nr

Sets or queries the data width for the incoming waveform. Changing the value of WFMinpre:BYT\_Nr also changes the value of WFMinpre:BIT\_Nr.

**Group** Waveform

**Syntax** WFMinpre:BYT\_Nr  
WFMinpre:BYT\_Nr?

**Arguments** <NR1> is an integer in the range of 1 to 2 that sets the number of bytes per point.

**Examples** WFMINPRE:BYT\_NR 1 sets the number of bytes per incoming waveform data point to 1, which is the default setting.  
WFMINPRE:BYT\_NR? might return :WFMINPRE:BYT\_NR 2 indicating that there are 2 bytes per incoming waveform data point.

## WFMinpre:ENCdg

Sets or queries the type of encoding for waveform data transferred with the CURVe command.

**Group** Waveform

**Syntax** WFMinpre:ENCdg {ASCii|BINary}  
WFMinpre:ENCdg?

<b>Arguments</b>	<p>ASCIi specifies ASCII curve data.</p> <p>BINary specifies that the incoming data is in a binary format whose further interpretation requires knowledge of BYT_NR, BIT_NR, BN_FMT, and BYT_OR.</p>
<b>Examples</b>	<p>WFMINPre:ENCdgASC specifies that the waveform data is in ASCII format.</p> <p>WFMINPre:ENCdg? might return :WFMINPRE:ENCDG BIN, indicating that the waveform data is in binary format.</p>

## WFMINpre:NR\_Pt?

Returns the number of points that are in the incoming waveform record.

**Group** Waveform

**Syntax** WFMINpre:NR\_Pt <NR1>  
WFMINpre:NR\_Pt?

**Related Commands** [CURVe](#) on page 85, [DATa](#) on page 87, [DATa:STARt](#) on page 90, [DATa:STOP](#) on page 91, [SAVe:WAVEform](#) on page 208, [SAVe:WAVEform:FILEFormat](#) on page 209, [WFMINpre:NR\\_Pt?](#) on page 251

**Arguments** <NR1> is the number of data points if WFMINpre:PT\_Fmt is set to Y. It is the number of min-max pairs if WFMINpre:PT\_Fmt is set to ENV.

**Examples** WFMINPRE:NR\_PT 10000 specifies that 10000 data points will be expected.  
WFMINPRE:NR\_PT ? might return :WFMINPRE:NR\_PT 2000 indicating that there are 2000 data points in the expected incoming waveform record.

## WFMinpre:XINcr

Sets or queries the horizontal interval between incoming waveform points in units specified by WFMinpre:XUNit.

**Group** Waveform

**Syntax** WFMinpre:XINcr <NR3>  
WFMinpre:XINcr?

**Arguments** <NR3> is the interval between points in the waveform record, in the units specified by WFMPre:XUNit.

**Examples** WFMINPRE:XINCR 3E-3 sets the interval between incoming waveform points to 3 ms.  
WFMINPRE:XINCR ? might return :WFMINPRE:XINCR 1.0000E-3 indicating that if WFMinpre:XUNit is set to "s", there is a 1 ms interval between incoming waveform points.

## WFMinpre:XUNit

Sets or queries the horizontal units of the incoming waveform.

**Group** Waveform

**Syntax** WFMinpre:XUNit <Qstring>  
WFMinpre:XUNit?

**Related commands**    [WFMOutpre:XUNit?](#) on page 264

**Arguments**    <Qstring> contains a maximum of three alpha characters that represent the horizontal unit of measure for the incoming waveform.

**Examples**    WFMINPRE:XUNIT "HZ" specifies that the horizontal units for the incoming waveform are hertz.

WFMINPRE:XUNIT? might return :WFMINPRE:XUNIT "s" indicating that the horizontal units for the incoming waveform are seconds.

## WFMInpre:XZEro

Sets or queries the position value, in XUNits, of the first sample of the incoming waveform, relative to the trigger.

The instrument sets WFMPre:XZEro to zero when:

- The display mode is set to XY.
- The DATA:SOURce is set to MATH FFT when the waveform is acquired.

---

**NOTE.** *The instrument uses XZEro when calculating cursor readouts.*

---

**Group**    Waveform

**Syntax**    WFMInpre:XZEro <NR3>  
WFMInpre:XZEro?

**Related commands**    [WFMInpre:XINcr](#) on page 252, [WFMInpre:BYT\\_Nr](#) on page 250, [WFMOutpre:XZEro?](#) on page 265

<b>Arguments</b>	<NR3> argument is the floating point value of the position, in XUNits, of the first sample in the incoming waveform. If XUNits is "s", <NR3> is the time of the first sample in the incoming waveform.
<b>Examples</b>	<p>WFMINPRE:XZERO 5.7E-6 indicates the trigger occurred 5.7 <math>\mu</math>s before the first sample in the waveform.</p> <p>WFMINPRE:XZERO? might return :WFMINPRE:XZero -7.5000E-6 indicating that the trigger occurs 7.5 <math>\mu</math>s after the first sample in the waveform.</p>

## WFMinpre:YMUlt

Sets or queries the vertical scale factor of the incoming waveform, expressed in YUNits per waveform data point level. For one byte waveform data, there are 256 data point levels. For two byte waveform data there are 65,536 data point levels. YMUlt, YOff, and YZero are used to convert waveform record values to YUNit values using the following formula (where dl is the data level; curve\_in\_dl is a data point in CURVe):  $\text{value\_in\_units} = ((\text{curve\_in\_dl} - \text{YOff\_in\_dl}) * \text{YMUlt}) + \text{YZero\_in\_units}$ .

<b>Group</b>	Waveform
<b>Syntax</b>	<p>WFMinpre:YMUlt &lt;NR3&gt;</p> <p>WFMinpre:YMUlt?</p>
<b>Related commands</b>	<p><a href="#">DATA:DESTination</a> on page 88, <a href="#">WFMinpre:BYT_Nr</a> on page 250, <a href="#">WFMinpre:YUNit</a> on page 256</p>
<b>Arguments</b>	<NR3> is the vertical scale factor per digitizing level of the incoming waveform points.

**Examples**     WFMINPRE:YMULT? might return :WFMINPRE:YMULT 40.0000E-3, which (if YUNit is "V") indicates that the vertical scale is 40 mV/digitizing level (1V/div for 8-bit data).

## WFMInpre:YOff

Sets or queries the vertical position of the incoming waveform in digitizing levels. Variations in this number are analogous to changing the vertical position of the waveform.

**Group**     Waveform

**Syntax**     WFMInpre:YOff <NR3>  
WFMInpre:YOff?

**Arguments**     <NR3> is a value expressed in digitizing levels.

**Examples**     WFMINPRE:YOFF 50 specifies that the zero reference point for the incoming waveform is 50 digitizing levels (2 divisions, for 8-bit data) above the center of the data range.  
  
WFMINPRE:YOFF? might return :WFMINPRE:YOFF 25 indicating the vertical position of the incoming waveform in digitizing levels.

## WFMinpre:YUNit

Sets or returns the vertical units of the incoming waveform.

**Group**      Waveform

**Syntax**      WFMinpre:YUNit <Qstring>  
WFMinpre:YUNit?

**Arguments**      <Qstring> contains a maximum of three alpha characters that represent the vertical unit of measure for the incoming waveform.

**Returns**      The query may return the following:

- Volts for volts
- U for unknown units (divisions)
- dB for decibels
- ? for unknown mask waveforms units
- A for amperes
- VA for volt  $\times$  amperes
- AA for amperes  $\times$  amperes
- VV for volts  $\times$  volts

**Examples**      WFMINPRE:YUNIT "A" specifies that the vertical units for the incoming waveform are Amperes.

WFMINPRE:YUNIT? might return :WFMINPRE:YUNIT "V" indicating that the vertical units for the incoming waveform are volts.



## WFMinpre:YZero

Sets or returns the vertical offset of the incoming waveform in units specified by WFMinpre:YUNit. Variations in this number are analogous to changing the vertical offset of the waveform.

**Group** Waveform

**Syntax** WFMinpre:YZero <NR3>  
WFMinpre:YZero?

**Related commands** [WFMinpre:YUNit](#) on page 256, [WFMinpre:YZero?](#) on page 268

**Arguments** <NR3> is offset, expressed in YUNits.

**Examples** WFMINPRE:YZERO 1.5E+0 specifies that the zero reference point for the incoming waveform is 1.5 V below the center of the data range (given that WFMinpre:YUNit is set to V).  
WFMINPRE:YZERO? might return :WFMINPRE:YZero 7.5000E-6 indicating that the zero reference for the incoming waveform is 7.5  $\mu$ V below the center of the data range (given that WFMinpre:YUNit is set to V).

## WFMOutpre?

Returns waveform transmission and formatting settings for the waveform specified by the DATA:SOURce command. Query only.

If the waveform specified by the DATA:SOURce command is not displayed, the instrument returns only the waveform transmission parameters (BYT\_Nr, BIT\_Nr, ENCDg, BN\_Fmt, BYT\_Or).

**Group**      Waveform

**Syntax**     WFMOutpre?

**Examples**    WFMOUTPRE? might return the waveform formatting data as:  
:WFMOUTPRE:BYT\_NR 2;BIT\_NR 16;ENCDG ASCII;BN\_FMT  
RI;BYT\_ORMSB;WFID "Ch1, DC coupling, 100.0mV/div, 4.000us/div,  
10000 points, Sample mode";NR\_PT 2000;PT\_FMT Y;XUNIT "s";XINCR  
4.0000E-9;XZERO - 20.0000E-6;PT\_OFF 0;YUNIT "V";YMULT  
15.6250E-6;YOFF : "6.4000E+3;YZERO 0.0000.

## WFMOutpre:BIT\_Nr

Sets and queries the number of bits per waveform point that outgoing waveforms contain, as specified by the DATA:SOURce command. Changing the value of WFMOutpre:BIT\_Nr also changes the values of WFMInpre:FILTERFreq and DATA:WIDTH.

**Group**      Waveform

**Syntax**    WFMOutpre:BIT\_Nr <NR1>  
               WFMOutpre:BIT\_Nr?

**Related commands**    [DATA:SOURce](#) on page 89, [DATA:WIDth](#) on page 92

**Arguments**    <NR1> is the number of bits per data point and can be 8 or 16.

**Examples**    WFMOUTPRE:BIT\_NR 16 sets the number of bits per waveform point to 16 for outgoing waveforms.  
                   WFMOUTPRE:BIT\_NR? might return :WFMOUTPRE:BIT\_NR 8 indicating that outgoing waveforms use 8 bits per waveform point.

## WFMOutpre:BN\_Fmt

Sets or returns the format of binary data for outgoing waveforms specified by the DATA:SOURce command. Changing the value of WFMOutpre:BN\_Fmt also changes the value of DATA:ENCdg.

**Group**    Waveform

**Syntax**    WFMOutpre:BN\_Fmt {RI|RP}  
               WFMOutpre:BN\_Fmt?

**Arguments**    RI specifies signed integer data point representation.  
                   RP specifies positive integer data point representation.

- Examples**      WFMOUTPRE:BN\_FMT RP specifies that outgoing waveform data will be in positive integer format.
- WFMOUTPRE:BN\_FMT? might return :WFMOUTPRE:BN\_FMT RI indicating that the outgoing waveform data is currently in signed integer format.

## WFMOutpre:BYT\_Nr

Sets or queries the data width for the outgoing waveform specified by the DATA:SOURce command. Changing WFMOutpre:BYT\_Nr also changes WFMOutpre:BIT\_Nr and DATA:WIDth.

- Group**      Waveform
- Syntax**      WFMOutpre:BYT\_Nr <NR1>  
WFMOutpre:BYT\_Nr?
- Related commands**      [DATA:SOURce](#) on page 89, [DATA:WIDth](#) on page 92, [WFMOutpre:BIT\\_Nr](#) on page 258
- Arguments**      <NR1> is the number of bytes per data point and can be 1 or 2.
- Examples**      WFMOUTPRE:BYT\_NR 1 sets the number of bytes per outgoing waveform data point to 1, which is the default setting.
- WFMOUTPRE:BYT\_NR? might return :WFMOUTPRE:BYT\_NR 2 indicating that there are 2 bytes per outgoing waveform data point.

## WFMOutpre:ENCdg

Sets and queries the type of encoding for outgoing waveforms.

**Group**     Waveform

**Syntax**     WFMOutpre:ENCdg {ASCIi|BINary}  
WFMOutpre:ENCdg?

**Related commands**     [WFMOutpre:BYT\\_Nr](#) on page 260, [WFMOutpre:BIT\\_Nr](#) on page 258

**Arguments**     ASCIi specifies that the outgoing data is to be in ASCII format. Waveforms will be sent as <NR1> numbers.  
  
BINary specifies that outgoing data is to be in a binary format whose further specification is determined by WFMOutpre:BYT\_Nr, WFMOutpre:BIT\_Nr, WFMOutpre:BN\_Fmt and WFMInpre:FILTERFreq.

**Examples**     WFMOutpre:ENCdg ASCIi sets the encoding to ASCII.  
  
WFMOutpre:ENCdg? might return WFMOutpre:ENCdg BINARY indicating the encoding is set to binary.

## WFMOutpre:NR\_Pt?

Returns the number of points for the DATA:SOURce waveform that will be transmitted in response to a CURVe? query. The query command will timeout and an error will be generated if the waveform specified by DATA:SOURce is not turned on. Query only.

**Group**      Waveform

**Syntax**     WFMOutpre:NR\_Pt?

**Related commands**    [CURVe](#) on page 85, [DATA](#) on page 87, [DATA:START](#) on page 90, [DATA:STOP](#) on page 91, [SAVE:WAVEform](#) on page 208, [SAVE:WAVEform:FILEFormat](#) on page 209, [WFMinpre:NR\\_Pt?](#) on page 251

**Examples**      WFMOUTPRE:NR\_PT? might return :WFMOUTPRE:NR\_PT 2000 indicating that there are 2000 data points to be sent.

## WFMOutpre:RECOrdlength?

Returns the record length for the source waveform as specified by the DATA:SOURce command. Query only.

**Group**      Waveform

**Syntax**     WFMOutpre:RECOrdlength?

**Examples**     WFMOUTPRE:RECORDLENGTH? might return :WFMOUTPRE:RECORDLENGTH 2000 indicating that 2000 is the source waveform record length.

## WFMOutpre:WFId?

Returns a string describing several aspects of the acquisition parameters for the waveform specified by the DATA:SOURce command. The query command will time out and an error will be generated if the waveform specified by DATA:SOURce is not turned on. Query only.

**Group**     Waveform

**Syntax**     WFMOutpre:WFId?

**Related commands**     [DATA:SOURce](#) on page 89

**Returns**     <QString> comprises the following comma-separated fields:

Source The source identification string as it appears in the front-panel scale factor readouts.

Coupling A string describing the vertical coupling of the waveform.

Vert Scale A string containing the vertical scale factor of the unzoomed waveform. The numeric portion will always be four digits. The examples cover all known internal units.

Horiz Scale A string containing the horizontal scale factor of the unzoomed waveform. The numeric portion will always be four digits. The examples cover all known internal units.

Record Length A string containing the number of waveform points available in the entire record. The numeric portion is given as an integer.

Acquisition Mode A string describing the mode used to acquire the waveform.

**Examples**      WFMOUTPRE:WFID? might return :WFMOUTPRE:WFID "Ch1, DC coupling, 100.0mVolts/div,500.0µs/div, 1000 points, Sample mode".

## WFMOutpre:XINcr?

Returns the horizontal point spacing in units of WFMOutpre:XUNit for the waveform specified by the DATA:SOURce command. This value corresponds to the sampling interval. The query command will time out and an error will be generated if the waveform specified by DATA:SOURce is not turned on. Query only.

**Group**      Waveform

**Syntax**      WFMOutpre:XINcr?

**Related commands**      [DATA:SOURce](#) on page 89, [WFMOutpre:XUNit?](#) on page 264

**Examples**      WFMOUTPRE:XINCR? might return :WFMOUTPRE:XINCR 10.0000E-6 indicating that the horizontal sampling interval is 10 µs/point.

## WFMOutpre:XUNit?

Returns the horizontal units for the waveform specified by the DATA:SOURce command. The query command will time out and an error will be generated if the waveform specified by DATA:SOURce is not turned on. Query only.

**Group**      Waveform



**Syntax**      WFMOutpre:XUNit?

**Related commands**    [DATA:SOURce](#) on page 89

**Examples**      WFMOUTPRE:XUNIT? might return :WFMOUTPRE:XUNIT "HZ" indicating that the horizontal units for the waveform are in Hertz.

## WFMOutpre:XZEro?

Returns the time coordinate of the first point in the outgoing waveform. This value is in units of WFMOutpre:XUNit?. The query command will time out and an error will be generated if the waveform specified by DATA:SOURce is not turned on. Query only.

**Group**      Waveform

**Syntax**      WFMOutpre:XZEro?

**Related commands**    [DATA:SOURce](#) on page 89, [WFMOutpre:XUNit?](#) on page 264

**Examples**      WFMOUTPRE:XZERO? might return :WFMOUTPRE:XZERO 5.6300E-9 indicating that the trigger occurred 5.63 ns before the first sample in the waveform record.

## WFMOutpre:YMUlt?

Returns the vertical scale factor per digitizing level in units specified by WFMOutpre:YUNit for the waveform specified by the Returns the vertical scale factor per digitizing level in units specified by WFMOutpre:YUNit for the waveform specified by the DATA:SOURce command. The query command will time out and an error is generated if the waveform specified by DATA:SOURce is not turned on. command. The query command will time out and an error is generated if the waveform specified by DATA:SOURce is not turned on. (Query Only)

**Group** Waveform

**Syntax** WFMOutpre:YMUlt?

**Related commands** [DATA:SOURce](#) on page 89, [WFMinpre:YMUlt](#) on page 254

**Examples** WFMOUTPRE:YMULT? might return :WFMOUTPRE:YMULT 4.0000E-3 indicating that the vertical scale for the corresponding waveform is 100 mV/div (for 8-bit waveform data).

## WFMOutpre:YOFf?

Returns the vertical position in digitizing levels for the waveform specified by the DATA:SOURce command. The query command will time out and an error will be generated if the waveform specified by DATA:SOURce is not turned on. Query only.

**Group** Waveform

**Syntax** WFMOutpre:YOFF?

**Related commands** [DATA:SOURce](#) on page 89, [WFMOutpre:BYT\\_Nr](#) on page 260

**Examples** WFMOUTPRE:YOFF? might return :WFMOUTPRE:YOFF -50.0000E+0 indicating that the position indicator for the waveform was 50 digitizing levels (2 divisions) below center screen (for 8-bit waveform data).

## WFMOutpre:YUNit?

Returns the vertical units for the waveform specified by the DATA:SOURce command. The query command will time out and an error will be generated if the waveform specified by DATA:SOURce is not turned on. Query only.

**Group** Waveform

**Syntax** WFMOutpre:YUNit?

**Related commands** [DATA:SOURce](#) on page 89

**Examples** WFMOUTPRE:YUNIT? might return :WFMOUTPRE:YUNIT "dB" indicating that the vertical units for the waveform are measured in decibels.

## WFMOutpre:YZero?

Returns the vertical offset in units specified by WFMOutpre:YUNit? for the waveform specified by the DATA:SOURce command. The query command will time out and an error will be generated if the waveform specified by DATA:SOURce is not turned on. Query only.

**Group** Waveform

**Syntax** WFMOutpre:YZero?

**Related commands** [DATA:SOURce](#) on page 89, [WFMOutpre:YUNit?](#) on page 267

**Examples** WFMOUTPRE:YZERO? might return :WFMOUTPRE:YZERO -100.0000E-3 indicating that vertical offset is set to –100 mV.

---

## Z commands

This section lists commands and queries that begin with the letter Z.

### ZOOM?

Returns the current vertical and horizontal positioning and scaling of the display.  
Query only.

**Group** Zoom

**Syntax** ZOOM?

**Examples** ZOOM? might return :ZOOM:MODE 1; :ZOOM:ZOOM1:STATE 1;SCALE  
20.0000E-9;POSITION 50.0000; FACTOR 10.0000;  
HORIZONTAL:POSITION 50.0000;SCALE 20.0000E-9.

### ZOOM{:MODE|:STATE}

Turns Zoom mode on or off. The Zoom mode query returns the current state of Zoom mode.

This command is equivalent to pressing the zoom button located on the front panel.

**Group** Zoom

**Syntax**    ZOOM{:MODE|:STATE} {ON|OFF|<NR1>}  
             ZOOM{:MODE|:STATE}?

**Arguments**    ON turns on Zoom mode.  
                 OFF turns off Zoom mode.  
                 <NR1> = 0 turns off Zoom mode; any other value turns on Zoom mode.

**Examples**    ZOOM:MODE OFF turns off Zoom mode.  
                 ZOOM:MODE? might return :ZOOM:MODE 1 indicating that Zoom mode is currently turned on.

## ZOOM:ZOOM1?

Returns the current horizontal positioning and scaling of the display. Query only.

**Group**        Zoom

**Syntax**        ZOOM:ZOOM1?

**Examples**    ZOOM:ZOOM1? might return :ZOOM:ZOOM1:STATE 1;SCALE  
                 20.0000E-9;POSITION 50.0000;FACTOR 10.0000;HORIZONTAL:POSITION  
                 50.0000;SCALE 20.0000E-9.

## ZOOM:ZOOM1:FACtor

Queries the zoom factor of a particular zoom box. Query only.

**Group** Zoom

**Syntax** ZOOM:ZOOM1:FACtor?

**Returns** <NR1> is the zoom factor of a zoom box.

**Examples** ZOOM:ZOOM1:FACtor? might return :ZOOM:ZOOM1:FACtor X5 indicating that the specified zoom factor is X5 of the acquired waveform.

## ZOOM:ZOOM1:HORizontal:POSition

Sets or queries the horizontal position of a specified zoom box.

**Group** Zoom

**Syntax** ZOOM:ZOOM1:HORizontal:POSition <NR1>  
ZOOM:ZOOM1:HORizontal:POSition?

**Arguments** <NR1> is 1 to 100.00 and is the percent of the upper window that is to the left of the screen center, when the zoom factor is 1× or greater.

**Examples**     `ZOOM:ZOOM1:HORizontal:POSition 50.00` sets the zoom reference pointer at 50% of the acquired waveform.

`ZOOM:ZOOM1:HORIZONTAL:POSITION?` might return `:ZOOM1:HORIZONTAL:POSITION 50.00`, indicating the reference pointer is at 50% of the acquired waveform.

## **ZOOM:ZOOM1:HORizontal:SCAle**

Sets or queries the zoom horizontal scale for the specified zoom.

**Group**     Zoom

**Syntax**     `ZOOMm:ZOOM1:HORizontal:SCAle <NR3>`  
`ZOOMm:ZOOM1:HORizontal:SCAle?`

**Arguments**     `<NR3>` is the amount of expansion in the horizontal direction and ranges from 1.0E-9 to 100.0.

**Examples**     `ZOOM:ZOOM1:HORIZONTAL:SCALE 5` sets the horizontal scale to 5 seconds per division.

`ZOOM:ZOOM2:HORIZONTAL:SCALE?` might return `:ZOOM2:HORIZONTAL:SCALE 1`, indicating that the horizontal scale is 1 second per division.



## ZOOm:ZOOM1:POSition

Sets or queries the horizontal position for the specified zoom.

**Group** Zoom

**Syntax** ZOOm:ZOOM1:POSition <NR3>  
ZOOm:ZOOM1:POSition?

**Arguments** <NR3> is a value from 0 to 100.00 and is the percent of the upper window that is to the left of screen center, when the zoom factor is 1× or greater

**Examples** ZOOm:ZOOM1:POSition 20 sets the percent of the upper window that is to the left of screen center to 20%.  
ZOOm:ZOOM1:POSition? might return :ZOOm:ZOOM1:POSition 25 indicating the percent of the upper window that is to the left of the screen center is 25%.

## ZOOm:ZOOM1:SCAle

Sets or queries the zoom horizontal scale for the specified zoom.

**Group** Zoom

**Syntax** ZOOm:ZOOM1:SCAle <NR3>  
ZOOm:ZOOM1:SCAle?

<b>Arguments</b>	<NR3> is the amount of expansion in the horizontal direction and ranges from 1.0E-9 to 100.0.
<b>Examples</b>	<p>ZOOM:ZOOM1:SCALE 5.0 sets the horizontal expansion of the specified Zoom to 5.</p> <p>ZOOM:ZOOM1:SCALE? might return ZOOM:ZOOM1:SCALE 5.0 indicating the zoom1 horizontal expansion is set to 5.</p>

## ZOOM:ZOOM1:STATE

Sets or queries the specified zoom on or off, where x is the integer representing the specified zoom window.

<b>Group</b>	Zoom
<b>Syntax</b>	<p>ZOOM:ZOOM1:STATE {ON OFF &lt;NR1&gt;}</p> <p>ZOOM:ZOOM1:STATE?</p>
<b>Arguments</b>	<p>ON turns the specified Zoom on.</p> <p>OFF turns the specified Zoom off.</p> <p>&lt;NR1&gt; = 0 disables the specified zoom; any other value enables the specified zoom</p>
<b>Examples</b>	<p>ZOOM:ZOOM1:STATE ON turns Zoom1 on.</p> <p>ZOOM:ZOOM1:STATE? might return :ZOOM:ZOOM1:STATE 1 indicating that Zoom1 is on.</p>

---

# Status and Events

The instrument provides a status and event reporting system for the GPIB, RS-232, and USB interfaces. This system informs you of certain significant events that occur within the instrument.

The instrument status reporting system consists of five 8-bit registers and two queues. This section describes these registers and components, and explains how the event handling system operates.

## Registers

- Overview** The registers in the event handling system fall into two functional groups:
- Status Registers contain information about the status of the instrument. They include the Standard Event Status Register (SESR).
  - Enable Registers determine whether selected types of events are reported to the Status Registers and the Event Queue. They include the Device Event Status Enable Register (DESER), the Event Status Enable Register (ESER), and the Service Request Enable Register (SRER).

**Status Registers** The Standard Event Status Register (SESR) and the Status Byte Register (SBR) record certain types of events that may occur while the instrument is in use. IEEE Std 488.2-1987 defines these registers.

Each bit in a Status Register records a particular type of event, such as an execution error or message available. When an event of a given type occurs, the instrument sets the bit that represents that type of event to a value of one. (You can disable bits so that they ignore events and remain at zero. See Enable Registers). Reading the status registers tells you what types of events have occurred.

**The Standard Event Status Register (SESR).** The SESR records eight types of events that can occur within the instrument. Use the \*ESR? query to read the SESR register. Reading the register clears the bits of the register so that the register can accumulate information about new events.

7	6	5	4	3	2	1	0
PON	URQ	CME	EXE	DDE	QYE	RQC	OPC

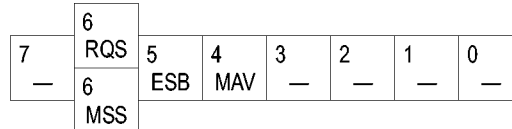
**Figure 3: The Standard Event Status Register (SESR)**

**Table 28: SESR bit functions**

Bit	Function	
7 (MSB)	PON	Power On. Shows that the instrument was powered on. On completion, the diagnostic self tests also set this bit.
6	URQ	User Request. Indicates that an application event has occurred. *See note.
5	CME	Command Error. Shows that an error occurred while the instrument was parsing a command or query.
4	EXE	Execution Error. Shows that an error executing a command or query.
3	DDE	Device Error. Shows that a device error occurred.
2	QYE	Query Error. Either an attempt was made to read the Output Queue when no data was present or pending, or that data in the Output Queue was lost.
1	RQC	Request Control. This is not used.
0 (LSB)	OPC	Operation Complete. Shows that the operation is complete. This bit is set when all pending operations complete following an *OPC command.

**The Status Byte Register (SBR).** Records whether output is available in the Output Queue, whether the instrument requests service, and whether the SESR has recorded any events.

Use a Serial Poll or the \*STB? query to read the contents of the SBR. The bits in the SBR are set and cleared depending on the contents of the SESR, the Event Status Enable Register (ESER), and the Output Queue. When you use a Serial Poll to obtain the SBR, bit 6 is the RQS bit. When you use the \*STB? query to obtain the SBR, bit 6 is the MSS bit. Reading the SBR does not clear the bits.



**Figure 4: The Status Byte Register (SBR)**

**Table 29: SBR bit functions**

Bit	Function	
7 (MSB)	-----	Not used.
6	RQS	Request Service. Obtained from a serial poll. Shows that the instrument requests service from the GPIB controller.
6	MSS	Master Status Summary. Obtained from *STB? query. Summarizes the ESB and MAV bits in the SBR.
5	ESB	Event Status Bit. Shows that status is enabled and present in the SESR.
4	MAV	Message Available. Shows that output is available in the Output Queue.
3	-----	Not used.
2	-----	Not used.
1–0	-----	Not used.

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 a bit in the Status Register, the corresponding bit in the Enable Register must be set to one. If the bit in the Enable Register is set to zero, the event is not recorded.

Various commands set the bits in the Enable Registers. The Enable Registers and the commands used to set them are described below.

**The Device Event Status Enable Register (DESER).** This register controls which types of events are reported to the SESR and the Event Queue. The bits in the DESER correspond to those in the SESR.

Use the DESE command to enable and disable the bits in the DESER. Use the DESE? query to read the DESER.

7	6	5	4	3	2	1	0
PON	URQ	CME	EXE	DDE	QYE	RQC	OPC

Figure 5: The Device Event Status Enable Register (DESER)

**The Event Status Enable Register (ESER).** This register controls which types of events are summarized by the Event Status Bit (ESB) in the SBR. Use the \*ESE command to set the bits in the ESER. Use the \*ESE? query to read it.

7	6	5	4	3	2	1	0
PON	URQ	CME	EXE	DDE	QYE	RQC	OPC

Figure 6: The Event Status Enable Register (ESER)

**The Service Request Enable Register (SRER).** This register controls which bits in the SBR generate a Service Request and are summarized by the Master Status Summary (MSS) bit.

Use the \*SRE command to set the SRER. Use the \*SRE? query to read the register. The RQS bit remains set to one until either the Status Byte Register is read with a Serial Poll or the MSS bit changes back to a zero.

7	6	5	4	3	2	1	0
—	—	ESB	MAV	—	—	—	—

Figure 7: The Service Request Enable Register (SRER)

**\*PSC Command** The \*PSC command controls the Enable Registers contents at power-on. Sending \*PSC 1 sets the Enable Registers at power on as follows:

- DESER 255 (equivalent to a DESe 255 command)
- ESER 0 (equivalent to an \*ESE 0 command)
- SRER 0 (equivalent to an \*SRE 0 command)

Sending \*PSC 0 lets the Enable Registers maintain their values in nonvolatile memory through a power cycle.

---

**NOTE.** To enable the PON (Power On) event to generate a Service Request, send \*PSC 0, use the DESe and \*ESE commands to enable PON in the DESER and ESER, and use the \*SRE command to enable bit 5 in the SRER. Subsequent power-on cycles will generate a Service Request.

---

## Queues

The \*PSC command controls the Enable Registers contents at power-on. Sending \*PSC 1 sets the Enable Registers at power on as follows:

**Output Queue** The instrument stores query responses in the Output Queue and empties this queue each time it receives a new command or query message after an <EOM>. The controller must read a query response before it sends the next command (or query) or it will lose responses to earlier queries.




---

**CAUTION.** When a controller sends a query, an <EOM>, and a second query, the instrument normally clears the first response and outputs the second while reporting a Query Error (QYE bit in the ESER) to indicate the lost response. A fast controller, however, may receive a part or all of the first response as well. To avoid this situation, the controller should always read the response immediately after sending any terminated query message or send a DCL (Device Clear) before sending the second query.

---

**Event Queue** The Event Queue stores detailed information on up to 20 events. If than 20 events stack up in the Event Queue, the 20th event is replaced by event code 350, "Queue Overflow."

Read the Event Queue with the EVENT? query (which returns only the event number), with the EVMSG? query (which returns the event number and a text description of the event), or with the ALLEV? query (which returns all the event numbers with a description of the event). Reading an event removes it from the queue.

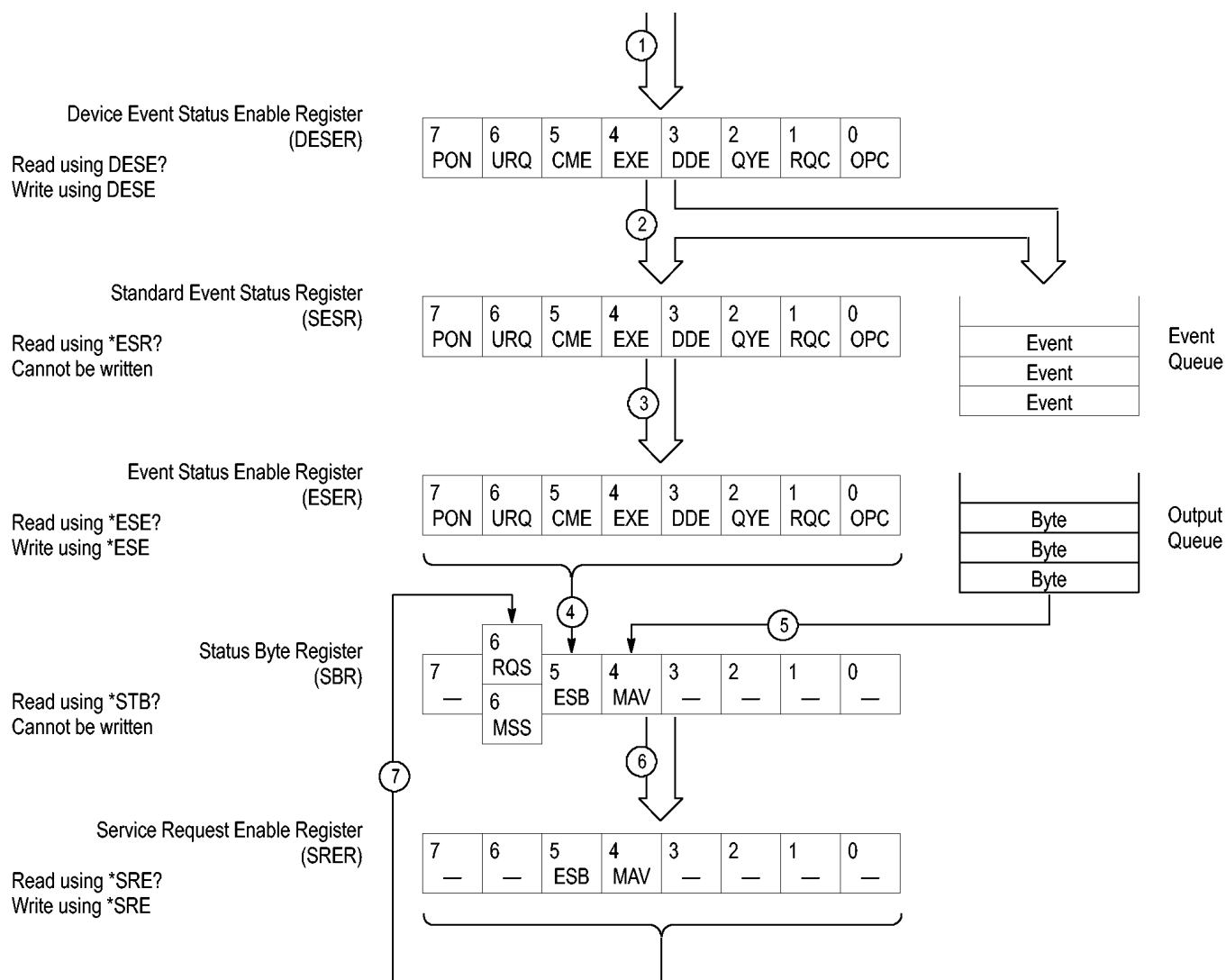
Before reading an event from the Event Queue, you must use the \*ESR? query to read the summary of the event from the SESR. This makes the events summarized by the \*ESR? read available to the EVENT? and EVMSG? queries, and empties the SESR.

Reading the SESR erases any events that were summarized by previous \*ESR? reads but not read from the Event Queue. Events that follow an \*ESR? read are put in the Event Queue but are not available until \*ESR? is used again.

## Event Handling Sequence

The following figure shows how to use the status and event handling system. In the explanation that follows, numbers in parentheses refer to numbers in the figure.





**Figure 8: Status and Event Handling Process**

When an event occurs, a signal is sent to the DESER (1). If that type of event is enabled in the DESER (that is, if the bit for that event type is set to 1), the appropriate bit in the SESR is set to one, and the event is recorded in the Event Queue (2). If the corresponding bit in the ESER is also enabled (3), then the ESB bit in the SBR is set to one (4).

When output is sent to the Output Queue, the MAV bit in the SBR is set to one (5).

When a bit in the SBR is set to one and the corresponding bit in the SRER is enabled (6), the MSS bit in the SBR is set to one and a service request is generated (7).

## Synchronization Methods

**Overview** Although most commands are completed almost immediately after being received by the instrument, some commands start a process that requires time. For example, once a single sequence acquisition command is executed, depending upon the applied signals and trigger settings, it may take an extended period of time before the acquisition is complete. Rather than remain idle while the operation is in process, the instrument will continue processing other commands. This means that some operations will not be completed in the order that they were sent.

Sometimes the result of an operation depends on the result of an earlier operation. A first operation must complete before the next one is processed. The instrument status and event reporting system is designed to accommodate this process.

The Operation Complete (OPC) bit of the Standard Event Status Register (SESR) can be programmed to indicate when certain instrument operations have completed and, by setting the Event Status Enable Register (ESER) to report OPC in the Event Status Bit (ESB) of the Status Byte Register (SBR) and setting the Service Request Enable Register (SRER) to generate service request upon a positive transition of the ESB, a service request (SRQ) interrupt can be generated when certain operations complete as described in this section.

The following instrument operations can generate an OPC:

**Table 30: Instrument operations that can generate OPC**

Command	Conditions
ACQuire:STATE ON or ACQuire:STATE RUN	Only when ACQuire:STOPAfter is set to SEquence
*CAL?	
CALibrate:CONTINUE	
CALibrate:FActory	
CALibrate:INTERNAL	
FActory	
HARDCopy START	
RECALL:SETUp <file as quoted string>	
RECALL:WAVEform <file as quoted string>	
*RST	
SAVe:IMAGe <file as quoted string>	
SAVe:SETUp <file as quoted string>	
SAVe:WAVEform <file as quoted string>	

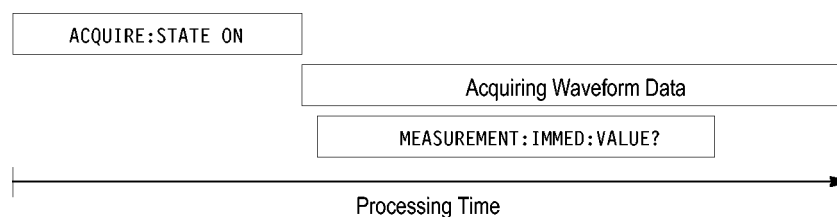
For example, a typical application might involve acquiring a single-sequence waveform and then taking a measurement on the acquired waveform. You could use the following command sequence to do this:

```

/** Set up conditional acquisition **/
ACQUIRE:STATE OFF
SELECT:CH1 ON
HORIZONTAL:RECORDLENGTH 1000
ACQUIRE:MODE SAMPLE
ACQUIRE:STOPAFTER SEQUENCE
/** Acquire waveform data **/
ACQUIRE:STATE ON
/** Set up the measurement parameters **/
MEASUREMENT:IMMED:TYPE AMPLITUDE
MEASUREMENT:IMMED:SOURCE CH 1
/** Take amplitude measurement **/
MEASUREMENT:MEAS1:VALUE?

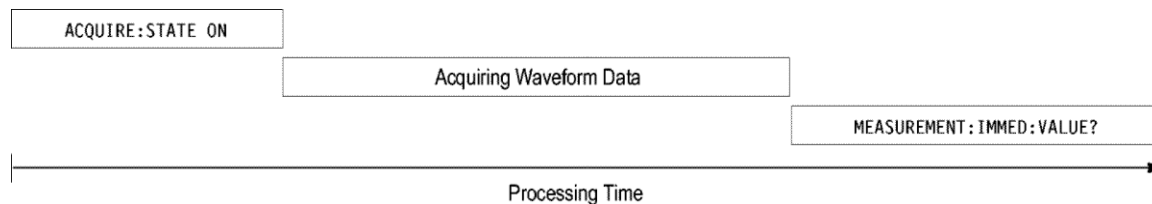
```

The acquisition of the waveform requires extended processing time. It may not finish before the instrument takes an amplitude measurement (see the following figure). This can result in an incorrect amplitude value.



**Figure 9: Command processing without using synchronization**

To be sure the instrument completes waveform acquisition before taking the measurement on the acquired data, you can synchronize the program.



**Figure 10: Processing sequence with synchronization**

You can use four commands to synchronize the operation of the instrument with your application program: \*WAI, BUSY, \*OPC, and \*OPC?

**Using the \*WAI Command**

The \*WAI command forces completion of previous commands that generate an OPC message. No commands after the \*WAI are processed before the OPC message(s) are generated

The same command sequence using the \*WAI command for synchronization looks like this:

```
/* Set up conditional acquisition */
ACQUIRE:STATE OFF
SELECT:CH1 ON
ACQUIRE:MODE SAMPLE
ACQUIRE:STOPAFTER SEQUENCE
/* Acquire waveform data */
ACQUIRE:STATE ON
/* Set up the measurement parameters */
MEASUREMENT:IMMED:TYPE AMPLITUDE
MEASUREMENT:IMMED:SOURCE CH1
/* Wait until the acquisition is complete before taking the measurement*/
*/
*WAI
/* Take amplitude measurement */
MEASUREMENT:IMMED:VALUE?
```

The controller can continue to write commands to the input buffer of the instrument, but the commands will not be processed by the instrument until all in-process OPC operations are complete. If the input buffer becomes full, the controller will be unable to write commands to the buffer. This can cause a time-out.

**Using the BUSY Query**

The BUSY? query allows you to find out whether the instrument is busy processing a command that has an extended processing time such as single-sequence acquisition.

The same command sequence, using the BUSY? query for synchronization, looks like this:

```
/* Set up conditional acquisition */
ACQUIRE:STATE OFF
SELECT:CH1 ON
ACQUIRE:MODE SAMPLE
ACQUIRE:STOPAFTER SEQUENCE
```

```

/* Acquire waveform data */
ACQUIRE:STATE ON
/* Set up the measurement parameters */
MEASUREMENT:IMMED:TYPE AMPLITUDE
MEASUREMENT:IMMED:SOURCE CH1
/* Wait until the acquisition is complete before taking the measurement */
While BUSY? keep looping
/* Take amplitude measurement */
MEASUREMENT:IMMED:VALUE?

```

This sequence lets you create your own wait loop rather than using the \*WAI command. The BUSY? query helps you avoid time-outs caused by writing too many commands to the input buffer. The controller is still tied up though, and the repeated BUSY? query will result in 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 conditional acquisition */
ACQUIRE:STATE OFF
SELECT:CH1 ON
ACQUIRE:MODE SAMPLE
ACQUIRE:STOPAFTER SEQUENCE
/* Enable the status registers */
DESE 1
*ESE 1
*SRE 0
/* Acquire waveform data */

```

```
ACQUIRE:STATE ON
```

```
/* Set up the measurement parameters */
```

```
MEASUREMENT:IMMED:TYPE AMPLITUDE
```

```
MEASUREMENT:IMMED:SOURCE CH1
```

```
/* Wait until the acquisition is complete before taking the measurement.*/
```

```
*OPC
```

```
While serial poll = 0, keep looping
```

```
/* Take amplitude measurement */
```

```
MEASUREMENT:IMMED:VALUE?
```

This technique requires less bus traffic than did looping on BUSY.

**Service Request Method:** Enable the OPC bit in the Device Event Status Enable Register (DESER) and the Event Status Enable Register (ESER) using the DESE and \*ESE commands.

You can also enable service requests by setting the ESB bit in the Service Request Enable Register (SRER) using the \*SRE command. When the operation is complete, the instrument will generate a Service Request.

The same command sequence using the \*OPC command for synchronization looks like this

```
/* Set up conditional acquisition */
```

```
ACQUIRE:STATE OFF
```

```
SELECT:CH1 ON
```

```
ACQUIRE:MODE SAMPLE
```

```
ACQUIRE:STOPAFTER SEQUENCE
```

```
/* Enable the status registers */
```

```
DESE 1
```

```
*ESE 1
```

```
*SRE 32
```

```
/* Acquire waveform data */
```

```
ACQUIRE:STATE ON
```

```
/* Set up the measurement parameters */
```

```
MEASUREMENT:IMMED:TYPE AMPLITUDE
```

```
MEASUREMENT:IMMED:SOURCE CH1
```

```
/* Wait until the acquisition is complete before taking the measurement*/
```

```
*OPC
```

The program can now do different tasks such as talk to other devices. The SRQ, when it comes, interrupts those tasks and returns control to this task.

```
/* Take amplitude measurement */
```

```
MEASUREMENT:IMMED:VALUE?
```

### Using the \*OPC? Query

The \*OPC? query places a 1 in the Output Queue once an operation that generates an OPC message is complete. The \*OPC? query does not return until all pending OPC operations have completed. Therefore, your time-out must be set to a time at least if the longest expected time for the operations to complete.

The same command sequence using the \*OPC? query for synchronization looks like this:

```
/* Set up single sequence acquisition */
```

```
ACQUIRE:STATE OFF
```

```
SELECT:CH1 ON
```

```
ACQUIRE:MODE SAMPLE
```

```
ACQUIRE:STOPAFTER SEQUENCE
```

```
/* Acquire waveform data */
```

```
ACQUIRE:STATE ON
```

```
/* Set up the measurement parameters */
```

```
MEASUREMENT:IMMED:TYPE AMPLITUDE
```

```
MEASUREMENT:IMMED:SOURCE CH1
```

```
/* Wait until the acquisition is complete before taking the measurement*/
```

```
*OPC?
```

Wait for read from Output Queue.

```
/* Take amplitude measurement */
```

```
MEASUREMENT:IMMED:VALUE?
```

This is the simplest approach. It requires no status handling or loops. However, you must set the controller time-out for longer than the acquisition operation.

**Messages** The information contained in the topic tabs above covers all the programming interface messages the instrument generates in response to commands and queries.

For most messages, a secondary message from the instrument gives detail about the cause of the error or the meaning of the message. This message is part of the message string and is separated from the main message by a semicolon.

Each message is the result of an event. Each type of event sets a specific bit in the SESR and is controlled by the equivalent bit in the DESER. Thus, each message is associated with a specific SESR bit. In the message tables, the associated SESR bit is specified in the table title, with exceptions noted with the error message text.

**No Event** The following table shows the messages when the system has no events or status to report. These have no associated SESR bit.

**Table 31: No Event messages**

Code	Message
0	No events to report; queue empty
1	No events to report; new events pending *ESR?

**Command Error** The following table shows the command error messages generated by improper syntax. Check that the command is properly formed and that it follows the rules in the section on command Syntax.

**Table 32: Command error messages (CME bit 5)**

Code	Message
100	Command error
101	Invalid character
102	Syntax error
103	Invalid separator
104	Data type error
105	GET not allowed
108	Parameter not allowed
109	Missing parameter
110	Command header error
112	Program mnemonic too long
113	Undefined header
120	Numeric data error
121	Invalid character in numeric
123	Exponent too large



Code	Message
124	Too many digits
130	Suffix error
131	Invalid suffix
134	Suffix too long
140	Character data error
141	Invalid character data
144	Character data too long
150	String data error
151	Invalid string data
152	String data too long
160	Block data error
161	Invalid block data
170	Command expression error
171	Invalid expression

**Execution Error**

The following table lists the execution errors that are detected during execution of a command.

**Table 33: Execution error messages (EXE bit 4)**

Code	Message
200	Execution error
221	Settings conflict
222	Data out of range
224	Illegal parameter value
241	Hardware missing
250	Mass storage error
251	Missing mass storage
252	Missing media
253	Corrupt media
254	Media full
255	Directory full
256	File name not found
257	File name error
258	Media protected
259	File name too long
270	Hardcopy error
271	Hardcopy device not responding
272	Hardcopy is busy

Code	Message
273	Hardcopy aborted
274	Hardcopy configuration error
280	Program error
282	Insufficient network printer information
283	Network printer not responding
284	Network printer server not responding
286	Program run time error
287	Print server not found
2200	Measurement error, Measurement system error
2201	Measurement error, Zero period
2202	Measurement error, No period, second waveform
2203	Measurement error, No period, second waveform
2204	Measurement error, Low amplitude, second waveform
2205	Measurement error, Low amplitude, second waveform
2206	Measurement error, Invalid gate
2207	Measurement error, Measurement overflow
2208	Measurement error, No backward Mid Ref crossing
2209	Measurement error, No second Mid Ref crossing
2210	Measurement error, No Mid Ref crossing, second waveform
2211	Measurement error, No backward Mid Ref crossing
2212	Measurement error, No negative crossing
2213	Measurement error, No positive crossing
2214	Measurement error, No crossing, target waveform
2215	Measurement error, No crossing, second waveform
2216	Measurement error, No crossing, target waveform
2217	Measurement error, Constant waveform
2219	Measurement error, No valid edge - No arm sample
2220	Measurement error, No valid edge - No arm cross
2221	Measurement error, No valid edge - No trigger cross

Code	Message
2222	Measurement error, No valid edge - No second cross
2223	Measurement error, Waveform mismatch
2224	Measurement error, WAIT calculating
2225	Measurement error, No waveform to measure
2226	Measurement error, Null Waveform
2227	Measurement error, Positive and Negative Clipping
2228	Measurement error, Positive Clipping
2229	Measurement error, Negative Clipping
2230	Measurement error, High Ref < Low Ref
2231	Measurement error, No statistics available
2233	Requested waveform is temporarily unavailable
2235	Math error, invalid math description
2240	Invalid password
2241	Waveform requested is invalid
2244	Source waveform is not active
2245	Saveref error, selected channel is turned off
2250	Reference error, the reference waveform file is invalid
2253	Reference error, too many points received
2254	Reference error, too few points received
2259	File too big
2260	Calibration error
2270	Alias error
2271	Alias syntax error
2273	Illegal alias label
2276	Alias expansion error
2277	Alias redefinition not allowed
2278	Alias header not found
2285	TekSecure(R) Pass
2286	TekSecure(R) Fail
2301	Cursor error, Off screen
2302	Cursor error, Cursors are off
2303	Cursor error, Cursor source waveform is off
2500	Setup error, file does not look like a setup file
2501	Setup warning, could not recall all values from external setup
2620	Mask error, too few points received

Code	Message
2760	Mark limit reached
2761	No mark present
2762	Search copy failed

**Device Error** The following table lists the device errors that can occur during instrument operation. These errors may indicate that the instrument needs repair.

**Table 34: Device error messages (DDE bit 3)**

Code	Message
310	System error
311	Memory error
312	PUD memory lost
314	Save/recall memory lost

**System Event** The following table lists the system event messages. These messages are generated whenever certain system conditions occur.

**Table 35: System event messages**

Code	Message
400	Query event
401	Power on (PON bit 7 set)
402	Operation complete (OPC bit 0 set)
403	User request (URQ bit 6 set)
404	Power fail (DDE bit 3 set)
405	Request control
410	Query INTERRUPTED (QYE bit 2 set)
420	Query UNTERMINATED (QYE bit 2 set)
430	Query DEADLOCKED (QYE bit 2 set)
440	Query UNTERMINATED after indefinite response (QYE bit 2 set)
468	Knob/Keypad value changed
472	Application variable changed

**Execution Warning** The following table lists warning messages that do not interrupt the flow of command execution. They also notify you of a possible unexpected results.

**Table 36: Execution warning messages (EXE bit 4)**

Code	Message
528	Parameter out of range
532	Curve data too long, Curve truncated
533	Curve error, Preamble values are inconsistent
540	Measurement warning, Uncertain edge
541	Measurement warning, Low signal amplitude
542	Measurement warning, Unstable histogram
543	Measurement warning, Low resolution
544	Measurement warning, Uncertain edge
545	Measurement warning, Invalid in minmax
546	Measurement warning, Need 3 edges
547	Measurement warning, Clipping positive/negative
548	Measurement warning, Clipping positive
549	Measurement warning, Clipping negative

**Table 37: Execution warning messages (EXE bit 4)**

Code	Message
540	Measurement warning
541	Measurement warning, Low signal amplitude
542	Measurement warning, Unstable histogram
543	Measurement warning, Low resolution
544	Measurement warning, Uncertain edge
545	Measurement warning, Invalid min max
546	Measurement warning, Need 3 edges
547	Measurement warning, Clipping positive/negative
548	Measurement warning, Clipping positive
549	Measurement warning, Clipping negative

**Internal Warning** The following table shows internal errors that indicate an internal fault in the instrument.

**Table 38: Internal warning messages**

Code	Message
600	Internal warning



---

# Programming Examples

The following series of commands and queries illustrate many of the most common commands and techniques.

To use these commands and queries over USB, you must use a program or routines that interface to the USBTMC driver on your PC. You can also use the PC Communications software that came on the CD with your instrument to get the same data without having to write programs. For operating information, you can launch the PC Communications software and refer to the online help.

To use these commands and queries over GPIB, you must use a program or routines that interface to the GPIB hardware in your computer. The software is usually supplied by the GPIB hardware manufacturer.

In these examples, data sent from the controller computer to the instrument is prefaced with the > symbol. Replies from the instrument have no preface.

```
> REM "Check for any messages, and clear them from the queue."  
> *ESR?  
128  
> ALLEV ?  
:ALLEV 401,"Power on; "  
> REM "Set the instrument to the default state."  
> FACTORY  
> REM "Set the instrument parameters that differ from the defaults."  
> CH1:VOLTS 2.0  
> HOR:MAIN:SCALE 100e-6  
> TRIG:MAIN:LEVEL 2.4  
> REM "Start a single sequence acquisition."  
> ACQUIRE:STOPAFTER SEQUENCE  
> ACQUIRE:STATE ON  
> REM "Wait for the acquisition to complete."  
> REM "Note: your controller program time-out must be set long enough to  
handle the wait."  
> *OPC?  
1  
> REM "Use the instrument built-in measurements to measure the waveform you  
acquired."  
> MEASU:IMMED:TYPE MEAN
```

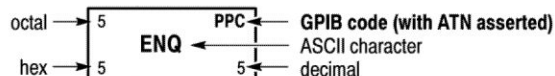
```
> MEASU:IMMED:VALUE?
:MEASUREMENT:IMMED:VALUE 2.4631931782E0
> REM "Be sure to use the *esr? query to check for measurement errors."
> MEASU:IMMED:TYPE FREQ
> MEASU:IMMED:VALUE?
:MEASUREMENT:IMMED:VALUE 9.9E37
> *ESR?
16
> ALLEV?
:ALLEV 2202,"Measurement error, No period found; "
> REM "Query out the waveform points, for later analysis on your controller
computer." > data:encdg ascii
> CURVE?
:CURVE 7,6,5,5,5,6,6,8 [...]
> REM "Query out the parameters used for calculating the times and voltages of
the waveform points."
> WFMPRE?
:WFMPRE:BYT_NR 1;BIT_NR 8;ENCDG ASC;BN_FMT RP;BYT_OR
MSB;NR_PT 2500; [...]
```



# ASCII Code Chart

B7 B6 B5 BITS B4 B3 B2 B1	0 0 0		0 0 1		0 1 0		0 1 1		1 0 0		1 0 1		1 1 0		1 1 1					
	CONTROL				NUMBERS SYMBOLS				UPPER CASE				LOWER CASE							
0 0 0 0	0 0	NUL	20 10	DLE	40 20	SP	60 30	0	100 40	TA0 @	120 50	TA16 P	140 60	SA0 ,	160 70	SA16 p				
0 0 0 1	1 1	GTL SOH	21 11	LL0 DC1	41 21	!	61 31	1	101 41	TA1 A	121 51	TA17 Q	141 61	SA1 a	161 71	SA17 q				
0 0 1 0	2 2	STX	22 12	DC2	42 22	"	62 32	2	102 42	TA2 B	122 52	TA18 R	142 62	SA2 b	162 72	SA18 r				
0 0 1 1	3 3	ETX	23 13	DC3	43 23	#	63 33	3	103 43	TA3 C	123 53	TA19 S	143 63	SA3 c	163 73	SA19 s				
0 1 0 0	4 4	SDC EOT	24 14	DCL DC4	44 24	\$	64 34	4	104 44	TA4 D	124 54	TA20 T	144 64	SA4 d	164 74	SA20 t				
0 1 0 1	5 5	PPC ENQ	25 15	PPU NAK	45 25	%	65 35	5	105 45	TA5 E	125 55	TA21 U	145 65	SA5 e	165 75	SA21 u				
0 1 1 0	6 6	ACK	26 16	SYN	46 26	&	66 36	6	106 46	TA6 F	126 56	TA22 V	146 66	SA6 f	166 76	SA22 v				
0 1 1 1	7 7	BEL	27 17	ETB	47 27	'	67 37	7	107 47	TA7 G	127 57	TA23 W	147 67	SA7 g	167 77	SA23 w				
1 0 0 0	10 8	GET BS	30 18	SPE CAN	50 28	(	70 38	8	110 48	TA8 H	130 58	TA24 X	150 68	SA8 h	170 78	SA24 x				
1 0 0 1	11 9	TCT HT	31 19	SPD EM	51 29	)	71 39	9	111 49	TA9 I	131 59	TA25 Y	151 69	SA9 i	171 79	SA25 y				
1 0 1 0	12 A	LF	32 1A	SUB	52 2A	*	72 3A	:	112 4A	TA10 J	132 5A	TA26 Z	152 6A	SA10 j	172 7A	SA26 z				
1 0 1 1	13 B	VT	33 1B	ESC	53 2B	+	73 3B	;	113 4B	TA11 K	133 5B	TA27 [	153 6B	SA11 k	173 7B	SA27 {				
1 1 0 0	14 C	FF	34 1C	FS	54 2C	,	74 3C	<	114 4C	TA12 L	134 5C	TA28 \ ]	154 6C	SA12 l	174 7C	SA28 				
1 1 0 1	15 D	CR	35 1D	GS	55 2D	-	75 3D	=	115 4D	TA13 M	135 5D	TA29 ]	155 6D	SA13 m	175 7D	SA29 }				
1 1 1 0	16 E	SO	36 1E	RS	56 2E	.	76 3E	>	116 4E	TA14 N	136 5E	TA30 ^	156 6E	SA14 n	176 7E	SA30 ~				
1 1 1 1	17 F	SI	37 1F	US	57 2F	/	77 3F	? UNL	117 4F	TA15 O	137 5F	UNT -	157 6F	SA15 o	177 7F	RUBOUT (DEL)				
	ADDRESSED COMMANDS				UNIVERSAL COMMANDS				LISTEN ADDRESSES				TALK ADDRESSES				SECONDARY ADDRESSES OR COMMANDS			

## KEY



**Tektronix**  
 REF: ANSI STD X3.4-1977  
 IEEE STD 488.1-1987  
 ISO STD 646-2973



---

# Factory setup

The following listing is the instrument response to the concatenated command FACTory;SET. This response describes the factory default setup in detail. (Carriage returns have been inserted for clarity.)

Items enclosed in ( ) parentheses are returned by the SET? query response, but are not changed by the FACTory command.

## TBS1000C Series Oscilloscopes

```
:HEADER 1;(:VERBOSE 1;)
:DATA:ENCDG RIBINARY;DESTINATION REFA;SOURCE CH1;START
1;STOP 2500;WIDTH 1;
(:LOCK NONE;)
:DISPLAY:FORMAT YT;STYLE VECTORS;PERSISTENCE 0;(CONTRAST
50);(INVERT OFF);
:ACQUIRE:MODE SAMPLE;NUMAVG 16;STATE 1;STOPAFTER
RUNSTOP;
:CH1:PROBE 10;SCALE 1.0E0;POSITION 0.0E0;COUPLING
DC;BANDWIDTH OFF;INVERT OFF;
:CH2:PROBE 10;SCALE 1.0E0;POSITION 0.0E0;COUPLING
DC;BANDWIDTH OFF;INVERT OFF;
:HORIZONTAL:VIEW MAIN;MAIN:SCALE 5.0E-4;POSITION 0.0E0;
:HORIZONTAL:DELAY:SCALE 5.0E-5;POSITION 0.0E0;
:TRIGGER:MAIN:MODE AUTO;TYPE EDGE;HOLDOFF:VALUE 5.0E-7;
:TRIGGER:MAIN:EDGE:SOURCE CH1;COUPLING DC;SLOPE RISE;
:TRIGGER:MAIN:VIDEO:SOURCE CH1;SYNC LINE;POLARITY
NORMAL;LINE 1;STANDARD NTSC;
:TRIGGER:MAIN:PULSE:SOURCE CH1;WIDTH:POLARITY
POSITIVE;WHEN EQUAL;WIDTH 1.0E-3;
:TRIGGER:MAIN:LEVEL 0.0E0;
:SELECT:CH1 1;CH2 0;MATH 0;REFA 0;REFB 0;REFC 0;REFD 0;
:CURSOR:FUNCTION OFF;SELECT:SOURCE CH1;
:CURSOR:VBARS:UNITS SECONDS;POSITION1 -2.0E-3;POSITION2
2.0E-3;
:CURSOR:HBARS:POSITION1 -3.2E0;POSITION2 3.2E0;
:MEASUREMENT:MEAS1:TYPE NONE;SOURCE CH1;
```

```
:MEASUREMENT:MEAS2:TYPE NONE;SOURCE CH1;  
:MEASUREMENT:MEAS3:TYPE NONE;SOURCE CH1;  
:MEASUREMENT:MEAS4:TYPE NONE;SOURCE CH1;  
:MEASUREMENT:MEAS5:TYPE NONE;SOURCE CH1;  
:MEASUREMENT:IMMED:TYPE PERIOD;SOURCE CH1;  
:MATH:DEFINE "CH1 - CH2";FFT:HORIZONTAL:POSITION 5.0E1;SCALE  
1.0E0;  
:MATH:FFT:VERTICAL:POSITION 0.0E0;SCALE 1.0E0;  
(<SAVE:IMAGE:FILEFORMAT BMP;>)  
(:LANGUAGE ENGLISH)
```

## Reserved words

The following words are reserved for the instrument.

\*CAL, \*CLS, \*DDT, \*ESE, \*ESR, \*IDN, \*LRN, \*OPC, \*PSC, \*RCL, \*RST, \*SAV, \*SRE, \*STB, \*TRG, \*TST, \*WAI, A0, A1, A2, A3, A4, A5, A6, A7, A8, A9, ABOrt, AC, ACLINE, ACQuire, ALL, ALLEv, ASC, ASCIi, AUTO, AUTOMATIC, AUTORange, AUTOSet, AVERage, B0, B1, B2, B3, B4, B5, B6, B7, B8, B9, BANDwidth, BATTERIES, BAUD, BIN, BIT\_Nr, BMP, BN\_Fmt, BOTH, BRIGHTness, BUBBLEJet, BUSY, BUTTON, BUTTONLIGHT, BYT\_Nr, BYT\_Or, Block, CALibrate, CARD, CENtronics, CH1, CH1CH2, CH2, CM10BY15, CM13BY18, CM15BY21, CM18BY24, CM6BY8, CM7BY10, CM9BY13, COMpare, CONDUCTION, CONTINUE, CONTINUOUS, CONTRast, COUpling, CR, CRLf, CRMs, CURRENTPRObe, CURSor, CURSORRms, CURVe, CWD, DATALOGging, DATE, DATEPRINT, DATa, DC, DCLIne, DEF, DEFINE, DEFLT, DEFault, DELay, DELEte, DELTa, DELay, DESE, DESKJet, DESTination, DIAg, DIR, DISplay, DOTs, DPU3445, DPU411, DPU412, DRAFT, DURAtion, E, EDGE, ENABle, ENCdg, ENGLish, ENV, EPSC60, EPSC80, EPSIMAGE, EPSOn, EQual, ERRLOG, EVEN, EVENT, EVMsg, EVQty, EXECute, EXT, EXT10, EXT5, FACtory, FALL, FALLINGedge, FASTPHOTO, FFT, FIELD, FILEFormat, FILESystem, FINE, FIRST, FLAg, FORCe, FORMat, FREESpace, FRENch, FREQuency, FREquency, FUNCTION, GASgauge, GERMan, GND, GPIb, HAGAKIPC, HAGAKIPCARD, HARDCopy, HARDFlagging, HARmonics, HBArS, HDELTA, HDR, HEADer, HERTz, HFRej, HOLDOff, HORizontal, HRMS, ID, IDPRINT, IMAGESIZE, IMAge, IMMEd, IN11BY17, IN2P5BY3P25, IN4BY6, IN8BY10, INDEX, INF, INIT, INKSaver, INTERLEAF, INTERNAL, INVERT, INVert, INside, ITALian, JAPANese, JOULES, JPEG, JPG, KOREan, L, L2, L4, LANGuage, LANdScape, LASERJet, LAYout, LETTER, LEVELS, LEVEl, LF, LFCr, LFRej, LIMit, LINE, LINENum, LOCK, LOG, LSB, MAIn, MANUAL, MATH, MAXImum, MEAN, MEASUrement, MINImum, MKDir, MM100BY150, MM54BY86, MODE, MSB, MULTICYcle, N, NEGAtive, NEXT, NOISerej, NONE, NONE, NORMAl, NOTEqual, NRMAL, NR\_Pt, NTSc, NUMACq, NUMAVg, NWIdth, ODD, OFF, ON, OUTside, PAL, PAPERSIZE, PAPERTYPE, PARity, PCX, PEAKdetect, PERCent, PERIod, PERSistence, PFPHASE, PHAse, PHOTO, PICTBridge, PK2pk, PLAIN, POLarity, PORT, PORTrait, PORTUguese, POSITIVE, POSition, POWERFACTOR, POWer, POWerANALYSIS, PRESENT, PRINTQUAL, PRINTS, PRObe, PT\_Fmt, PT\_Off, PULse, PWidth, RECAIl, RECOrdlength, REFx, REM, REName, RESUlt, RI, RIBinary, RISINgedge, RISe, RLE, RMDir, RMS, ROLL100MM, ROLL127MM, ROLL210MM, ROLL89MM, RP, RPBinary, RS232, RUN, RUNSTop, SAMple, SAVESAll, SAVESImage, SAVE, SAVEIMAge, SAVEWFM, SCALE, SECOnds, SECdiv, SELEct, SEQuence, SET, SETLevel, SETTings, SETUp, SHOW, SIGNAL, SIMPLifiedchinese, SINGLECYcle, SLOPe, SOFTFlagging, SOURCE, SOURCE1, SOURCE2, SOURce, SOURces, SPANish, SRIBinary, SRPBinary, STANDard, START, STATE, STATUS, STOP, STOPAfter, STYle, SWLoss, SYNC, TARget, TEMPLate, TERMinator, THDF, THDR, THINKjet, TIFF, TIME, TIME, TOFFEND, TOFFSTART, TOLerance, TONEND,

TONSTART, TOTAL, TRADitionalchinese, TRANsmit, TRIGger,  
TRUEPOWER, TURNOFF, TURNON, TYPE, UNIts, UNLock, USB, VALue,  
VAR, VBArS, VDELtA, VECtors, VERBoSe, VERTical, VIDEo, VIEW,  
VIOLation, VOLts, VSAT, WATTS, WAVEform, WAVEFORMANALYSIS,  
WAVEform, WAVFrm, WFCREST, WFCYCRMS, WFFREQ, WFIId, WFMPRe,  
WHEN, WIDth, WINDOW, XINcr, XUNit, XY, XZEro, Y, YMuLt, YOFF, YT,  
YUNit, YZEro, ZONE,

---

# Glossary

## Glossary terms

<b>ASCII</b>	Acronym for the American Standard Code for Information Interchange. Controllers transmit commands to the digitizing instrument using ASCII character encoding.
<b>Address</b>	A 7-bit code that identifies an instrument on the communication bus. The digitizing instrument must have a unique address for the controller to recognize and transmit commands to it.
<b>Backus-Naur Form (BNF)</b>	A standard notation system for command syntax. The syntax in this manual use BNF notation.
<b>Controller</b>	A computer or other device that sends commands to and accepts responses from the digitizing instrument.
<b>EOM</b>	A generic acronym referring to the end-of-message terminator. For GPIB, the end-of-message terminator is either an EOI or the ASCII code for line feed (LF). For USB, the end-of-message terminator is the EOM bit in a USBTMC message.
<b>IEEE</b>	Acronym for the Institute of Electrical and Electronics Engineers.
<b>USB</b>	An acronym for Universal Serial Bus.
<b>USBTMC</b>	An acronym for USB Test and Measurement Class.
<b>USB488</b>	The USBTMC subclass specification that implements an IEEE488-like interface over USB.





# Index

- \*CAL?, 53
- \*CLS, 72
- \*ESE, 106
- \*ESR?, 107
- \*IDN?, 152
- \*LRN?, 154
- \*OPC, 191
- \*PSC, 193
- \*RCL, 195
- \*RST, 202
- \*SAV, 203
- \*SRE, 217
- \*STB?, 218
- \*WAI, 247

## A

- ACQuire?, 37
- ACQuire:MAXSamplerate, 38
- ACQuire:MODE, 38
- ACQuire:NUMAcq?, 40
- ACQuire:NUMAVg, 41
- ACQuire:STATE, 41
- ACQuire:STOPAfter, 42
- ALias, 43, 47
- ALias:CATalog?, 44
- ALias:DEFine, 45
- ALias:DELEte, 46, 47
- ALias:DELEte:ALL, 46
- ALLEv?, 48
- Arguments, 10
- ASCII, 3
- AUTOSet, 49
- AUTOSet:ENABLE, 49

## B

- BNF (Backus Naur form), 3
- BUSY?, 51

## C

- CALibrate:INTERNAL, 54

- CALibrate:INTERNAL:START, 54
- CALibrate:INTERNAL:STATus?, 55
- CALibrate:RESults?, 56
- CALibrate:RESults:SPC?, 56
- CH<x>?, 57
- CH<x>:AMPSVIAVOLTs :Factor, 59
- CH<x>:AMPSVIAVOLTs:ENABLE, 58
- CH<x>:BANdwidth, 59
- CH<x>:COUPling, 60
- CH<x>:DESKew, 61
- CH<x>:INVert, 62
- CH<x>:LABel, 62
- CH<x>:OFFSet, 63
- CH<x>:POSition, 64
- CH<x>:PRObe, 65
- CH<x>:PRObe:GAIN, 66
- CH<x>:PRObe:ID?, 67
- CH<x>:PRObe:ID:SERnumber?, 67
- CH<x>:PRObe:ID:TYPE?, 68
- CH<x>:PRObe:SIGnal, 68
- CH<x>:PRObe:UNIts?, 69
- CH<x>:SCALE, 69
- CH<x>:VOLts, 70
- CH<x>:YUNit, 71
- CLEARMenu, 72
- Command
  - syntax:BNF (Backus Naur form), 3
- Command and Query Structure, 4
- Command syntax
  - BNF (Backus Naur form), 3
- CURSor?, 73
- CURSor:ENABLE, 74
- CURSor:FUNCTION, 75
- CURSor:HBArs?, 76
- CURSor:HBArs:DELTA?, 76
- CURSor:HBArs:POSITION<x>, 77
- CURSor:HBArs:UNIts, 78
- CURSor:VBArS?, 80
- CURSor:VBArS:DELTA?, 81
- CURSor:VBArS:HPOS<x>?, 82
- CURSor:VBArS:POSITION<x>, 82

CURSor:VBArS:UNIts, 83  
CURSor:VBArS:VDELtA?, 84  
CURVe, 85

## D

DATA, 87  
DATA:DESTination, 88  
DATA:SOURce, 89  
DATA:START, 90  
DATA:STOP, 91  
DATA:WIDTH, 92  
DATE, 93  
DESE, 94  
DIAG:FAN, 95  
DIAG:LOOP:OPTion, 95  
DIAG:RESUlt:FLAg?, 97  
DIAG:RESUlt:LOG?, 97  
DIAG:SElect, 98  
DIAG:SElect:<function>, 99  
DIAG:STATE, 99  
DIAG:TEMPVAL, 100  
DISplay:GRAticule, 100  
DISplay:INTENSITy:BACKLight, 101  
DISplay:PERsistence:STATe, 102  
DISplay:PERsistence:VALUe, 102

## E

ERRLOG:FIRST?, 105  
ERRLOG:NEXT?, 105  
Ethernet command group, 16  
Event handling, 275  
EVENT?, 108  
EVMsg?, 108  
EVQty?, 109  
Example programming, 296  
Examples  
    Programming, 296

## F

FACtory, 111  
Factory setup

    detailed description, 299  
FEAEN:PASSWORD, 112  
FFT?, 113  
FFT:HORizontal:POSition, 113  
FFT:HORizontal:SCAle, 114  
FFT:SOURce, 114  
FFT:SRCWFM, 115  
FFT:VERTical:POSition, 116  
FFT:VERTical:SCAle, 116  
FFT:VERTical:UNIts, 117  
FFT:VType, 117  
FFT:WINDow, 118  
FILESystem?, 118  
FILESystem:CWD, 119  
FILESystem:DELEte, 120  
FILESystem:DIR?, 121  
FILESystem:FORMat, 122  
FILESystem:FREESpace?, 122  
FILESystem:MKDir, 123  
FILESystem:READFile, 124  
FILESystem:REName, 124  
FILESystem:RMDir, 125  
FILESystem:WRITEFile, 126  
FPAnel:PRESS, 127  
FPAnel:TURN, 128

## H

HDR, 131  
HEADer, 131  
HELPevery:ACQuire, 132  
HELPevery:ALL, 133  
HELPevery:CURsor, 133  
HELPevery:FFT, 134  
HELPevery:MATH, 134  
HELPevery:MEASUrement, 135  
HELPevery:REFeRence, 136  
HELPevery:TRIGger, 136  
HELPevery:UTILity, 137  
HELPevery:VERTical, 138  
HORizontal?, 138  
HORizontal:ACQLENGTH, 139  
HORizontal:DELay:POSition, 140  
HORizontal:DIVisions, 140

HORizontal:MAIn:DElay:MODE, 141  
 HORizontal:MAIn:DElay:STATe, 142  
 HORizontal:MAIn:DElay:TIME, 143  
 HORizontal:MAIn:SAMPLERate, 144  
 HORizontal:MAIn:SECdiv, 145  
 HORizontal:MAIn:UNIts:STRing, 146  
 HORizontal:PREViewstate, 146  
 HORizontal:RECOrdlength, 147  
 HORizontal:RECOrdlength:Auto, 147  
 HORizontal:RESOLution, 148  
 HORizontal:ROLL, 149  
 HORizontal:SCALe, 144  
 HORizontal:TRIGger:POSition, 149

## I

ID?, 151  
 IEEE Std. 488.2-1987, 3

## L

LANGuage, 153  
 LOCK, 154

## M

MATH?, 155  
 MATH:DEFINE, 156  
 MATH:HORizontal:POSition, 157  
 MATH:HORizontal:SCALe, 157  
 MATH:HORizontal:UNIts, 158  
 MATH:LABel, 159  
 MATH:VERTical:POSition, 159  
 MATH:VERTical:SCALe, 160  
 MATH:VERTical:UNIts, 161  
 MEASUrement?, 161  
 MEASUrement:CLEARSHOT, 163  
 MEASUrement:ENABLE, 163  
 MEASUrement:GATing, 164  
 MEASUrement:IMMed?, 165  
 MEASUrement:IMMed:DElay?, 165  
 MEASUrement:IMMed:DElay:EDGE<x>, 166  
 MEASUrement:IMMed:SOURce1, 167  
 MEASUrement:IMMed:SOURce2, 168

MEASUrement:IMMed:TYPE, 169  
 MEASUrement:IMMed:UNIts?, 171  
 MEASUrement:IMMed:VALue?, 172  
 MEASUrement:MEAS<x>?, 173  
 MEASUrement:MEAS<x>:DElay?, 173  
 MEASUrement:MEAS<x>:DElay:EDGE<x>, 174  
 MEASUrement:MEAS<x>:SOURce1, 175  
 MEASUrement:MEAS<x>:SOURce2, 175  
 MEASUrement:MEAS<x>:STATE, 176  
 MEASUrement:MEAS<x>:TYPE, 177  
 MEASUrement:MEAS<x>:UNIts?, 180  
 MEASUrement:MEAS<x>:VALue?, 181  
 MEASUrement:REFLevel?, 182  
 MEASUrement:REFLevel:ABSolute:LOW, 182  
 MEASUrement:REFLevel:ABSolute:MID1, 183  
 MEASUrement:REFLevel:ABSolute:MID2, 184  
 MEASUrement:REFLevel:METHod, 185  
 MEASUrement:REFLevel:PERCent:HIGH, 186  
 MEASUrement:REFLevel:PERCent:LOW, 187  
 MEASUrement:REFLevel:PERCent:MID1, 188  
 MEASUrement:REFLevel:PERCent:MID2, 189  
 MEASUrement:SNAPSHOT, 190  
 MEASUrement:SOURceSHOT, 190  
 Message  
     handling, 275  
 Mnemonics, 9

## P

Programming  
     examples, 296  
 Programming examples, 296

## R

RECALL:SETUp, 196  
 RECALL:WAVEForm, 197  
 REF<x>?, 198  
 REF<x>:DATE, 198  
 REF<x>:HORizontal:DElay:TIME?, 199  
 REF<x>:HORizontal:SCALe?, 200  
 REF<x>:POSition?, 200  
 REF<x>:TIME?, 199  
 REF<x>:VERTical:POSition?, 201  
 REF<x>:VERTical:SCALe?, 201

## S

- SAVe:IMAge, 204
- SAVe:IMAge:FILEFormat, 205
- SAVe:IMAge:LAYout, 206
- SAVe:SETUp, 207
- SAVe:WAVEform, 208
- SAVe:WAVEform:FILEFormat, 209
- SElect:CH<x>, 210
- SElect:CONTRol, 211
- SElect:FFT, 212
- SElect:MATH, 213
- SElect:REF<x>, 214
- SET?, 215
- SETUP<x>:DATE?, 216
- SETUP<x>:TIME?, 216
- Setups
  - factory:TBS2000, 300
- Status, 275
- Syntax
  - BNF (Backus Naur form), 3

## T

- TEKSecure, 219
- Time, 219
- TRIGger, 220
- TRIGger:A, 221
- TRIGger:A:EDGE?, 222
- TRIGger:A:EDGE:COUpling, 222
- TRIGger:A:EDGE:SLOpe, 223
- TRIGger:A:EDGE:SOUrce, 224
- TRIGger:A:HOLDOff?, 225
- TRIGger:A:HOLDOff:Time, 225
- TRIGger:A:LEVel, 226
- TRIGger:A:LEVel:CH<x>, 227
- TRIGger:A:LOWerthreshold:CH<x>, 227
- TRIGger:A:MODE, 228
- TRIGger:A:PULse?, 229
- TRIGger:A:PULse:CLAss, 230
- TRIGger:A:PULse:SOUrce, 232
- TRIGger:A:PULse:WIDth?, 231
- TRIGger:A:PULse:WIDth:POLarity, 231
- TRIGger:A:PULse:WIDth:WHEN, 232

- TRIGger:A:PULse:WIDth:WIDth, 234
- TRIGger:A:RUNT, 235
- TRIGger:A:RUNT:POLarity, 235
- TRIGger:A:RUNT:SOUrce, 236
- TRIGger:A:RUNT:WHEn, 237
- TRIGger:A:RUNT:WIDth, 238
- TRIGger:A:TYPe, 238
- TRIGger:A:UPPerthreshold:CH<x>, 239
- TRIGger:FREQuency?, 240
- TRIGger:STATE?, 240

## U

- UNLock, 243

## V

- VERBose, 245

## W

- WAVFrm?, 248
- WFMInpre?, 248
- WFMInpre:BIT\_Nr, 249
- WFMInpre:BYT\_Nr, 250
- WFMInpre:ENCdg, 250
- WFMInpre:NR\_Pt?, 251
- WFMInpre:XINcr, 252
- WFMInpre:XUNit, 252
- WFMInpre:XZEro, 253
- WFMInpre:YMUlt, 254
- WFMInpre:YOFF, 255
- WFMInpre:YUNit, 256
- WFMInpre:YZEro, 257
- WFMOutpre?, 258
- WFMOutpre:BIT\_Nr, 258
- WFMOutpre:BN\_Fmt, 259
- WFMOutpre:BYT\_Nr, 260
- WFMOutpre:ENCdg, 261
- WFMOutpre:NR\_Pt?, 262
- WFMOutpre:RECOrdlength?, 262
- WFMOutpre:WFId?, 263
- WFMOutpre:XINcr?, 264
- WFMOutpre:XUNit?, 264

WFMOutpre:XZEro?, 265  
WFMOutpre:YMUlt?, 266  
WFMOutpre:YOFF?, 266  
WFMOutpre:YUNit?, 267  
WFMOutpre:YZEro?, 268

## Z

ZOOM?, 269

ZOOm{:MODE|:STATE}, 269  
ZOOm:ZOOM1, 270  
ZOOM:ZOOM1:FACtor, 271  
ZOOM:ZOOM1:HORizontal:POSition, 271  
ZOOM:ZOOM1:HORizontal:SCAle, 272  
ZOOm:ZOOM1:POSition, 273  
ZOOm:ZOOM1:SCAle, 273  
ZOOM:ZOOM1:STATE, 274

