Programmer Manual

Tektronix

TDS Family Digitizing Oscilloscopes
TDS 400A, 510A, 500D, 600B, & 700D Series

063-3002-00

Table of Contents

Tektronix TDS Family Digitizing Oscilloscopes Programmer Manual (TDS 400A, 510A, 500D, 600B, & 700D Series)

063-3002-00

The following Help Topics are available:

For Help on Help, Press F1

Alphabetic Command List

Command Descriptions

You can use commands to either set instrument features or query instrument values. You can use some commands to do both, some to only set, and some to only query. This manual marks set only commands with the words "No Query Form" following the command name. It marks query only commands with a question mark appended to the header, and includes the words "Query Only" following the command name.

This help file spells out headers, mnemonics, and arguments with the minimal spelling shown in upper case. For example, to use the abbreviated form of the ACQuire:MODe command, just type ACQ:MOD.

ACQuire?
ACQuire:AUTOSAve
ACQuire:MODe
ACQuire:NUMACq?
ACQuire:NUMAVg
ACQuire:NUMEnv
ACQuire:REPEt
ACQuire:STATE
ACQuire:STOPAfter
ALIas
ALIas:CATalog?
ALIas:DEFIne
ALIas:DELEte
ALIas:DELEte:ALL
ALIas:DELEte:NAMe
ALIas:STATE
ALLEv?
ALLOcate?
ALLOcate:WAVEform?
ALLOcate:WAVEform:FREE?
ALLOcate:WAVEform:REF<x>
APPMenu
APPMenu:LABel
APPMenu:LABel:BOTTOM<x>
APPMenu:LABel:RIGHT<x>
APPMenu:TITLe
AUTOSet
BELl
BUSY?
*CAL?
CH<x>?
CH<x>:BANdwidth
CH<x>:COUPling
CH<x>:DESKew
CH<x>:IMPedance

TRIGger:MAIn:LOGIc:SETHold:CLOCk:LEVel
TRIGger:MAIn:LOGIc:SETHold:CLOCk:SOUrce
TRIGger:MAIn:LOGIc:SETHold:DATa:LEVel
TRIGger:MAIn:LOGIc:SETHold:DATa:SOUrce
TRIGger:MAIn:LOGIc:SETHold:HOLDTime
TRIGger:MAIn:LOGIc:SETHold:SETTime
TRIGger:MAIn:LOGIc:STATE:INPut:CH4
TRIGger:MAIn:LOGIc:STATE:WHEn
TRIGger:MAIn:LOGIc:THReshold?
TRIGger:MAIn:LOGIc:THReshold:CH<x>
TRIGger:MAIn:LOGIc:WHEn
TRIGger:MAIn:MODe
TRIGger:MAIn:PULse?
TRIGger:MAIn:PULse:CLAss
TRIGger:MAIn:PULse:GLItch?
TRIGger:MAIn:PULse:GLItch:FILTer
TRIGger:MAIn:PULse:GLItch:POLarity
TRIGger:MAIn:PULse:GLItch:WIDth
TRIGger:MAIn:PULse:RUNT?
TRIGger:MAIn:PULse:RUNT:POLarity
TRIGger:MAIn:PULse:RUNT:THReshold?
TRIGger:MAIn:PULse:RUNT:THReshold:BOTh
TRIGger:MAIn:PULse:RUNT:THReshold:HIGH
TRIGger:MAIn:PULse:RUNT:THReshold:LOW
TRIGger:MAIn:PULse:RUNT:WHEn
TRIGger:MAIn:PULse:RUNT:WIDth
TRIGger:MAIn:PULse:SLEWRate:DELTATime
TRIGger:MAIn:PULse:SLEWRate:POLarity
TRIGger:MAIn:PULse:SLEWRate:SLEWRate?
TRIGger:MAIn:PULse:SLEWRate:THReshold:BOTh
TRIGger:MAIn:PULse:SLEWRate:THReshold:HIGH
TRIGger:MAIn:PULse:SLEWRate:THReshold:LOW
TRIGger:MAIn:PULse:SLEWRate:WHEn
TRIGger:MAIn:PULse:SOUrce
TRIGger:MAIn:PULse:TIMEOut:POLarity
TRIGger:MAIn:PULse:TIMEOut:TIMe
TRIGger:MAIn:PULse:WIDth?
TRIGger:MAIn:PULse:WIDth:HIGHLimit
TRIGger:MAIn:PULse:WIDth:LOWLimit
TRIGger:MAIn:PULse:WIDth:POLarity
TRIGger:MAIn:PULse:WIDth:WHEn
TRIGger:MAIn:TYPe
TRIGger:MAIn:VIDeo?
TRIGger:MAIn:VIDeo:BY
TRIGger:MAIn:VIDeo:FIELD
TRIGger:MAIn:VIDeo:FIELDType
TRIGger:MAIn:VIDeo:FLEXformat?
TRIGger:MAIn:VIDeo:FLEXformat:FIELDRATE

TRIGger:MAIn:VIDeo:FLEXformat:FIELDS
TRIGger:MAIn:VIDeo:FLEXformat:LINES
TRIGger:MAIn:VIDeo:FLEXformat:NEGSynchwidth
TRIGger:MAIn:VIDeo:FLEXformat:V1STArttime
TRIGger:MAIn:VIDeo:FLEXformat:V1STOptime
TRIGger:MAIn:VIDeo:FLEXformat:V2STArttime
TRIGger:MAIn:VIDeo:FLEXformat:V2STOptime
TRIGger:MAIn:VIDeo:HDTv
TRIGger:MAIn:VIDeo:HOLdoff?
TRIGger:MAIn:VIDeo:HOLdoff:VALue
TRIGger:MAIn:VIDeo:INTERLAce
TRIGger:MAIn:VIDeo:LINE
TRIGger:MAIn:VIDeo:LINES
TRIGger:MAIn:VIDeo:NTSc
TRIGger:MAIn:VIDeo:PAL
TRIGger:MAIn:VIDeo:SCAN
TRIGger:MAIn:VIDeo:SCANPeriod
TRIGger:MAIn:VIDeo:SOUrce
TRIGger:MAIn:VIDeo:STANdard
TRIGger:MAIn:VIDeo:SYNc
TRIGger:MAIn:VIDeo:SYStem
TRIGger:MAIn:VIDeo:TIMe
TRIGger:STATE?
*TRG
*TST?
UNLock
VERBose

*WAI
WAVFrm?
WFMPre?
WFMPre:BIT_Nr
WFMPre:BN_Fmt
WFMPre:BYT_Nr
WFMPre:BYT_Or
WFMPre:ENCdg
WFMPre:PT_Fmt
WFMPre:PT_Off
WFMPre:XINcr
WFMPre:YMUlt
WFMPre:YOFf
WFMPre:YZEro
WFMPre:<wfm>?
WFMPre:<wfm>:NR_Pt
WFMPre:<wfm>:PT_Fmt
WFMPre:<wfm>:PT_Off
WFMPre:<wfm>:WFId
WFMPre:<wfm>:XINcr

Command Groups

This section lists oscilloscope commands by functional groups.

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

Acquisition Commands

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

Table 2-4: Acquisition Commands

Header	Description
ACQuire?	Return acquisition parameters
ACQuire:AUTOSAve	Save waveforms to reference memory (TDS 500D, 600B, 700D)
ACQuire:MODe	Acquisition mode
ACQuire:NUMACq?	Return # of acquisitions obtained
ACQuire:NUMAVg	Number of acquisitions for average
ACQuire:NUMEnv	Number of acquisitions for envelope
ACQuire:REPet	Repetitive acquisition mode (TDS 400A, 510A, 500D, & 700D)
ACQuire:STATE	Start or stop acquisition system
ACQuire:STOPAfter	Acquisition control

Alias Commands

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

Table 2-5: Alias Commands

Header	Description
ALIas	Turn the alias state on and off
ALIas:CATalog?	Return a list of aliases
ALIas:DEFIne	Create a new alias
ALIas:DELEte	Remove an alias
ALIas:DELEte:ALL	Remove all aliases
ALIas:DELEte:NAMe	Remove a named alias
ALIas:STATE	Turn the alias state on and off

Application Menu Commands

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

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

Table 2-6: Application Menu Commands

Header	Description
APPMenu	Display the application menu
APPMenu:LABel	Return or remove all application menu button labels
APPMenu:LABel:BOTTOM<x>	Label for a bottom menu button
APPMenu:LABel:RIGHT<x>	Label for a side menu button
APPMenu:TITLe	Create a title for the application menu

Calibration and Diagnostic Commands

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

Table 2-7: Calibration and Diagnostic Commands

Header	Description
*CAL?	Perform an internal self-calibration
DIAg:RESULt:FLAg?	Return diagnostic tests status
DIAg:RESULt:LOG?	Return diagnostic test sequence results
DIAg:SELect:ACQUISition	Acquisition system diagnostic test sequence
DIAg:SELect:ALL	Diagnostic test sequence for Acquisition, Processor, Display, and Front panel
DIAg:SELect:CPU	Processor diagnostic test sequence
DIAg:SELect:DISplay	Display system diagnostic test sequence
DIAg:SELect:FPAnel	Front panel diagnostic test sequence
DIAg:STATE	Control of diagnostic tests

Cursor Commands

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

Cursor Commands

Header	Description
CURSor?	Return cursor settings
CURSor:FUNCtion	Cursors on or off; select cursor type
CURSor:HBArs?	Return H bar settings
CURSor:HBArs:DELTa?	Return distance between H bars
CURSor:HBArs:POSITION<x>	Position a horizontal cursor
CURSor:HBArs:POSITION<x>Pcnt	Position a horizontal cursor in units of % of vertical range. (TDS 400A Only)
CURSor:HBArs:SELect	Set which cursor the knob controls
CURSor:HBArs:UNIts	Set H bar units
CURSor:MODe	Set cursor tracking mode
CURSor:PAIred	Position paired cursors; also, return settings
CURSor:PAIred:HDELTA?	Return horizontal distance between 1st and 2nd paired cursors
CURSor:PAIred:HPOS1?	Return horizontal position of 1st paired cursor
CURSor:PAIred:HPOS2?	Return horizontal position of 2nd paired cursor
CURSor:PAIred:POSITION<x>	Set or return vbar position of the 1st or 2nd paired cursor
CURSor:PAIred:POSITION<x>Pcnt	Position the horizontal paired cursor in units of % of record length (TDS 400A Only)
CURSor:PAIred:SELect	Select active paired cursor
CURSor:PAIred:UNITS	Set paired cursor units
CURSor:PAIred:VDELTA?	Return vertical distance between 1st and 2nd paired cursors
CURSor:VBArs	Position vertical bar cursors
CURSor:VBArs:DELTa?	Return horizontal distance between cursors
CURSor:VBArs:POSITION<x>	Position a vertical cursor
CURSor:VBArs:POSITION<x>Pcnt	Position a vertical cursor in units of % of record length (TDS 400A Only)
CURSor:VBArs:SELect	Set which cursor the knob controls
CURSor:VBArs:UNITS	Set vertical cursors to period or frequency or to lines with option 05 video on the TDS 510A, 500D, 600B, & 700D
CURSor:VBArs:UNITSTring?	Return unit string for the vertical bar cursor (TDS 400A Only)

Display Commands

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

Table 2-9: Display Commands

Header	Description
CLEARMenu	Clear menus from display
DISplay?	Return display settings
DISplay:CLOCk	Control the display of the date/time stamp
DISplay:COLOr:CONTRast	Control collision contrast (TDS 6X4B & 700D)
DISplay:COLOr:MAP:<item_name>:BYCONTents	Set color for math or ref waveform to the color of the waveform content (TDS 6X4BA & 700D)
DISplay:COLOr:MAP:<item_name>:TO	Set color for math or ref waveform to specific color index (TDS 6X4B & 700D)
DISplay:COLOr:PALEtte:PERSistence	Set current persistence palette to a preset persistence palette (TDS 6X4B & 700D)
DISplay:COLOr:PALEtte:REGular	Set current palette to a preset palette (TDS 6X4B & 700D)
DISplay:COLOr:PALEtte:RESETALL	Reset all palettes to their factory default settings (TDS 6X4B & 700D)
DISplay:COLOr:PALEtte:<palette_name>:RESET	Reset a selected palette to its factory default settings (TDS 6X4B & 700D)
DISplay:COLOr:PALEtte:<palette_name>:<item_name>	Set the color of a selected item on a selected palette (TDS 6X4B & 700D)
DISplay:FILTer	Displayed data interpolation
DISplay:FORMat	YT, XY, or XYZ display
DISplay:GRAticule	Graticule style
DISplay:INStavu:AUTOBright	Set automatic DPO brightness on or off (TDS 500D & 700D)
DISplaly:INStavu:BRIGHTness	Set the DPO brightness level (TDS 500D & 700D)
DISplaly:INStavu:CONTRast	Set the DPO contrast level (TDS 500D & 700D)
DISplaly:INStavu:ACCUmulate	Set the accumulation depth for DPO, mask counting, and histograms

	(TDS 500D & 700D)
DISplaly:INStavu:MAP	Select intensity or percentages to map display indexes to the acquisition database
DISplaly:INStavu:PERCent:P	Control the maximum value for a color index in the percent mapping mode
DISplay:INStavu:PERSistence	Set DPO persistence type - variable, infinite, or no persistence (TDS 500D & 700D)
DISplay:INStavu:STYle	Set DPO waveform dots or vector style (TDS 500D & 700D)
DISplay:INStavu:VARpersist	Set DPO variable persistence decay time (TDS 500D & 700D)
DISplay:INTENSITy?	Return intensity settings
DISplay:INTENSITy:CONTRast	Set waveform intensified zone brightness (TDS 4X0A, 510A, 500D, & 6X0B)
DISplay:INTENSITy:OVERALL	Set main brightness (TDS 4X0A, 510A, 500D, & 6X0B)
DISplay:INTENSITy:TEXt	Set text brightness
DISplay:INTENSITy:WAVEform	Set waveform brightness
DISplay:MODe	Set normal or DPO display mode (TDS 500D & 700D)
DISplay:PERSistence	Set variable persistence decay time
DISplay:STYle	Waveform dots, vector, infinite persistence, or variable persistence
DISplay:TRIGBar	Control the display of the trigger bar/s on screen
DISplay:TRIGT	Control the display of the trigger indicator on screen
MESSage	Remove text from the message window
MESSage:BOX	Set size and location of message window
MESSage:SHOw	Remove and display text in the message window
MESSage:STATE	Control display of message window

File System Commands

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

Table 2-10: File System Commands

Header	Description
FILESystem:COPy	Copy file to new file
FILESystem:CWd	Set directory path
FILESystem:DELEte	Delete named file
FILESystem:DELWarn	Set front-panel delete warning
FILESystem:DIR	Make directory
FILESystem:FORMat	Format named drive
FILESystem:FREESpace	Return free space on current drive
FILESystem:MKDir	Make new directory
FILESystem:OVERWrite	Set file-overwrite protection
FILESystem:PRint	Print file to port
FILESystem:READFile	Copy from file to GPIB port (TDS 500D & 700D)
FILESystem:REName	Assign new name to file
FILESystem:RMDir	Delete named directory
FILESystem:WRITEFile	Copy from GPIB port to file (TDS 500D & 700D)

Hardcopy Commands

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

Table 2-11: Hardcopy Commands

Header	Description
HARDCopy	Start or terminate hardcopy
HARDCopy:FILEName	Select file to send hardcopy data to (File System Only)
HARDCopy:FORMat	Hardcopy output format
HARDCopy:LAYout	Hardcopy orientation
HARDCopy:PALEtte	Select palette to use when making hardcopy (TDS 6X4B & 700D)
HARDCopy:PORT	Hardcopy port for output

Histogram Commands

Histogram Commands let you select the type of histogram, what part of the waveform should go into the histogram, and histogram statistics. Table 2-12 lists these commands.

Table 2-12: Histogram Commands

Header	Description
HIStogram?	Queries the state of all settable parameters related to histograms (TDS 500D & 700D)
HIStogram:BOX	Defines the left, top, right, and bottom positions of the histogram box, in source waveform coordinates (TDS 500D & 700D)
HIStogram:BOXPcnt	Define the left, top, right, and bottom positions of the histogram box, in percentage coordinates (TDS 500D & 700D)
HIStogram:COUNt	Zeros the counts for histograms and starts recounting (TDS 500D & 700D)
HIStogram:DISplay	Selects the way the histogram is displayed, if at all (TDS 500D & 700D)
HIStogram:MODe	Selects the type of histogram to be made - or turn the histogram off (TDS 500D & 700D)
HIStogram:SIZe	Controls the width (or height) of the histogram on the screen (TDS 500D & 700D)
HIStogram:SOUrce	Selects which trace to use when histograms are turned on (TDS 500D & 700D)

Horizontal Commands

Horizontal commands control the time bases of the oscilloscope. You can set the time per division (or time per point) of both the main and delay time bases. You can also set the record lengths. Table 2-13 lists these commands.

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

Table 2-13: Horizontal Commands

Header	Description
HORizontal?	Return horizontal settings
HORizontal:ACQDURACTION?	Returns the overall time interval covered by the live channels (TDS 500D & 700D)
HORizontal:ACQLENGTH?	Returns the acquisition length - either the extended one when extended acquisition length mode is on or the record length when this mode is off (TDS 500D & 700D)
HORizontal:CLOck	Enable internal or external clocks (TDS 400A)
HORizontal:CLOck:MAXRate	Set maximum external clock rate (TDS 400A)
HORizontal:DELay?	Return delay time base settings
HORizontal:DELay:MODe	Delay time base mode
HORizontal:DELay:SCAle	Delay time base time per division
HORizontal:DELay:SECdiv	Same as HORizontal:DELay:SCAle
HORizontal:DELay:TIMe	Delay time
HORizontal:DELay:TIMe:RUNSAfter	Time to wait in delay-runs-after-main mode
HORizontal:DELay:TIMe:TRIGAfter	Time to wait in delay-runs-after-trigger mode
HORizontal:EXTDACQ	Enable or disable extended acquisition length mode when DPO mode is not on. (TDS 500D & 700D)
HORizontal:FASTframe:COUNt	Select FastFrame count (TDS 500D & 700D)
HORizontal:FASTframe:FRAMELock	Turns Frame Lock on or off
HORizontal:FASTframe:LENgth	Select length of each FastFrame frame (TDS 500D & 700D)
HORizontal:FASTframe:POSition	Select FastFrame frame to display (TDS 500D & 700D)

HORizontal:FASTframe:REF	Select the reference frame number
HORizontal:FASTframe:STATE	Setup FastFrame acquisition (TDS 500D & 700D)
HORizontal:FASTframe:TIMEStamp:STATE	
HORizontal:FASTframe:TIMEStamp:POSition?	Returns the absolute trigger date and time for the position frame
HORizontal:FASTframe:TIMEStamp:REF?	Returns the absolute trigger date and time for the reference frame
HORizontal:FASTframe:TIMEStamp:FRAME?	Returns the time stamp for the given frame number
HORizontal:FASTframe:TIMEStamp:DELTa?	Returns the relative time between the triggers of the reference and position frames
HORizontal:FASTframe:TIMEStamp:BETWeen?	Returns the relative time between the triggers of the two specified frames
HORizontal:FASTframe:TIMEStamp:SNAPShot	Displays the TimeStamp Snapshot list and overlay menu
HORizontal:FASTframe:TIMEStamp:CLEARSNapshot	Clears the TimeStamp Snapshot list and overlay menu from the display
HORizontal:FITtoscreen	Setup waveform compress
HORizontal:MAIn?	Return main time per division
HORizontal:MAIn:SCAle	Main time base time per division
HORizontal:MAIn:SECdiv	Same as HORizontal:MAIn:SCAle
HORizontal:MODe	Turn delay time base on or off
HORizontal:POSition	Portion of waveform to display
HORizontal:RECOrdlength	Number of points in waveform record
HORizontal:RECORDSTART	Set the selected channel's record start that is used in extended acquisition length mode. Record start controls where, in the extended

	acquisition, the displayed waveform starts (TDS 500D & 700D)
HORizontal:ROLL	Set roll mode to auto or off (TDS 400A)
HORizontal:SCAle	Same as HORizontal:MAIn:SCAle
HORizontal:SECdiv	Same as HORizontal:MAIn:SCAle
HORizontal:TRIGger?	Return trigger position
HORizontal:TRIGger:POSition	Main time base trigger position

Limit Test Commands

The Limit Test commands let you automatically compare each incoming waveform against a template waveform. You set an envelope of limits around a waveform and let the oscilloscope find the waveforms that fall outside those limits. When the oscilloscope finds such a waveform, it can generate a hardcopy, ring a bell, stop and wait for your input, or do any combination of these actions. Table 2-14 lists these commands.

Table 2-14: Limit Test Commands

Header	Description
LIMit:BELl	Ring bell when limit is exceeded
LIMit:COMpare:CH<x>	Template to compare waveform to
LIMit:COMpare:MATH<x>	Template to compare math waveform to (TDS 510A, 500D, 600B, & 700D)
LIMit:HARDCopy	Make hardcopy when limit exceeded
LIMit:STATE	Turn limit testing on or off
LIMit:TEMPLate	Template to compare waveform to
LIMit:TEMPLate:DESTination	Reference storage for template waveform
LIMit:TEMPLate:SOUrce	Template waveform source
LIMit:TEMPLate:TOLerance:HORizontal	Tested waveform horizontal tolerance
LIMit:TEMPLate:TOLerance:VERTical	Tested waveform vertical tolerance

Mask Commands

Mask commands control standard masks, user-defined masks, and testing against masks. A mask is a polygonal region on the screen. Every vertical line on the screen must intersect the polygon in zero, one, or two places, but never in more than two. You must break up more complicated polygons into separate masks. Unlike Limit Testing, the inside of a mask is the region where waveform data would not normally fall. A telecom standard requires two, three, or four of these masks. Pulse standards always have two masks. Standards with eye patterns usually have three masks, but some have four. Table 2-15 lists these commands.

Table 2-15: Mask Commands

Header	Description
MASK?	Return mask parameters (TDS 500D & 700D)
MASK:AUTOSet:MODe	Controls whether or not an autoset will automatically be done after a standard mask is selected (TDS 500D & 700D)
MASK:AUTOSet:OFFSETAdj	Controls whether or not a standard-mask autoset has the freedom to adjust vertical offset so that the waveform matches the mask (TDS 500D & 700D)
MASK:AUTOSet:STANdard	Autosets a mask to a specific standard. (TDS 500D & 700D)
MASK:COUNt	Clears mask counts and restarts counting) (TDS 500D & 700D)
MASK:COUNt:STATE	Controls whether or not mask counting occurs (TDS 500D & 700D)
MASK:COUNt:TOTal?	Returns the sum of all the hits in all the defined tasks (TDS 500D & 700D)
MASK:COUNt:WAVEFORMS?	Returns the number of waveforms that have contributed to mask counting (TDS 500D & 700D)
MASK:DISplay	Controls whether or not defined masks are displayed on the screen (TDS 500D & 700D)
MASK:FILTer	Controls whether or not a digital filter will be run on the waveform data (TDS 500D & 700D)
MASK:INVert	Makes masks inverted with respect to standard orientation (TDS 500D &

	700D)
MASK:MARgin:PERCent	Defines the mask margin in percent (TDS 500D & 700D)
MASK:MARgin:STATE	Turns mask margins on or off (TDS 500D & 700D)
MASK:MASK<x>	With delete argument, deletes all points in the specified mask. With query form, queries the state of settable parameters of the specified mask (TDS 500D & 700D)
MASK:MASK<x>:COUNt?	Returns the number of hits in the specified mask (TDS 500D & 700D)
MASK:MASK<x>:NR_Pt?	Returns the number of points in the specified mask (TDS 500D & 700D)
MASK:MASK<x>:POInts	Defines points in user coordinates in the specified mask (TDS 500D & 700D)
MASK:MASK<x>:POINTSPcnt	Defines points in a mask in percentage coordinates (TDS 500D & 700D)
MASK:PASSFail:BELl	Defines when the pass/fail bell will sound (TDS 500D & 700D)
MASK:PASSFail:STATE	Controls whether mask pass/fail testing occurs (TDS 500D & 700D)
MASK:PASSFail:STATUs	Returns whether mask pass/fail testing is passing, passed, or failed (TDS 500D & 700D)
MASK:PASSFail:THReshold	Defines the number of counts which are considered a mask testing failure (TDS 500D & 700D)
MASK:PASSFail:WAVEform	Defines the minimum number of waveforms to be acquired for mask pass/fail counting (TDS 500D & 700D)
MASK:SOUrce	Selects which trace will be compared against the mask(s) during counting (TDS 500D & 700D)
MASK:STANdard	Delete any currently existing mask(s) and replace with standard mask(s) (TDS 500D & 700D)

MASK:TBPosition	Set the time base position to the value of the argument (TDS 500D & 700D)

Measurement Commands

Measurement commands control the automated measurement system. Up to four automated measurements can be displayed on the screen. In the commands, these four measurement readouts are named MEAS<x>, where <x> can be 1, 2, 3, or 4. Table 2-16 lists these commands.

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

Whether you use displayed or immediate measurements, you use the VALue? query to obtain measurement results.

Measurement commands can set and query measurement parameters. You can assign some parameters, such as waveform sources, differently for each measurement readout. Other parameters, such as reference levels, have only one value, which applies to all measurements.

Table 2-16: Measurement Commands

Header	Description
MEASUrement?	Return all measurement parameters
MEASUrement:CLEARSNapshot	Remove measurement snapshot from the display
MEASUrement:GATING	Set or query measurement gating
MEASUrement:IMMed?	Return immediate measurement parameters
MEASUrement:IMMed:DELay?	Return information on immediate delay measurement
MEASUrement:IMMed:DELay:DIRection	Search direction to use for delay measurements
MEASUrement:IMMed:DELay:EDGE1	Specify which waveform edge to use for delay measurements
MEASUrement:IMMed:DELay:EDGE2	Specify which waveform edge to use for delay measurements
MEASUrement:IMMed:SOURCE[1]	Specify the channel (or histogram with the TDS 500D & 700D) to take measurement from
MEASUrement:IMMed:SOURCE2	Specify the second channel to take measurement from

	(delay or "to" channel)
MEASUrement:IMMed:TYPe	Specify the measurement to be taken
MEASUrement:IMMed:UNITS?	Return measurement units
MEASUrement:IMMed:VALue?	Return measurement result
MEASUrement:MEAS<x>?	Return parameters on measurement
MEASUrement:MEAS<x>:COUNt?	Returns the number of values accumulated for this measurement since the last statistical reset (TDS 500D & 700D)
MEASUrement:MEAS<x>:DELay?	Return delay measurement parameters.
MEASUrement:MEAS<x>:DELay:DIRection	Search direction to use for delay measurements
MEASUrement:MEAS<x>:DELay:EDGE1	Which waveform edge to use for delay measurements
MEASUrement:MEAS<x>:DELay:EDGE2	Which waveform edge to use for delay measurements
MEASUrement:MEAS<x>:MAXimum?	Returns the maximum value found for this measurement since the last statistical reset (TDS 500D & 700D)
MEASUrement:MEAS<x>:MEAN?	Returns the mean value accumulated for this measurement since the last statistical reset (TDS 500D & 700D)
MEASUrement:MEAS<x>:MINImum?	Returns the minimum value found for this measurement since the last statistical reset (TDS 500D & 700D)
MEASUrement:MEAS<x>:SOURCE[1]	Channel (or histogram with the TDS 500D & 700D) to take measurement from
MEASUrement:MEAS<x>:SOURCE2	Second channel (or

	histogram with the TDS 500D & 700D) to take measurement from (delay or "to" channel)
MEASUrement:MEAS<x>:STATE	Turn measurement display on or off
MEASUrement:MEAS<x>:STDdev	Returns the standard deviation of values accumulated for this measurement since the last statistical reset (TDS 500D & 700D)
MEASUrement:MEAS<x>:TYPe	The measurement to be taken
MEASUrement:MEAS<x>:UNITS?	Return units to use for measurement
MEASUrement:MEAS<x>:VALue?	Return measurement result
MEASUrement:METHod	Method for calculating reference levels
MEASUrement:REFLevel?	Return reference levels
MEASUrement:REFLevel:ABSolute:HIGH	The top level for risetime (90% level)
MEASUrement:REFLevel:ABSolute:LOW	The low level for risetime (10% level)
MEASUrement:REFLevel:ABSolute:MID	Mid level for measurements
MEASUrement:REFLevel:ABSolute:MID2	Mid level for delay measurements
MEASUrement:REFLevel:METHod	Method to assign HIGH and LOW levels: either % or absolute volts
MEASUrement:REFLevel:PERCent:HIGH	The top level for risetime (90% level)
MEASUrement:REFLevel:PERCent:LOW	The low level for risetime (10% level)
MEASUrement:REFLevel:PERCent:MID	Mid level for measurements
MEASUrement:REFLevel:PERCent:MID2	Mid level for delay measurements

MEASUrement:SNAPShot	Display measurement snapshot
MEASUrement:STATIstics:MODe	Controls operation and display of measurement statistics (TDS 500D & 700D)
MEASUrement:STATIstics:WEIghting	Controls the responsiveness of mean and standard deviation to waveform changes (TDS 500D & 700D)

Miscellaneous Commands

Miscellaneous commands do not fit into other categories. Table 2-17 lists these commands.

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

Table 2-17: Miscellaneous Commands

Header	Description
AUTOSet	Automatic instrument setup
BELl	Audio alert
DATE	Set date
*DDT	Define group execute trigger (GET)
FACtory	Reset to factory default
HDR	Same as HEADer
HEADer	Return command header with query
*IDN?	Identification
*LRN?	Learn device setting
LOCk	Lock front panel (local lockout)
NEWpass	Change password for User Protected Data
PASSWord	Access to change User Protected Data
REM	No action; remark only
SET?	Same as *LRN?
TEKSecure	Initialize waveforms and setups
TIMe	Set time
*TRG	Perform Group Execute Trigger (GET)
*TST?	Self-test
UNLock	Unlock front panel (local lockout)
VERBose	Return full command name or minimum spellings with query

RS-232 Commands

RS-232 commands allow you to utilize the serial communications port (available with the RS-232/Centronics Hardcopy Interface). Table 2-18 lists these commands.

Table 2-18: RS-232 Commands

Header	Description
RS232?	Return RS-232 parameters
RS232:BAUd	Set baud rate
RS232:HARDFlagging	Set hard flagging
RS232:PARity	Set parity
RS232:SOFTFlagging	Set soft flagging
RS232:STOPBits	Set # of stop bits

Save and Recall Commands

Save and Recall commands allow you to store and retrieve internal waveforms and settings. When you "save a setup," you save all the settings of the oscilloscope. When you then "recall a setup," the oscilloscope restores itself to the state it was in when you originally saved that setting. Table 2-19 lists these commands.

Table 2-19: Save and Recall Commands

Header	Description
ALLOcate?	Return the number of allocated and unallocated data points
ALLOcate:WAVEform?	Return the number of allocated data points
ALLOcate:WAVEform:FREE?	Return the number of unallocated data points
ALLOcate:WAVEform:REF<x>	Specify the number of allocated data points
DELEte:SETUp	Delete a stored setup
DELEte:WAVEform	Delete a stored waveform
*RCL	Recall settings
RECAll:ACQDATA	Replace the indicated channel's live acquisition data with data from indicated file (TDS 500D & 700D)
RECALL:IMAGEHistogram	Recall an image histogram from disk
RECAll:SETUp	Recall saved instrument settings
RECAll:WAVEform	Recall saved waveform (File System Only)
*SAV	Save settings
SAVe:ACQDATA	Write out acquisition data to the indicated file (TDS 500D & 700D)
SAVe:IMAGEHistogram	Save an image histogram to disk
SAVe:SETUp	Save instrument settings
SAVe:WAVEform	Save waveform
SAVe:WAVEform:FILEFormat	Specifies the file format for saved waveforms (Not available for the TDS 510A)

Status and Error Commands

Status and error commands allow you to determine the status of the oscilloscope and to control events. Table 2-20 lists these commands.

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

Table 2-20: Status and Error Commands

Header	Description
ALLEv?	Return all events
BUSY?	Return scope status
*CLS	Clear status
DESE	Device event status enable
*ESE	Event status enable
*ESR?	Return standard event status register
EVENT?	Return event code
EVMsg?	Return event code and message
EVQty?	Return number of events in queue
ID?	Identification
*OPC	Operation complete
*OPT?	Return installed options (TDS 510A, 500D, 600B, & 700D)
*PSC	Power-on status clear
*PUD	Query or set User Protected Data
*RST	Reset
*SRE	Service request enable
*STB?	Read status byte
*WAI	Wait to continue

Trigger Commands

Trigger commands control all aspects of oscilloscope triggering. Table 2-21 lists these commands.

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

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

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

Table 2-21: Trigger Commands

Header	Description
TRIGger	Force trigger event; Return parameters.
TRIGger:DELay	Delay trigger level to 50%
TRIGger:DELay:BY	Delay by time or events
TRIGger:DELay:EDGE?	Return delay trigger parameters
TRIGger:DELay:EDGE:COUPling	Delay trigger coupling
TRIGger:DELay:EDGE:SLOpe	Delay trigger slope
TRIGger:DELay:EDGE:SOUrce	Delay trigger source
TRIGger:DELay:EVENTS?	Return delay trigger event parameters
TRIGger:DELay:EVENTS:COUNt	Delay by events count
TRIGger:DELay:LEVel	Delay trigger level
TRIGger:DELay:TIMe	Time for delay by time
TRIGger:DELay:TYPe	Delay trigger, edge
TRIGger:MAIn	Main trigger level to 50%
TRIGger:MAIn:COMMunication:AMI:PULSEForm	AMI pulse form setting (an Eye Diagram, AMI pulse form setting (an Isolated +1, or an Isolated -1) (TDS 500D & 700D)
TRIGger:MAIn:COMMunication:AMI:THReshold:HIGH	AMI threshold high level (TDS 500D & 700D)
TRIGger:MAIn:COMMunication:AMI:THReshold:LOW	AMI threshold low level (TDS 500D & 700D)
TRIGger:MAIn:COMMunication	

:BITRate	Communication trigger bit rate (TD 500D & 700D)
TRIGger:MAIn:COMMunication:CMI:PULSEForm	CMI pulse form (TDS 500D & 700D)
TRIGger:MAIn:COMMunication:CODe	Select which line code the comm trigger should expect on the incoming signal (TDS 500D & 700D)
TRIGger:MAIn:COMMunication:NRZ:PULSEForm	NRZ pulse form (TDS 500D & 700D)
TRIGger:MAIn:COMMunication:SOUrce	Communication trigger source channel (TDS 500D & 700D)
TRIGger:MAIn:COMMunication:STANdard	Communication trigger standard which identifies the code and bit rate (TDS 500D & 700D)
TRIGger:MAIn:EDGE?	Return main edge trigger parameters
TRIGger:MAIn:EDGE:COUPling	Main trigger coupling
TRIGger:MAIn:EDGE:SLOpe	Main trigger slope
TRIGger:MAIn:EDGE:SOUrce	Main trigger source
TRIGger:MAIn:HOLDOff?	Main trigger holdoff value
TRIGger:MAIn:HOLDOff:ACTUal?	Return main trigger holdoff value in seconds (TDS 500D, 600B, & 700D)
TRIGger:MAIn:HOLDOff:BY	Main trigger holdoff default (TDS 500D, 600B, & 700D)
TRIGger:MAIn:HOLDOff:TIMe	Main trigger holdoff time (TDS 500D, 600B, & 700D)
TRIGger:MAIn:HOLDOff:VALue	Main trigger holdoff value (TDS 400A & 510A)
TRIGger:MAIn:LEVel	Main trigger level
TRIGger:MAIn:LOGIc?	Returns main logic trigger parameters (TDS 510A, 500D, 600B, & 700D)
TRIGger:MAIn:LOGIc:CLAss	Logic trigger input usage (TDS 510A, 500D, 600B, & 700D)
TRIGger:MAIn:LOGIc:FUNCtion	Logic trigger input combining (TDS 510A, 500D, 600B, & 700D)

TRIGger:MAIn:LOGIc:INPut?	Return main logic trigger input settings (TDS 510A, 500D, 600B, & 700D)
TRIGger:MAIn:LOGIc:INPut:CH<x>	Logic trigger expected channel state (TDS 510A, 500D, 600B, & 700D)
TRIGger:MAIn:LOGIc:PATtern:INPut:CH4	Logic trigger expected for channel 4 pattern class (TDS 510A, 500D, 600B, & 700D)
TRIGger:MAIn:LOGIc:PATtern:WHEn	Main logic pattern trigger condition (TDS 510A, 500D, 600B, & 700D)
TRIGger:MAIn:LOGIc:PATtern:WHEn:LESSLimit	Maximum time the selected pattern may be true and still generate main logic pattern trigger (TDS 510A, 500D, 600B, & 700D)
TRIGger:MAIn:LOGIc:PATtern:WHEn:MORELimit	Minimum time the selected pattern may be true and still generate main logic pattern trigger (TDS 510A, 500D, 600B, & 700D)
TRIGger:MAIn:LOGIc:SETHold:CLOCk:EDGE	Clock edge polarity for setup and hold violation triggering (TDS 500D, 600B, & 700D)
TRIGger:MAIn:LOGIc:SETHold:CLOCk:LEVel	Setup/Hold clock voltage trigger level (TDS 500D, 600B, & 700D)
TRIGger:MAIn:LOGIc:SETHold:CLOCk:SOUrce	Setup/Hold clock input source (TDS 500D, 600B, & 700D)
TRIGger:MAIn:LOGIc:SETHold:DATa:LEVel	Setup/Hold data level (TDS 500D, 600B, & 700D)
TRIGger:MAIn:LOGIc:SETHold:DATa:SOUrce	Setup/Hold data input data channel (TDS 500D, 600B, & 700D)
TRIGger:MAIn:LOGIc:SETHold:HOLDTime	Setup/Hold trigger hold time (TDS 500D, 600B, & 700D)
TRIGger:MAIn:LOGIc:SETHold	

:SETTime	Setup/Hold trigger set time (TDS 500D, 600B, & 700D)
TRIGger:MAIn:LOGIc:STATE:INPut:CH4	Logic trigger expected for channel 4 state class (TDS 510A, 500D, 600B, & 700D)
TRIGger:MAIn:LOGIc:STATE:WHEn	When the logic trigger occurs (on true or false) (TDS 510A, 500D, 600B, & 700D)
TRIGger:MAIn:LOGIc:THReshold?	Return main logic thresholds (TDS 510A, 500D, 600B, & 700D)
TRIGger:MAIn:LOGIc:THReshold:CH<x>	Logic trigger thresholds (TDS 510A, 500D, 600B, & 700D)
TRIGger:MAIn:LOGIc:WHEn	Logic trigger on combination true or false (TDS 510A, 500D, 600B, & 700D)
TRIGger:MAIn:MODe	Main trigger mode
TRIGger:MAIn:PULse?	Return pulse trigger parameters (TDS 510A, 500D, 600B, & 700D)
TRIGger:MAIn:PULse:CLAss	Pulse trigger class (TDS 510A, 500D, 600B, & 700D)
TRIGger:MAIn:PULse:GLItch?	Returns glitch trigger parameters (TDS 510A, 500D, 600B, & 700D)
TRIGger:MAIn:PULse:GLItch:FILTer	Glitch filter on and off (TDS 510A, 500D, 600B, & 700D)
TRIGger:MAIn:PULse:GLItch:POLarity	Glitch filter positive, negative, or both (TDS 510A, 500D, 600B, & 700D)
TRIGger:MAIn:PULse:GLItch:WIDth	Glitch trigger with differentiation between glitch and valid pulse (TDS 510A, 500D, 600B, & 700D)
TRIGger:MAIn:PULse:RUNT?	Return runt trigger parameters (TDS 510A, 500D, 600B, & 700D)

TRIGger:MAIn:PULse:RUNT:POLarity	Runt trigger positive, negative, or both (TDS 510A, 500D, 600B, & 700D)
TRIGger:MAIn:PULse:RUNT:THReshold?	Return runt trigger thresholds (TDS 510A, 500D, 600B, & 700D)
TRIGger:MAIn:PULse:RUNT:THReshold:BOTh	Trigger level switching thresholds (TDS 500D, 600B, & 700D)
TRIGger:MAIn:PULse:RUNT:THReshold:HIGH	Upper limit for runt pulse (TDS 510A, 500D, 600B, & 700D)
TRIGger:MAIn:PULse:RUNT:THReshold:LOW	Lower limit for runt pulse (TDS 510A, 500D, 600B, & 700D)
TRIGger:MAIn:PULse:RUNT:WHEn	Runt pulse width type to check for (TDS 500D, 600B, & 700D)
TRIGger:MAIn:PULse:RUNT:WIDth	Minimum width for valid main pulse runt trigger (TDS 500D, 600B, & 700D)
TRIGger:MAIn:PULse:SLEWRate:DELTATime	Slew rate trigger delta time (TDS 500D, 600B, & 700D)
TRIGger:MAIn:PULse:SLEWRate:POLarity	Slew rate trigger polarity (TDS 500D, 600B, & 700D)
TRIGger:MAIn:PULse:SLEWRate:SLEWRate?	Return slew rate value (TDS 500D, 600B, & 700D)
TRIGger:MAIn:PULse:SLEWRate:THReshold:BOTh	Upper and lower slew rate trigger thresholds (TDS 500D, 600B, & 700D)
TRIGger:MAIn:PULse:SLEWRate:THReshold:HIGH	Upper limit for slew rate pulse (TDS 500D, 600B, & 700D)
TRIGger:MAIn:PULse:SLEWRate:THReshold:LOW	Lower limit for slew rate pulse (TDS 500D, 600B, & 700D)
TRIGger:MAIn:PULse:SLEWRate:WHEn	Slewing signal type to

	check for (TDS 500D, 600B, & 700D)
TRIGger:MAIn:PULse:SOUrce	Pulse trigger channel (TDS 510A, 500D, 600B, & 700D)
TRIGger:MAIn:PULse:TIMEOut:POLarity	Pulse timeout trigger polarity (TDS 500D, 600B, & 700D)
TRIGger:MAIn:PULse:TIMEOut:TIMe	Pulse timeout trigger time (TDS 500D, 600B, & 700D)
TRIGger:MAIn:PULse:WIDth?	Return trigger pulse width parameters (TDS 510A, 500D, 600B, & 700D)
TRIGger:MAIn:PULse:WIDth:HIGHLimit	Pulse trigger maximum pulse width (TDS 510A, 500D, 600B, & 700D)
TRIGger:MAIn:PULse:WIDth:LOWLimit	Pulse trigger minimum pulse width (TDS 510A, 500D, 600B, & 700D)
TRIGger:MAIn:PULse:WIDth:POLarity	Pulse trigger positive, negative, or both (TDS 510A, 500D, 600B, & 700D)
TRIGger:MAIn:PULse:WIDth:WHEn	Pulse trigger when pulse detected or when not detected (TDS 510A, 500D, 600B, & 700D)
TRIGger:MAIn:TYPe	Set main trigger edge, logic, pulse, or, with options, video and communications
TRIGger:MAIn:VIDeo?	Returns video trigger parameters (Option 05)
TRIGger:MAIn:VIDeo:BY	Set video trigger delay mode (TDS 400A Option 05)
TRIGger:MAIn:VIDeo:FIELD	Set video trigger field
TRIGger:MAIn:VIDeo:FIELDType	Set video trigger field type (TDS 510A, 500D, 600B, & 700D Option 05)
TRIGger:MAIn:VIDeo:FLEXformat?	Return FlexFormat parameters (TDS 510A, 500D, 600B, & 700D Option 05)
TRIGger:MAIn:VIDeo:FLEXfor	

mat:FIELDRATE	Set FlexFormat frames per second (TDS 510A, 500D, 600B Option 05)
TRIGger:MAIn:VIDeo:FLEXformat:FIELDS	Set FlexFormat video fields (TDS 510A, 500D, 600B, & 700D Option 05)
TRIGger:MAIn:VIDeo:FLEXformat:LINES	Set FlexFormat lines in a frame (TDS 510A, 500D, 600B, & 700D Option 05)
TRIGger:MAIn:VIDeo:FLEXformat:NEGSynchwidth	Set FlexFormat negative sync width (TDS 510A, 500D, 600B, & 700D Option 05)
TRIGger:MAIn:VIDeo:FLEXformat:V1STArttime	Set time from positive (+) edge of tri-sync pulse for the last line in the selected field to the leading edge (-) of the first negative vertical sync pulse (TDS 510A, 500D, 600B, & 700D Option 05)
TRIGger:MAIn:VIDeo:FLEXformat:V1STOptime	Set time from positive (+) edge of tri-sync pulse for the last line in the selected field (t0) to trailing edge (positive) of the first negative vertical sync pulse (TDS 510A, 500D, 600B, & 700D Option 05)
TRIGger:MAIn:VIDeo:FLEXformat:V2STArttime	Set time from t0 to the leading edge (positive) of the second vertical sync pulse (TDS 510A, 500D, 600B, & 700D Option 05)
TRIGger:MAIn:VIDeo:FLEXformat:V2STOptime	Set time from t0 to the trailing edge (positive) of the second negative vertical sync pulse (TDS 510A, 500D, 600B, & 700D Option 05)
TRIGger:MAIn:VIDeo:HDTv	High definition TV format (TDS 510A, 500D, 600B, & 700D Option 05)
TRIGger:MAIn:VIDeo:HOLdoff?	Return video trigger holdoff (TDS 400A Option 05)

TRIGger:MAIn:VIDeo:HOLdoff:VALue	Set video trigger holdoff value (TDS 400A Option 05)
TRIGger:MAIn:VIDeo:INTERLAce	Set video trigger interlace format (TDS 400A Option 05)
TRIGger:MAIn:VIDeo:LINE	Set video trigger delay in terms of a number of lines (Option 05)
TRIGger:MAIn:VIDeo:LINES	Set video trigger delay in terms of a number of lines (TDS 400A Option 05)
TRIGger:MAIn:VIDeo:NTSC	Select color or mono NTSC (TDS 510A, 500D, 600B, & 700D Option 05 Only)
TRIGger:MAIn:VIDeo:PAL	Select color or mono PAL (TDS 510A, 500D, 600B, & 700D Option 05 Only)
TRIGger:MAIn:VIDeo:SCAN	Set video trigger scan rate (TDS 400A Option 05)
TRIGger:MAIn:VIDeo:SCANPeriod	Set video scan period (TDS 400A Option 05)
TRIGger:MAIn:VIDeo:SOUrce	Select video trigger source (Option 05)
TRIGger:MAIn:VIDeo:STANdard	Select video trigger standard (TDS 510A, 500D, 600B, & 700D Option 05)
TRIGger:MAIn:VIDeo:SYNc	Select video trigger sync polarity (Option 05)
TRIGger:MAIn:VIDeo:SYStem	Select video trigger class (TDS 400A Option 05)
TRIGger:MAIn:VIDeo:TIMe	Set video trigger delay time (TDS 400A Option 05)
TRIGger:STATE?	Return trigger system status

Vertical Commands

Vertical commands control the display of channels and of main and reference waveforms. Table 2-22 lists these commands.

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

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

Table 2-22: Vertical Commands

Header	Description
CH<x>?	Return vertical parameters
CH<x>:BANdwidth	Channel bandwidth
CH<x>:COUPling	Channel coupling
CH<x>:DESKew	Channel deskew time (TDS 500D, 600B, & 700D)
CH<x>:IMPedance	Channel impedance
CH<x>:OFFSet	Channel offset
CH<x>:POSition	Channel position
CH<x>:PROBE?	Return channel probe attenuation
CH<x>:PROBECal?	Return channel probe cal status (TDS500D, 600B, & 700D)
CH<x>:PROBEFunc:EXTAtten	Sets the state of the external attenuation for the specified channel to the specified value (TDS 500D & 700D)
CH<x>:PROBEFunc:EXTDBatten	Sets the state of the external attenuation for the specified channel to the specified value in dB (TDS 500D & 700D)
CH<x>:SCAle	Channel volts per div
CH<x>:VOLts	Same as CH<x>:SCAle
MATH<x>?	Return math waveform definition
MATH<x>:DEFine	Define math waveform
MATH<x>:NUMAVg	Acquisition number at which to begin exponential averaging (TDS 510A, 500D, 600B, & 700D; some models require Option 2F)
MATH<x>:PROCessing	Math waveform averaging on or off (TDS 510A, 500D, 600B, & 700D; some models require Option 2F)

SELect?	Return selected waveform
SELect:<wfm>	Set selected waveform
SELect:CONTROl	Front-panel channel selector

Waveform Commands

Waveform commands let you transfer waveform data points to and from the oscilloscope. Waveform data points are a collection of values that define a waveform. One data value usually represents one data point in the waveform record. When working with enveloped waveforms, each data value is either the min or max of a min/max pair. Before you transfer waveform data, you must specify the data format, record length, and waveform locations. Table 2-23 lists these commands.

Table 2-23: Waveform Commands

Header	Description
CURVe	Transfer waveform data
DATa	The format and location of the waveform data that is transferred with the CURVe command
DATa:DESTination	Destination for waveforms sent to the oscilloscope
DATa:ENCdg	Waveform data encoding method
DATa:SOUrce	Source of CURVe? data
DATa:STARt	Starting point in waveform transfer
DATa:STOP	Ending point in waveform transfer
DATa:TARget	Same as DATa:DESTination
DATa:WIDth	Byte width of waveform points
WAVFrm?	Return waveform preamble and data
WAVPre?	Return waveform format data
WFMPre:BIT_Nr	Preamble bit width of waveform points
WFMPre:BN_Fmt	Preamble binary encoding type
WFMPre:BYT_Nr	Preamble byte width of waveform points
WFMPre:BYT_Or	Preamble byte order of waveform points
WFMPre:CRVchk	Preamble checksum of waveform points
WFMPre:ENCdg	Preamble encoding method
WFMPre:NR_Pt	Number of points in the curve
WFMPre:PT_Fmt	Format of curve points
WFMPre:PT_Off	Trigger position
WFMPre:WFId	Curve identifier
WFMPre:XINcr	Horizontal sampling interval

WFMPre:XMUlt	Horizontal scale factor
WFMPre:XOFf	Horizontal offset
WFMPre:XUNit	Horizontal units
WFMPre:XZEro	Horizontal origin offset
WFMPre:YMUlt	Vertical scale factor
WFMPre:YOFf	Vertical offset
WFMPre:YUNit	Vertical units
WFMPre:YZEro	Offset voltage
WFMPre:ZMUlt	Z-axis scale factor
WFMPre:ZOFf	Z-axis offset
WFMPre:ZUNit	Z-axis units
WFMPre:ZZEro	Z-axis origin offset
WFMPre:<wfm>:NR_Pt	Number of points in the curve
WFMPre:<wfm>:PT_Fmt	Format of curve points
WFMPre:<wfm>:PT_Off	Trigger position
WFMPre:<wfm>:WFId	Curve identifier
WFMPre:<wfm>:XINcr	Horizontal sampling interval
WFMPre:<wfm>:XUNit	Horizontal units
WFMPre:<wfm>:XZEro	Horizontal units (TDS 500D, 600B, & 700D)
WFMPre:<wfm>:YMUlt	Vertical scale factor
WFMPre:<wfm>:YOFf	Vertical offset
WFMPre:<wfm>:YUNit	Vertical units
WFMPre:<wfm>:YZEro	Offset voltage

Waveform Data Formats

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

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

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

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

Each data value requires two to seven characters: one to five characters to represent the value, another character, if the value is negative, to represent a minus sign, and a comma to separate the data points.

An example ASCII waveform data string may look like this:

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

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

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

The defined binary formats also specify the order in which the bytes are transferred. The four binary formats are RIBinary, RPBinary, SRIbinary, and SRPbinary.

RIBinary is a signed-integer format, where the most significant byte is transferred first, and RPBinary is a positive-integer format where the most significant byte is transferred first. SRIbinary and SRPbinary formats correspond to RIBinary and RPBinary formats respectively but use a swapped byte order where the least significant byte is transferred first. The byte order is ignored when DATa:WIDth is set to 1.

Waveform Data/Record Lengths

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

When transferring data into the oscilloscope, you must specify the location of the first data point within the waveform record. For example, when you set DATa:STARt to 1, data points will be stored starting with the first point in the record, and when you set DATa:STARt to 500, data will be stored starting at the 500th point in the record. The oscilloscope will ignore DATa:STOP when reading in data as it will stop reading data when it has no more data to read or when it has reached the specified record length.

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

Waveform Data Locations and Memory Allocation

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

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

Waveform Preamble

Each waveform that you transfer has an associated waveform preamble that contains information such as the horizontal scale, the vertical scale, and other settings in place when the waveform was created. Refer to the WFMPre command descriptions for more information about the waveform preamble.

Scaling Waveform Data

Once you transfer the waveform data to the controller, you can convert the data points into voltage values for analysis using information from the waveform preamble. The GETWFM program on the diskette that comes with this manual describes how to scale data.

Transferring Waveform Data from the TDS Family Oscilloscope

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

1. Select the waveform source(s) using the DATa:SOUrce command. If you want to transfer multiple waveforms, select more than one source.

2. Specify the waveform data format using DATa:ENCdg.

3. Specify the number of bytes per data point using DATa:WIDth.

4. Specify the portion of the waveform that you want to transfer using DATa:STARt and DATa:STOP.

5. Transfer waveform preamble information using WFMPRe? query.

6. Transfer waveform data from the oscilloscope using the CURVe?query.

Transferring Waveform Data to the Oscilloscope

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

1. Specify waveform reference memory using DATa:DESTination.

2. Specify the memory size for the reference location specified in Step 1 using the ALLOcate:WAVEFORM:REF<x> command.

3. Specify the waveform data format using DATa:ENCdg.

4. Specify the number of bytes per data point using DATa:WIDth.

5. Specify first data point in the waveform record using DATa:STARt.

6. Transfer waveform preamble information using WFMPRe:<wfm>.

7. Transfer waveform data to the oscilloscope using CURVe.

Extended-Acquisition Length Mode (TDS 500D & 700D, Option 2M)

Waveform commands also work with extended-acquisition-length mode. This mode lets the oscilloscope acquire extended acquisition lengths of 2M, 4M, and 8M while maintaining waveform record lengths limited to 500K.

In other words, the acquisition data for a channel may be bigger than a waveform you view. Then, after the data is acquired, you can move to and display any portion of the data.

This mode lets you select a subsection of the larger acquisition data. You can select the starting and ending positions in the acquisition data and the waveform record.

NOTE: To function properly, extended acquisition mode forces some oscilloscope modes and settings to new values. Also, measurements, gating, and cursors are restricted to the current waveform record.

Table 2-24 below shows the correlation between the number of channels in use and the waveform record lengths.

Table 2-24: Table of Extended-Acquisition-Length Mode Channels and Record Lengths

# of Channels in Use	Extended Acquisition Length	Waveform Record Length
1 of 4 (not on the TDS 520D and TDS 724D)	8 M samples	500 K samples
2 of 4 (or on the TDS 520D and TDS 724D: 1 of 2)	4 M samples	250 K samples
3 or 4 of 4 (or on the TDS 520D and TDS 724D: 2 of 2)	2 M samples	100 K samples

From a software development viewpoint, to define how much of the extended acquisition length data to include, you can use the DATa:STARt and DATa:STOP commands.

To define the waveform record within the extended acquisition data, you can use the HORizontal:RECORDSTART or HORizontal:POSition command.

The CURVe? query will return the extended acquisition data as you defined it with DATa:STARt and DATa:STOP. With extended-acquisition-mode set to on, this can be much larger than the waveform record. With extended-acquisition-mode set to off, this is equal to the waveform record.

Commands related to extended-acquisition-length mode operations include:
CURVe, DATa RECORDSHAp, DATa SOUrce, DATa:STARt, DATa:STOP, HORizontal:ACQDURATION?, HORizontal:ACQLENGTH, HORizontal:EXTDACQ, HORizontal:POSition, HORizontal:RECORDSTART, RECAll:ACQDATA, SAVe:ACQDATA, and WFMPre:<wfm?>.

NOTE: Extended-acquisition-length mode CURVe? queries can easily return more data than will fit on a 1.44 Mbyte floppy. For example, to return an entire 8 M extended acquisition mode record in ASCII format can require over 26.8 Mbytes of hard disk space. It can also take over 25 minutes to do this on some models.

Zoom Commands

Zoom commands let you expand and position the waveform display horizontally and vertically without changing the time base or vertical settings. Table 2-25 lists these commands.

Table 2-25: Zoom Commands

Header	Description
ZOOm	Reset zoom parameters to defaults
ZOOm:DUAl	Turn dual zoom mode on and off (Not on TDS 510A)
ZOOm:DUAl:OFFSet	Adjust the requested horizontal offset between the centers of the main and second zoom boxes (Not on TDS 510A)
ZOOm:GRAticule	Select between the upper and lower graticule for use by the zoom preview state (Not on TDS 510A)
ZOOm:HORizontal:LOCk	Horizontal zoom lock
ZOOm:HORizontal:POSition	Horizontal zoom position
ZOOm:HORizontal:SCAle	Horizontal zoom scale
ZOOm:STATE	Turn zoom mode on or off (you can also turn on the preview mode - not on the TDS 510A)
ZOOm:VERTical:POSition	Vertical zoom position
ZOOm:VERTical:SCAle	Vertical zoom scale

Syntax

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

Backus-Naur Grammar Overview

This help file describes commands and queries using Backus-Naur Form (BNF) notation.

This file uses the following BNF symbols shown in Table 2-1 below:

Table 2-1: BNF Symbols and Meanings

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

Command and Query Structure

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

Most commands have both a set form and a query form. The query form of the command differs from the set form by its question mark on the end. For example, the set command ACQuire:MODe has a query form ACQuire:MODe?. Not all commands have both a set and a query form. Some commands have set only and some have query only.

A command message is a command or query name followed by any information the oscilloscope needs to execute the command or query. Command messages may contain five element types, defined in the Table 2-2 shown below.

Table 2-2: Command Message Elements

Symbol	Meaning
<Header>	The basic command name. If the header ends with a question mark, the command is a query. The header may begin with a colon (:) character. If the command is concatenated with other commands, the beginning colon is required. Never use the beginning colon with command headers beginning with a star (*).
<Mnemonic>	A header sub-function. Some command headers have only one mnemonic. If a command header has multiple mnemonics, a colon (:) character always separates them from each other.
<Argument>	A quantity, quality, restriction, or limit associated with the header. Some commands have no argument while others have multiple arguments. A <Space> separates arguments from the header. A <Comma> separates arguments from each other.
<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.

Commands

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

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

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

Queries

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

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

You can specify a query command at any level within the command tree unless otherwise noted. These branch queries return information about all the mnemonics below the specified branch or level. For example, MEASUrement:MEAS<x>:DELay:DIRection? returns the starting point and direction of the edge of a delayed measurement, while MEASUrement:MEAS<x>:DELay? returns the current settings of all delayed measurement parameters, and MEASUrement:MEAS<x>? returns all the measurement parameters for the specified measurement.

Headers in Query Responses

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

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

Query	Header Off Response	Header On Response
APPMenu:TITLe?	"Test Setup"	:APPMENU:TITLE "Test Setup"
ACQuire:NUMAVg?	100	:ACQUIRE:NUMAVG 100

Clearing the TDS Family Oscilloscope

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

Argument Types

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

Numeric Arguments

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

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

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

Quoted String Arguments

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

"this is a quoted string"

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.

2. Use the same type of quote character to open and close the string:

   "this is a valid string"

3. You can mix quotation marks within a string as long as you follow the previous rule:

   "this is an 'acceptable' string"

4. You can include a quote character within a string simply by repeating the quote. For example,

   "here is a "" mark"

5. Strings can have upper or lower case characters.

6. If you use a GPIB network, you cannot terminate a quoted string with the END message before the closing delimiter.

7. A carriage return or line feed imbedded in a quoted string does not terminate the string, but is treated as just another character in the string.

8. The maximum length of a quoted string returned from a query is 1000 characters.

   Here are some invalid strings:

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

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

Block Arguments

Several oscilloscope commands use a block argument form:

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

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

Constructed Mnemonics

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

Application Menu Mnemonics

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

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

Color Index Mnemonics

Commands can specify a color index as a mnemonic in the header. The color index is specified in this way:

Symbol	Meaning
P<x>	A color index specifier; <x> is either 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, or 15

Cursor Position Mnemonics

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

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

Mask Specifier Mnemonics

Commands can specify which mask to set or query as a mnemonic in the header. The masks are specified in this way:

Symbol	Meaning
MASK<x>	A mask specifier; <x> is either 1, 2, 3, 4, 5, 6, 7, or 8

Measurement Specifier Mnemonics

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

Symbol	Meaning
MEAS<x>	A measurement specifier; <x> is either 1 [top], 2, 3, or 4 [bottom].

Channel Mnemonics

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

Symbol	Meaning
CH<x>	A channel specifier; <x> is either 1, 2, 3, or 4. For the TDS 520D, 620B, 680B, and TDS 724D, CH3 and CH4 represent the front-panel inputs labeled AUX 1 and AUX 2 respectively.

Math Waveform Mnemonics

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

Symbol	Meaning
MATH<x>	A math waveform specifier; <x> is 1, 2, or 3.

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 either 1, 2, 3, or 4.

Waveform Mnemonics

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

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

===

Command Entry

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

Abbreviating Commands

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

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

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

Concatenating Commands

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

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

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

   TRIGger:MODe NORMal;:ACQuire:NUMAVg 10

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

   ACQuire:MODe ENVelope; NUMAVg 10

   The longer version works equally well:

   ACQuire:MODe ENVelope;:ACQuire:NUMAVg 10

3. Never precede a star (*) command with a colon:

   ACQuire:MODe ENVelope;*TRG

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

   ACQuire:MODe ENVelope;*TRG;NUMAVg 10

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

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

   DISplay:INTENsity:TEXt?;WAVEform?

   will return either

   :DISPLAY:INTENSITY:TEXT 80;:DISPLAY:INTENSITY:WAVEFORM 90

   if header is on or

   80;90

   if header is off.

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

   ACQuire:MODe NORMal;NUMAVg?;STATE?

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

   Here are some invalid concatenations:

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

Message Terminators

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

Symbol	Meaning
<EOM>	Message terminator

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

ACQuire?

(Query Only)

Description: Returns all the current acquisition parameters.

Group: Acquisition

Syntax: ACQuire?

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

ACQuire:AUTOSAve

(TDS 500D, 600B, & 700D Only)

Description: Saves waveforms in reference memory when acquisition completes. Sending this command is equivalent to setting **Autosave Single Seq** in the Acquire menu and the corresponding side menu **Off** or **On** items.

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

When extended-acquisition-length mode is on, this command can still set and return values. However, this `ACQuire:AUTOSave` feature will not actually work while extended-acquisition-length mode is on.

Group: Acquisition

Syntax 1: ACQuire:AUTOSAve { OFF **|** ON **|** <NR1> }

Syntax 2: ACQuire:AUTOSAve?

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

ON or <NR1> turns autosave mode on.

Example 1: `ACQUIRE:AUTOSAVE 1`
turns autosave mode on.

Example 2: `ACQUIRE:AUTOSAVE OFF`
turns autosave off.

Example 3: `ACQUIRE:AUTOSAVE?`
might return `1`, indicating that autosave mode is on.

ACQuire:MODe

Description: Sets or queries the acquisition mode of the oscilloscope. This affects all live waveforms. Sending this command is equivalent to setting **Mode** in the Acquire menu.

Waveforms are the displayed data point values taken from acquisition intervals. Each acquisition interval represents a time duration set by the horizontal scale (time per division). The oscilloscope sampling system always samples at the maximum rate, and so an acquisition interval may include more than one sample.

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

When extended-acquisition-length mode is on, this command can still set and return values. However, the oscilloscope will treat all modes, except Peak Detect, as Sample mode.

Group: Acquisition

Related Commands: ACQuire:NUMAVg, ACQuire:NUMENv, CURVe?, DATa:WIDth

Syntax 1:
For the TDS 400A, 510A, 500D, & 700D:
ACQuire:MODe { SAMple **|** PEAKdetect **|** HIRes **|** AVErage **|** ENVelope }

For the TDS 600B:
ACQuire:MODe { SAMple **|** PEAKdetect **|** AVErage **|** ENVelope }

Syntax 2:
For all TDS:
ACQuire:MODe?

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

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

HIRes - (for the TDS 400A, 510A, 500D, & 700D) specifies Hi Res mode, where the displayed data point value is the average of all the samples taken during the acquisition interval. This is a form of averaging, where the average comes from a single waveform acquisition. The number of samples taken during the acquisition interval determines the number of data values that compose the average.

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

ENVelope - specifies envelope mode, where the resulting waveform shows the PEAKdetect range of data points from several separate waveform acquisitions. The number of waveform acquisitions that go into making up the envelope waveform is set or queried using the ACQuire:NUMENv command.

Example 1: `ACQUIRE:MODE ENVELOPE`
sets the acquisition mode to display a waveform that is an envelope of many individual waveform acquisitions.

Example 2: `ACQUIRE:MODE?`
might return `ENVELOPE`.

ACQuire:NUMACq?

(Query Only)

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

Group: Acquisition

Related Commands: ACQuire:STATE

Syntax: ACQuire:NUMACq?

Returns: <NR1>

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

ACQuire:NUMAVg

Description: Sets the number of waveform acquisitions that make up an averaged waveform. Sending this command is equivalent to setting the **Average** count in the Acquisition Mode side menu.

Group: Acquisition

Related Commands: ACQuire:MODe

Syntax 1: ACQuire:NUMAVg <NR1>

Syntax 2: ACQuire:NUMAVg?

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

Example 1: `ACQUIRE:NUMAVG 10`
specifies that an averaged waveform will show the result of combining 10 separately acquired waveforms.

Example 2: `ACQUIRE:NUMAVG?`
might return 75, indicating that there are 75 acquisitions specified for averaging.

ACQuire:NUMEnv

Description: Sets the number of waveform acquisitions that make up an envelope waveform. Sending this command is equivalent to setting the **Envelope** count in the Acquisition Mode side menu.

Group: Acquisition

Related Commands: ACQuire:MODe

Syntax 1: ACQuire:NUMEnv { <NR1> | INFInite }

Syntax 2: ACQuire:NUMEnv?

Argument 1: <NR1> NOT = 0 is the number of waveform acquisitions, from 1 to 2000. The envelope will restart after the specified number of envelopes have been acquired or when the ACQuire:STATE RUN command is sent.

Argument 2: INFInite or <NR1> = 0 specifies continuous enveloping.

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

Example 1: `ACQUIRE:NUMENV 10`
specifies that an enveloped waveform will show the result of combining 10 separately acquired waveforms.

Example 2: `ACQUIRE:NUMENV?`
might return `0`, indicating that acquisitions are acquired infinitely for enveloped waveforms.

ACQuire:REPEt

(TDS 400A, 510A, 500D, 600B, 700D Only)

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

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

Group: Acquisition

Related Commands: ACQuire:STATE, ACQuire:STOPAfter

Syntax 1: ACQuire:REPEt { OFF | ON | <NR1> }

Syntax 2: ACQuire:REPEt?

Argument 1: OFF or <NR1> = 0 turns repetitive mode off.

Argument 2: ON or <NR1> NOT = 0 turns repetitive mode on.

Example 1: `ACQUIRE:REPET 1`
turns repetitive mode on.

Example 2: `ACQUIRE:REPET OFF`
turns repetitive mode off.

Example 3: `ACQUIRE:REPET?`
might return 1, indicating that repetitive signal acquisition mode is on.

ACQuire:STATE

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

Group: Acquisition

Related Commands: ACQuire:NUMACq? , ACQuire:REPEt, ACQuire:STOPAfter

Syntax 1: ACQuire:STATE { OFF **|** ON **|** RUN **|** STOP **|** <NR1> }

Syntax 2: ACQuire:STATE?

Argument 1: OFF or STOP or <NR1> = 0 stops acquisitions.

Argument 2: ON or RUN or <NR1> NOT = 0 starts acquisition and display of waveforms. If the command was issued in the middle of an acquisition sequence (for instance averaging or enveloping), RUN restarts the sequence, discarding any data accumulated prior to the STOP. It also resets the number of acquisitions.

Example 1: `ACQUIRE:STATE RUN`
starts acquisition of waveform data and resets the number of acquisitions count (NUMACQ) to zero.

Example 2: `ACQUIRE:STATE?`
returns either `0` or `1`, depending on whether the acquisition system is running.

ACQuire:STOPAfter

Description: Tells the oscilloscope when to stop taking acquisitions. Sending this command is equivalent to setting **Stop After** in the Acquire menu.

When extended-acquisition-length mode is on, this command can still set and return values. However, this `ACQuire:STOPAfter` feature will not actually work while extended-acquisition-length mode is on.

Group: Acquisition

Related Commands: ACQuire:MODe, ACQuire:STATE, ACQuire:REPEt

Syntax 1: ACQuire:STOPAfter { RUNSTop | SEQuence | LIMit }

Syntax 2: ACQuire:STOPAfter?

Argument 1: RUNSTop specifies that the run and stop state should be determined by the user pressing the front-panel **RUN/STOP** button.

Argument 2: SEQuence specifies "single sequence" operation, where the oscilloscope stops after it has acquired enough waveforms to satisfy the conditions of the acquisition mode. For example, if the acquisition mode is set to sample, and the horizontal scale is set to a speed that allows real-time operation, then the oscilloscope will stop after digitizing a waveform from a single trigger event. However, if the acquisition mode is set to average 100 waveforms, then the oscilloscope will stop only after all 100 waveforms have been acquired. The ACQuire: STATE command and the front-panel **RUN/STOP** button will also stop acquisition when the oscilloscope is in single sequence mode.

Argument 3: LIMit specifies the oscilloscope stops after the limit test condition is met.

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

Example 1: `ACQUIRE:STOPAFTER RUNSTop`
sets the oscilloscope to stop acquisition when the user presses the
front-panel **RUN/STOP** button.

Example 2: `ACQUIRE:STOPAFTER?`
might return `SEQUENCE`.

ALIas

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

Group: Alias

Syntax 1: ALIas { OFF | ON | <NR1> }

Syntax 2: ALIas?

Argument 1: OFF or <NR1> = 0 turns alias expansion off. If a defined alias label is sent when ALIas is OFF, an execution error (110, "Command header error") will be generated.

Argument 2: ON or <NR1> NOT = 0 turns alias expansion on. When a defined alias is received, the specified command sequence is substituted for the alias and executed.

Example 1: `ALIAS ON`
turns the alias feature on.

Example 2: `ALIAS?`
returns `1`
when aliases are on.

ALIas:CATalog?

(Query Only)

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

Group: Alias

Syntax: ALIas:CATalog?

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

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

ALIas:DEFIne

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

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

Group: Alias

Syntax 1: ALIas:DEFIne <QString><Comma>{ <QString> **|** <Block> }

Syntax 2: ALIas:DEFIne? <QString>

Argument 1: 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 < or = 12 characters.

Argument 2: 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 < or = 80 characters.

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

Example 1:
```
ALIAS:DEFINE "ST1",":RECALL:SETUP 5;:AUTOSET EXECUTE;:SELECT:CH1 ON"
```
defines an alias named "ST1" that sets up the oscilloscope.

Example 2:
```
ALIAS:DEFINE? "ST1"
```
might return
```
:ALIAS:DEFINE "ST1",#239:RECALL:SETUP 5;:AUTOSET EXECUTE;:SELECT:CH1 ON
```

ALIas:DELEte

(No Query Form)

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

Group: Alias

Syntax: ALIas:DELEte <QString>

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

Example: `ALIAS:DELETE "SETUP1"`
deletes the alias named SETUP1.

ALIas:DELEte:ALL

(No Query Form)

Description: Deletes all existing aliases.

Group: Alias

Syntax: ALIas:DELEte:ALL

Example: `ALIAS:DELETE:ALL`
deletes all aliases.

ALIas:DELEte:NAMe

(No Query Form)

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

Group: Alias

Syntax: ALIas:DELEte:NAMe <QString>

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

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

ALIas:STATE

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

Group: Alias

Syntax 1: ALIas:STATE { OFF | ON | <NR1> }

Syntax 2: ALIas:STATE?

Argument 1: OFF or <NR1> = 0 turns alias expansion off. If a defined alias is sent when ALIas:STATE is OFF, a command error (102) will be generated.

Argument 2: ON or <NR1> NOT = 0 turns alias expansion on. When a defined alias is received, the specified command sequence is substituted for the alias and executed.

Example 1: `ALIAS:STATE OFF`
turns the command alias feature off.

Example 2: `ALIAS:STATE?`
returns `0` when alias mode is off.

ALLEv?

(Query Only)

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

Group: Status and error

Related Commands: *CLS, DESE, *ESE, *ESR, EVENT?, EVMsg?, EVQty, *SRE, *STB?

Syntax: ALLEv?

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

<EventCode><Comma><QString>[<Comma><EventCode><Comma><QString>...]

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

<Command> is the command that caused the error and may be returned when a command error is detected by the oscilloscope. As much of the command will be returned as possible without exceeding the 60 character limit of the <Message> and <Command> strings combined. The command string is right-justified.

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

ALLOcate?

(Query Only)

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

Group: Save and Recall

Syntax: ALLOcate?

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

ALLOcate:WAVEform?

(Query Only)

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

Group: Save and Recall

Syntax: ALLOcate:WAVEform?

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

ALLOcate:WAVEform:FREE?

(Query Only)

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

Group: Save and Recall

Syntax: ALLOcate:WAVEform:FREE?

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

Example: `ALLOCATE:WAVEFORM:FREE?`
might return `520` indicating that there are approximately 500 data points available for allocation. The extra 20 are used for administration purposes.

ALLOcate:WAVEform:REF<x>

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

Group: Save and Recall

Syntax 1: ALLOcate:WAVEform:REF<x> <NR1>

Syntax 2: ALLOcate:WAVEform:REF<x>?

Argument 1: <NR1> = 0 is returned when the reference location is empty.

Argument 2: <NR1> NOT = 0 specifies the number of data points. Table 2-23 below shows the number of data points supported for reference locations by TDS model. In the TDS 500D and 600B, all invalid values less than the maximum will be forced to the next highest valid value, and those higher than the maximum will be forced to the maximum. For example, 15002 points on a TDS 520D with option 1M will allocate 50000 points of data for the reference. No complete references are stored for 500000 data points in the TDS 700D.

Table 2-23: Waveform Data Points Supported for Reference Locations

Configuration	TDS 420A, TDS 430A, TDS 460A	TDS 510A, TDS 520D, TDS 724D	TDS 600B	TDS 540D, TDS 754D, TDS 784D
Standard	500, 1000, 2500, 5000, 15000, 30000 (4 refs)	500, 1000, 2500, 5000, 15000, 50000	500, 1000, 2500, 5000, 15000	500, 1000, 2500, 5000, 15000, 50000 (4 refs)
Option 1M (not on the TDS 510A) or 2M (TDS 500D & 700D only)	60000 (2 refs), 120000 (1 ref)	TDS 520D & 724D only: 75000 (3 refs) 100000 (2 refs) 130000 (2 refs) 250000 (1 ref)		75000 (3 refs), 100000 (2 refs), 130000 (2 refs), 250000 (1 ref), 500000 (no complete refs)

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

Example 2: `ALLOCATE:WAVEFORM:REF1?`
might return `500`

APPMenu

Description: Displays the user-definable Application menu, and the query returns the current Application menu labels and title. Sending this command is equivalent to pressing the front-panel **APPLICATION** button.

Group: Application Menu

Related Commands: CLEARMenu, *ESR, EVENT?

Syntax 1: APPMenu ACTivate

Syntax 2: APPMenu?

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

Once the Application menu is activated, whenever a front-panel menu button is pressed an event is generated that tells which button was pressed.

Menu button presses will also generate Service Requests when the URQ bit is enabled in DESER and ESER and the ESB bit is enabled in SRER.

Example: `APPMENU ACTIVATE`
displays the application menu.

APPMenu:LABel

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

Group: Application Menu

Syntax 1: APPMenu:LABel CLEar

Syntax 2: APPMenu:LABel?

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

Example: `APPMENU:LABEL CLEAR`
clears the user-defined menu labels from the display.

APPMenu:LABel:BOTTOM<x>

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

Group: Application Menu

Syntax 1: APPMenu:LABel:BOTTOM<x> <QString>

Syntax 2: APPMenu:LABel:BOTTOM<x>?

Argument: <QString> is the menu button label and can include any legal TDS character. The maximum length of the label is 1000 characters. The TDS displays the label in the area above the specified main menu button.

The TDS displays the label on a single line and centers it, both vertically and horizontally, within the label area. You can embed a line feed character in the string to position the label on multiple lines. You can also use white space tab characters to position the label within a line.

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

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

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

The label area is 45 pixels high and 90 pixels wide. The length of the label that fits in the label area depends on the contents of the label, because the width of characters varies. The label area is about 10 characters wide and 3 lines high.

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

Example: `APPMENU:LABEL:BOTTOM3 "SETUP1"`
assigns the label "SETUP1" to the third main menu button.

APPMenu:LABel:RIGHT<x>

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

Group: Application Menu

Syntax 1: APPMenu:LABel:RIGHT<x> <QString>

Syntax 2: APPMenu:LABel:RIGHT<x>?

Argument: <QString> is the menu button label and can include any legal TDS character. The maximum length of the label is 1000 characters. The label is displayed in the area to the left of the specified side menu button. Refer to the APPMenu:LABel:BOTTOM<x> command description for more information on defining menu labels.

The label area is 72 pixels high and 112 pixels wide. The length of the label that fits in the label area depends on the contents of the label, because the width of characters varies. The label area is about 12 characters wide and 2 lines high.

Example: `APPMENU:LABEL:RIGHT1 "TEST ON"` displays the label `"TEST ON"` next to the top side menu button.

APPMenu:TITLe

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

Group: Application Menu

Related Commands: APPMenu, APPMenu:LABel

Syntax 1: APPMenu:TITLe <QString>

Syntax 2: APPMenu:TITLe?

Argument: <QString> is the side menu title and can include any legal TDS character. The maximum length of the title is 1000 characters. The APPMenu:LABel:BOTTOM<x> command description provides information on defining menu labels.

The label area is 40 pixels high and 112 pixels wide. The length of the label that fits in the label area depends on the contents of the label, because the width of characters varies. The label area is about 12 characters wide and 4 lines high.

Example 1: `APPMENU:TITLE "Custom Menu"`
displays the title `"Custom Menu"` on the screen.

Example 2: `APPMENU:TITLE?`
might return `"Test Setup"` for the current application menu title.

AUTOSet

(No Query Form)

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

This command will turn the extended-acquisition-length mode to off.

Group: Miscellaneous

Syntax: AUTOSet EXECute

Example: `EXECute` autosets the displayed waveform.

BELl

(No Query Form)

Description: Beeps the audio indicator of the oscilloscope.

Group: Miscellaneous

Syntax: BELl

Example: `BELL`
rings the bell.

BUSY?

(Query Only)

Description: Returns the status of the oscilloscope. This command allows you to synchronize the operation of the oscilloscope with your application program. Synchronization Methods are described separately in this help system.

Group: Status and error

Related Commands: *OPC, *WAI

Syntax: BUSY?

Returns 1: <NR1> = 0 means that the oscilloscope is not busy processing a command whose execution time is extensive. These commands are listed in Table 2-24.

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

Table 2-24: Commands that Affect BUSY? Response

Operation	Command
Single sequence acquisition	ACQuire:STATE ON or ACQuire:STATE RUN (when ACQuire:STOPAfter is set to SEQuence)
Hardcopy output	HARDCopy STARt

Example: BUSY?
might return 1, indicating that the instrument is busy.

CAL?

(Query Only)

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

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

Group: Calibration and Diagnostic

Syntax: *CAL?

Returns 1: <NR1> = 0 indicates that the calibration completed without any errors detected.

Returns 2: <NR1> NOT = 0 indicates that the calibration did not complete successfully.

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

CH<x>?

(Query Only)

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

Group: Vertical

Syntax: CH<x>?

Example: `CH1?`
might return the string

```
:CH1:SCALE 10.0E-3;POSITION 0.0E+0;OFFSET 0.0E+0;COUPLING
DC;IMPEDANCE MEG;BANDWIDTH FULL
```

for channel 1.

CH<x>:BANdwidth

Description: Sets or queries the bandwidth setting of the specified channel. Sending this command is equivalent to setting **Bandwidth** in the Vertical menu. The TDS 794D has TWEnty, ONEFifty, and FULl settings with the P6339A probe, and it has FULl without the probe.

Group: Vertical

Syntax 1: CH<x>:BANdwidth { TWEnty | HUNdred | ONEFifty | TWOfifty | FULl }

Syntax 2: CH<x>:BANdwidth?

Argument 1: TWEnty sets the channel bandwidth to 20 MHz (requires a P6339A probe on the TDS 794D).

Argument 2: HUNdred sets the channel bandwidth to 100 MHz (TDS 400A & 510A)

Argument 2: ONEFifty sets the channel bandwidth to 150 MHz (TDS 794D with P6339A only)

Argument 3: TWOfifty sets the channel bandwidth to 250 MHz (TDS 500D, 600B, & 700D).

Argument 4: FULl sets the channel bandwidth to the full bandwidth of the oscilloscope.

Example 1: `CH2:BANDWIDTH TWENTY`
sets the bandwidth of channel 2 to 20 MHz.

Example 2: `CH1:BANDWIDTH?`
might return `FULL`, which indicates that there is no bandwidth limiting on channel 1.

CH<x>:COUPling

Description: Sets or queries the input attenuator coupling setting of the specified channel. Sending this command is equivalent to setting **Coupling** in the Vertical menu.

Group: Vertical

Related Commands: CH<x>:IMPedance

Syntax 1: CH<x>:COUPling { AC | DC | GND }

Syntax 2: CH<x>:COUPling?

Argument 1: AC sets the specified channel to AC coupling. TDS 794D oscilloscopes require a P6339A probe to use AC coupling.

Argument 2: DC sets the specified channel to DC coupling.

Argument 3: GND sets the specified channel to ground. Only a flat ground-level waveform will be displayed.

Example 1: `CH1:COUPLING AC`
sets AC coupling on channel 1.

Example 2: `CH3:COUPLING?`
might return `DC`, indicating that channel 3 is set to DC coupling.

CH<x>:DESKew

(TDS 500D, 600B, & 700D Only)

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

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

When extended-acquisition-length mode is on, this command can still set and return values. However, this deskew feature will not actually work while extended-acquisition-length mode is on.

Group: Vertical

Syntax 1: CH<x>:DESKew <NR3>

Syntax 2: CH<x>:DESKew?

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

CH<x>:IMPedance

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

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

Group: Vertical

Related Commands: CH<x>:COUPling

Syntax 1: CH<x>:IMPedance { FIFty **|** MEG }

Syntax 2: CH<x>:IMPedance?

Argument 1: FIFty sets the specified channel to 50 ohm impedance. Fifty is not available on a TDS 794D if a P6339A probe is attached.

Argument 2: MEG sets the specified channel to 1 M-ohm impedance. Meg is not available on a TDS 794D unless a P6339A probe is attached.

Example 1: `CH1:IMPEDANCE FIFty`
establishes 50 ohm impedance on channel 1.

Example 2: `CH3:IMPEDANCE?`
might return `MEG`, indicating that channel 3 is set to 1 M-ohm impedance.

CH<x>:OFFSet

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

Group: Vertical

Related Commands: CH<x>:POSition

Syntax 1: CH<x>:OFFSet <NR3>

Syntax 2: CH<x>:OFFSet?

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

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

CH<x>:SCAle	OFFSet Range
1 mV/div - 99.5 mV/div	+-1 V
100 mV/div - 995 mV/div	+-10 V
1 V/div - 10 V/div	+100 V

Table 2-26: Offset Ranges for the TDS 520D & 724D (Aux 1 & Aux 2) using a 1x Probe

CH<x>:SCAle	OFFSet Range
50 mV/div & 100 mV/div	+-0.5 V
500 mV/div & 1 V/div	+-5.0 V
5 V/div & 10 V/div	+50 V

Table 2-27: Offset Ranges for the TDS 794D

TDS 794D CH<x>:SCAle	OFFSet Range	TDS 794D with P6339A Probe CH<x>:SCAle	OFFSet Range
10 mV/div & 50 mV/div	+-0.5 V	10 mV/div & 500 mV/div	+-10.0 V
50.5 mV/div & 100 mV/div	+-0.25 V	501 mV/div & 5 V/div	+-100.0 V
101 V/div & 500 mV/div	+5 V	5.01 V/div & 100 V/div	+1000 V
505 mV/div & 1 V/div	+2.5 V		

Example 1: `CH1:OFFSET 0.5E+00`
lowers the channel 1 displayed waveform by 0.5 volts.

Example 2: `CH1:OFFSET?`
might return `500.0E-3`, indicating that the current channel 1 offset is 0.5 volts.

CH<x>:POSition

Description: Sets or queries the vertical position of the specified channel. The position value is applied to the signal before it is digitized. Sending this command is equivalent to setting **Position** in the Vertical menu or adjusting the front-panel **Vertical Position** knob.

Group: Vertical

Related Command: CH<x>:OFFSet

Syntax 1: CH<x>:POSition <NR3>

Syntax 2: CH<x>:POSition?

Argument: <NR3> is the desired position in divisions from the center graticule. The range is + or - 5 divisions.

Example 1: `CH2:POSITION 1.3E+00`
positions the channel 2 input signal 1.3 divisions above the center of the display.

Example 2: `CH1:POSITION?`
might return `-1.3E+00`, indicating that the current position of channel 1 is at -1.3 divisions.

CH<x>:PROBE?

(Query Only)

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

Group: Vertical

Syntax: CH<x>:PROBE?

Returns: <NR3>

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

CH<x>:PROBECal?

(TDS 500D, 600B, & 700D Only) (Query Only)

Description: Queries the probe cal status for the selected channel.

Group: Vertical

Syntax: CH<x>:PROBECal?

Output 1: FALI signifies that probe cal has failed for the selected channel.

Output 2: INIT signifies the probe cal has not yet been run for the selected channel.

Output 3: PASS signifies that probe cal has passed for the selected channel.

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

CH<x>:PROBEFunc:EXTAtten:<NR3>

(TDS 500D, 600B, & 700D Only)

Description: Sets and queries the external attenuation for the specified channel to the specified value. This command deals with the attenuation factor rather than the gain factor. This is unlike the CH<x>:PROBE? command.

Group: Vertical

Syntax: CH<x>:PROBEFunc:EXTAtten:<NR3>

Argument: <NR3> is the attenuation value specified as a multiplier in the range 1.00E-6 to 1.00E+6. The default is 1.00.

Example: `CH2:PROBEFUNC:EXTATTEN 15.00E+0`
sets the external attenuation to 15X.

CH<x>:PROBEFunc:EXTBatten:<NR3>

(TDS 500D, 600B, & 700D Only)

Description: Sets and queries the external attenuation for the specified channel to the specified value in dB. Note that 1X = 0dB, 10X = 20 dB, 100X = 40 dB, etc..

Group: Vertical

Syntax: CH<x>:PROBEFunc:EXTBatten:<NR3>

Argument: <NR3> is the attenuation value specified in the range -120.00 dB to 120.00 dB. The default is 0.00 dB.

Example: `CH2:PROBEFUNC:EXTDB 2.5`
sets the external attenuation to 2.5 dB.

CH<x>:SCAle

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

Group: Vertical

Related Command: CH1:VOLts

Syntax 1: CH<x>:SCAle <NR3>

Syntax 2: CH<x>:SCAle?

Argument: <NR3> is the gain in volts per division. The range is 1 mV per division to 10 V per division when using a 1x probe, except for the TDS 794D. The range for the TDS 794D is 10 mV per division to 1 V per division when using a 1x probe, and it is 10 mV per division to 100 V per division when using a P6339A probe.

Example 1: `CH4:SCALE 100E-03`
sets the channel 4 gain to 100 mV per division.

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

CH<x>:VOLts

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

Group: Vertical

Related Command: CH1:SCAle

Syntax 1: CH<x>:VOLts <NR3>

Syntax 2: CH<x>:VOLts?

Example 1: `CH4:VOLTS 100E-03`
sets the channel 4 gain to 100 mV per division.

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

CLEARMenu

(No Query Form)

Description: Clears the current menu from the display. Sending this command is equivalent to pressing the **CLEAR MENU** button on the front panel.

Group: Display

Syntax: CLEARMenu

Example: `CLEARMENU`
clears the menu from the display.

*CLS

(No Query Form)

Description: Clears the oscilloscope status data structures.

Group: Status and Error

Related Commands: DESE, *ESE, *ESR?, EVENT?, EVMsg?, *SRE, *STB?

Syntax: *CLS

The *CLS command clears the following:

o the Event Queue

o the Standard Event Status Register (SESR)

o the Status Byte Register (except the MAV bit; see below)

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 will clear the output queue and thus MAV. *CLS does not clear the output queue or MAV. (A complete discussion of these registers and bits, and of event handling in general, begins in the section on Status and Events.)

*CLS can suppress a Service Request that is to be generated by an *OPC. This will happen if a hardcopy output or single sequence acquisition operation is still being processed when the *CLS command is executed.

CURSor?

(Query Only)

Description: Returns all current cursor settings.

Group: Cursor

Syntax: CURSor?

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

CURSor:FUNCtion

Description: Selects and displays the cursor type. Cursors are attached to the selected waveform. Sending this command is equivalent to setting **Function** in the Cursor menu.

Group: Cursor

Related Command: SELect:CONTROl

Syntax 1: CURSor:FUNCtion { HBArs | OFF | VBArs | PAIred }

Syntax 2: CURSor:FUNCtion?

Argument 1: HBArs specifies horizontal bar cursors that measure volts.

Argument 2: OFF removes the cursors from the display.

Argument 3: VBArs specifies vertical bar cursors that measure time.

Argument 4: PAIred specifies paired cursors that measure both time and volts.

Example: `CURSOR:FUNCtion VBARS`
selects vertical bar type cursors.

CURSor:HBArs?

(Query Only)

Description: Returns the current settings for the horizontal bar cursors.

Group: Cursor

Syntax: CURSor:HBArs?

Example: `CURSOR:HBARS?`
might return `:CURSOR:HBARS:POSITION1 0;POSITION2 0;SELECT CURSOR1.`

CURSor:HBArs:DELTa?

(Query Only)

Description: Returns the voltage difference between the two horizontal bar cursors.

Group: Cursor

Syntax: CURSor:HBArs:DELTa?

Returns: <NR3>

Example: `CURSOR:HBARS:DELTA?`
might return `5.08E+0` for the voltage difference between the two cursors.

CURSor:HBArs:POSITION<x>

Description: Positions a horizontal bar cursor.

Group: Cursor

Syntax 1: CURSor:HBArs:POSITION<x> <NR3>

Syntax 2: CURSor:HBArs:POSITION<x>?

Argument: <NR3> specifies the cursor position relative to ground, in volts.

Example 1: `CURSOR:HBARS:POSITION1 25.0E-3`
positions one of the horizontal cursors at 25.0 mV.

Example 2: `CURSOR:HBARS:POSITION2?`
might return `-64.0E-3`, indicating that one of the horizontal bar cursors is at -64.0 mV.

CURSor:HBArs:POSITION<x>Pcnt

(TDS 400A Only)

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

Group: Cursor

Syntax 1: CURSor:HBArs:POSITION<x>Pcnt <NR3>

Syntax 2: CURSor:HBArs:POSITION<x>Pcnt?

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

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

CURSor:HBArs:SELect

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

Group: Cursor

Syntax 1: CURSor:HBArs:SELect { CURSOR1 **|** CURSOR2 }

Syntax 2: CURSor:HBArs:SELect?

Argument 1: CURSOR1 selects the first horizontal bar cursor.

Argument 2: CURSOR2 selects the second horizontal bar cursor.

Example 1: `CURSOR:HBARS:SELECT CURSOR1`
selects the first horizontal bar cursor as the active cursor.

Example 2: `CURSOR:HBARS:SELECT?`
returns CURSOR1 when the first cursor is the active cursor.

CURSor:HBArs:UNITS

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

Group: Cursor

Syntax 1: CURSor:HBArs:UNITS { BASe **|** IRE }

Syntax 2: CURSor:HBArs:UNITS?

Argument 1: BASe specifies base as the unit of measure.

Argument 2: IRE specifies IRE as the unit of measure. These units are typically used with video signals.

Example 1: `CURSOR:HBARS:UNITS BASE`
sets the units for the horizontal bar cursors to base.

Example 2: `CURSOR:HBARS:UNITS?`
returns `IRE` when the horizontal bar cursor units are IRE.

CURSor:MODe

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

Group: Cursor

Related Command: CURSor:FUNCtion

Syntax 1: CURSor:MODe { TRACk **|** INDependent }

Syntax 2: CURSor:MODe?

Argument 1: TRACk ties the two cursors together as you move the general purpose knob.

Argument 2: INDependent frees the two cursors to move separately.

Example 1: `CURSOR:MODE TRACK`
specifies that the cursors positions move in unison.

Example 2: `CURSOR:MODE?`
might return `:TRACK` showing the two cursors move in unison.

CURSor:PAIred

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

In extended-acquisition-length mode, the cursors are limited to the waveform record section of the acquisition data.

Group: Cursor

Related Commands: DATa:STARt, DATa:STOP

Syntax 1: CURSor:PAIred:SNAp

Syntax 2: CURSor:PAIred?

Argument: SNAp positions the paired cursors at DATa:STARt and DATa:STOP.

Example 1: `CURSOR:PAIRED:SNAP`
specifies the positions of the cursors are at the current DATA:START and DATA:STOP values.

Example 2: `CURSOR:PAIRED?`
might return
`:CURSOR:PAIRED:UNITS BASE;POSITION1 -2.00E-3;POSITION2 2.00E-3;SELECT CURSOR2`

CURSor:PAIred:HDELTA

(Query Only)

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

Group: Cursor

Related Command: CURSor:FUNCtion

Syntax: CURSor:PAIred:HDELTA?

Example: `CURSOR:PAIRED:HDELTA?`
might return `5.08E+0` for the voltage difference between the two cursors.

CURSor:PAIred:HPOS1

(Query Only)

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

Group: Cursor

Related Command: CURSor:FUNCtion

Syntax: CURSor:PAIred:HPOS1?

Example: `CURSOR:PAIRED:HPOS1?`
might return `-64.0E-3`, indicating that the first cursor is at -64.0 mV.

CURSor:PAIred:HPOS2

(Query Only)

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

Group: Cursor

Related Command: CURSor:FUNCtion

Syntax: CURSor:PAIred:HPOS2?

Example: `CURSOR:PAIRED:HPOS2?`
might return `-64.0E-3`, indicating the second cursor is at -64.0 mV.

CURSor:PAIred:POSITION<x>

Description: Sets or queries the vertical bar (time) position of the first or second paired cursor. x is either 1 or 2 and refers to the first or second cursor.

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

In extended-acquisition-length mode, the paired cursor position must be within the waveform record (as opposed to the entire extended acquisition length) for any change to happen.

Group: Cursor

Related Command: CURSor:FUNCtion, CURSor:VBArs:UNITS

Syntax 1: CURSor:PAIred:POSITION<x> < NR3 >

Syntax 2: CURSor:PAIred:POSITION<x>?

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

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

Example 1: `CURSOR:PAIRED:POSITION1 9.00E-6`
specifies the first paired cursor is at 9 us.

Example 2: `CURSOR:POSITION1?`
might return `1.00E-6`, indicating that the first paired cursor is at 1 us.

CURSor:PAIred:POSITION<x>Pcnt

(TDS 400A Only)

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

Group: Cursor

Syntax 1: CURSor:PAIred:POSITION<x>Pcnt < NR3 >

Syntax 2: CURSor:PAIred:POSITION<x>Pcnt?

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

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

CURSor:PAIred:SELect

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

Group: Cursor

Syntax 1: CURSor:PAIred:SELect { CURSOR1 **|** CURSOR2 }

Syntax 2: CURSor:PAIred:SELect?

Argument 1: CURSOR1 specifies the first paired cursor.

Argument 2: CURSOR2 specifies the second paired cursor.

Example 1: `CURSOR:PAIRED:SELECT CURSOR2`
selects the second paired cursor as the active cursor.

Example 2: `CURSOR:PAIRED:SELECT?`
returns `CURSOR1` when the first paired cursor is the active cursor.

CURSor:PAIred:UNITS

Description: Sets or queries the units for the paired cursors. Sending this command is equivalent to setting **Amplitude** in the Cursor menu.

Group: Cursor

Syntax 1: CURSor:PAIred:UNITS { BASe **|** IRE }

Syntax 2: CURSor:PAIred:UNITS?

Argument 1: BASe specifies base as the unit of measure.

Argument 2: IRE specifies IRE as the unit of measure. These units are typically used with video signals.

Example 1: `CURSOR:PAIRED:UNITS BASE`
sets the units for the paired cursors to base.

Example 2: `CURSOR:PAIRED:UNITS?`
returns `IRE` when the paired cursor units are IRE.

CURSor:PAIred:VDELTA

(Query Only)

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

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

Group: Cursor

Related Command: CURSor:FUNCtion

Syntax: CURSor:PAIred:VDELTA?

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

CURSor:VBArs

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

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

In extended-acquisition-length mode, the cursors are limited to the waveform record section of the acquisition data.

Group: Cursor

Related Commands: DATa:STARt, DATa:STOP, MEASUrement:GATing

Syntax 1: CURSor:VBArs SNAp

Syntax 2: CURSor:VBArs?

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

Example 1: `CURSOR:VBARS SNAP`
specifies that the cursors positions are the same as the current DATA:START and DATA:STOP values.

Example 2: `CURSOR:VBARS?`
might return
`:CURSOR:VBARS:UNITS SECONDS;POSITION1 1.00E-6;POSITION2 9.00E-6;SELECT CURSOR2.`

CURSor:VBArs:DELTa?

(Query Only)

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

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

Group: Cursor

Related Command: CURSor:VBArs:UNITS

Syntax: CURSor:VBArs:DELTa?

Returns: <NR3>

Example: `CURSOR:VBARS:DELTa?`
might return `1.064E+00`, indicating that the time between the vertical bar cursors is 1.064 seconds.

CURSor:VBArs:POSITION<x>

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

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

Group: Cursor

Related Command: CURSor:VBArs:UNITS

Syntax 1: CURSor:VBArs:POSITION<x> <NR3>

Syntax 2: CURSor:VBArs:POSITION<x>?

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

Example 1: `CURSOR:VBARS:POSITION2 9.00E-6`
positions one of the vertical bar cursors at 9 ms.

Example 2: `CURSOR:VBARS:POSITION1?`
might return `1.00E-6`, indicating a vertical bar cursors is at 1 ms.

CURSor:VBArs:POSITION<x>Pcnt

(TDS 400A Only)

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

Group: Cursor

Syntax 1: CURSor:VBArs:POSITION<x>Pcnt <NR3>

Syntax 2: CURSor:VBArs:POSITION<x>Pcnt?

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

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

CURSor:VBArs:SELect

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

Group: Cursor

Syntax 1: CURSor:VBArs:SELect { CURSOR1 | CURSOR2 }

Syntax 2: CURSor:VBArs:SELect?

Argument 1: CURSOR1 specifies the first vertical bar cursor.

Argument 2: CURSOR2 specifies the second vertical bar cursor.

Example 1: `CURSOR:VBARS:SELECT CURSOR2`
selects the second vertical bar cursor as the active cursor.

Example 2: `CURSOR:VBARS:SELECT?`
returns `CURSOR1` when the first vertical bar cursor is the active cursor.

CURSor:VBArs:UNITS

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

Group: Cursor

Related Command:
CURSor:VBArs:DELTa?, CURSor:VBArs:POSITION<x>

Syntax 1: CURSor:VBArs:UNITS { SECOnds **|** HERtz **|** LINE (with option 05) **|** BASe (TDS 400A only) **|** INVert (TDS 400A only) }

Syntax 2: CURSor:VBArs:UNITS?

Argument 1: SEConds specifies seconds as the unit of measure. For the TDS 400A, this argument is available only for backward compatibility. If a TDS 400A receives this argument, it will convert it to BASE or INVERT depending on the selected waveform. The TDS 400A will not output this argument in response to a query.

Argument 2: HERtz specifies hertz as the unit of measure. For the TDS 400A, this argument is available only for backward compatibility. If a TDS 400A receives this argument, it will convert it to BASE or INVERT depending on the selected waveform. The TDS 400A will not output this argument in response to a query.

Argument 3: LINE specifies a video line as the unit of measure. This is most useful if you have option 05 video trigger installed. On some models, if you do not have option 05, use of this argument will generate an error message. On the TDS 400A, if you do not have option 05, the TDS 400A will use the NTSC standard for converting from time to lines. In order for the measurement to be accurate in the TDS 400A when TRIGGER:MAIN:VIDEO:SYSTEM is set to custom, you must set TRIGGER:MAIN:VIDEO:SCANPERIOD.

Argument 4: BASe (TDS 400A only) specifies the base (or natural) units of the waveform as the unit of measurement. If internal clock is enabled, the base units are seconds. If external clock is enabled, the base units are clocks.

Argument 5: INVert (TDS 400A only) specifies 1/base (or natural) units of the waveform as the unit of measurement. If internal clock is enabled, the base units are 1/seconds. If external clock is enabled, the base units are 1/clocks.

Example 1: `CURSOR:VBARS:UNITS SECONDS`
sets the units for the vertical bar cursors to seconds.

Example 2: `CURSOR:VBARS:UNITS?`
returns `HERTZ` when the vertical bar cursor units are hertz.

CURSor:VBArs:UNITSTring?

(TDS 400A Only) (Query Only)

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

Group: Cursor

Syntax: CURSor:VBArs:UNITSTring?

Example: `CURSOR:VBARS:UNITSTRING?`
might returns "`ms`", indicating the units are milliseconds.

CURVe

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

In extended-acquisition-length mode, the query form of this command can transfer acquisition record (as opposed to waveform) data from the oscilloscope in binary or ASCII format. Of course, to do this, you must specify a valid channel with the DATa:SOUrce command.

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

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

A description of the waveform transfer process starts in the section on the waveform command group.

Group: Waveform

Related Commands: DATa, WFMPre

Syntax 1: CURVe { <Block> | <asc curve> }

Syntax 2: CURVe?

Argument: <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 including checksum. If width is 1 then all bytes on the bus are single data points. If width is 2 then all bytes on the bus are 2-byte pairs. Use the DATa:WIDth command to set the width. <data> is the curve data. <newline> is a single byte newline character at the end of the data. See the GETWFM.C or GETWFM.BAS examples in the accompanying disk for more specifics.

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

Example: `CURVE?`
might return, for ASCII data:
```
CURVE 0,0,0,0,-1,1,0,-1,0,0,-1,0,0,-1,0,-1, -1,1,0,0,0,-1,0,0,-
1,0,1,1,0,-1,0,0,-1,0,0,-1,0,0
```

DATa

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

Group: Waveform

Related Commands: CURVE, WAVFrm

Syntax 1: DATa { INIT **|** SNAp **|** RECORDSNAp (TDS 500D & 700D) }

Syntax 2: DATa?

Argument 1: INIT initializes the waveform data parameters to their factory defaults.

Argument 2: SNAp sets DATa:STARt and DATa:STOP to match the current vertical bar cursor positions.

Argument 3: RECORDSNAp sets DATa:STARt and DATa:STOP for the first and last points of the selected channel's waveform/extended acquisition. When extended acquisition length (2M, 4M, and 8M) mode is on and the selected channel is a live channel, this is how to set up to get a waveform's data over GPIB. It also works when extended acquisition length mode is off.

Example 1: `DATA SNAP`
assigns the current position of the vertical bar cursors to DATA:START and DATA:STOP.

Example 2: `DATA?`
might return the string
```
:DATA:ENCDG RPBINARY;DESTINATION REF4; SOURCE REF4;START 1;STOP
500;WIDTH 2
```

DATa:DESTination

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

Group: Waveform

Syntax 1: DATa:DESTination REF<x>

Syntax 2: DATa:DESTination?

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

This means that you cannot have a channel as a destination. This is true even for extended-acquisition-length mode.

Example 1: `DATA DESTINATION REF3`
stores incoming waveform data in reference memory 3.

Example 2: `DATA:DESTINATION?`
might return `REF2` as the reference memory location that is currently selected.

DATa:ENCdg

Description: Sets or queries the format of the waveform data. This command is equivalent to setting WFMPre:ENCdg, WFMPre:BN_Fmt, and WFMPre:BYT_Or as shown in Table 2-27. Setting the DATa:ENCdg value causes the corresponding WFMPre values to be updated and vice versa.

Group: Waveform

Related Commands: WFMPre:ENCdg, WFMPre:BN.FMT, WFMPre:BYT_Or

Syntax 1: DATa:ENCdg { ASCIi **|** RIBinary **|** RPBinary **|** SRIbinary **|** SRPbinary}

Syntax 2: DATa:ENCdg?

Argument 1: ASCIi specifies the ASCII representation of signed integer (RIBinary) data. If this is the value at power-on, the WFMPre values for BN_Fmt, BYT_Or, and ENCdg are set as RP, MSB, and ASC respectively.

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

The range is -128 to 127 when DATa:WIDth is 1. Center screen is zero. The range is -32768 to 32767 when DATa:WIDth is 2. The upper limit is one division above the top of the screen and the lower limit is one division below the bottom of the screen.

Argument 3: RPBinary specifies positive integer data-point representation with the most significant byte transferred first.

The range is 0 to 255 when DATa:WIDth is 1. Center screen is 127. The range is 0 to 65,535 when DATa:WIDth is 2. The upper limit is one division above the top of the screen and the lower limit is one division below the bottom of the screen.

Argument 4: SRIbinary is the same as RIBinary except that the byte order is swapped, meaning that the least significant byte is transferred first. This format is useful when transferring data to IBM compatible PCs.

Argument 5: SRPbinary is the same as RPBinary except that the byte order is swapped, meaning that the least significant byte is transferred first. This format is useful when transferring data to IBM compatible PCs.

Table 2-27**:** DATa and WFMPre Parameter Settings

	WFMPre Settings		
DATa:ENCdg Setting	:ENCdg	:BN_Fmt	:BYT_Or
ASCIi	ASC	N/A	N/A
RIBinary	BIN	RI	MSB
RPBinary	BIN	RP	MSB
SRIbinary	BIN	RI	LSB
SRPbinary	BIN	RP	LSB

Example 1: `DATA:ENCDG RPBINARY`
sets the data encoding format to be positive integer where the most significant byte is transferred first.

Example 2: `DATA:ENCDG?`
might return `SRPBINARY` for the format of the waveform data.

DATa:SOUrce

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

In extended-acquisition-length mode, the MATH1 through MATH3 sources can not be used. Also, in this mode, CH1 through CH4 refer to extended-acquisition-length data. In contrast, when not in extended-acquisition-length mode, CH1 through CH4 refer only to the waveform record.

Group: Waveform

Syntax 1: DATa:SOUrce <wfm>[<Comma><wfm>]...

Syntax 2: DATa:SOUrce?

Argument: <wfm> is the location of the waveform data that will be transferred from the oscilloscope to the controller.

Example 1: `DATA:SOURCE REF2, CH2, MATH1, CH1`
specifies that four waveforms will be transferred in the next CURVE? query. The order that the data will be transferred is CH1, CH2, MATH1, and then REF2.

Example 2: `DATA:SOURCE?`
might return `REF3`, indicating the source for the waveform data that is transferred using a CURVE? query.

DATa:STARt

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

Group: Waveform

Related Commands: CURVe?, DATa:STOP

Syntax 1: DATa:STARt <NR1>

Syntax 2: DATa:STARt?

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

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

In extended-acquisition-length mode, <NR1> ranges up to the extended acquisition length as opposed to the record length.

Example 1: `DATA:START 10`
specifies that the waveform transfer will begin with data point 10.

Example 2: `DATA:START?`
might return `214` as the first waveform data point that will be transferred.

DATa:STOP

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

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

Group: Waveform

Related Commands: CURVe?

Syntax 1: DATa:STOP <NR1>

Syntax 2: DATa:STOP?

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

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

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

In extended-acquisition-length mode, <NR1> ranges up to the extended acquisition length as opposed to the record length.

Example 1: `DATA:STOP 15000`
specifies that the waveform transfer will stop at data point 15000.

Example 2: `DATA:STOP?`
might return `14900` as the last data point that will be transferred.

DATa:TARget

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

Group: Waveform

Related Command: CURVe

Syntax 1: DATa:TARget REF<x>

Syntax 2: DATa:TARget?

DATa:WIDth

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

Group: Waveform

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

Syntax 1: DATa:WIDth <NR1>

Syntax 2: DATa:WIDth?

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

<NR1> = 2 specifies that there are 2 bytes (16 bits) per point. This format is useful for AVErage and HIRes (two bytes per point) waveform. If used for ENVelope, PEAKdetect, or SAMple (one byte per point), the least significant byte is always zero.

Example: `DATA:WIDth 1`
sets the data width to 1 byte per data point for CURVe data.

DATE

Description: Sets or queries the date that the oscilloscope can display.

Group: Miscellaneous

Related Commands: DISplay, CLOCk, TIMe

Syntax 1: DATE <QString>

Syntax 2: DATE?

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

Example: `DATE "1993-01-24"`
specifies that the date is set to January 24th, 1993.

*DDT

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

Group: Miscellaneous

Related Commands: ALIAS:DEFINE, *TRG

Syntax 1: *DDT { <Block> **|** <QString> }

Syntax 2: *DDT?

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

Example: *DDT #0ACQUIRE:STATE RUN;BELL<EOI>
specifies that the acquisition system will be started and the bell rings each time a *TRG command is sent.

DELEte:SETUp

(No Query Form)

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

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

Group: Save and Recall

Related Commands: *RCL, RECAll:SETUp, *RST, *SAV, SAVe:SETUp, TEKSecure

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

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

Argument 2: ALL specifies all the stored setups.

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

DELEte:WAVEform

(No Query Form)

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

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

Group: Save and Recall

Related Commands: RECAll:WAVEform, SAVe:WAVEform, TEKSecure

Syntax: DELEte:WAVEform { REF<x> | ALL }

Argument 1: REF<x> (REF1, REF2, REF3 or REF) specifies one of the reference memory locations.

Argument 2: ALL specifies all the stored waveforms.

Example 1: `DELETE:WAVEFORM ALL`
removes all the waveforms stored in reference memory.

Example 2: `DELETE:WAVEFORM REF2`
removes the waveform stored at REF2.

DESE

Description: Sets and queries the bits in the Device Event Status Enable Register (DESER). The DESER is the mask that determines whether events are reported to the Standard Event Status Register (SESR), and entered into the Event Queue. For a more detailed discussion of the use of these registers, see the Registers section of this document.

Group: Status and Error

Related Commands: *CLS, *ESE, *ESR?, EVENT?, EVMsg?, *SRE, *STB?

Syntax 1: DESE <NR1>

Syntax 2: DESE?

Argument: <NR1>
is a value in the range from 0 to 255. The binary bits of the DESER are set according to this value. For example, DESE 209 sets the DESER to the binary value 11010001 (that is, the most significant bit in the register is set to 1, the next most significant bit to 1, the next bit to 0, etc.).

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

NOTE: Setting the DESER and the ESER to the same value allows only those codes to be entered into the Event Queue and summarized on the ESB bit (bit 5) of the Status Byte Register. Use the *ESE command to set the ESER. A discussion of event handling begins on page .

Example 1: `DESE 209`
sets the DESER to binary 11010001, which enables the PON, URQ, EXE, and OPC bits.

Example 2: `DESE?`
might return the string :DESE 186, showing that the DESER contains the binary value 10111010.

DIAg:RESUlt:FLAg?

(Query Only)

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

Group: Calibration and Diagnostic

Syntax: DIAg:RESUlt:FLAg?

Argument 1: PASS indicating that all of the selected diagnostic tests have passed.

Argument 2: FAIl indicating that at least one of the selected diagnostic tests has failed.

Example: `DIAG:RESULT:FLAG?`
returns either PASS or `FAIl`.

DIAg:RESUlt:LOG?

(Query Only)

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

Group: Calibration and Diagnostic

Syntax: DIAg:RESUlt:LOG?

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

Example: `DIAG:RESULT:LOG?`
might return
`:DIAG:RESULT:LOG "pass--Processor,pass--Display,pass--FP/Proc Interface,FAIL--Front Panel"`

DIAg:SELect:ACQUISition

(No Query Form)

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

Group: Calibration and Diagnostic

Syntax: DIAg:SELect:ACQUISition ALL

Argument: ALL selects functional, memory, and register tests.

DIAg:SELect:ALL

(No Query Form)

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

Group: Calibration and Diagnostic

Syntax: DIAg:SELect:ALL ALL

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

DIAg:SELect:CPU

(No Query Form)

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

Group: Calibration and Diagnostic

Syntax: DIAg:SELect:CPU ALL

Argument: ALL selects functional, memory, and register tests.

DIAg:SELect:DISplay

(No Query Form)

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

Group: Calibration and Diagnostic

Syntax: DIAg:SELect:DISplay ALL

Argument: ALL selects functional, memory, and register tests.

DIAg:SELect:FPAnel

(No Query Form)

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

Group: Calibration and Diagnostic

Syntax: DIAg:SELect:FPAnel ALL

Argument: ALL selects self diagnostic tests.

DIAg:STATE

(No Query Form)

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

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

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

Group: Calibration and Diagnostic

Syntax: DIAg:STATE EXECute

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

The DIAg:STATE EXECute command clears the following locations:

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

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

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

Example: `DIAg:STATE EXECUTE`
executes all the diagnostic tests that have been selected.

DISplay?

(Query Only)

Description: Returns the current display settings.

Group: Display

Syntax: DISplay?

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

DISplay:CLOCk

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

Group: Display

Syntax 1: DISplay:CLOCk { OFF | ON | <NR1> }

Syntax 2: DISplay:CLOCk?

Argument 1: <OFF> or <NR1> = 0 removes the clock from the display.

Argument 2: <ON> or <NR1> NOT = 0 displays the clock on the display.

Example 1: `DISPLAY:CLOCK ON`
sets the display to show time and date.

Example 2: `DISPLAY:CLOCK?`
might return `1`
indicating that the display shows time and date.

DISplay:COLOr:CONTRast

(TDS 6X4B & 700D Only)

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

Group: Display

Syntax 1: DISplay:COLOr:CONTRast { OFF **|** ON **|** <NR1> }

Syntax 2: DISplay:COLOr:CONTRast?

Argument 1: <OFF> or <NR1> = 0 turns off collision contrast.

Argument 2: <ON> or <NR1> NOT = 0 turns on collision contrast.

Example: `DISPLAY:COLOR:CONTRAST ON`
turns on the contrast option.

DISplay:COLOr:MAP:<item name>:BYCONTents

(TDS 6X4B & 700D Only)

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

Group: Display

Syntax 1: DISplay:COLOr:MAP:{ MATH1 **|** MATH2 **|** MATH3 **|** REF1 **|** REF2 **|** REF3 **|** REF4 }:BYCONTents { OFF **|** ON **|** <NR1> }

Syntax 2: DISplay:COLOr:MAP:{ MATH1 **|** MATH2 **|** MATH3 **|** REF1 **|** REF2 **|** REF3 **|** REF4 }:BYCONTents?

Argument 1: <OFF> or <NR1> = 0 means set the color by index (NOT by contents).

Argument 2: <ON> or <NR1> NOT = 0 means set the color by contents (NOT by index).

Example 1: `DISPLAY:COLOR:MAP:MATH1:BYCONTENTS ON`
sets the display of math waveform # 1 to the color of the waveform used to define math waveform # 1. If two waveforms were used to defined math waveform # 1, then the color of the first mentioned waveform is used.

Example 2: `DISPLAY:COLOR:MAP:MATH1:BYCONTENTS?`
might return ON.

DISplay:COLOr:MAP:<item name>:TO

(TDS 6X4B & 700D Only)

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

Group: Display

Syntax 1: DISplay:COLOr:MAP:{ MATH1 | MATH2 | MATH3 | REF1 | REF2 | REF3 | REF4 }:TO { CH1 | CH2 | CH3 | CH4 | MATH | REF }

Syntax 2: DISplay:COLOr:MAP:TO?

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

Example 2: `DISPLAY:COLOR:MAP:MATH1:TO?`
might return CH3.

DISplay:COLOr:PALEtte:PERSistence

(TDS 6X4B & 700D Only)

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

Group: Display

Syntax 1: DISplay:COLOr:PALEtte:PERSistence { TEMPErature **|** SPECTral **|** GRAYscale }

Syntax 2: DISplay:COLOr:PALEtte:PERSistence?

Example: `DISPLAY:COLOR:PALETTE:PERSISTENCE TEMPERATURE`
sets temperature palette as the current persistence palette.

DISplay:COLOr:PALEtte:<persistPalette>:P<x>

(TDS 500D & 700D Only)

Description: Sets the hue, lightness, and saturation values for persistence palette color indexes.

Group: Display

Syntax: 1: DISplay:COLOr:PALEtte:{ GRAYscale | SPECTral | TEMPErature }:P<x> { <NR1>,<NR1>,<NR1> | RESET}

Syntax 2: DISplay:COLOr:PALEtte: :{ GRAYscale | SPECTral | TEMPErature }:P<x> { <NR1>,<NR1>,<NR1> | RESET}?

Argument 1: BACKGround specifies the display background color.

Argument 2: CH1 specifies the channel 1 waveform and associated text color.

Argument 3: CH2 specifies the channel 2 waveform and associated text color.

Argument 4: CH3 specifies the channel 3 waveform and associated text color.

Argument 5: CH4 specifies the channel 4 waveform and associated text color.

Argument 6: MATH specifies the default math waveforms and associated text color.

Argument 7: REF specifies the default reference waveforms and associated text color.

Argument 8: TEXt specifies the text color in all menus and all readouts not associated with a waveform.

Argument 9: SCROLLTEXT specifies text in the scroll bar. For example, this covers the scroll bars used in the file system.

Argument 10: ZONe specifies the color of the intensified zone on the waveform when the time base is intensified and the color of the real samples when the display style is **Intensified Samples**.

Argument 11: COLLision specifies the color of the areas where waveforms overlap each other. You can turn collision marking on or off with the Items, **Waveforms, Options, Collision Contrast** menu item.

Argument 12: GRAticule specifies the color of the graticule. Use the **Settings, Display, Graticule** option to select the graticule type.

Argument 13: SCROLLBAR specifies the color of the scrollbar. For example, you will find a scrollbar in various file system menus.

Argument 14: HISTOMASK (TDS 700D) specifies the color of the histogram's box, the histogram itself, and masks, in the different palettes.

Argument 15: <NR1>, <NR1>, <NR1> specifies the desired colors in terms of hue, lightness and saturation values.

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

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

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

Example: `DISPLAY:COLOr:PALEtte:HARDCOPY:CH1 120,50,100` sets the CH1 color of the hardcopy palette to 120 hue, 50 lightness, and 100 saturation. This provides a pure red.

DISplay:COLOr:PALEtte:REGular

(TDS 6X4B & 700D Only)

Description: Sets the current palette to one of the preset palettes. Sending this command is equivalent to selecting **Palette** from the main Color menu and **Normal**, **Bold**, **Hardcopy**, or **Monochrome** from the resulting side menu.

Group: Display

Syntax 1: DISplay:COLOr:PALEtte:REGular { NORMal **|** BOLd **|** HARDCopy **|** MONo }

Syntax 2: DISplay:COLOr:PALEtte:REGular?

Example: `DISPLAY:COLOR:PALETTE:REGULAR HARDCOPY`
sets the current palette to the hardcopy palette.

DISplay:COLOr:PALEtte:RESETALL

(TDS 6X4B & 700D Only) (No Query Form)

Description: Restores all palettes to their factory default setting. Sending this command is equivalent to selecting **Restore Colors** from the main Color menu and **Reset All Palettes to Factory** from the resulting side menu.

Group: Display

Syntax: DISplay:COLOr:PALEtte:RESETALL

Example: `DISPLAY:COLOR:PALETTE:RESETALL`
resets the palette to the factory default setting.

DISplay:COLOr:PALEtte:<palette name>:RESET

(TDS 6X4B & 700D Only) (No Query Form)

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

Group: Display

Syntax: DISplay:COLOr:PALEtte:{ NORMal **|** MONo **|** BOLd **|** HARDCopy }:RESET

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

DISplay:COLOr:PALEtte:<palette name>:<item name>

(TDS 6X4B & 700D Only)

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

Group: Display

Syntax: 1: DISplay:COLOr:PALEtte:{ NORMal | MONo | BOLd | HARDCopy }:{ BACKGround | CH1 | CH2 | CH3 | CH4 | MATH | REF | TEXT | SCROLLTEXT | ZONe | COLLision | GRAticule | SCROLLBAR | HISTOMASK (TDS 700D) } { <NR1>,<NR1>,<NR1> }

Syntax 2: DISplay:COLOr:PALEtte:{ NORMal | MONo | BOLd | HARDCopy }:{ BACKGround | CH1 | CH2 | CH3 | CH4 | MATH | REF | TEXT | SCROLLTEXT | ZONe | COLLision | GRAticule | SCROLLBAR | HISTOMASK (TDS 700D) }?

Argument 1: BACKGround specifies the display background color.

Argument 2: CH1 specifies the channel 1 waveform and associated text color.

Argument 3: CH2 specifies the channel 2 waveform and associated text color.

Argument 4: CH3 specifies the channel 3 waveform and associated text color.

Argument 5: CH4 specifies the channel 4 waveform and associated text color.

Argument 6: MATH specifies the default math waveforms and associated text color.

Argument 7: REF specifies the default reference waveforms and associated text color.

Argument 8: TEXt specifies the text color in all menus and all readouts not associated with a waveform.

Argument 9: SCROLLTEXT specifies text in the scroll bar. For example, this covers the scroll bars used in the file system.

Argument 10: ZONe specifies the color of the intensified zone on the waveform when the time base is intensified and the color of the real samples when the display style is **Intensified Samples**.

Argument 11: COLLision specifies the color of the areas where waveforms overlap each other. You can turn collision marking on or off with the Items, **Waveforms, Options, Collision Contrast** menu item.

Argument 12: GRAticule specifies the color of the graticule. Use the **Settings, Display, Graticule** option to select the graticule type.

Argument 13: SCROLLBAR specifies the color of the scrollbar. For example, you will find a scrollbar in various file system menus.

Argument 14: HISTOMASK (TDS 700D) specifies the color of the histogram's box, the histogram itself, and masks, in the different palettes.

Argument 15: <NR1>, <NR1>, <NR1> specifies the desired colors in terms of hue, lightness and saturation values.

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

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

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

Example: `DISPLAY:COLOr:PALEtte:HARDCOPY:CH1 120,50,100` sets the CH1 color of the hardcopy palette to 120 hue, 50 lightness, and 100 saturation. This provides a pure red.

DISplay:FILTer

Description: Sets or queries the type of interpolation to use for the display. Sending this command is equivalent to setting **Filter** in the Display menu.

Group: Display

Related Commands: DISplay:STYle

Syntax 1: DISplay:FILTer { LINEAr **|** SINX }

Syntax 2: DISplay:FILTer?

Argument 1: LINEAr specifies linear interpolation where acquired points are connected with straight lines.

Argument 2: SINX specifies sin(x)/x interpolation where acquired points are fit to a curve.

Example 1: `DISPLAY:FILTER LINEAR`
sets the interpolation filter type to linear.

Example 2: `DISPLAY:FILTER?`
returns either `LINEAR` or `SINX`, indicating the type of interpolation filter.

DISplay:FORMat

Description: Sets or queries the display format. Sending this command is equivalent to setting **Format** in the Display menu.

Group: Display

Syntax 1: DISplay:FORMat { XY **|** YT **|** XYZ }

Syntax 2: DISplay:FORMat?

Argument 1: XY displays the voltage of one waveform against the voltage of another. The sources that make up an XY waveform are predefined and are listed in Table 2-28. Displaying one source causes its corresponding source to be displayed.

Table 2-28: XY Format Pairs

X-Axis Source	Y-Axis Source
Ch 1	Ch 2
Ch 3 (or AUX 1) (All models except TDS 430A)	Ch 4 (or AUX 2) (All models except TDS 430A)
Ref 1	Ref 2
Ref 3	Ref 4

Argument 2: YT sets the display to a voltage versus time format and is the normal mode.

Argument 3: XYZ format (DPO mode only) compares the voltage levels of the CH 1 (X) and CH 2 (Y) waveform records point-by-point as in XY format. The displayed waveform intensity is modulated by the CH 3 (Z) waveform record. XYZ format is not triggered. XYZ format is not available on 2+2 channel oscilloscopes. If the requested record length in normal mode is large enough so that the number of channels that are on is limited, then XYZ (and XY) will not work even in a four channel oscilloscope. A -5 division signal (including position and offset) on CH 3 produces a blank screen; a +5 division signal produces full intensity.

Example 1: `DISPLAY:FORMAT YT`
selects a voltage versus time format for the display.

Example 2: `DISPLAY:FORMAT?`
might return XY for the display format.

DISplay:GRAticule

Description: Selects the type of graticule that will be displayed. Sending this command is equivalent to setting **Graticule** in the Display menu.

Group: Display

Syntax 1: DISplay:GRAticule { CROSSHair | FRAme | FULl | GRId | NTSC | PAL}

Syntax 2: DISplay:GRAticule?

Argument 1: CROSSHair specifies a frame and cross hairs.

Argument 2: FRAme specifies just a frame.

Argument 3: FULl specifies a frame, a grid, and cross hairs.

Argument 4: GRId specifies a frame and a grid.

Argument 5: NTSC specifies a special NTSC frame.

Argument 6: PAL specifies a special PAL frame.

Example 1: `DISPLAY:GRATICULE GRID`
sets the graticule type to display a frame and a grid.

Example 2: `DISPLAY:GRATICULE?`
returns `FULL` when all graticule elements (grid, frame, and cross hairs) are selected.

DISplay:INStavu:ACCUmulate

(TDS 500D & 700D Only)

Description: Selects the length of the counters in the waveform array that record the number of times acquisition signals occur at each display pixel. DPO, mask-counting, and histogram modes use the counters. In DPO mode, the accumulated numbers and the intensity controls determine the gray-scale information for each displayed pixel.

Group: Display

Syntax 1: DISplay:INStavu:ACCUmulate { SHALlow **|** DEEP }

Syntax 2: DISplay:INStavu:ACCUmulate?

Argument 1: SHALlow sets the accumulation depth to 21 bits or, if histograms or mask counting is active, 32 bits. Shallow allows the image to saturate faster.

Argument 2: DEEP sets the accumulation depth to 64 bits. Accumulation depths greater than 21 bits reduce the screen update rate, disable variable persistence and no persistence, and enable SAve/Recall of the image array.

Example: `DISPLAY:INSTAVU:ACCUmulate?`
might return `DEEP` indicating the accumulation depth is 64 bits.

DISplay:INStavu:AUTOBright

(TDS 500D & 700D Only)

Description: Automatically sets the display brightness in DPO mode.

Group: Display

Syntax 1: DISplay:INStavu:AUTOBright { On **|** Off **|** <NR1> }

Syntax 2: DISplay:INStavu:AUTOBright?

Argument 1: <ON> or <NR1> NOT = 0 automatically sets the maximum display brightness to the value of the most frequent event. Accumulation depths greater than 21 bits always use the ON mode.

Argument 2: <OFF> or <NR1> = 0 lets the display brightness depend on the trigger rate.

Example: `DISPLAY:INSTAVU:AUTOBRIGHT?`
might return `1` indicating the DPO autobright mode is on.

DISplay:INStavu:BRIGHTness

(TDS 500D & 700D Only)

Description: Sets the DPO mode brightness.

Group: Display

Syntax 1: DISplay:INStavu:BRIGHTness <NR3>

Syntax 2: DISplay:INStavu:BRIGHTness?

Argument: <NR3> specifies the brightness of DPO mode waveforms that are displayed on the screen. The range is from 0 to 100. Higher brightness settings result in less frequent events being more easily seen. This command has no effect on the display until/unless DISplay:INStavu:MAP INTENSITy is selected.

Example: `DISPLAY:INSTAVU:BRIGHTNESS?`
might return `:DISPLAY:INSTAVU:BRIGHTNESS 40.0E+0.`

DISplay:INStavu:CONTRast

(TDS 500D & 700D Only)

Description: Sets the DPO mode contrast.

Group: Display

Syntax 1: DISplay:INStavu:CONTRast <NR3>

Argument: <NR3> specifies the contrast of DPO mode waveforms that are displayed on the screen. The range is from 0 to 100. Low contrast settings makes less frequent events display at the same brightness as frequent events. This command has no effect on the display until/unless DISplay:INStavu:MAP INTENSITy is selected.

Example: `DISPLAY:INSTAVU:CONTRAST?`
might return `:DISPLAY:INSTAVU:CONTRAST 10.0E+0.`

DISplay:INStavu:MAP

(TDS 500D & 700D Only)

Description: Selects the method used to map the acquisition database to display indexes.

Group: Display

Syntax 1: DISplay:INStavu:MAP { PERCent | INTENSITy }

Syntax 2: DISplay:INStavu:MAP?

Argument 1: PERCent allows you to set Color Mapping percentages using the DISplay:INStavu:PERCent command.

Argument 2: INTENSITy allows you to set the DPO mode display intensity using the DPO Brightness and Contrast controls.

Example: `DISPLAY:INSTAVU:MAP?`
might return `DISPLAY:INSTAVU:MAP:PERCENT` indicating the DPO mode display intensity can be set using the DISplay:INStavu:PERCent command.

DISplay:INStavu:PERCent:P<x>

(TDS 500D & 700D Only)

Description: Selects the maximum value for a color index when you are in the PERCENT Color Mapping mode.

Group: Display

Syntax 1: DISplay:INStavu:PERCent:P<x> { <NR3> | RESET }

Syntax 2: DISplay:INStavu:PERCent:P<x>?

Argument 1: <NR3> sets the maximum percentage for a color index when the PERCENT Color Mapping modes is selected.

Argument 2: RESET sets the maximum percentage for a color index to the factory default.

Example: `DISPLAY:INSTAVU:PERCent:P5?`
might return the value `26.667E+0,` indicating the maximum percentage for color index 5 is 27.

DISplay:INStavu:PERSistence

(TDS 500D & 700D Only)

Description: Selects the persistence mode to use with DPO mode.

Group: Display

Syntax 1: DISplay:INStavu:PERSistence { INFPersist | VARpersist | NOPersist }

Syntax 2: DISplay:INStavu:PERSistence?

Argument 1: INFPersist accumulates data points on the display indefinitely.

Argument 2: VARpersist leaves acquired data points on the display for a period of time specified by DISplay:INStavu:VARpersist.

Argument 3: NOPersist displays only the most recent acquisition, erasing the display each time a new waveform array is displayed.

Example: `DISPLAY:INSTAVU:PERSISTENCE?`
might return `INFPERSIST` indicating the infinite persistence mode is on.

DISplay:INStavu:STYle

(TDS 500D & 700D Only)

Description: Selects how the data is displayed with DPO mode.

Group: Display

Syntax 1: DISplay:INStavu:STYle { DOTs | VECtors }

Syntax 2: DISplay:INStavu:STYle?

Argument 1: DOTs displays individual data points.

Argument 2: VECtors connects adjacent data points. Old points are immediately replaced by new ones.

Example: `DISPLAY:INSTAVU STYLE?`
might return `DOTS` indicating that the display shows individual waveform data points.

DISplay:INStavu:VARpersist

(TDS 500D & 700D Only)

Description: Sets the length of time that data points are displayed when DISplay:INStavu:PERSistence is set to VARpersist. This affects the display only.

Group: Display

Related Command:
DISplay:INStavu:PERSistence

Syntax 1: DISplay:INStavu:VARpersist <NR3>

Syntax 2: DISplay:INStavu:VARpersist?

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

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

DISplay:INTENSITy?

(Query Only)

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

Group: Display

Syntax: DISplay:INTENSITy?

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

DISplay:INTENSITy:CONTRast

(TDS 400A, 510A, 500D, & 6X0B Only)

Description: Sets the intensity of the intensified zone on a waveform. Sending this command is equivalent to setting **Contrast** in the Display Intensity side menu.

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

Group: Display

Related Command: HORizontal:MODe

Syntax 1: DISplay:INTENSITy:CONTRast <NR1>

Syntax 2: DISplay:INTENSITy:CONTRast?

Argument 1: <NR1> ranges from 100 to 250 percent.

Example: `DISPLAY:INTENSITY:CONTRAST 140`
sets the intensity of the intensified portion of a waveform.

DISplay:INTENSITy:OVERALL

(TDS 400A, 510A, 500D, & 6X0B Only)

Description: Sets the intensity of the entire display. Sending this command is equivalent to setting **Overall** in the Display Intensity side menu.

Group: Display

Syntax 1: DISplay:INTENSITy:OVERALL <NR1>

Syntax 2: DISplay:INTENSITy:OVERALL?

Argument: <NR1> ranges from 20 to 100 percent.

Example 1: `DISplay:INTENSITY:OVERALL 50`
sets the intensity of the display to the middle of the range.

Example 2: `DISplay:INTENSITY:OVERALL?`
might return `75` as the overall display intensity.

DISplay:INTENSITy:TEXt

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

Group: Display

Syntax 1: DISplay:INTENSITy:TEXt <NR1>

Syntax 2: DISplay:INTENSITy:TEXt?

Argument: <NR1> ranges from 20 to 100 percent.

Example: `DISPLAY:INTENSITY:TEXT 100`
sets the intensity of the text to the brightest level.

DISplay:INTENSITy:WAVEform

Description: Sets the intensity of the waveforms. Sending this command is equivalent to setting **Waveform** in the Display Intensity side menu.

Group: Display

Syntax 1: DISplay:INTENSITy:WAVEform <NR1>

Syntax 2: DISplay:INTENSITy:WAVEform?

Argument: <NR1> ranges from 20 to 100 percent.

Example: `DISPLAY:INTENSITY:WAVEFORM?`
might return `60` as the intensity of the waveform.

DISplay:MODe

(TDS 500D & 700D Only)

Description: Selects whether or not to turn on DPO mode.

Group: Display

Syntax 1: DISplay:MODe { INStavu **|** NORMal }

Syntax 2: DISplay:MODe?

Argument 1: INSTavu turns on DPO mode. This mode can help you view infrequent deviations in a signal. It only uses a 500 point record length, no averaging, and no enveloping.

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

Argument 2: NORMal turns on the normal (non-DPO) mode.

Example 1: `DISPLAY:MODE INSTAVU`
turns on InstaVu.

Example 2: `DISPLAY:MODE?`
might return `INSTAVU` indicating that the DPO mode is on.

DISplay:PERSistence

Description: Sets the length of time that data points are displayed when DISplay:STYle is set to VARpersist. This affects the display only and is equivalent to setting **Variable Persistence** in the Display Style side menu.

Group: Display

Related Command: DISplay:STYle

Syntax 1: DISplay:PERSistence <NR3>

Syntax 2: DISplay:PERSistence?

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

Example: `DISPLAY:PERSISTENCE 3`
specifies that the waveform points are displayed on the screen for 3 seconds before they fade.

DISplay:STYle

Description: Selects how the data is displayed. Sending this command is equivalent to setting **Style** in the Display menu.

Group: Display

Related Command: DISplay:PERSistence

Syntax 1: DISplay:STYle { DOTs **|** INFPersist **|** INTENSIFied **|** VARpersist **|** VECtors }

Syntax 2: DISplay:STYle?

Argument 1: DOTs displays individual data points.

Argument 2: INFPersist accumulates data points on the display indefinitely. The display is reset when the style or acquisition is reset.

Argument 3: VARpersist leaves acquired data points on the display for a period of time specified by DISplay:PERSistence.

Argument 4: VECtors connects adjacent data points. Old points are immediately replaced by new ones.

Argument 5: INTENSIFied causes the display to show acquired (non-interpolated) samples with brighter or different colored dots than the rest of the waveform.

Example 1: `DISPLAY:STYLE INFPERSIST`
sets the display to indefinitely accumulate data points on the screen.

Example 2: `DISPLAY:STYLE?`
might return `DOTS` indicating that the display shows individual waveform data points.

DISplay:TRIGBar

Description: Controls the display of the trigger bar indicator/s. The bar indicates where the trigger will occur, in voltage.

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

Group: Display

Syntax 1: DISplay:TRIGBar { OFF **|** SHORT **|** LONG }

Syntax 2: DISplay:TRIGBar?

Argument 1: OFF removes the trigger bar indicator from the display.

Argument 2: SHORT displays a short arrow at the right side of the graticule for each displayed trigger signal.

Argument 3: LONG displays a horizontal line in the center of the graticule for each displayed trigger signal.

Example: `DISPLAY:TRIGBAR LONG`
sets the display to show long trigger bar indicator (or indicators).

DISplay:TRIGT

Description: Controls the display of the trigger indicator. Sending this command is equivalent to setting the **Display `T' @ Trigger Point** in the Readout Options side menu. The query form returns an ON (1) or an OFF (0).

Group: Display

Syntax 1: DISplay:TRIGT { ON | OFF | <NR1> }

Syntax 2: DISplay:TRIGT?

Argument 1: <OFF> or <NR1> = 0 removes the trigger indicator from the display.

Argument 2: <ON> or <NR1> NOT = 0 displays a trigger indicator on each of the displayed waveforms. The trigger indicator is in reverse video for the selected waveform.

Example 1: `DISPLAY:TRIGT ON`
sets the display to show trigger indicators.

Example 2: `DISPLAY:TRIGT?`
might return 1 indicating that the display shows trigger indicators.

*ESE

Description: Sets and queries the bits in the Event Status Enable Register (ESER). The ESER prevents events from being reported to the Status Byte Register (STB). For a more detailed discussion of the use of these registers, see the section on Registers.

Group: Status and Error

Related Commands: *CLS, DESE, *ESR?, EVENT?, EVMsg?, *SRE, *STB?

Syntax 1: *ESE <NR1>

Syntax 2: *ESE?

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

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

NOTE: Setting the DESER and the ESER to the same value allows only those codes to be entered into the Event Queue and summarized on the ESB bit (bit 5) of the Status Byte Register. Use the DESE command to set the DESER. A discussion of event handling begins on page .

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

Example 2: `*ESE?`
might return the string `*ESE 186`, showing that the ESER contains the binary value 10111010.

*ESR?

(Query Only)

Description: Returns the contents of the Standard Event Status Register (SESR). *ESR? also clears the SESR (since reading the SESR clears it). For a more detailed discussion of the use of these registers, see the section on Registers.

Group: Status and Error

Related Commands: ALLEv?, *CLS, DESE, *ESE, EVENT?, EVMsg?, *SRE, *STB?

Syntax: *ESR?

Example: `*ESR?`
might return the value 213, showing that the SESR contains binary 11010101.

EVENT?

(Query Only)

Description: 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. For more information, see the section on Event Handling.

Group: Status and Error

Related Commands:
ALLev?, *CLS, DESE, *ESE, *ESR?, EVMsg?, *SRE, *STB?

Syntax: EVENT?

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

EVMsg?

(Query Only)

Description: Removes from the Event Queue a single event code associated with the results of the last *ESR? read, and returns the event code along with an explanatory message. For more information, see the section on Event Handling.

Group: Status and Error

Related Commands:
ALLEv?, *CLS, DESE, *ESE, *ESR?, EVENT?, *SRE, *STB

Syntax: EVMsg?

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

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

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

where <Command> is the command that caused the error and may be returned when a command error is detected by the oscilloscope. As much of the command will be returned as possible without exceeding the 60 character limit of the <Message> and <Command> strings combined. The command string is right-justified.

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

EVQty?

(Query Only)

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

Group: Status and Error

Related Commands: ALLEv?, EVENT?, EVMsg?

Syntax: EVQty?

Example: `EVQTY?`
might return 3 as the number of event codes in the Event Queue.

FACtory

(No Query Form)

Description: Resets the oscilloscope to its factory default settings. Sending this command is equivalent to selecting **Recall Factory Setup** in the Waveform Save/Recall menu.

Group: Miscellaneous

Related Commands: *PSC, *RCL, RECAll:SETUp, *RST, *SAV, SAVe:SETUp

Syntax: FACtory

Setting the oscilloscope to factory default includes:

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

The FACtory command does not alter the following:

- The state of the GPIB (IEEE Std 488.1-1987) interface.
- The selected GPIB address.
- Calibration data that affects device specifications.
- Protected user data.
- Stored settings.
- The current password (if implemented).

FILESystem:COPy

(No Query Form) (File System Only)

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

Group: File system

Syntax: FILESystem:Copy { <source file path>,<destination file path> **|** <source directory path>,<destination file path> **|** <source directory path>,<destination directory path> }

Argument 1: <file path> is a quoted string that defines the file name and path. Input the file path using the form <drive>/<dir>/<filename>. <drive> and one or more <dir>s are optional. If you do not specify them, the TDS will copy the file in the current directory. <filename> stands for a filename of up to 8 characters and can be followed by a period (".") and a 3-character extension. You can also use the inclusive filename *.* in the source file path to copy all files.

Argument 2: <directory path> is a quoted string that defines the directory. Input the directory using the form <drive>/<dir>/<directory name>. <drive> and one or more <dir>s are optional. If you do not specify them, the TDS will copy the directory in the current directory. <directory name> stands for a directory name of up to 8 characters and can be followed by a period (".") and a 3-character extension.

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

Example 1: `FILESYSTEM:COPY "TEK00001.SET","fd0:/TEK00001.SET"`
copies the file named TEK00001.SET on the current drive to a file named TEK00001.SET on the drive fd0: in the root directory.

Example 2: `FILESYSTEM:COPY "fd0:/YOURDIR/TEK00001.SET","fd0:/MYDIR"`
copies the file named TEK00001.SET on the fd0: drive and the YOURDIR directory to the MYDIR directory on the same drive.

Example 3: `FILESYSTEM:COPY "YOURDIR","fd0:/MYDIR"`
copies the files in the YOURDIR directory in the current directory to the MYDIR directory on the fd0: drive.

FILESystem:CWD

(File System Only)

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

Group: File system

Syntax 1: FILESystem:CWD <directory path>

Syntax 2: FILESystem:CWD?

Example 1: `FILESYSTEM:CWD "fd0:/MYDIR"`
will define fd0:/MYDIR as the current directory.

Example 2: `FILESYSTEM:CWD?`
might return fd0:/MYDIR if that is the current directory.

FILESystem:DELEte

(No Query Form) File System Only)

Description: Deletes a named file.

Group: File system

Syntax: FILESystem:DELEte <file path>

Argument 1: <file path> is a quoted string that defines the file name and path. Input the file path using the form <drive>/<dir>/<filename>. <drive> and one or more <dir>s are optional. If you do not specify them, the TDS will delete the file in the current directory. <filename> stands for a filename of up to 8 characters and can be followed by a period (".") and a 3-character extension. You can also use the inclusive filename *.* to delete all files.

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

Example 1: `FILESYSTEM:DELETE "NOT-MINE.SET"`
deletes the file named NOT-MINE.SET on the default drive and directory.

Example 2: `FILESYSTEM:DELETE "*.*"`
deletes all the files in the default directory on the default drive.

FILESystem:DELWarn

(File System Only)

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

Group: File system

Syntax 1: FILESystem:DELWarn { ON | OFF | <NR1> }

Syntax 2: FILESystem:DELWarn?

Argument 1: ON or <NR1> NOT = 0 turns on the front-panel delete warning.

Argument 2: OFF or <NR1> = 0 turns off the front-panel delete warning.

Example 1: `FILESYSTEM:DELWARN OFF`
disables the front-panel delete warning.

Example 2: `FILESYSTEM:DELWARN?`
might return 0 indicating the front-panel warning is disabled.

FILESystem:DIR

(Query Only) (File System Only)

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

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

Group: File system

Syntax: FILESystem:DIR?

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

FILESystem:FORMat

(No Query Form) (File System Only)

Description: Formats a named drive.

Group: File system

Syntax: FILESystem:FORMat <drive name>

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

Example: `FILESYSTEM:FORMAT "fd0:"`
formats the media on drive fd0:.

FILESystem:FREESpace

(Query Only) (File System Only)

Description: Returns the amount of freespace (in bytes) on the current drive.

Group: File system

Syntax: FILESystem:FREESpace?

Example: `FILESYSTEM:FREESpace?`
might return `0` as the amount of freespace available if the drive was full.

FILESystem:MKDir

(No Query Form) (File System Only)

Description: Make a new directory.

Group: File system

Syntax: FILESystem:MKDir <directory path>

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

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

Example: `FILESYSTEM:MKDIR "NEATPICS"`
creates the directory named NEATPICS on the current drive.

FILESystem:OVERWrite

(File System Only)

Description: Turns on or off the file overwrite protection. Turning on file-overwrite protection prevents writing over existing files.

Group: File system

Syntax 1: FILESystem:OVERWrite { ON **|** OFF **|** <NR1> }

Syntax 2: FILESystem:OVERWrite?

Argument 1: ON or <NR1> NOT = 0 turns on the file-overwrite protection.

Argument 2: OFF or <NR1> = 0 turns off the file-overwrite protection.

Example 1: `FILESYSTEM:OVERWRITE OFF`
lets you overwrite existing files.

Example 2: `FILESYSTEM:OVERWRITE?`
might return `0` indicating you cannot overwrite existing files.

FILESystem:PRInt

(No Query Form) (File System Only)

Description: Prints a named file to the named port.

Group: File system

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

Argument 1: <file path> is a quoted string that defines the file name and path. Input the file path using the form <drive>/<dir>/<filename>. <drive> and one or more <dir>s are optional. If you do not specify them, the TDS will print the file in the current directory. <filename> stands for a filename of up to 8 characters and can be followed by a period (".") and a 3-char extension.

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

Argument 2: GPIb specifies that the hardcopy is sent out the GPIB port.

Argument 3: CENtronics specifies that the hardcopy is sent out the Centronics port (Option 13 RS232/Centronics Hardcopy Interface only).

Argument 4: RS232 specifies that the hardcopy is sent out the RS232 port (RS232/Centronics Hardcopy Interface only).

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

FILESystem:READFile

(TDS 500D & 700D Only)

Description: Copies a named file to the GPIB port.

Group: File system

Syntax: FILESystem:READFile <file path>

Argument 1: <file path> is a quoted string that defines the file to read. Input the file path using the form <drive>/<dir>/<filename>. <drive> and one or more <dir>s are optional. If you do not specify them, the TDS will look for the filename in the current directory.

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

Example: `FILESYSTEM:READFILE "hd0:TEK00000.IBM"`
sends a hard-disk-based file named TEK00000.IBM out the GPIB port.

FILESystem:REName

(No Query Form) (File System Only)

Description: Assigns a new name to a file.

Group: File system

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

Argument 1: <old file path> is a quoted string that defines the file to rename. Input the file path using the form <drive>/<dir>/<filename>. <drive> and one or more <dir>s are optional. If you do not specify them, the TDS will look for the filename in the current directory.

Argument 2: <new file path> is a quoted string that defines the new name of the file. Input the file path using the form <drive>/<dir>/<filename>. <drive> and one or more <dir>s are optional. If you do not specify them, the TDS will place the newly named file in the current directory.

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

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

FILESystem:RMDir

(No Query Form) (File System Only)

Description: Deletes a named directory.

Group: File system

Syntax: FILESystem:RMDir <directory path>

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

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

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

FILESystem:WRITEFile

(TDS 500D & 700D Only)

Description: Copies the GPIB port block data to a named file.

Group: File system

Syntax: FILESystem:WRITEFile <file path>,<Block>

Argument 1: <file path> is a quoted string that defines the file to read. Input the file path using the form <drive>/<dir>/<filename>. <drive> and one or more <dir>s are optional. If you do not specify them, the TDS will look for the filename in the current directory.

<Block> is a block of data bytes that uses the form shown below:

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

<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.

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

Example 1: `FILESYSTEM:WRITEFILE "hd0:/hello1.txt”,#21hello world"`
writes the text string “hello world” to a file named HELLO1.TXT on the oscilloscope’s hard disk. It uses a format that defines the number of characters to transfer.

Example 2: `FILESYSTEM:WRITEFILE "hd0:/HELLO2.TXT”,#0hello world`
writes the text string “hello world” to a file named HELLO2.TXT on the oscilloscope’s hard disk. It uses a format that does not define the number of characters to transfer.

HARDCopy

Description: Sends a copy of the screen display followed by and EOI to the port specified by HARDCopy:PORT. The format and layout of the output is specified with the HARDCopy:FORMat and HARDCopy:LAYout commands. Sending this command is equivalent to pressing the front-panel **HARDCOPY** button.

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

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

Group: Hardcopy

Syntax 1: HARDCopy { ABOrt | CLEARSpool | STARt }

Syntax 2: HARDCopy?

Argument 1: ABOrt terminates the hardcopy output in process.

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

Argument 2: CLEARSpool clears the printer output spooler.

Argument 3: STARt initiates a screen copy that is sent to the controller where it can be stored in a file or redirected to a printing device.

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

Example: `HARDCOPY ABORT`
stops any hardcopy output that is in process.

HARDCopy:FILEName

(File System Only)

Description: Selects the file to send the hardcopy data to on the next hardcopy command (HARDCOPY START). Sending this command is equivalent to setting the target file name in the Hardcopy menu.

Group: Hardcopy

Related Command: HARDCopy

Syntax 1: HARDCopy:FILEName <file path>

Syntax 2: HARDCopy:FILEName?

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

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

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

Example 1: `HARDCOPY:FILENAME "TEK.IBM"`
selects TEK.IBM as the selected file name.

Example 2: `HARDCOPY:FILENAME?`
might return `TEK.IBM` as the selected file name.

Example 3: `HARDCOPY:FILENAME "TEK??.IBM"`
selects TEK as the selected file name with a numeric, two-digit suffix. The TDS might return `TEK00.IBM` as the first file, `TEK01.IBM` as the second.

HARDCopy:FORMat

Description: Selects the output data format for hardcopies. Sending this command is equivalent to setting **Format** in the Hardcopy menu.

Group: Hardcopy

Syntax 1: HARDCopy:FORMat { BMP | BMPCOLOR | DESKJET | DESKJETC (not on TDS 400A & 510A) | DPU411 | DPU412 | EPSCOLImg (not on TDS 400A & 510A) | DESKJETC (not on TDS 400A & 510A) | EPSColor | EPSImage | EPSMono | EPSOn | HPGl | INTERLeaf | LASERJet | PCX | PCXCOLOR | RLE (not on TDS 400A & 510A) | THInkjet | TIFf }

Syntax 2: HARDCopy:FORMat?

On monochrome instruments, the following formats are mapped to a monochrome near equivalent:

PCXCOLOR -> PCX
BMPCOLOR -> BMP
RLE -> BMP
EPSCOLIMG -> EPSIMAGE
DESKJETC -> DESKJET
(The DESKJETC argument is not the TDS 400A and 510A)

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

Example 1: `HARDCOPY:FORMAT HPGL`
sets the hardcopy output format to HPGL.

Example 2: `HARDCOPY:FORMAT?`
might return `INTERLEAF` as the final hardcopy output format.

HARDCopy:LAYout

Description: Selects the printing orientation. Sending this command is equivalent to setting **Layout** in the Hardcopy menu.

Group: Hardcopy

Syntax 1: HARDCopy:LAYout { LANdscape | PORTRait }

Syntax 2: HARDCopy:LAYout?

Argument 1: LANDscape specifies that the bottom of the hardcopy is along the longest side of the page.

Argument 2: PORTRait specifies that the bottom of the hardcopy is along the short side of the page. This is the standard format.

Example: `HARDCOPY:LAYOUT?`
might return `PORTRAIT` as the hardcopy page-layout format.

HARDCopy:PALEtte

(TDS 644B, 684B, & 700D)

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

Group: Hardcopy

Related Command: HARDCopy

Syntax 1: HARDCopy:PALEtte { CURRent **|** HARDCopy }

Syntax 2: HARDCopy:PALEtte

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

HARDCopy:PORT

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

Group: Hardcopy

Related Command: HARDCopy, LIMit:HARDCopy

Syntax 1: HARDCopy:PORT { CENtronics| FILE (Option 1F File System only) | GPIb | RS232 }

Syntax 2: HARDCopy:PORT?

Argument 1: CENtronics specifies that the hardcopy is sent out the Centronics port (available with the RS232/Centronics Hardcopy Interface).

Argument 2: FILE specifies that the hardcopy is stored in the file named in the HARDCOPY:FILENAME command.

Argument 3: GPIb specifies that the hardcopy is sent out the GPIB port.

Argument 4: RS232 specifies that the hardcopy is sent out the RS232 port (Option 13 RS232/Centronics Hardcopy Interface only).

Example: `HARDCOPY:PORT?`
might return `GPIB` as the selected hardcopy output port.

HDR

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

Group: Miscellaneous

Syntax 1: HDR { <NR1> | OFF | ON }

Syntax 2: HDR?

HEADer

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

Group: Miscellaneous

Related Command: VERBose

Syntax 1: HEADer { <NR1> **|** OFF **|** ON }

Syntax 2: HEADer?

Argument 1: ON or <NR1> NOT = 0 sets the Response Header Enable State to true. This causes the oscilloscope to include headers on applicable query responses. You can then use the query response as a command.

Argument 2: OFF or <NR1> = 0 sets the Response Header Enable State to false. This causes the oscilloscope to omit headers on query responses, so that only the argument is returned.

Example 1: `HEADER OFF`
causes the oscilloscope to omit headers from query responses.

Example 2: `HEADER?`
might return the value 1, showing that the Response Header Enable State is true.

HIStogram?

(TDS 500D & 700D Only) (Query Only)

Description: Returns histogram parameters. In other words, it queries the state of all settable histogram parameters.

Group: Histogram

Syntax: HIStogram?

Example: `HISTOGRAM?`
might return
```
:HISTOGRAM:SOURCE CH1;MODE VERTICAL;DISPLAY LOG;BOXPCNT
30.0E+0,25.1E+0,70.0E+0,75.2E+0;SIZE 1.00E+0
```

HIStogram:BOX

(TDS 500D & 700D Only)

Description: Defines the left, top, right, and bottom positions of the histogram box, in source waveform coordinates. Any value outside the display is clipped to the display boundary and generates an error.

Group: Histogram

Syntax 1: HIStogram:BOX <NR3>, <NR3>, <NR3>, <NR3>

Syntax 2: HIStogram:BOX?

Argument 1: <NR3> is the left position of the histogram box, in source waveform coordinates.

Argument 2: <NR3> is the top position of the histogram box, in source waveform coordinates.

Argument 3: <NR3> is the right position of the histogram box, in source waveform coordinates.

Argument 4: <NR3> is the bottom position of the histogram box, in source waveform coordinates.

Example: `HISTOGRAM:BOX 1e-9, 0.250, 2e-9, 0.500`
would define the coordinates of the histogram box in source waveform coordinates

HIStogram:BOXPcnt

(TDS 500D & 700D Only)

Description: Defines the left, top, right, and bottom positions of the histogram box, in percentage coordinates. The upper left has the value 0, 0 and the lower right has the value 100, 100 when the horizontal trigger position is 50%. Any value outside the range of 0 to 100 is clipped and generates an error.

Group: Histogram

Syntax 1: HIStogram:BOXPcnt <NR3>, <NR3>, <NR3>, <NR3>

Syntax 2: HIStogram:BOXPcnt?

Argument 1: <NR3> is the left position of the histogram box, in percentage coordinates. 30 is the default value.

Argument 2: <NR3> is the top position of the histogram box, in percentage coordinates. 75 is the default value.

Argument 3: <NR3> is the right position of the histogram box, in percentage coordinates. 70 is the default value.

Argument 4: <NR3> is the bottom position of the histogram box, in percentage coordinates 25 is the default value.

Example: `HISTOGRAM:BOXPCNT 50.2, 75.6, 50.8, 30.4`
would define the coordinates of a histogram box in percentage coordinates

HIStogram:COUNt

(TDS 500D & 700D Only)

Description: Zeros the counts for histograms. If histograms are on then the counts start counting up again.

Group: Histogram

Syntax: HIStogram:COUNt RESET

Argument: RESET indicates the need to zero the count.

Example: `HISTOGRAM:COUNT RESET`
would zero the count.

HIStogram:DISplay

(TDS 500D & 700D Only)

Description: Selects the way the histogram is displayed, if at all.

Group: Histogram

Syntax 1: HIStogram:DISplay { LINEAr | LOG | OFF }

Syntax 2: HIStogram:DISplay?

Argument 1: LINEAr to display the count in each bin. This is the default value.

Argument 2: LOG to display the log of the count in each bin. Log scaling provides better visual detail for bins with low counts.

Argument 3: OFF to turn off histogram displays. Histogram counting and measurements can continue. The histogram box is not turned off.

Example: `HISTOGRAM:DISPLAY LINEAR`
would display the count in each bin.

HIStogram:MODe

(TDS 500D & 700D Only)

Description: Selects the type of histogram to create or turns the histogram off.

Group: Histogram

Syntax 1: HIStogram:MODe { OFF | HORizontal | VERTical }

Syntax 2: HIStogram:MODe?

Argument 1: OFF means turn off. This is the default value.

Argument 2: HORizontal means select a horizontally positioned histogram that shows time distribution.

Sending this command with this argument will turn off the extended acquisition length and DPO modes.

Argument 3: VERTical means select a vertically positioned histogram that shows voltage (or amps, etc.) distribution.

Sending this command with this argument will turn off the extended acquisition length and DPO modes.

Example: `HISTOGRAM:MODE OFF`
would turn off the histogram mode.

HIStogram:SIZe

(TDS 500D & 700D Only)

Description: Controls the width or height of the histogram on the screen.

Group: Histogram

Syntax 1: HIStogram:SIZe <NR3>

Syntax 2: HIStogram:SIZe?

Argument: <NR3> varies from 0.1 to 10.0 in div units. It varies from 0.1 to 8.0 in HORIZONTAL mode and from 0.1 to 10.0 in VERTICAL mode. Resolution is to the nearest pixel.

Example: `HISTOGRAM:SIZE 2.0`
would set the size of the histogram to 2.0 divs.

HIStogram:SOUrce

(TDS 500D & 700D Only)

Description: Selects which trace will be compared against the histogram box, when histograms are turned on. CH1-4 need not be ON for histograms to run. You might want the CH OFF so you can see a full-screen histogram and not have waveform data confuse the display.

Group: Histogram

Syntax 1: HIStogram:SOUrce CH<x>

Syntax 2: HIStogram:SOUrce?

Argument: <x> indicates CH #. It may be 1, 2, 3, or 4.

Example: `HISTOGRAM:SOURCE CH1`
would enable CH1 to be compared against the histogram box.

HORizontal?

(Query Only)

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

Group: Horizontal

Syntax: HORizontal?

Example: `HORIZONTAL?`
might return the string `:HORIZONTAL:MODE MAIN;RECORDLENGTH 500; POSITION 5.0E+0;TRIGGER:POSITION 50;:HORIZONTAL:MAIN:SCALE 1.0E-6;:HORIZONTAL:DELAY:MODE RUNSAFTER;SCALE 1.0E-6;TIME: 16.0E-9`

HORizontal:ACQDURATION?

(TDS 500D & 700D Only) (Query Only)

Description: Returns the overall time interval covered by the live channels. This works for both extended acquisition length mode on (acquisition length = # points) and off (record length = acquisition length = # points).

Group: Horizontal

Syntax: HORizontal:ACQDURATION?

Returns: In seconds, # points * (time/point). The default is 5 ms.

HORizontal:ACQLENGTH?

(TDS 500D & 700D Only) (Query Only)

Description: Returns the acquisition length, either the extended one when the extended acquisition length mode is on or the record length when this mode is off.

Group: Horizontal

Syntax: HORizontal:ACQLENGTH?

Returns: The acquisition length. 500 is the default value.

Example: `HORIZONTAL:ACQLENGTH?`
might return: `500`.

HORizontal:CLOck

(TDS 400A Only)

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

Group: Horizontal

Syntax 1: HORizontal:CLOck { INTernal **|** EXTernal }

Syntax 2: HORizontal:CLOck?

Argument 1: INTernal enables the internal clock mode.

Argument 2: EXTernal enables the external clock mode.

Example: `HORIZONTAL:CLOCK INTERNAL`
enables the internal clocks.

HORizontal:CLOck:MAXRate

(TDS 400A Only)

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

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

Group: Horizontal

Syntax 1: HORizontal:CLOck:MAXRate <NR3>

Syntax 2: HORizontal:CLOck:MAXRate?

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

Example: `HORIZONTAL:CLOCK:MAXRATE 50`
sets the maximum external clock rate to 50 clocks per second.

HORizontal:DELay?

(Query Only)

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

Group: Horizontal

Related Commands: HORizontal?, HORizontal:DELay:MODe?, HORizontal:DELay:SCAle?, HORizontal:DELay:SECdiv?, HORizontal:DELay:TIMe?

Syntax: HORizontal:DELay?

Example: `HORIZONTAL:DELAY?`
might return the delay parameters
`:HORIZONTAL:DELAY:MODE RUNSAFTER;SCALE 1.0E-6;TIME: 16.0E-9`

HORizontal:DELay:MODe

Description: Selects the mode for the delayed time base. Sending this command is equivalent to setting **Time Base** in the Horizontal menu.

Group: Horizontal

Related Command: HORizontal:DELay:TIMe

Syntax 1: HORizontal:DELay:MODe { RUNSAfter **|** TRIGAfter }

Syntax 2: HORizontal:DELay:MODe?

Argument 1: RUNSAfter specifies that the delayed time base runs a user-specified amount of delay time after the main trigger event.

Argument 2: TRIGAfter specifies that the delayed time base is triggerable after the main time base triggers.

Example: `HORIZONTAL:DELAY:MODE?`
returns either `RUNSAFTER` or `TRIGAFTER`, indicating the delayed time base mode.

HORizontal:DELay:SCAle

Description: Sets the time per division for the delayed time base. Sending this command is equivalent to setting **Delayed Scale** in the Horizontal Scale side menu.

On the TDS 400A, changes made while the external clock is enabled do not take effect until the internal clock is enabled. Also, when the external clock is enabled, the query form of this command returns an <NR3> value representing '50'.

Group: Horizontal

Related Command: HORizontal:DELay:SECdiv

Syntax 1: HORizontal:DELay:SCAle <NR3>

Syntax 2: HORizontal:DELay:SCAle?

Argument: <NR3> is the time per division. For the TDS 784D, the range is 10 s to 200 ps. For the other TDS 500D and 700D, the range is 10 s to 500 ps. For the TDS 600B, the range is 10 s to 200 ps. For the TDS 400A, the range is 20 s to 1 ns. This are values that are in a 1-2-5 sequence. Values that are not in a 1-2-5 sequence (1-2.5-5 on the TDS 620B, 640B, & 644B) will be set to the closest valid value. If the delayed time base scale is set slower than the main time base scale, both the main and delayed time base scales will be set to the delay scale value.

Example 1: `HORIZONTAL:DELAY:SCALE 2.0E-6`
sets the delay scale to 2 us per division.

Example 2: `HORIZONTAL:DELAY:SCALE 9.0E-6`
sets the delay scale to 10 us per division. Since 9 us is not a valid value within the 1-2-5 sequence (1-2.5-5 on the TDS 620B, 640B, & 644B), it is automatically set to the closest valid value.

Example 3: `HORIZONTAL:DELAY:SCALE?`
might return `1.0E-3`, indicating that the delay time is 1 ms per division.

HORizontal:DELay:SECdiv

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

Group: Horizontal

Syntax 1: HORizontal:DELay:SECdiv <NR3>

Syntax 2: HORizontal:DELay:SECdiv?

HORizontal:DELay:TIMe

Description: Sets or queries the delay time to wait after the main trigger before the delayed time base begins. Sending this command is equivalent to setting **Delayed Runs After Main** in the **Time Base** side menu of the Horizontal menu.

Group: Horizontal

Related Commands: HORizontal:DELay:MODe, HORizontal:DELay:TIMe:RUNSAfter?, HORizontal:DELay:TIMe:TRIGAfter?

Syntax 1: HORizontal:DELay:TIMe <NR3>

Syntax 2: HORizontal:DELay:TIMe?

Argument: <NR3> is the time, in seconds, between the main trigger and the delayed trigger. The range on the TDS 510A, 500D, 600B, & 700D is from 16 ns to 250 seconds with a resolution of 4 ns. The range on the TDS 400A is from 10 ns to 20 seconds with a resolution of 10 ns.

Example 1: `HORIZONTAL:DELAY:TIME 2.0E-3`
sets the delay time between the main and delayed time base to 2 ms.

Example 2: `HORIZONTAL:DELAY:TIME?`
might return `:HORIZONTAL:DELAY:TIME:16.0E-9` for the delay time.

HORizontal:DELay:TIMe:RUNSAfter

Description: Sets or queries the delay time to wait after the main trigger before the delayed time base begins. Sending this command is equivalent to setting **Delayed Runs After Main** in the **Time Base** side menu of the Horizontal menu.

Group: Horizontal

Related Command: HORizontal:DELay:MODe

Syntax 1: HORizontal:DELay:TIMe:RUNSAfter <NR3>

Syntax 2: HORizontal:DELay:TIMe:RUNSAfter?

Argument: <NR3> is the time, in seconds, between the main trigger and the delayed trigger. The range is from 16 ns (10 ns on the TDS 400A) to 250 seconds (20 s on the TDS 400A) with a resolution of 4 ns.

Example: `HORIZONTAL:DELAY:TIME:RUNSAFTER 2.0E-3`
sets the delay time between the main and delayed time base to 2 ms.

HORizontal:DELay:TIMe:TRIGAfter

Description: Sets the delay time to wait in the trigger after delay mode.
This is the time that must pass before a delayed trigger is accepted.
Sending this command is equivalent to setting **Delay by Time** time in the Delayed Trigger menu.

Group: Horizontal

Related Command: HORizontal:DELay:MODe

Syntax 1: HORizontal:DELay:TIMe:TRIGAfter <NR3>

Syntax 2: HORizontal:DELay:TIMe:TRIGAfter?

Argument: <NR3> is the delay time, in seconds. The range is from 16 ns (60 ns on the TDS 400A to 250 seconds (20 s on the TDS 400A) with a resolution of 4 ns 10 ns down to 110 ns on the TDS 400A).

Example 1: `HORIZONTAL:DELAY:TIME:TRIGAFTER 4.0E-6`
sets the delay time to 4 us.

Example 2: `HORIZONTAL:DELAY:TIME:TRIGAFTER?`
might return `1.000E-3`, indicating that the delay time is 1 ms.

HORizontal:EXTDACQ

(TDS 500D & 700D Only)

Description: Enable or disable extended-acquisition-length mode. The DPO display mode must be off in order to turn the extended-acquisition-length mode on.

Group: Horizontal

Syntax 1: HORizontal:EXTDACQ { <NR1> **|** OFF **|** ON }

Syntax 2: HORizontal:EXTDACQ?

Argument 1: indicates OFF if it’s a 0 or ON if it’s a 1 (or any other nonzero value).

Argument 2: ON means turn on.

Argument 3: OFF means turn off.

Returns: The query form only returns 0 or 1. The default is 0.

Example: `HORIZONTAL:EXTDACQ ON`
would enable extended-acquisition-length mode (if DPO mode was not on).

HORizontal:FASTframe:COUNt

(TDS 500D & 700D Only)

Description: Sets or queries FastFrame frame count. Sending this command is equivalent to setting **FastFrame Setup** in the Horizontal menu and the **Frame Count** menu item in the side menu. FastFrame, also known as memory segmentation, captures a series of triggered acquisitions with minimal intervening time.

Group: Horizontal

Syntax 1: HORizontal:FASTframe:COUNT <NR1>

Syntax 2: HORizontal:FASTframe:COUNT?

Argument: <NR1> indicates the number of frames to acquire.

Example: `HORIZONTAL:FASTFRAME:COUNT 2`
Sets up FastFrame mode to acquire two frames (segments) of data.

HORizontal:FASTframe:FRAMELock

(TDS 500D & 700D Only)

Description: Turns FastFrame frame lock on or off.

Group: Horizontal

Syntax 1: HORizontal:FASTframe:FRAMELock { ON | OFF | <NR1>}

Syntax 2: HORizontal:FASTframe:FRAMELock?

Argument 1: <NR1> indicates OFF if it's a 0 or ON if it's a 1
(or any other non-zero value).

Argument 2: ON keeps the reference and position frames the same relative distance from each other when either frame number is adjusted.

Argument 3: OFF allows the reference and position frames to be adjusted independently.

Example 1: `HORIZONTAL:FASTFRAME:FRAMELOCK ON`
turns on FastFrame Lock frames.

Example 2: `HORIZONTAL:FASTFRAME:FRAMELOCK?`
might return 1 indicating that FastFrame Lock frames is on.

HORizontal:FASTframe:LENgth

(TDS 500D & 700D Only)

Description: Setup length of each FastFrame frame. Sending this command is equivalent to setting **FastFrame Setup** in the Horizontal menu and the **Frame Length** menu item in the side menu. FastFrame, also known as memory segmentation, lets users capture a series of triggered acquisitions with minimal intervening time between them.

Group: Horizontal

Syntax 1: HORizontal:FASTframe:LENgth { <NR1> }

Syntax 2: HORizontal:FASTframe:LENgth?

Argument: <NR1> indicates the frame (segment) length.

Example: `HORIZONTAL:FASTFRAME: LENGTH 250`
Sets up the FastFrame mode to acquire frames of 250 samples each.

HORizontal:FASTframe:POSition

(TDS 500D & 700D Only)

Description: Display the selected FastFrame frame. Sending this command is equivalent to selecting **Horiz Pos** in the Horizontal menu, selecting the **Frame** menu item in the side menu, and entering a value with the keypad or the general purpose knob. FastFrame, also known as memory segmentation, lets users capture a series of triggered acquisitions with minimal intervening time between them.

Group: Horizontal

Syntax 1: HORizontal:FASTframe:POSition <NR1>

Syntax 2: HORizontal:FASTframe:POSition?

Argument: <NR1> indicates the selected frame to display.

Example: `HORIZONTAL:FASTFRAME:POSITION 25`
Selects the 25th FastFrame frame to display.

HORizontal:FASTframe:REF

(TDS 500D & 700D Only)

Description: Sets the reference frame number. The reference frame is the starting frame in calculating the delta timestamp. Sending this command is equivalent to setting **Fast Frame** in the Horizontal menu and the **Reference Frame** menu item in the side menu.

Group: Horizontal

Syntax 1: HORizontal:FASTframe:REF <NR1>

Syntax 2: HORizontal:FASTframe:REF?

Argument 1: <NR1> is the reference frame number. If the argument is less than 1 or greater than the frame count, the value is clipped, but no error is generated.

Example 1: `HORIZONTAL:FASTFRAME:REF 1`
sets the reference frame for FastFrame TimeStamp readouts to 1.

Example 2: `HORIZONTAL:FASTFRAME:REF?`
might return 3 indicating that the reference frame for FastFrame TimeStamps is 3.

HORizontal:FASTframe:STATE

(TDS 500D & 700D Only)

Description: Set up FastFrame acquisition. Sending this command is equivalent to setting **FastFrame Setup** in the Horizontal menu and the **FastFrame** menu item in the side menu.

FastFrame, also known as memory segmentation, lets you capture a series of triggered acquisitions with minimal time between them. In FastFrame mode, the oscilloscope is ready to accept a continuous burst of triggers 400 ms after the controller sends the ACQuire:STATE RUN command.

When extended-acquisition-length mode or DPO mode is on, this command can still set and return values. However, FastFrame acquisitions will not occur.

Group: Horizontal

Syntax 1: HORizontal:FASTframe:STATE { ON **|** OFF **|** <NR1>}

Syntax 2: HORizontal:FASTframe:STATE?

Argument 1: <NR1> indicates OFF if it's a 0 or ON if it's a 1 (or any other non-zero value).

Argument 2: ON turns on FastFrame.

Argument 3: OFF turns off FastFrame.

Example: `HORIZONTAL:FASTFRAME:STATE ON`
turns on FastFrame.

HORizontal:FASTframe:TIMEStamp:BETWeen?

(TDS 500D & 700D Only) (Query Only) (No LRN)

Description: Returns the relative time between the triggers of two specified frames. If FastFrame is not on or either time stamp is not available, a parameter error is generated.

Group: Horizontal

Syntax: HORizontal:FASTframe:TIMEStamp:BETWeen? <NR1>, <NR1>

Returns: The relative time between the triggers of two specified frames.

Example: `HORIZONTAL:FASTFRAME:TIMESTAMP:BETWEEN? 1,5`
might return: `“45:23:11.987 654 321 042”`.

HORizontal:FASTframe:TIMEStamp:CLEARSNapshot

(TDS 500D & 700D Only) (No Query Form) (No LRN)

Description: Clears the TimeStamp Snapshot display list and overlay menu if they are displayed.

Group: Horizontal

Syntax: HORizontal:FASTframe:TIMEStamp:CLEARSNapshot

Example: `HORIZONTAL:FASTFRAME:TIMESTAMP:CLEARSNAPSHOT`
clears the TimeStamp Snapshot if displayed.

HORizontal:FASTframe:TIMEStamp:DELTa?

(TDS 500D & 700D Only) (Query Only) (No LRN)

Description: Returns the relative time between the triggers of the reference and position frames. If FastFrame is not on or if either time stamp is not available, a parameter error is generated.

Group: Horizontal

Syntax: HORizontal:FASTframe:TIMEStamp:DELTa?

Returns: The relative time between the triggers of the reference and position frames.

Example: `HORIZONTAL:FASTFRAME:TIMESTAMP:DELTA?`
might return: `“45:23:11.987 654 321 042”`.

HORizontal:FASTframe:TIMEStamp:FRAme?

(TDS 500D & 700D Only) (Query Only) (No LRN)

Description: Returns the absolute trigger date and time for the requested frame (in text format). If FastFrame is not on, or if the time stamp is not available, a parameter error is generated.

Group: Horizontal

Syntax: HORizontal:FASTframe:TIMEStamp:FRAme? <NR1>

Argument 1: <NR1> is the frame number for which the timestamp will be returned.

Returns: The absolute trigger date and time for the requested frame (in text format).
The displayed time uses the following format:

@Pos: xxx	Position (or reference) frame number
DD MMM YYYY	Date (day, month, and year)
HH:MM:SS.mmm,	Clock time (hours, minutes, seconds, and milliseconds)
uuu,nnn,ppp	Fraction of a second (to picoseconds)

Example: `HORIZONTAL:FASTFRAME:TIMESTAMP:FRAME? 3`
might return: `“4 Apr 1998;08:47:24.123 456 789 112”`.

HORizontal:FASTframe:TIMEStamp:POSition?

(TDS 500D & 700D Only) (Query Only) (No LRN)

Description: Returns the absolute trigger date and time for the position frame (in text format). If FastFrame is not on or either time stamp is not available, a parameter error is generated.

Group: Horizontal

Syntax: HORizontal:FASTframe:TIMEStamp:POSition?

Returns: The absolute trigger date and time for the position frame.
The displayed time uses the following format:

@Pos: xxx	Position (or reference) frame number
DD MMM YYYY	Date (day, month, and year)
HH:MM:SS.mmm,	Clock time (hours, minutes, seconds, and milliseconds)
uuu,nnn,ppp	Fraction of a second (to picoseconds)

Example: `HORIZONTAL:FASTFRAME:TIMESTAMP:POSITION?`
might return: `“4 Apr 1998;08:47:24.123 456 789 112”`.

HORizontal:FASTframe:TIMEStamp:REF?

(TDS 500D & 700D Only) (Query Only) (No LRN)

Description: Returns the absolute trigger date and time for the reference frame (in text format). If FastFrame is not on or if the time stamp is not available, a parameter error is generated.

Group: Horizontal

Syntax: HORizontal:FASTframe:TIMEStamp:REF?

Returns: The absolute trigger date and time for the reference frame.
The displayed time uses the following format:

@Pos: xxx	Position (or reference) frame number
DD MMM YYYY	Date (day, month, and year)
HH:MM:SS.mmm,	Clock time (hours, minutes, seconds, and milliseconds)
uuu,nnn,ppp	Fraction of a second (to picoseconds)

Example: `HORIZONTAL:FASTFRAME:TIMESTAMP:REF?`
might return: `“4 Apr 1998;08:47:24.123 456 789 112”`.

HORizontal:FASTframe:TIMEStamp:SNAPShot

(TDS 500D & 700D Only) (No Query) (No LRN)

Description: Displays the TimeStamp Snapshot display list and overlay menu on the oscilloscope display. If a Snapshot is already displayed, this command behaves like the **Again** button of FastFrame TimeStamp Snapshot.

Group: Horizontal

Syntax: HORizontal:FASTframe:TIMEStamp: SNAPShot

Example: `HORIZONTAL:FASTFRAME:TIMESTAMP:SNAPSHOT`
displays the TimeStamp Snapshot.

HORizontal:FASTframe:TIMEStamp:STATE

(TDS 500D & 700D Only)

Description: Turns the TimeStamp mode on or off. FastFrame mode being off does not preclude TimeStamp mode from being on. However, if FastFrame mode is off, turning on TimeStamp mode has no effect until FastFrame is turned on.

Group: Horizontal

Syntax 1: HORizontal:FASTframe:TIMEStamp:STATE { ON **|** OFF **|** <NR1>}

Syntax 2: HORizontal:FASTframe:TIMEStamp:STATE?

Argument 1: <NR1> indicates OFF if it's a 0 or ON if it's a 1
(or any other non-zero value).

Argument 2: ON turns on FastFrame TimeStamps.

Argument 3: OFF turns off FastFrame TimeStamps.

Example: `HORIZONTAL:FASTFRAME:TIMESTAMP:STATE ON`
turns on FastFrame TimeStamps.

HORizontal:FITtoscreen

Description: Set-up horizontal-waveform compression. Sending this command is equivalent to setting **Record Length** in the Horizontal menu and the **Fit to screen** menu item in the side menu. Waveform compress lets you fit a captured waveform to the visible screen. It provides the same functionality as if you were in zoom mode and changed the time per division until the waveform fit the screen.

In extended-acquisition-length mode, entering the zoom preview state causes fit-to-screen to be on. If the oscilloscope exits the extended-acquisition-length mode or zoom preview state, the fit-to-screen mode returns back to its prior setting.

Group: Horizontal

Syntax 1: HORizontal:FITtoscreen { ON | OFF | <NR1>}

Syntax 2: HORizontal:FITtoscreen?

Argument 1: <NR1> indicates OFF if it's a 0. It indicates ON if it's a non-zero value.

Argument 2: ON turns on waveform compress.

Argument 3: OFF turns off waveform compress.

Example: `HORIZONTAL:FITTOSCREEN ON`
turns on waveform compress.

HORizontal:MAIn?

(Query Only)

Description: Returns the time per division of the main time base. The commands HORizontal:MAIn:SECdiv and HORizontal:MAIn:SCAle are identical so only HORizontal:MAIn:SCAle will be returned.

Group: Horizontal

Related Commands:
HORizontal:SCAle, HORizontal:SECdiv, HORizontal:MAIn:SECdiv

Syntax: HORizontal:MAIn?

Example: `HORIZONTAL:MAIN?`
might return `:HORIZONTAL:MAIN:SCALE 1.0E-6.`

HORizontal:MAIn:SCAle

Description: Sets the time per division for the main time base. Sending this command is equivalent to setting **Main Scale** in the Horizontal Scale side menu.

On the TDS 400A, changes made while the external clock is enabled do not take effect until the internal clock is enabled. Also, when the external clock is enabled, the query form of this command returns an <NR3> value representing '50'.

In extended-acquisition-length mode, no matter what value you set, the oscilloscope will only use the closest real-time, non-interpolated value.

Group: Horizontal

Related Commands:
HORizontal:DELay:SCAle, HORizontal:DELay:SECdiv, HORizontal:MAIn:SECdiv

Syntax 1: HORizontal:MAIn:SCAle <NR3>

Syntax 2: HORizontal:MAIn:SCAle?

Argument: <NR3> is the time per division. For the TDS 400A series, the range is 20 s to 1 ns. For the TDS 510A, 500D, 600B, and 700D, the range is 10 s to 500 ps (or 200 ps on the TDS 784D), in a 1-2-5 sequence.

Example: `HORIZONTAL:MAIN:SCALE 2E-6`
sets the main scale to 2 us per division.

HORizontal:MAIn:SECdiv

Description: 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 oscilloscopes.

Group: Horizontal

Related Commands:
HORizontal:DELay:SCAle, HORizontal:DELay:SECdiv, HORizontal:MAIn:SCAle

Syntax 1: HORizontal:MAIn:SECdiv <NR3>

Syntax 2: HORizontal:MAIn:SECdiv?

HORizontal:MODe

Description: Selects whether the horizontal display uses the main or delayed time base or both. Sending this command is equivalent to setting **Time Base** in the Horizontal menu.

Group: Horizontal

Related Command:
DISplay:INTENSITy:CONTRast

Syntax 1: HORizontal:MODe { DELAYEd **|** INTENSIFied **|** MAIn }

Syntax 2: HORizontal:MODe?

Argument 1: DELAYEd means that the selected waveform is horizontally scaled relative to the delayed time base.

Argument 2: INTENSIFied uses both the main and delay scales to display the waveform. The portion of the waveform that would be displayed in DELAYEd mode is intensified. The level of intensity is set by the DISplay:INTENSITy:CONTRast command.

Argument 3: MAIn means that the waveform is horizontally scaled relative to the main time base.

Example 1: `HORIZONTAL:MODE DELAYED`
uses the delayed horizontal scale to display the waveform.

Example 2: `HORIZONTAL:MODE?`
might return `INTENSIFIED`, indicating that the waveform is displayed using both the main and delayed time base scale.

HORizontal:POSition

Description: Positions the waveform horizontally on the display. This is used for both main and delayed time bases. Sending this command is equivalent to adjusting the front-panel **Horizontal Position** knob or setting the position in the Horizontal Position side menu.

Group: Horizontal

Syntax 1: HORizontal:POSition <NR3>

Syntax 2: HORizontal:POSition?

Argument: <NR3> is from 0 to 100, and is the percent of the waveform that is displayed left of the center graticule.

In extended-acquisition-length mode, it is the percent of the acquisition that is displayed left of the center graticule.

Example: `HORIZONTAL:POSITION 10`
sets the horizontal position of the waveform such that 10% of the waveform is to the left of screen center.

HORizontal:RECOrdlength

Description: Sets the number of data points that are acquired for each record. Sending this command is equivalent to setting **Record Length** in the Horizontal menu.

When extended-acquisition-length mode is on, this command can still set and return values. However, this feature will not actually work while extended-acquisition-length mode is on.

Group: Horizontal

Syntax 1: HORizontal:RECOrdlength <NR1>

Syntax 2: HORizontal:RECOrdlength?

Argument: <NR1> Table 2-30 below shows supported values for <NR1>.

Table 2-30:Record Length Values (<NR1>)

Configuration	TDS 420A TDS 430A TDS 460A	TDS 510A TDS 520D TDS 724D	TDS 600B	TDS 540D TDS 580D TDS 754D TDS 784D TDS 794D
Standard	500, 1000, 2500, 5000, 15000, 30000	500, 1000, 2500, 5000, 15000, 50000	500, 1000, 2500, 5000, 15000	500, 1000, 2500, 5000, 15000, 50000
Option 1M (not on the TDS 510A) or 2M (TDS 500D & 700D only)	60000, 120000	TDS 520D & 724D: 75000, 100000, 130000, 250000		75000, 100000, 130000, 250000, 500000

Example 1: `HORIZONTAL:RECORDLENGTH 2500`
specifies that 2500 data points will be acquired for each record.

Example 2: `HORIZONTAL:RECORDLENGTH?`
might return `15000` as the number of data points per record.

HORizontal:RECORDSTART

(TDS 500D & 700D Only)

Description: Set the selected live channel's record start that is to be used in the extended-acquisition-length mode. In other words, it indicates where the waveform starts within the extended acquisition record.

Group: Horizontal

Syntax 1: HORizontal:RECORDSTART <NR3>

Syntax 2: HORizontal:RECORDSTART?

Argument: <NR3> is the units of percent. It ranges from 0% to 93.75% (for 4M and 8M) or 95% (for 2M).

If fit-to-screen is on or extreme zoom factors are on, then the recordstart is automatically set to 0.

Returns: The query form returns a value from 0 to 93.75 (%) for 4M and 8M or 95 (%) for 2M. This is the figure for a specific sample.

Example: `HORIZONTAL:RECORDSTART 23`
would set the record to start at 23%.

HORizontal:ROLL

(TDS 400A Only)

Description: Sets roll mode to auto (on) or off.

Group: Horizontal

Syntax 1: HORizontal:ROLL { OFF **|** ON **|** AUTO **|** <NR1> }

Syntax 2: HORizontal:ROLL?

Argument 1: OFF or <NR1> = 0 turns off roll mode.

Argument 2: ON or AUTO or <NR1> NOT = 0 turns on roll mode if the time/division is set appropriately.

HORizontal:SCAle

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

Group: Horizontal

Syntax 1: HORizontal:SCAle <NR3>

Syntax 2: HORizontal:SCAle?

HORizontal:SECdiv

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

Group: Horizontal

Syntax 1: HORizontal:SECdiv <NR3>

Syntax 2: HORizontal:SECdiv?

HORizontal:TRIGger?

(Query Only)

Description: Returns the horizontal trigger parameter.

Group: Horizontal

Syntax: HORizontal:TRIGger?

Example: `HORIZONTAL:TRIGGER?`
might return `:HORIZONTAL:TRIGGER:POSITION 50`.

HORizontal:TRIGger:POSition

Description: Sets or queries the position of the trigger. Sending this command is equivalent to setting **Trigger Position** in the Horizontal menu.

Group: Horizontal

Syntax 1: HORizontal:TRIGger:POSition <NR1>

Syntax 2: HORizontal:TRIGger:POSition?

Argument: <NR1> is from 0 to 100 %.

Example: `HORIZONTAL:TRIGGER:POSITION?`
might return `50`.

ID?

(Query Only)

Description: Returns identifying information about the instrument and its firmware.

Group: Status and Error

Related Command: *IDN?

Syntax: ID?

Returns: The instrument id in the following format:

TEK/<model number>,CF:92.1CT,FV:<firmware version number>

Example: `ID?`
might return `TEK/TDS544A,CF:92.1CT,FV:3.0`

*IDN?

(Query Only)

Description: Returns the oscilloscope identification code.

Group: Miscellaneous

Related Command: ID?

Syntax: *IDN?

Returns: The instrument id in the following format:

TEKTRONIX,<model number>,0,CF:92.1CT FV:<firmware version number>

Example: `*IDN?`
might return the response `TEKTRONIX,TDS544A,0,CF:92.1CT FV:2.0`

LIMit:BELl

Description: Rings the bell when the waveform data exceeds the limits set in the limit test, if the limit state is on.

Group: Limit Test

Related Command: LIMit:COMpare:CH<x>, LIMit:STATE

Syntax 1: LIMit:BELl { OFF | ON | <NR1> }

Syntax 2: LIMit:BELl?

Argument 1: OFF or <NR1> = 0 turns off ringing the bell when any waveform data exceeds the limits set by the limit test.

Argument 2: ON or <NR1> NOT = 0 turns on ringing the bell.

Example 1: `LIMit:BELl ON`
specifies that the bell is to ring when any waveform data exceeds the limits specified in the limit test.

Example 2: `LIMit:BELl?`
returns either `0` or `1`, indicating whether the bell is to ring when any waveform data exceeds the limits specified in the limit test.

LIMit:COMpare:CH<x>

Description: Sets or queries the template against which to compare the waveform acquired through the specified channel. The template can be a waveform saved in any of the reference locations REF1 through REF4, or none.

Group: Limit Test

Related Commands:
CURve, LIMit:COMpare:MATH<x>, LIMit:TEMPLate, LIMit:TEMPLate:DESTination, LIMit:TEMPLate:SOUrce, WFMPre

Syntax 1: LIMit:COMpare:CH<x> { NONe | REF<x> }

Syntax 2: LIMit:COMpare:CH<x>?

Argument 1: REF<x> is a reference waveform.

Argument 2: NONe specifies that no template testing is to be done for the specified channel.

Example 1: `LIMIT:COMPARE:CH1 REF1`
specifies REF1 as the template waveform against which to compare waveforms acquired using CH1.

Example 2: `LIMIT:COMPARE:CH2?`
might return `LIMIT:COMPARE:CH2 REF4`, indicating that waveforms acquired using CH2 will be compared to the template waveform stored in REF4.

LIMit:COMpare:MATH<x>

(TDS 510A, 500D, 600B, & 700D)

Description: Sets or queries the template against which to compare a math waveform. The template can be a waveform saved in any of the reference locations REF1 through REF4, or none.

Group: Limit Test

Related Commands:
CURve, LIMit:TEMPLate, LIMit:TEMPLate:DESTination, LIMit:TEMPLate:SOUrce, WFMPre

Syntax 1: LIMit:COMpare:MATH<x> { NONe | REF<x> }

Syntax 2: LIMit:COMpare:MATH<x>?

Argument 1: REF<x> is a reference waveform.

Argument 2: NONe specifies that no template testing is to be done for the specified channel.

Argument 3: MATH<x> is a math waveform.

Example 1: `LIMIT:COMPARE:CH1 REF1`
specifies REF1 as the template waveform against which to compare waveforms acquired using CH1.

Example 2: `LIMIT:COMPARE:CH2?`
might return `LIMIT:COMPARE:CH2 REF4`, indicating that waveforms acquired using CH2 will be compared to the template waveform stored in REF4.

LIMit:HARDCopy

Description: Executes a hardcopy operation on the waveform when any waveform data exceeds the limits set in the limit test, if the limit state is on. The hardcopy operation uses the port, and prints in the format and layout, specified using the HARDCopy commands.

Group: Limit Test

Related Commands: LIMit:COMpare:CH<x>, LIMit:STATE, HARDCopy

Syntax 1: LIMit:HARDCopy { OFF **|** ON **|** <NR1> }

Syntax 2: LIMit:HARDCopy?

Argument 1: ON or <NR1> NOT = 0 turns on the hardcopy operation for the waveform when any waveform data exceeds the limits set by the limit test.

Argument 2: OFF or <NR1> = 0 turns off the hardcopy operation.

Example 1: `LIMit:HARDCopy ON`
specifies that the hardcopy operation occurs for the waveform when any waveform data exceeds the limits specified in the limit test.

Example 2: `LIMit:HARDCopy?`
returns either `0` or `1`, indicating whether the hardcopy operation occurs for the waveform when any waveform data exceeds the limits specified in the limit test.

LIMit:STATE

Description: Turns limit testing on or off, or queries whether limit testing is in effect.

When extended-acquisition-length mode or DPO mode is on, this command can still set and return values. However, this feature will not actually function.

You cannot turn limit:state to on when extended-acquisition-length mode is already on.

Group: Limit Test

Related Commands:
CURve, LIMit:BELl, LIMit:COMpare:CH<x>, LIMit:HARDCopy, LIMit:TEMPLate, WFMPre

Syntax 1: LIMit:STATE { OFF **|** ON **|** <NR1> }

Syntax 2: LIMit:STATE?

Argument 1: OFF or <NR1> = 0 turns off limit testing.

Argument 2: ON or <NR1> NOT = 0 turns on limit testing.

Example 1: `LIMit:STATE ON`
specifies that limit testing of waveforms is in effect.

Example 2: `LIMit:STATE?`
returns either `0` or `1`, indicating whether limit testing of waveforms is in effect.

LIMit:TEMPLate

(No Query Form)

Description: Creates a template which you can use for limit testing. You can compare the waveform acquired through the specified channel against this template. The template can be a waveform saved in any of the reference locations REF1 through REF4, or none.

Group: Limit Test

Related Commands:
LIMit:TEMPLate:DESTination, LIMit:TEMPLate:SOUrce

Syntax: LIMit:TEMPLate STORe

Argument: STORe creates a template with the specified source waveform and tolerances, and stores it in the destination reference waveform to be used in limit testing comparisons.

Example: `LIMIT:TEMPLate STORe`
creates a template with the specified source waveform and tolerances, and stores it in the destination reference waveform to be used in limit testing comparisons.

LIMit:TEMPLate:DESTination

Description: Sets or queries the destination reference waveform that the LIMit:TEMPLate STORe command will use.

Group: Limit Test

Related Commands: LIMit:COMpare:CH<x>, LIMit:TEMPLate, LIMit:TEMPLate:SOUrce

Syntax 1: LIMit:TEMPLate:DESTination REF<x>

Syntax 2: LIMit:TEMPLate:DESTination?

Argument: REF<x> specifies the reference waveform destination in which the template waveform is to be stored.

Example: `LIMIT:TEMPLate:DESTination REF2`
specifies that the template waveform referred to with the LIMit:TEMPLate STORe command is stored as the REF2 waveform.

LIMit:TEMPLate:SOUrce

Description: Sets or queries the channel, math waveform, or reference waveform that the LIMit:TEMPLate STORe command will use.

Group: Limit Test

Related Commands: LIMit:COMpare:CH<x>, LIMit:TEMPLate, LIMit:TEMPLate:DESTination

Syntax 1: LIMit:TEMPLate:SOUrce { CH<x> | MATH<x> | REF<x> }

Syntax 2: LIMit:TEMPLate:SOUrce?

Argument 1: CH<x> specifies that the template waveform is the waveform currently being acquired using the specified channel.

Argument 2: MATH<x> specifies that the template waveform is the waveform currently stored as the specified math waveform.

Argument 3: REF<x> specifies that the template waveform is the waveform currently stored as the specified reference waveform.

Example 1: `LIMIT:TEMPLate:SOUrce CH2`
specifies that the template waveform for limit tests is the waveform currently acquired using channel 2.

Example 2: `LIMIT:TEMPLate:SOUrce?`
might return MATH3, specifying that the template waveform for limit tests is the waveform currently stored as the MATH3 waveform.

LIMit:TEMPLate:TOLerance:HORizontal

Description: Sets or queries the amount, in units of horizontal divisions, by which the source waveform is varied horizontally when creating the destination waveform.

Group: Limit Test

Related Command: LIMit:COMpare:CH<x>

Syntax 1: LIMit:TEMPLate:TOLerance:HORizontal <NR3>

Syntax 2: LIMit:TEMPLate:TOLerance:HORizontal?

Argument: <NR3> is the amount, in horizontal divisions, by which the current waveform is allowed to deviate from the template waveform without being deemed to have exceeded the limits set in the limit test. The range is 0 to 5 divisions.

Example 1: `LIMIT:TEMPLate:TOLerance:HORizontal 1.0`
specifies that the current waveform is deemed to be close enough to the template waveform if it is within +-1.0 horizontal division.

Example 2: `LIMIT:TEMPLate:TOLerance:HORizontal?`
might return `1.0`, specifying that the current waveform is deemed to be close enough to the template waveform if it is within +-1.0 horizontal division.

LIMit:TEMPLate:TOLerance:VERTical

Description: Sets or queries the amount, in vertical divisions, by which the source waveform is varied vertically when creating the destination waveform.

Group: Limit Test

Related Command: LIMit:COMpare:CH<x>

Syntax 1: LIMit:TEMPLate:TOLerance:VERTical <NR3>

Syntax 2: LIMit:TEMPLate:TOLerance:VERTical?

Argument: <NR3> is the amount, in vertical divisions, by which the current waveform is allowed to deviate from the template waveform without exceeding the limits set in the limit test. The range is 0 to 5 divisions.

Example 1: `LIMIT:TEMPLate:TOLerance:VERTical 1.0`
specifies that the current waveform is deemed to be close enough to the template waveform if it is within + or - 1.0 vertical division from the template waveform.

Example 2: `LIMIT:TEMPLate:TOLerance:VERTical?`
might return `1.0`, specifying that the current waveform is deemed to be close enough to the template waveform if it is within + or - 1.0 vertical division.

LOCk

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

Group: Miscellaneous

Related Commands: UNLock, Remote Enable Group, Local Lockout Group

Syntax 1: LOCk { ALL | NONe }

Syntax 2: LOCk?

Argument 1: ALL disables all front panel controls.

Argument 2: NONe enables all front-panel controls. This is equivalent to the UNLock ALL command.

NOTE: If the oscilloscope is in the Remote With Lockout State (RWLS), the LOCk NONe command has no effect. For more information, see the ANSI/IEEE Std. 488.1-1987 Standard Digital Interface for Programmable Instrumentation, section 2.8.3 on RL State Descriptions.

Example 1: `LOCK ALL`
locks the front panel controls.

Example 2: `LOCK?`
returns `NONe` when the front panel controls are enabled by this command.

*LRN?

(Query Only)

Description: Returns a string listing the oscilloscope settings, except for configuration information for the calibration values. You can use this string to return the oscilloscope to the state it was in when you made the *LRN? query.

Group: Miscellaneous

Related Commands: HEADer, SET?, VERBose

Syntax: *LRN?

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

Example: `*LRN?`
a partial response might look like this:
```
:ACQUIRE:STATE 1;MODE SAMPLE;NUMENV 10;NUMAVG 16;REPET 1;STOPAFTER
RUNSTOP;:DIAG:LOOP:OPTION ONCE;COUNT 1;:DIAG:STATE HALT;:HEADER
1;:VERBOSE 1;:CURSOR:FUNCTION OFF;VBARS:UNITS SECONDS;POSITION1 1.00E-
6;POSITION2 9.00E-6;SELECT CURSOR1;
```

MASK?

(TDS 500D & 700D Only) (Query Only)

Description: Returns the status of all settable mask parameters.

Group: Mask

Syntax: MASK?

Returns:
might return

```
:MASK:MASK1:POINTSP
11.400000000E+0,12.531328321E+0,8.820e+01,8.840e+01,0.0
E+0,1.120e+01,0.0E+0;:MASK:MASK2:POINTSP
22.800000000E+0,50.125313283E+0,3.820e+01,6.617e+01,6.140e+01,6.6
17e+01,7.680e+01,5.013e+01,6.140e+-1,3.383e01,3.383e+0
1;:MASK:MASK3:POINTSP
11.400000000E+0,87.719298246E+0,8.820e+01,8.772e+01,8.840e+01,1.0
03e+02,1.120e+01,1.003e+02;:MSK:MASK4:POINTSP
0.0E+0,0.0E+0;:MASK:MASK5:POINTSP
0.0E+0,0.0E+0;:MASK:MASK6:POINTSP
0.0E+0,0.0E+0;:MASK:MASK7:POINTSP
0.0E+0,0.0E+0;:MSAK:MASK8:POINTSP
0.0E+0,0.0E+0;:MASK:STAN
0C1;DIS 1;COUN:STATE 0;:MASK:FILT ENA;AUTOS:OFFSETA 1;MOD
MAN;:MASK:SOU CH1;TBP 0.0E+0
```

MASK:AUTOSet:MODe

(TDS 500D & 700D Only)

Description: Controls whether or not an autoset will automatically be done after a standard mask is selected. If mask counting is being done against a math trace, the mask autoset will not run.

Group: Mask

Syntax 1: MASK:AUTOSet:MODe { AUTO | MANual }

Syntax 2: MASK:AUTOSet:MODe?

Argument 1: AUTO makes autoset run automatically when a standard mask is selected from the front panel. Autoset does not keep running. It runs once when a standard is selected.

Argument 2: MANual means that the mask autoset algorithm for a standard mask will be run only if the user presses the autoset button or sends the AUTOSET START command. If a standard mask is not currently selected, the mask autoset algorithm will not run and the normal autoset will run when the user presses the autoset button. Manual is the default value.

Example: `MASK:AUTOSET:MODE AUTO`
makes autoset run automatically when a standard mask is selected from the front panel.

MASK:AUTOSet:OFFSETAdj

(TDS 500D & 700D Only)

Description: Controls whether or not a standard mask autoset adjusts vertical offset so the waveform matches the mask. Such adjustment can be needed if ground (0 V) in the oscilloscope does not match ground for the input signal. For the following standards, offset will not be adjusted if this feature is OFF: T1.102, DS-0, E1, E2, and E3. For all other standards, offset will be adjusted regardless of how this feature is set.

Group: Mask

Syntax 1: MASK:AUTOSet:OFFSETAdj { ON | OFF | <NR1> }

Syntax 2: MASK:AUTOSet:OFFSETAdj?

Argument 1: OFF or <NR1> = 0 prevents a standard-mask autoset from adjusting vertical offset for standard masks. This is the default value.

Argument 2: ON or <NR1> not = 0 allows a standard-mask autoset to adjust vertical offset for standard masks.

Example: `MASK:AUTOSET:OFFSETADJ ON`
allows a standard-mask autoset to adjust vertical offset for standard masks.

MASK:AUTOSet:STANdard

(TDS 500D & 700D Only)

Selects a standard mask for autoset. You can modify a standard mask and still force the oscilloscope to autoset to the mask.

Group: Mask

Syntax 1: MASK:AUTOSet:STANdard { OC1 | OC3 | OC12 | DS0Single | DS0Double | DS0Contra | DS0Timing | DS1Rate | E1Symmetrical | E1Coax | E2 | E3 | DS3Rate | E4_0 | E4_1 | E5 | STM1E_0 |STM1E_1 | DS1 | DS1A | DS1C | DS2 | DS3 | DS3Time | DS4NA |DS4NA_Max | STS1Eye | STS1Pulse | STS3 | STS3_Max | FC133 | FC266 | FC531 | FC1063 | FC133E | FC266E | FC531E | FC1063E | D2 | D1 | VIDEO360 | ENET1250 | ENET100UTP | ENET100STP | ENET10IDTime | ENET10IDVolt | ENET10IDFull |ENET10LKTime | ENET10LKVolt | ENET10LKFull | FDDI | NONe | USERMask }

Syntax 2: MASK:AUTOSet:STANdard?

Argument 1: OC1 selects the mask specified by the OC1/STM0 standard (51.84 Mb/s).

Argument 2: OC3 selects the mask specified by the OC3/STM1 standard (155.52 Mb/s).

Argument 3: OC12 selects the mask specified by OC12/STM4 standard (622.08 Mb/s).

Argument 4: DS0Single selects the mask specified by the DS-0 Single Pulse standard (64 kb/s).

Argument 5: DS0Double selects the mask specified by the DS-0 Double Pulse standard (64 kb/s).

Argument 6: DS0Contra selects the mask specified by the DS-0 Data Pulse, Contradirectional standard (64 kb/s).

Argument 7: DS0Timing selects the mask specified by the DS-0 Timing Pulse, Contradirectional standard (64 kb/s).

Argument 8: DS1Rate selects the mask specified by the ITU G.703 standard (1.544 Mb/s).

Argument 9: DS3Rate selects the mask specified by the ITU G.703 standard (44.736 Mb/s).

Argument 10: E1Symmetrical selects the mask specified by the E1 Symmetrical Pair standard (2.048 Mb/s).

Argument 11: E1Coax selects the mask specified by the E1 Coaxial Pair standard (2.048 Mb/s).

Argument 12: E2 selects the mask specified by the E2 standard (8.448 Mb/s).

Argument 13: E3 selects the mask specified by the E3 standard (34.368 Mb/s).

Argument 14: ITU44 selects the mask specified by the ITU G.703 standard (44.736 Mb/s).

Argument 15: E4_0 selects the mask specified by the E4 Binary 0 standard (139.26 Mb/s).

Argument 16: E4_1 selects the mask specified by the E4 Binary 1 standard (139.26 Mb/s).

Argument 17: E5 selects the mask specified by the E5 CEPT standard (565 Mb/s).

Argument 18: STM1E_0 selects the mask specified by the G.703 Binary 0 standard (155.52 Mb/s). The SONET standard GR-253-CORE calls this mask STS-3. ANSI standard T1.102 has STS-3 eye pattern standards.

Argument 19: STM1E_1 selects the mask specified by the G.703 Binary 1 standard (155.52 Mb/s). The SONET standard GR-253-CORE calls this mask STS-3. ANSI standard T1.102 has STS-3 eye pattern standards.

Argument 20: DS1 selects the mask specified by the DS1 standard (1.544 Mb/s).

Argument 21: DS1A selects the mask specified by the DS1A standard (2.048 Mb/s).

Argument 22: DS1C selects the mask specified by the DS1C standard (3.152 Mb/s).

Argument 23: DS2 selects the mask specified by the DS2 standard (6.312 Mb/s).

Argument 24: DS3 selects the mask specified by the DS3 standard (44.736 Mb/s).

Argument 25: DS3Time selects the mask specified by the DS3 standard (44.736 Mb/s). Mask removes 1% of the points from each end of the DS3 Full Mask to fit mask within 5 ns per division.

Argument 26: DS4NA selects the mask specified by the DS4NA eye standard (139.26 Mb/s).

Argument 27: DS4NA_Max selects the mask specified by the DS4NA Maximum equipment output eye standard (139.26 Mb/s).

Argument 28: STS1Eye selects the mask specified by the STS-1 Eye standard (51.84 Mb/s).

Argument 29: STS1Pulse selects the mask specified by the STS-1 Pulse standard (51.84 Mb/s).

Argument 30: STS3 selects the mask specified by the STS-3 eye standard (155.52 Mb/s).

Argument 31: STS3_Max selects the mask specified by the STS-3 Maximum equipment output eye standard (155.52 Mb/s).

Argument 32: FC133 selects the mask specified by the Fibre Channel 133 Optical standard (132.7 Mb/s).

Argument 33: FC266 selects the mask specified by the Fibre Channel 266 Optical standard (265.6 Mb/s).

Argument 34: FC531 selects the mask specified by the Fibre Channel 531 Optical standard (531.2 Mb/s).

Argument 35: FC1063 selects the mask specified by the Fibre Channel 1063 Optical standard (1.0635 Gb/s).

Argument 36: FC133E selects the mask specified by the Fibre Channel 133 Electrical standard (132.7 Mb/s).

Argument 37: FC266E selects the mask specified by the Fibre Channel 266 Electrical standard (265.6 Mb/s).

Argument 38: FC531E selects the mask specified by the Fibre Channel 531 Electrical standard (531.2 Mb/s).

Argument 39: FC1063E selects the mask specified by the Fibre Channel 1063 Electrical standard (1.0635 Gb/s).

Argument 40: D2 selects the mask implied by the ANSI/SMPTE 259M 4fsc NTSC standard (143.1818 Mb/s).

Argument 41: D1 selects the mask implied by the ANSI/SMPTE 259M 4:2:2 component serial video standard (270 Mb/s).

Argument 42: VIDEO360 selects the mask implied by the ANSI/SMPTE 259M 4:2:2 component serial video standard (360 Mb/s).

Argument 43: ENET1250 selects the mask implied by the IEEE Draft P802.3z/D3 standard (1.25 Gb/s).

Argument 44: ENET100UTP selects the mask implied by the ANSI X3.263-1995 for Information Technology standard (125 Mb/s).

Argument 45: ENET100STP selects the mask implied by the ANSI X3.263-1995 for Information Technology standard (125 Mb/s).

Argument 46: ENET10IDTime selects the mask implied by the ANSI/IEEE 802.3 Fifth Edition 1996 - Information Technology - Telecommunications and Information Exchange standard (10 Mb/s). This mask displays the first 20% of the IDL Full Mask, so that you can see that part of the signal in detail.

Argument 47: ENET10IDVolt selects the mask implied by the ANSI/IEEE 802.3 Fifth Edition 1996 - Information Technology - Telecommunications and Information Exchange standard (10 Mb/s). This mask changes the Vertical scale to zoom in on the middle 20% of the signal while looking at the Full mask on the time axis.

Argument 48: ENET10IDFull selects the mask implied by the ANSI/IEEE 802.3 Fifth Edition 1996 - Information Technology - Telecommunications and Information Exchange standard (10 Mb/s).

Argument 49: ENET10LKTime selects the mask implied by the ANSI/IEEE 802.3 Fifth Edition 1996 - Information Technology - Telecommunications and Information Exchange standard (10 Mb/s). This mask displays the first 20% of the Link Test Full Mask, so that you can see that part of the signal in detail.

Argument 50: ENET10LKVolt selects the mask implied by the ANSI/IEEE 802.3 Fifth Edition 1996 - Information Technology - Telecommunications and Information Exchange standard (10 Mb/s). This mask changes the Vertical scale to zoom in on the middle 20% of the signal while looking at the Full mask on the time axis.

Argument 51: ENET10LKFull selects the mask implied by the ANSI/IEEE 802.3 Fifth Edition 1996 - Information Technology - Telecommunications and Information Exchange standard (10 Mb/s).

Argument 52: FDDI selects the mask implied by the FDDI standard (125 Mb/s).

Argument 53: NONe causes the normal autoset to occur.

Argument 54: USERMask is a no-op when received by the SET form of this command. It is allowed because a SET? can return it.

Example: `MASK:AUTOSET:STANDARD OC3`
selects the OC3 standard for autoset.

MASK:COUNt

(TDS 500D & 700D Only) (No Query Form)

Description: Zeros the counts for the masks. If Count Masks is on, then counting starts again.

Group: Mask

Syntax: MASK:COUNt RESET

Argument: RESET zeros the mask counts.

Example: `MASK:COUNT RESET`
clears the mask counts.

MASK:COUNt:STATE

(TDS 500D & 700D Only)

Description: Controls whether or not mask counting is being done.

Group: Mask

Syntax 1: MASK:COUNt:STATE { ON **|** OFF **|** <NR1> }

Syntax 2: MASK:COUNt:STATE?

Argument 1: OFF or <NR1> = 0 turns off mask counting. This is the default state. If the channel to count against is not on, mask counting is not turned on and an error is generated. If DPO mode is on, mask counting is not turned on and an error is generated.

Argument 2: ON or <NR1> not = 0 turns on mask counting.

Example: `MASK:COUNT:STATE ON`
turns on mask counting.

MASK:COUNt:TOTal?

(TDS 500D & 700D Only) (Query Only) (no LRN)

Description: Returns the sum of all the hits in all the defined masks. If masks overlap (no standard masks overlap) and a hit occurs in the overlap region, the total hits will count that hit multiple times. It returns the current value, which may be displayed if the correct mask menu is up, of this number.

Group: Mask

Syntax: MASK:COUNt:TOTal?

Returns: <NR1>

Example: `MASK:COUNT:TOTAL?`
might return: `MASK:COUNT:TOTAL 57`

MASK:COUNt:WAVEFORMS?

(TDS 500D & 700D Only) (Query Only) (no LRN)

Description: Returns the number of waveforms that have contributed to mask counting. This can be much smaller than the ACQUIRE:NUMACQ number since the acquire number just counts how many triggers have been processed. The time per div may be fast enough that 100 or more triggers are required to be processed to fill one waveform. Even then, it may not be completely filled.

Group: Mask

Syntax: MASK:COUNt:WAVEFORMS?

Returns: <NR1>

Example: `MASK:COUNT:WAVEFORMS`
might return: `MASK:COUNT:WAVEFORMS 286568`

MASK:DISplay

(TDS 500D & 700D Only)

Description: Controls whether or not defined masks are displayed on the screen. This is useful for temporarily turning off user-defined masks without deleting them.

Group: Mask

Syntax 1: MASK:DISplay { OFF **|** ON **|** <NR1> }

Syntax 2: MASK:DISplay?

Argument 1: OFF or <NR1> = 0 removes the masks from the display. Turning this off turns mask counting off. Selecting Zoom Preview turns this off.

Argument 2: ON or <NR1> not = 0 displays the masks on the display. This is the default value. Turning mask counting on turns this on. Selecting any standard mask turns this on. MASK:MASK<x>:POINTSPCNT or MASK:MASK<x>:POINTS turns this on.

Example 1: `MASK:DISPLAY ON`
sets the display to show the defined masks.

Example 2: `MASK:DISPLAY?`
might return `1` indicating that the display shows masks.

MASK:FILTer

(TDS 500D & 700D Only)

Description: Controls whether a digital filter will be run on the waveform data. The filter simulates expensive optical hardware. Different hardware would be used for each of the optical standards. The digital filter will run only if ENABLE is the mode and one of the optical eye-pattern standard masks is selected, There are several other restrictions:

- One of the seven optical standards must be selected in the Measure/Masks/MaskType/Standard Mask menu. In other words, any one of the three SONET SDH or any of the four fibre channel standards from the Fibre Chan menu must be selected.
- The time per division setting must be the correct value for the mask which has been selected.
- The record length must be set equal to 500.
- Only one channel can be on. Mask autoset may force channels off in order to allow the mask to be turned on.
- The oscilloscope bandwidth must be set to full.

Group: Mask

Syntax 1: MASK:FILTer { ENAble | DISAble }

Syntax 2: MASK:FILTer?

Argument 1: ENAbled enables the digital filter. ENAbled is the default value.

Argument 2: DISAbled disables the digital filter.

Example: `MASK:FILTER ENABLED`
enables the digital filter.

MASK:INVert

(TDS 500D & 700D Only)

Description: Controls whether the defined masks appear inverted on the display. Inverted masks are useful for testing the opposite polarity of a pulse. Once on, the current mask and all following masks are displayed inverted (if the inverted form exists) until the invert command is turned off.

Group: Mask

Syntax 1: MASK:INVert { ON **|** OFF **|** <NR1> }

Syntax 2: MASK:INVert?

Argument 1: OFF or <NR1> = 0 turns off mask inversion. The default is OFF.

Argument 2: ON or <NR1> not = 0 turns on mask inversion.

Example: `MASK:INVERT ON`
turns on mask inversion.

MASK:MARgin:PERCent

(TDS 500D & 700D Only)

Description: Sets or queries the mask-margin percentage.

Group: Mask

Syntax 1: MASK:MARgin:PERCent <NR3>

Syntax 2: MASK:MARgin:PERCent?

Argument: <NR3> is -50.0 to +50.0. The default is 5. A positive margin percentage will cause the signal space to become smaller and make the mask pass/fail test harder to pass. A percentage that would cause a displayed mask to cross another mask or to go off the screen is not allowed, instead the oscilloscope restores the last valid margin.

Example: `MASK:MARGIN:PERCENT 25`
sets the mask margin percent to 25.

MASK:MARgin:STATE

(TDS 500D & 700D Only)

Description: Turns mask margins on or off.

Group: Mask

Syntax 1: MASK:MARgin:STATE { ON **|** OFF **|** <NR1> }

Syntax 2: MASK:MARgin:STATE?

Argument 1: OFF or <NR1> = 0 turns off mask margins. The default is OFF. The currently displayed margined mask is erased and the original mask is displayed.

Argument 2: ON or <NR1> not = 0 turns on mask margins. The status of the currently displayed standard mask is changed to User.

Example: `MASK:MARGIN:STATE ON`
turns on mask margins.

MASK:MASK<x>

(TDS 500D & 700D Only)

Description: In the set form, it deletes all points in the specified mask.
In the query form, it returns the state of all settable parameters of the specified mask.

Group: Mask

Syntax 1: MASK:MASK<x> DELEte

Syntax 2: MASK:MASK<x>?

Argument: DELEte deletes all points in the specified mask

Returns: (query form) the coordinates of the specified mask in percentage units.

Example 1: `MASK:MASK3 DELETE`
deletes the points in mask 3.

Example 2: `MASK:MASK1?`
might return:

```
MASK:MASK1:POINTSP
11.400000000E+0,12.531328321E+0,8.820e+01,1.253e+01,8.840e+01,0.0
E+0,1.120e+01,0.0E+0
```

MASK:MASK<x>:COUNt?

(TDS 500D & 700D Only) (Query Only) (no LRN)

Description: Returns the number of hits in the specified mask. This will be zero unless the MASK:COUNT:STATE is ON (or was ON).

Group: Mask

Syntax: MASK:MASK<x>:COUNt?

Returns: <NR1> is the number of hits in the specified mask. The count is never larger than 2**32-1 (4.29496e9).

Example: `MASK:MASK3:COUNT?`
might return:
`MASK:MASK3:COUNT 1478`

MASK:MASK<x>:NR_Pt?

(TDS 500D & 700D Only) (Query Only) (no LRN)

Description: Returns the number of points in the specified mask. One use of this query is to see how many pairs a MASK:MASK<x>:POINTSPCNT command will return.

Group: Mask

Syntax: MASK:MASK<x>:NR_Pt?

Returns: <NR1> is the number of points in the specified mask.

Example: `MASK:MASK3:NR_PT?`
might return:
`MASK:MASK3:NR_PT 5`

MASK:MASK<x>:POINTS

(TDS 500D & 700D Only) (no LRN)

Description: Defines points in the specified mask, in user coordinates. Any currently existing points in the mask are deleted.

Group: Mask

Syntax 1: MASK:MASK<x>:POINTS <NR3>, <NR3> [, <NR3>, <NR3>]

Syntax 2: MASK:MASK<x>:POINTS?

Argument: <NR3> refers to the coordinates of one of the vertexes in the mask. Each pair of <NR3> is the horizontal and vertical coordinates of a mask vertex. The order of the pairs has no effect on the mask created. If the vertical or horizontal scale or position is changed after this command and then the query form of this command is issued, then the <NR3> returned from the TDS oscilloscope will not be the same. The general rule for how the boundary of the mask is generated from the input pairs is that an imaginary line connects the top-left point with the bottom-right point. All points below this imaginary line are part of the bottom boundary curve. All other points are part of the top boundary curve. If just one pair is input then they are ignored and the mask is marked as undefined. The default is no points in the mask.

If any point is outside the display area, an error is generated and the point is clipped to the nearest boundary. If more than 50 points are specified, an error is generated and the extra points are ignored.

If the specified mask is undefined, 0,0 is returned.

Example: `MASK:MASK7:POINTS -2.3e-9, 44e-3, -2.5e-9, 47e-3, 1.2e-9, 40e-3` defines the points in mask 7.

MASK:MASK<x>:POINTSPcnt

(TDS 500D & 700D Only)

Description: Defines points in the specified mask in percentage coordinates. The upper leftmost point is 0,0 and the point in the lower right is 100,100. Any currently existing points in the mask are deleted.

Group: Mask

Syntax 1: MASK:MASK<x>:POINTSPcnt <NR3>, <NR3> [, <NR3>, <NR3>]

Syntax 2: MASK:MASK<x>:POINTSPcnt?

Argument: <NR3> refers to the coordinates of one of the vertices in the mask. Each pair of <NR3> values is the horizontal and vertical coordinates of a mask vertex. The order of the pairs has no effect on the mask created. Resolution of the <NR3> value is to the nearest pixel. 500 pixels horizontally implies 0.2% resolution. 200 pixels vertically implies 0.5% resolution. The general rule for how the boundary of the mask is generated from the input pairs is that an imaginary line connects the top-left point with the bottom-right point. All points below this imaginary line are part of the bottom boundary curve. All other points are part of the top boundary curve. Points are connected left to right. If just one pair is input, then they are ignored and the mask is marked as undefined. The default is no points in the mask.

If any point is outside of 0..100, an error is generated and the point is clipped to the nearest boundary. If more than 50 points are specified, an error is generated and the extra points are ignored.

Returns: <NR3> that define the mask coordinates. If the specified mask is undefined, 0,0 is returned.

Example: `MASK:MASK7:POINTSPCNT 20.4, 10.5,90, 10.5, 50, 80`
defines the points in mask 7.

MASK:PASSFail:BELl

(TDS 500D & 700D Only)

Description: Turns the bell on or off for mask test failed or mask test complete.

Group: Mask

Syntax 1: MASK:PASSFail:BELl { OFF | FAIl | COMPLete }

Syntax 2: MASK:PASSFail:BELl?

Argument 1: OFF or <NR1> = 0 turns off the pass/fail bell. The default is OFF.

Argument 2: FAIl turns on the pass/fail bell for a failed mask test.

Argument 3: COMPLete turns on the pass/fail bell for a failed mask test and for a complete mask test.

Example: `MASK:PASSFAIL:BELL OFF`
turns off the bell for mask testing.

MASK:PASSFail:STATE

(TDS 500D & 700D Only)

Description: Turns conditional mask counting on or off. The Pass/Fail Test side menu button will reflect either Passing or Failed based on the number of hits in the mask for the requested number of waveforms. A failed test generates an error message.

Group: Mask

Syntax 1: MASK:PASSFail:STATE { ON **|** OFF **|** <NR1> }

Syntax 2: MASK:PASSFail:STATE?

Argument 1: OFF or <NR1> = 0 turns off mask counting. The default is OFF.

Argument 2: ON or <NR1> not = 0 turns on mask counting.

Example: `MASK:PASSFAIL:STATE ON`
turns on mask counting.

MASK:PASSFail:STATUs?

(TDS 500D & 700D Only) (Query Only) (no LRN)

Description: Returns the result of the conditional mask test.

Group: Mask

Syntax: MASK:PASSFail:STATUs?

Returns: PASSING, PASSED, or FAILED is the status of the mask test.

Example: `MASK:PASSFAIL:STATUS?`
might return: `MASK:PASSFAIL:STATUS PASSING.`

MASK:PASSFail:THReshold

(TDS 500D & 700D Only)

Description: Sets or queries the minimum number of mask hits that is a test failure.

Group: Mask

Syntax 1: MASK:PASSFail:THReshold <NR1>

Syntax 2: MASK:PASSFail:THReshold?

Arguments: <NR1> is the minimum number of mask hits, from 1 to (2**30).

Example: `MASK:PASSFAIL:THRESHOLD 2`
sets the minimum number of mask hits to 2.

MASK:PASSFail:WAVEform

(TDS 500D & 700D Only)

Description: Sets or queries the minimum number of waveforms to be acquired for mask pass/fail counting.

Group: Mask

Syntax 1: MASK:PASSFail:WAVEform <NR3>

Syntax 2: MASK:PASSFail:WAVEform?

Argument: <NR3> is the minimum number of waveforms to be acquired for mask pass/fail counting. The range of acquired waveforms is 1 to 2**(30). The default is 20.

Example: `MASK:PASSFAIL:WAVEFORM 12`
sets the minimum number of waveforms to 12.

MASK:SOUrce

(TDS 500D & 700D Only)

Description: Selects which trace will be compared against the mask(s) when counting is turned on.

Group: Mask

Syntax 1: MASK:SOUrce CH<x>

Syntax 2: MASK:SOUrce?

Argument: <x> indicates CH #. It may be 1, 2, 3, or 4. 1 is the default.

Example: `MASK:SOURCE CH1`
would enable CH1 to be compared against the mask.

MASK:STANdard

(TDS 500D & 700D Only)

Description: Deletes the existing mask and sets the standard mask. If MASK:COUNT:STATE is ON, then mask counting starts. The query form queries the standard mask.

Group: Mask

Syntax 1: MASK:STANdard { OC1 | OC3 | OC12 | DSOSingle | DSODouble | DSOContra | DSOTiming | DS1Rate | E1Symmetrical | E1Coax | E2 | E3 | DS3Rate | E4_0 | E4_1 | E5 | STM1E_0 | STM1E_1 | DS1 | DS1A | DS1C | DS2 | DS3 | DS3Time | DS4NA | DS4NA_Max | STS1Eye | STS1Pulse | STS3 | STS3_Max | FC133 | FC266 | FC531 | FC1063 | FC133E | FC266E | FC531E | FC1063E | D2 | D1 | VIDEO360 | ENET1250 | ENET100UTP | ENET100STP | ENET10IDTime | ENET10IDVolt | ENET10IDFull |ENET10LKTime | ENET10LKVolt | ENET10LKFull | FDDI | NONe | USERMask }

Syntax 2: MASK:STANdard?

Argument 1: OC1 means delete any user defined masks and then create masks 1 ..3 as specified by OC1/STM0 standard (51.84 Mb/s).

Argument 2: .OC3 means delete any user defined masks and then create masks 1 .. 3 as specified by OC3/STM1 standard (155.52 Mb/s).

Argument 3: OC12 means delete any user defined masks and then create masks 1 .. 3 as specified by OC12/STM4 standard (622.08 Mb/s).

Argument 4: DS0Single means delete any user defined masks and then create masks 1 .. 2 as specified by the DS-0 Single Pulse standard (64 kb/s).

Argument 5: .DS0Double means delete any user defined masks and then create masks 1 .. 2 as specified by the DS-0 Double Pulse standard (64 kb/s).

Argument 6: DS0Countra means delete any user defined masks and then create masks 1 .. 2 as specified by the DS-0 Data Pulse, Contradirectional standard (64 kb/s).

Argument 7: DS0Timing means delete any user defined masks and then create masks 1 .. 2 as specified by the DS-0 Timing Pulse, Contradirectional standard (64 kb/s).

Argument 8: DS1Rate means delete any user defined masks and then create the masks as specified by the ITU G.703 standard (1.544 Mb/s).

Argument 9: E1Symmetrical means delete any user defined masks and then create masks 1 .. 2 as specified by the E1 Symmetrical Pair standard (2.048 Mb/s).

Argument 10: E1Coax means delete any user defined masks and then create masks 1 .. 2 as specified by the E1 Coaxial Pair standard (2.048 Mb).

Argument 11: E2 means delete any user defined masks and then create masks 1 .. 2 as specified by the E2 standard (8.448 Mb/s).

Argument 12: E3 means delete any user defined masks and then create masks 1 .. 2 as specified by the E3 standard (34.368 Mb/s).

Argument 13: DS3Rate means delete any user defined masks and then create the masks as specified by the ITU G.703 standard (44.736 Mb/s).

Argument 14: E4_0 means delete any user defined masks and then create masks 1 .. 2 as specified by the E4 Binary 0 standard (139.26 Mb/s).

Argument 15: E4_1 means delete any user defined masks and then create masks 1 ..2 as specified by the E4 Binary 1 standard (139.26 Mb/s).

Argument 16: E5 means delete any user defined masks and then create masks 1 .. 2 as specified by the E5 CEPT standard (565 Mb/s).

Argument 17: STM1E_0 means delete any user defined masks and then create masks 1 .. 2 as specified by the G.703 Binary 0 standard (155.52 Mb). The SONET standard GR-23-CORE calls this masks STS-3. ANSI standard T1.102 has STS-3 eye pattern standards.

Argument 18: STM1E_1 means delete any user defined masks and then create masks 1 .. 2 as specified by the G.703 Binary 1 standard (155.52 Mb/s). The SONET standard GR-23-CORE calls this masks STS-3. ANSI standard T1.102 has STS-3 eye pattern standards.

Argument 19: DS1 means delete any user defined masks and then create masks 1 .. 2 as specified by the DS1 standard (1.544 Mb/s).

Argument 20: DS1A means delete any user defined masks and then create masks 1 .. 2 as specified by the DS1 standard (2.048 Mb/s).

Argument 21: DS1C means delete any user defined masks and then create masks 1 .. 2 as specified by the DS1 standard (3.152 Mb/s).

Argument 22: DS2 means delete any user defined masks and then create masks 1 .. 2 as specified by the DS2 standard (6.312 Mb/s).

Argument 23: DS3 means delete any user defined masks and then create masks 1 .. 2 as specified by the DS3 standard (44.736 Mb/s)

Argument 24: DS3Time means delete any user defined masks and then create the masks as specified by the DS3 standard (44.736 Mb/s). Mask removes 1% of the points from each end of the DS3 Full Mask to fit mask within 5 ns per division.

Argument 25: DS4NA means delete any user defined masks and then create masks 1 .. 4 as specified by the DS4NA eye standard (139.26 Mb/s).

Argument 26: DS4NA_Max means delete any user defined masks and then create masks 1 .. 4 as specified by the DS4NA Maximum equipment output eye standard (139.26 Mb/s).

Argument 27: STS1Eye means delete any user defined masks and then create masks 1 .. 4 as specified by the STS-1 Eye standard (51.84 Mb/s).

Argument 28: STSPulse means delete any user defined masks and then create masks 1 .. 2 as specified by the STS-1 Pulse standard (51.84 Mb/s).

Argument 29: STS3 means delete any user defined masks and then create masks 1 .. 2 as specified by the STS-3 eye standard (155.52 Mb/s).

Argument 30: STS3_Max means delete any user defined masks and then create masks 1 .. 4 as specified by the STS-3 Maximum equipment output eye standard (155.52 Mb/s).

Argument 31: FC133 means delete any user defined masks and then create masks 1 .. 3 as specified by the Fibre Channel 133 Optical standard (132.7 Mb/s).

Argument 32: FC266 means delete any user defined masks and then create masks 1 .. 3 as specified by the Fibre Channel 266 Optical standard (265.6 Mb/s).

Argument 33: FC531 means delete any user defined masks and then create masks 1 .. 3 as specified by the Fibre Channel 531 Optical standard (531.2 Mb/s).

Argument 34: FC1063 means delete any user defined masks and then create masks 1 .. 3 as specified by the Fibre Channel 1063 Optical standard (1.0635 Gb/s).

Argument 35: FC133E means delete any user defined masks and then create masks 1 .. 3 as specified by the Fibre Channel 133 Electrical standard (132.7 Mb/s).

Argument 36: FC266E means delete any user defined masks and then create masks 1 .. 3 as specified by the Fibre Channel 266 Electrical standard (265.6 Mb/s).

Argument 37: FC531E means delete any user defined masks and then create masks 1 .. 3 as specified by the Fibre Channel 531 Electrical standard (531.2 Mb/s).

Argument 38: FC1063E means delete any user defined masks and then create masks 1 .. 3 as specified by the Fibre Channel 1063 Electrical standard (1.0635 Gb/s).

Argument 39: D2 means delete any user defined masks and then create masks 1 .. 3 as specified by the ANSI/SMPTE 259M 4fsc NTSC standard (143.1818 Mb/s).

Argument 40: D1 means delete any user defined masks and then create masks 1 .. 3 as specified by the ANSI/SMPTE 259M 4:2:2 component serial video standard (270 Mb/s).

Argument 41: VIDEO360 means delete any user defined masks and then create the masks as implied by the ANSI/SMPTE 259M 4:2:2 component serial video standard (360 Mb/s).

Argument 42: ENET1250 means delete any user defined masks and then create the masks as implied by the IEEE Draft P802.3z/D3 standard (1.25 Gb/s).

Argument 43: ENET100UTP means delete any user defined masks and then create the masks as implied by the ANSI X3.263-1995 for Information Technology standard (125 Mb/s).

Argument 44: ENET100STP means delete any user defined masks and then create the masks as implied by the ANSI X3.263-1995 for Information Technology standard (125 Mb/s).

Argument 45: ENET10IDTime means delete any user defined masks and then create the masks as implied by the ANSI/IEEE 802.3 Fifth Edition 1996 - Information Technology - Telecommunications and Information Exchange standard (10 Mb/s). This mask displays the first 20% of the IDL Full Mask, so that you can see that part of the signal in detail.

Argument 46: ENET10IDVolt means delete any user defined masks and then create the masks as implied by the ANSI/IEEE 802.3 Fifth Edition 1996 - Information Technology - Telecommunications and Information Exchange standard (10 Mb/s). This mask changes the Vertical scale to zoom in on the middle 20% of the signal while looking at the Full mask on the time axis.

Argument 47: ENET10IDFull means delete any user defined masks and then create the masks as implied by the ANSI/IEEE 802.3 Fifth Edition 1996 - Information Technology - Telecommunications and Information Exchange standard (10 Mb/s).

Argument 48: ENET10LKTime means delete any user defined masks and then create the masks as implied by the ANSI/IEEE 802.3 Fifth Edition 1996 - Information Technology - Telecommunications and Information Exchange standard (10 Mb/s). This mask displays the first 20% of the Link Test Full Mask, so that you can see that part of the signal in detail.

Argument 49: ENET10LKVolt means delete any user defined masks and then create the masks as implied by the ANSI/IEEE 802.3 Fifth Edition 1996 - Information Technology - Telecommunications and Information Exchange standard (10 Mb/s). This mask changes the Vertical scale to zoom in on the middle 20% of the signal while looking at the Full mask on the time axis.

Argument 50: ENET10LKFull means delete any user defined masks and then create the masks as implied by the ANSI/IEEE 802.3 Fifth Edition 1996 - Information Technology - Telecommunications and Information Exchange standard (10 Mb/s).

Argument 51: FDDI means delete any user defined masks and then create masks 1 .. 2 as specified by the FDDI standard (125 Mb/s).

Argument 52: NONE causes all masks to be deleted. This even includes user masks.

Argument 53: USERMask is a no-op when received by the SET form of this command. It is allowed because a SET? can get it.

Example: `MASK:STANDARD OC3`
selects the OC3 standard.

MASK:TBPosition

(TDS 500D & 700D Only) (No Query Form)

Description: Set the time base position. Use this command to reposition the waveform against the mask.

Group: Mask

Syntax: MASK:TBPosition <NR3>

Argument: <NR3> is the time base position in seconds. The maximum range is + or -5 divisions. Increasing the value moves the trigger position and waveform left on the screen. Decreasing it moves the trigger position and waveform right. Minimum resolution is 1 ps. The oscilloscope will round any number entered to the nearest 1 ps. The default is 0 seconds.

Example: `MASK:TBPOSITION 1.44e-9`
moves the waveform appropriately.

MATH<x>?

(Query Only)

Description: Returns the definition for the math waveform specified by <x>.

Group: Vertical

Syntax: MATH<x>?

MATH<x>:DEFIne

Description: Allows the user to define new waveforms using mathematical expressions. Sending this command is equivalent to selecting **Change Math waveform definition** in the Math<x> side menu.

Group: Vertical

Syntax 1: MATH<x>:DEFIne <QString>

Syntax 2: MATH<x>:DEFIne?

Argument: <QString> contains the mathematical expression. The expression can include any amount of white space. Expressions can be either single or dual waveform expressions. <src> and <function> elements are case independent.

The format for a single waveform expression is:

<function>(<source> [, <window>, <scaling>, <phase suppression>])

The format for a dual waveform expression is:

<source><operator><source>

where:

<function> ::= INV | DIF | FFT | INT

- INVert (for invert): inverts the defined waveform.

- DIFferentiate (available with Advanced DSP Math only): takes the derivative of the selected waveform.

- FFT (available with Advanced DSP Math only): provides an FFT of the selected waveform. It uses the format: "FFT(<source>, <window>, <scaling>, <phase suppression>)" where the window, scaling, and phase suppression arguments in the parentheses are optional. You can specify these arguments in any order.

  <source> refers to a signal channel. Valid choices are: CH1, CH2, CH3, CH4, REF1, REF2, REF3, or REF4.

  (The TDS 430A does not include CH3 and CH4.)

  <window> refers to an FFT window. Valid choices are: RECtangular, HAMming, HANning, or BLAckmanharris.

  <scaling> provides vertical scaling. Valid choices are: LOGrms, LINearrms, DEGreesphase, or RADiansphase.

  <Phase suppression> is of the range: -100 dB to 100 dB.

- INTegrate (available on instruments with the Option 2F Advanced DSP Math only): takes the integral of the selected waveform.

- <operator> ::= { + | - | * | / (not available on TDS 400A)}

- <source> ::= { CH<x> | REF<x> }

Example 1: `MATH2:DEFINE "Ch1 + cH2"`
adds channel 1 and channel 2, and stores the result in MATH2.

Example 2: `MATH1:DEFINE "INV( ref4 )"`
inverts the waveform stored in reference memory location 4 storing the result in MATH1.

Example 3: `MATH1:DEFINE "FFT( CH1 )"`
takes an FFT on the waveform from channel 1 and stores the result in MATH1.

Example 4: `MATH1:DEFINE "FFT( CH1, HAMM, LINEARRMS, 20 )"`
takes an FFT from channel 1, using the HAMMING algorithm, with linear rms scaling, and 20 dB phase suppression. The result is stored in MATH1.

Example 5: `MATH1:DEFINE?`
might return "Ch2*Ref2" as the expression that defines MATH1.

MATH<x>:NUMAVg

(TDS 510A, 500D, 600B, & 700D) (Some models require Option 2F)

Description: Allows the user to declare at what acquisition number the averaging algorithm will begin exponential averaging. Prior to that acquisition number, the algorithm uses stable averaging. Sending this command is equivalent to selecting **Average** in the Math<x> side menu and entering a value with the general purpose knob or the keypad.

Group: Vertical

Syntax 1: MATH<x>:NUMAVg { <NR1> }

Syntax 2: MATH<x>:NUMAVg?

Argument: < NR1 > specifies the number of times to successively average the math waveform before completing an acquisition.

Example 1: `MATH2:NUMAVG 10`
Successively averages math waveform 2 by 10 times.

Example 2: `MATH2:NUMAVG?`
might return `10` indicating 10 math 2 waveforms are successively averaged before a single acquisition occurs.

MATH<x>:PROCessing

(TDS 510A, 500D, 600B, & 700D) (Some models require Option 2F)

Description: Allows the user to turn on or off averaging for the math waveform specified by <x>. Sending this command is equivalent to selecting **No Extended Process** or **Average** in the Math<x> side menu.

Math averaging allows the oscilloscope to successively average any acquisition-related math waveform. This can help reduce noise in a math waveform.

Group: Vertical

Syntax 1: MATH<x>:PROCessing { OFF **|** AVErage }

Syntax 2: MATH<x>:PROCessing?

Argument 1: OFF turns off waveform averaging.

Argument 2: AVErage turns on waveform averaging.

Example 1: `MATH1:PROCESSING OFF`
ensures that waveform averaging is not in use on math waveform 1.

Example 2: `MATH1:PROCESSING AVERAGE`
turns on waveform averaging on math waveform 1.

MEASUrement?

(Query Only)

Description: Returns all measurement parameters.

Group: Measurement

Syntax: MEASUrement?

Example: `MEASUREMENT?`
might return `:MEASUREMENT:MEAS1:STATE 0;TYPE PERIOD;UNITS "s";SOURCE1 CH1;SOURCE2 CH1;DELAY:EDGE1 RISE;EDGE2 RISE;DIRECTION FORWARDS;:MEASUREMENT:MEAS2:STATE 0;TYPE PERIOD;UNITS "s";SOURCE1 CH1;SOURCE2 CH1;DELAY:EDGE1 RISE;EDGE2 RISE;DIRECTION FORWARDS;:MEASUREMENT:MEAS3:STATE 0;TYPE PERIOD;UNITS "s";SOURCE1 CH1;SOURCE2 CH1;DELAY:EDGE1 RISE;EDGE2 RISE;DIRECTION FORWARDS;:MEASUREMENT:MEAS4:STATE 0;TYPE PERIOD;UNITS "s";SOURCE1 CH1;SOURCE2 CH1;DELAY:EDGE1 RISE;EDGE2 RISE;DIRECTION FORWARDS;:MEASUREMENT:IMMED:TYPE PERIOD;UNITS "s";SOURCE1 CH1;SOURCE2 CH1;DELAY:EDGE1 RISE;EDGE2 RISE;DIRECTION FORWARDS;:MEASUREMENT:METHOD HISTOGRAM;REFLEVEL:METHOD PERCENT;ABSOLUTE:HIGH 0.0E+0;LOW 0.0E+0;MID 0.0E+0;MID2 0.0E+0;:MEASUREMENT:REFLEVEL:PERCENT:HIGH 90.0E+0;LOW 10.0E+0;MID 50.0E+0;MID2 50.0E+0`

MEASUrement:CLEARSNapshot

Description: Removes the measurement snapshot display.

Group: Measurement

Syntax: MEASUrement:CLEARSNapshot

Example: `MEASUREMENT:CLEARSNAPSHOT`

MEASUrement:GATing

Description: Sets or queries measurement gating.

Group: Measurement

Related Command: CURSor:VBARS

Syntax 1: MEASUrement:GATing { ON **|** OFF **|** <NR1> }

Syntax 2: MEASUrement:GATing?

Argument 1: ON (or 1) turns on measurement gating.

Argument 2: OFF (or 0) turns off measurement gating.

Example 1: `MEASUREMENT:GATING ON`
turns gating on.

Example 2: `MEASUREMENT:GATING?`
might return `MEASUREMENT:GATING 1` showing gating is turned on.
It might also return `MEASUREMENT:GATING 0` showing gating is turned off.

MEASUrement:IMMed?

(Query Only)

Description: Returns all immediate measurement setup parameters.

Group: Measurement

Syntax: MEASUrement:IMMed?

Example: `MEASUREMENT:IMMED?`
might return `:MEASUREMENT:IMMED:TYPE PERIOD;UNITS "s"; SOURCE1 CH1;SOURCE2 CH1;DELAY:EDGE1 RISE;EDGE2 RISE; DIRECTION FORWARDS`

MEASUrement:IMMed:DELay?

(Query Only)

Description: Returns information about the immediate delay measurement.

Group: Measurement

Syntax: MEASUrement:IMMed:DELay?

Example: `MEASUREMENT:IMMED:DELAY?`
might return
`:MEASUREMENT:IMMED:DELAY:EDGE1 RISE;EDGE2 RISE; DIRECTION FORWARDS`

MEASUrement:IMMed:DELay:DIREction

Description: Sets or queries the starting point and direction that determines the delay "to" edge when taking an immediate delay measurement. Use the MEASUrement:IMMed:SOURCE2 command to specify the delay "to" waveform.

Group: Measurement

Syntax 1: MEASUrement:IMMed:DELay:DIREction {BACkwards | FORWards}

Syntax 2: MEASUrement:IMMed:DELay:DIREction?

Argument 1: BACkwards means that the search starts at the end of the waveform and looks for the last rising or falling edge in the waveform. The slope of the edge is specified by MEASUrement:IMMed:DELay:EDGE2.

Argument 2: FORWards means that the search starts at the beginning of the waveform and looks for the first rising or falling edge in the waveform. The slope of the edge is specified by MEASUrement:IMMed:DELay:EDGE2.

Example 1: `MEASUREMENT:IMMED:DELAY:DIRECTION FORWARDS`
starts searching from the beginning of the waveform record.

Example 2: `MEASUREMENT:IMMED:DELAY:DIRECTION?`
returns either `BACkwards` or `FORWARDS`.

MEASUrement:IMMed:DELay:EDGE1

Description: Sets or queries the slope of the edge that is used for the delay "from" waveform when taking an immediate delay measurement. The waveform is specified by MEASUrement:IMMed:SOURCE1.

Group: Measurement

Related Command: MEASUrement:IMMed:SOURCE1

Syntax 1: MEASUrement:IMMed:DELay:EDGE1 { FALL **|** RISe }

Syntax 2: MEASUrement:IMMed:DELay:EDGE1?

Argument 1: FALL specifies the falling edge.

Argument 2: RISe specifies the rising edge.

Example 1: `MEASUREMENT:IMMED:DELAY:EDGE1 RISE`
specifies that the rising edge be used for the immediate delay measurement.

Example 2: `MEASUREMENT:IMMED:DELAY:EDGE1?`
returns either `RISE` or `FALL`.

MEASUrement:IMMed:DELay:EDGE2

Description: Sets or queries the slope of the edge that is used for the delay "to" waveform when taking an immediate delay measurement. The waveform is specified by MEASUrement:IMMed:SOURCE2.

Group: Measurement

Related Command: MEASUrement:IMMed:SOURCE2

Syntax 1: MEASUrement:IMMed:DELay:EDGE2 { FALL **|** RISe }

Syntax 2: MEASUrement:IMMed:DELay:EDGE2?

Argument 1: FALL specifies the falling edge.

Argument 2: RISe specifies the rising edge.

Example 1: `MEASUREMENT:IMMED:DELAY:EDGE2 RISE`
specifies that the rising edge be used for the immediate delay measurement.

Example 2: `MEASUREMENT:IMMED:DELAY:EDGE2?`
returns `FALL` showing that the falling or negative edge of the waveform is used for the immediate delay measurement.

MEASUrement:IMMed:SOURCE[1]

Description: Sets or queries the source for all single channel immediate measurements and specifies the source to measure "from" when taking an immediate delay measurement or phase measurement.

Group: Measurement

Syntax 1: MEASUrement:IMMed:SOURCE[1] { CH<x> | MATH<x> | REF<x> | HIStogram (TDS500D and 700D }

Syntax 2: MEASUrement:IMMed:SOURCE[1]?

Argument 1: CH<x> is an input channel.

Argument 2: MATH<x> is a math waveform.

Argument 3: REF<x> is a reference waveform.

Argument 4: HIStogram (TDS 500D & 700D) is a histogram.

Example: `MEASUREMENT:IMMED:SOURCE MATH1`
specifies `MATH1` as the immediate measurement source.

MEASUrement:IMMed:SOURCE2

Description: Specifies the source to measure "to" when taking an immediate delay measurement or phase measurement.

Group: Measurement

Syntax 1: MEASUrement:IMMed:SOURCE2 { CH<x> | MATH<x> | REF<x> | HIStogram (TDS500D and 700D }

Syntax 2: MEASUrement:IMMed:SOURCE2?

Argument 1: CH<x> is an input channel.

Argument 2: MATH<x> is a math waveform.

Argument 3: REF<x> is a reference waveform.

Argument 4: HIStogram (TDS 500D & 700D) is a histogram.

Example 1: `MEASUREMENT:IMMED:SOURCE2 REF3`
sets the waveform in reference memory location 3 as the delay "to" source when making delay measurements.

Example 2: `MEASUREMENT:IMMED:SOURCE2?`
might return `MATH1`.

MEASUrement:IMMed:TYPe

Description: Specifies the immediate measurement.

Group: Measurement

Syntax 1: MEASUrement:IMMed:TYPe { AMPlitude | AREA | BURst | CARea | CMEan | CRMs | DELay | EXTINCTDB (TDS 500D & 700D) | EXTINCTPCT (TDS 500D & 700D) | EXTINCTRATIO (TDS 500D & 700D) | FALL | FREQuency | HIGH | HITS (TDS 500D & 700D) | LOW | MAXimum | MEAN | MEANDBM (TDS 500D & 700D) | MEDian (TDS 500D & 700D) | MINImum | NDUty | NOVershoot | NWIdth | PDUTy | PEAKHits (TDS 500D & 700D) | PERIod | PHAse | PK2pk | POVershoot | PWIdth | RISe | RMS | SIGMA[1-3] (TDS 500D & 700D) STDdev (TDS 500D & 700D) | WAVEFORMS (TDS 500D & 700D) }

Syntax 2: MEASUrement:IMMed:TYPe?

Argument 1: AMPlitude is the high value minus the low value.

Argument 2: AREA is the area between the curve and ground over the entire waveform.

Argument 3: BURst is the time from the first MidRef crossing to the last MidRef crossing.

Argument 4: CARea (cycle area) is the area between the curve and ground over one cycle.

Argument 5: CMEan is the arithmetic mean over one cycle.

Argument 6: CRMs is the true Root Mean Square voltage over one cycle.

Argument 7: DELay is the time between the MidRef crossings of two different waveforms.

Argument 8: EXTINCTDB (TDS 500D & 700D) is 10.0*log(10)(Extinction Rati0

Argument 9: EXTINCTPCT (TDS 500D & 700D) is 100/Extinction Ration

Argument 10: EXTINCTRATIO (TDS 500D & 700D) is High/Low

Argument 11: FALL is the time that it takes for the falling edge of a pulse to fall from a HighRef value to a LowRef value of its final value.

Argument 12: FREQuency is the reciprocal of the period measured in Hertz.

Argument 13: HIGH is the 100% reference level.

Argument 14: HITS (TDS 500D & 700D) displays the number of acquired points within or on the histogram box.

Argument 15: LOW is the 0% reference level.

Argument 16: MAXimum is the highest amplitude (voltage).

Argument 17: MEAN is the arithmetic mean over the entire waveform.

Argument 18: MEANDBM (TDS 500D & 700D) is the median of all acquired points within or on the histogram box. Half of the acquired points are greater than and half less than this value.

Argument 19: MEDian (TDS 500D & 700D) is the median of all acquired points within or on the histogram box. Half of the acquired points are greater than and half less than this value.

Argument 20: MINImum is the lowest amplitude (voltage).

Argument 21: NDUty is the ratio of the negative pulse width to the signal period expressed as a percentage.

Argument 22: NOVershoot is the negative overshoot, expressed as:

NOVershoot = 100 x ((Low - Minimum)/Amplitude)

Argument 23: NWIdth is the distance (time) between MidRef (usually 50%) amplitude points of a negative pulse.

Argument 24: PDUTy is the ratio of the positive pulse width to the signal period expressed as a percentage.

Argument 25: PEAKHits (TDS 500D & 700D) displays the number of acquired points in the largest bin of the histogram.

Argument 26: PERIod is the time, in seconds, it takes for one complete signal cycle to happen.

Argument 27: PHAse is the phase difference from the selected waveform to the designated waveform.

Argument 28: PK2pk is the absolute difference between the maximum and minimum amplitude. It can be used with both general purpose and histogram measurements. For histogram measurements, it is the value of the highest nonzero bin minus the value of the lowest nonzero bin.

Argument 29: POVershoot is the positive overshoot, expressed as:

Argument 30: PWIdth is the distance (time) between MidRef (usually 50%) amplitude points of a positive pulse.

Argument 31: RISe is the time that it takes for the leading edge of a pulse to rise from a low reference value to a high reference value of its final value.

Argument 32: RMS is the true Root Mean Square voltage.

Argument 33: SIGMA[1-3] (TDS 500D & 700D) is the percentage of points in the histogram which are within 1, 2, or 3 standard deviations of the histogram mean. In a normal distribution, 66% of the points are within +-1 standard deviation, 95% of the points are within +- 2 standard deviations, and 97.5% of the points are within +-3 standard deviations of the histogram mean.

Argument 34: STDdev (TDS 500D & 700D) is the standard deviation of all acquired points within or on the histogram box.

Argument 35: WAVEFORMS (TDS 500D & 700D) is the number of waveforms that have contributed to the histogram. A waveform is counted even if no points were within the histogram box.

Example: `MEASUREMENT:IMMED:TYPE FREQUENCY`
defines the immediate measurement to be a frequency measurement.

MEASUrement:IMMed:UNITS?

(Query Only)

Description: Returns the units for the immediate measurement.

Group: Measurement

Related Command: MEASUrement:IMMed:TYPe

Syntax: MEASUrement:IMMed:UNITS?

Return Value: <QString>
returns "V"
for volts, "s"
for seconds, "Hz"
for hertz, "VV"
for volts2,
or "%" for percent.

On the TDS 400A, <Qstring> also returns “c” for clocks, “Vc” fr volt-locks, or “/c? for 1/clks.

Example: `MEASUREMENT:IMMED:UNITS?`
might return `"s"`, indicating that the units for the immediate measurement are seconds.

MEASUrement:IMMed:VALue?

(Query Only)

Description: Immediately executes the immediate measurement specified by the MEASUrement:IMMed:TYPe command. The measurement is taken on the source(s) specified by a MEASUrement:IMMed:SOURCE command.

Group: Measurement

Syntax: MEASUrement:IMMed:VALue?

Return Value: <NR3>

MEASUrement:MEAS<x>?

(Query Only)

Description: Returns all measurement parameters for the displayed measurement specified by <x>.

Group: Measurement

Syntax: MEASUrement:MEAS<x>?

Example: `MEASUREMENT:MEAS3?`
might return:
`:MEASUREMENT:MEAS3:STATE 0;TYPE PERIOD;UNITS "s";SOURCE1 CH1;SOURCE2 CH2;DELAY:EDGE1 RISE;EDGE2 RISE;DIRECTION FORWARDS.`

MEASUrement:MEAS<x>:COUNt?

(TDS 500D & 700D Only) (Query Only)

Description: Returns the number of values accumulated for this measurement since the last statistical reset. Some values may have been ignored because they generated an error.

Group: Measurement

Syntax: MEASUrement:MEAS<x>:COUNt?

Returns: <NR3>

Example: `MEASUREMENT:MEAS1:COUNT?`
might return: `:MEASU:MEAS1:COUN 3.247000000E+3`

MEASUrement:MEAS<x>:DELay?

(Query Only)

Description: Returns the delay measurement parameters for the measurement specified by <x>.

Group: Measurement

Syntax: MEASUrement:MEAS<x>:DELay?

Example: `MEASUREMENT:MEAS3:DELAY?`
might return `:MEASUREMENT:MEAS3:DELAY:EDGE1 RISE;EDGE2 RISE;DIRECTION FORWARDS`.

MEASUrement:MEAS<x>:DELay:DIREction

Description: Sets or queries the starting point and direction that determines the delay "to" edge when taking a delay measurement. The waveform is specified by MEASUrement:MEAS<X>:SOURCE2. Sending this command is equivalent to setting the direction in the **Delay Edges & Direction** side menu.

Group: Measurement

Syntax 1: MEASUrement:MEAS<x>:DELay:DIREction { BACkwards **|** FORWards }

Syntax 2: MEASUrement:MEAS<x>:DELay:DIREction?

Argument 1: BACkwards means that the search starts at the end of the waveform and looks for the last rising or falling edge in the waveform. The slope of the edge is specified by MEASUrement:MEAS<x>:DELay:EDGE2.

Argument 2: FORWards means that the search starts at the beginning of the waveform and looks for the first rising or falling edge in the waveform. The slope of the edge is specified by MEASUrement:MEAS<x>:DELay:EDGE2.

Example 1: `MEASUREMENT:MEAS1:DELAY:DIRECTION BACKWARDS`
starts searching from the end of the waveform record.

Example 2: `MEASUREMENT:MEAS3:DELAY:DIRECTION?`
might return `FORWARDS` for the search direction.

MEASUrement:MEAS<x>:DELay:EDGE1

Description: Sets or queries the slope of the edge that is used for the delay "from" waveform when taking a delay measurement. The waveform is specified by MEASUrement:MEAS<x>:SOURCE1. Sending this command is equivalent to selecting the edges in the **Delay Edges & Direction** side menu.

Group: Measurement

Syntax 1: MEASUrement:MEAS<x>:DELay:EDGE1 { FALL | RISe }

Syntax 2: MEASUrement:MEAS<x>:DELay:EDGE1?

Argument 1: FALL specifies the falling edge.

Argument 2: RISe specifies the rising edge.

Example 1: MEASUREMENT:MEAS3:DELAY:EDGE1 RISE
specifies that the rising edge be used for measurement 3.

Example 2: `MEASUREMENT:MEAS1:DELAY:EDGE1?`
returns either `RISE` or `FALL` for measurement 1.

MEASUrement:MEAS<x>:DELay:EDGE2

Description: Sets or queries the slope of the edge that is used for the delay "to" waveform when taking a delay measurement. The waveform is specified by MEASUrement:MEAS<x>:SOURCE2. Sending this command is equivalent to selecting the edges in the **Delay Edges & Direction** side menu.

Group: Measurement

Syntax 1: MEASUrement:MEAS<x>:DELay:EDGE2 { FALL | RISe }

Syntax 2: MEASUrement:MEAS<x>:DELay:EDGE2?

Argument 1: FALL specifies the falling edge.

Argument 2: RISe specifies the rising edge.

Example 1: `MEASUREMENT:MEAS2:DELAY:EDGE2 RISE`
specifies that the rising edge be used for the second delay measurement.

Example 2: `MEASUREMENT:MEAS2:DELAY:EDGE2?`
might return `FALL` showing that the falling or negative edge of the waveform is used for the second measurement.

MEASUrement:MEAS<x>:MAXimum?

(TDS 500D & 700D Only) (Query Only)

Description: Returns the maximum value found for this measurement since the last statistical reset.

Group: Measurement

Syntax: MEASUrement:MEAS<x>:MAXimum?

Returns: <NR3>

Example: `MEASUREMENT:MEAS3:MAXIMUM?`
might return: `:MEASU:MEAS3:MAX 4.18E-9`

MEASUrement:MEAS<x>:MEAN?

(TDS 500D & 700D Only) (Query Only)

Description: Returns the mean value accumulated for this measurement since the last statistical reset.

Group: Measurement

Syntax: MEASUrement:MEAS<x>:MEAN?

Returns: <NR3>

Example: `MEASUREMENT:MEAS1:MEAN?`
might return: `:MEASU:MEAS1:MEAN 514.71E-9`

MEASUrement:MEAS<x>:MINImum?

(TDS 500D & 700D Only) (Query Only)

Description: Returns the minimum value found for this measurement since the last statistical reset.

Group: Measurement

Syntax: MEASUrement:MEAS<x>:MINImum?

Returns: <NR3>

Example: `MEASUREMENT:MEAS1:MINIMUM?`
might return: `:MEASU:MEAS1:MINI 1.75E-9`

MEASUrement:MEAS<x>:SOURCE[1]

Description: Sets or queries the source for all single channel measurements and specifies the source to measure "from" when taking a delay measurement or phase measurement.

Group: Measurement

Syntax 1: MEASUrement:MEAS<x>:SOURCE[1] { CH<x> **|** MATH<x> **|** REF<x> **|** HIStogram (TDS 500D and 700D) }

Syntax 2: MEASUrement:MEAS<x>:SOURCE[1]?

Argument 1: CH<x> is an input channel.

Argument 2: MATH<x> is a math waveform.

Argument 3: REF<x> is a reference waveform.

Argument 4: (TDS 500D & 700D) HIStogram is a histogram

Example: `MEASUREMENT:MEAS2:SOURCE1 MATH1`
specifies MATH1 as the measurement 2 source.

MEASUrement:MEAS<x>:SOURCE2

Description: Sets or queries the source to measure "to" when taking a delay measurement or phase measurement. Sending this command is equivalent to setting the source in the **Delay from Selected Wfm** side menu or the Phase from Selected Wfm side menu.

Group: Measurement

Syntax 1: MEASUrement:MEAS<x>:SOURCE2 { CH<x> | MATH<x> | REF<x> | HIStogram (TDS 500D and 700D) }

Syntax 2: MEASUrement:MEAS<x>:SOURCE2?

Argument 1: CH<x> is an input channel.

Argument 2: MATH<x> is a math waveform.

Argument 3: REF<x> is a reference waveform.

Argument 4: (TDS 500D & 700D) HIStogram is a histogram

Example 1: `MEASUREMENT:MEAS4:SOURCE2 CH<x>`
sets channel 1 as the delay "to" source when making delay measurements.

Example 2: `MEASUREMENT:MEAS2:SOURCE2?`
might return `MATH1`.

MEASUrement:MEAS<x>:STATE

Description: Controls the measurement system. The source specified by MEASUrement:MEAS<x>:SOURCE1 must be selected for the measurement to be displayed. The source can be selected using the SELect:CH<x> command.

Group: Measurement

Syntax 1: MEASUrement:MEAS<x>:STATE { OFF **|** ON **|** <NR1> }

Syntax 2: MEASUrement:MEAS<x>:STATE?

Argument 1: OFF or <NR1> = 0 turns measurements off. You can also turn the state off by deselecting the source.

Argument 2: ON or <NR1> NOT = 0 turns measurements on.

Example 1: `MEASUREMENT:MEAS1:STATE ON`
turns measurement defined as MEAS1 on.

Example 2: `MEASUREMENT:MEAS4:STATE?`
returns either `0` or `1`, indicating the state of MEAS4.

MEASUrement:MEAS<x>:STDdev?

(TDS 500D & 700D Only)

Description: Returns the standard deviation of values accumulated for this measurement since the last statistical reset.

Group: Measurement

Syntax: MEASUrement:MEAS<x>:STDdev?

Returns: <NR3>

Example: `MEASUREMENT:MEAS1:STDDEV?`
might return: `:MEASU:MEAS1:STD 21.0E-12`

MEASUrement:MEAS<x>:TYPe

Description: Sets or queries the measurement type for the measurement specified by MEAS<x>. Sending this command is equivalent to selecting the measurement in the **Select Measurement** side menu.

Group: Measurement

Syntax 1: MEASUrement:MEAS<x>:TYPe { AMPlitude | AREA | BURst | CARea | CMEan | CRMs | DELay | EXTINCTDB (TDS 500D & 700D) | EXTINCTPCT (TDS 500D & 700D) | EXTINCTRATIO (TDS 500D & 700D) | FALL | FREQuency | HIGH | HITS (TDS 500D & 700D) | LOW | MAXimum | MEAN | MEANDBM (TDS 500D & 700D) | MEDian (TDS 500D & 700D) | MINImum | NDUty | NOVershoot | NWIdth | PDUTy | PEAKHits (TDS 500D & 700D) | PERIod | PHAse | PK2pk | POVershoot | PWIdth | RISe | RMS | SIGMA[1-3] (TDS 500D & 700D) STDdev (TDS 500D & 700D) | WAVEFORMS (TDS 500D & 700D) }

Syntax 2: MEASUrement:MEAS<x>:TYPe?

Argument 1: AMPlitude is the high value minus the low value or HIGH - LOW.

Argument 2: AREA is the area between the curve and ground over the entire waveform.

Argument 3: BURst is the time from the first MidRef crossing to the last MidRef crossing.

Argument 4: CARea (cycle area) is the area between the curve and ground over one cycle.

Argument 5: CMEan is the arithmetic mean over one cycle.

Argument 6: CRMs is the true Root Mean Square voltage over one cycle.

Argument 7: DELay is the time between the MidRef crossings of two different waveforms.

Argument 8: EXTINCTDB (TDS 500D & 700D) is 10.0*log(10)(Extinction Rati0

Argument 9: EXTINCTPCT (TDS 500D & 700D) is 100/Extinction Ration

Argument 10: EXTINCTRATIO (TDS 500D & 700D) is High/Low

Argument 11: FALL is the time that it takes for the falling edge of a pulse to fall from a HighRef value to a LowRef value of its final value.

Argument 12: FREQuency is the reciprocal of the period measured in Hertz.

Argument 13: HIGH is the 100% reference level.

Argument 14: HITS (TDS 500D & 700D) displays the number of acquired points within or on the histogram box.

Argument 15: LOW is the 0% reference level.

Argument 16: MAXimum is the highest amplitude (voltage).

Argument 17: MEAN is the arithmetic mean over the entire waveform.

Argument 18: MEANDBM (TDS 500D & 700D) is the median of all acquired points within or on the histogram box. Half of the acquired points are greater than and half less than this value.

Argument 19: MEDian (TDS 500D & 700D) is the median of all acquired points within or on the histogram box. Half of the acquired points are greater than and half less than this value.

Argument 20: MINImum is the lowest amplitude (voltage).

Argument 21: NDUty is the ratio of the negative pulse width to the signal period expressed as a percentage.

Argument 22: NOVershoot is the negative overshoot, expressed as:

NOVershoot = 100 x ((Low - Minimum)/Amplitude)

Argument 23: NWIdth is the distance (time) between MidRef (usually 50%) amplitude points of a negative pulse.

Argument 24: PDUTy is the ratio of the positive pulse width to the signal period expressed as a percentage.

Argument 25: PEAKHits (TDS 500D & 700D) displays the number of acquired points in the largest bin of the histogram.

Argument 26: PERIod is the time, in seconds, it takes for one complete signal cycle to happen.

Argument 27: PHAse is the phase difference from the selected waveform to the designated waveform.

Argument 28: PK2pk is the absolute difference between the maximum and minimum amplitude.

Argument 29: POVershoot is the positive overshoot, expressed as:

Argument 30: PWIdth is the distance (time) between MidRef (usually 50%) amplitude points of a positive pulse.

Argument 31: RISe is the time that it takes for the leading edge of a pulse to rise from a low reference value to a high reference value of its final value.

Argument 32: RMS is the true Root Mean Square voltage.

Argument 33: SIGMA[1-3] (TDS 500D & 700D) is the percentage of points in the histogram which are within 1, 2, or 3 standard deviations of the histogram mean. In a normal distribution, 66% of the points are within +-1 standard deviation, 95% of the points are within +- 2 standard deviations, and 97.5% of the points are within +-3 standard deviations of the histogram mean.

Argument 34: STDdev (TDS 500D & 700D) is the standard deviation of all acquired points within or on the histogram box.

Argument 35: WAVEFORMS (TDS 500D & 700D) is the number of waveforms that have contributed to the histogram. A waveform is counted even if no points were within the histogram box.

Example: `MEASUREMENT:MEAS3:TYPE RMS`
specifies MEAS3 to calculate the Root Mean Square voltage.

MEASUrement:MEAS<x>:UNITS?

(Query Only)

Description: Returns the units for the measurement specified by MEASUrement:MEAS<x>:TYPe.

Group: Measurement

Syntax: MEASUrement:MEAS<x>:UNITS?

Return Value: <QString>
returns "V" for volts,
returns "s" for seconds,
returns "HZ" for hertz,
returns "VV" for volts2, or
returns "%" for percent.

Example: `MEASUREMENT:MEAS3:UNITS?`
might return `"%"`, indicating the units for Measurement 3 are percent.

MEASUrement:MEAS<x>:VALue?

(Query Only)

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

NOTE: This value is a display value and will be updated perhaps every 1/3 second. If you are acquiring a long acquisition record, the TDS may take longer than this time to update.

Group: Measurement

Syntax: MEASUrement:MEAS<x>:VALue?

Returns: <NR3>

MEASUrement:METHod

Description: Sets or queries the method used to calculate the 0% and 100% reference level. Sending this command is equivalent to setting the **High-Low Setup** in the Measure menu.

Group: Measurement

Syntax 1: MEASUrement:METHod { HIStogram | MINMax }

Syntax 2: MEASUrement:METHod?

Argument 1: HIStogram sets the high and low waveform levels statistically using a histogram algorithm.

Argument 2: MINMax sets the high and low waveform levels to MAX and MIN, respectively.

Example 1: `MEASUREMENT:METHOD HISTOGRAM`
specifies that the high and low reference levels are set statistically.

Example 2: `MEASUREMENT:METHOD?`
returns `MINMAX` when the reference levels are set to MIN and MAX.

MEASUrement:REFLevel?

(Query Only)

Description: Returns the reference levels.

Group: Measurement

Syntax: MEASUrement:REFLevel?

MEASUrement:REFLevel:ABSolute:HIGH

Description: Sets or queries the high reference level, and is the 100% reference level when MEASUrement:REFLevel:METHod is set to ABSolute. Sending this command is equivalent to setting the **Level Setup** in the Measure menu.

Group: Measurement

Syntax 1: MEASUrement:REFLevel:ABSolute:HIGH <NR3>

Syntax 2: MEASUrement:REFLevel:ABSolute:HIGH?

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

Example: `MEASUREMENT:REFLEVEL:ABSOLUTE:HIGH 1.71`
sets the high reference level to 1.71 V.

MEASUrement:REFLevel:ABSolute:LOW

Description: Sets or queries the low reference level, and is the 0% reference level when MEASUrement:REFLevel:METHod is set to ABSolute. Sending this command is equivalent to setting the **Level Setup** in the Measure menu.

Group: Measurement

Syntax 1: MEASUrement:REFLevel:ABSolute:LOW <NR3>

Syntax 2: MEASUrement:REFLevel:ABSolute:LOW?

Argument: <NR3> is the low reference level, in volts. The default is 0.0 V.

Example: `MEASUREMENT:REFLEVEL:ABSOLUTE:LOW?`
might return `0.0E+0` as the low reference level.

MEASUrement:REFLevel:ABSolute:MID

Description: Sets or queries the mid reference level, and is the 50% reference level when MEASUrement:REFLevel:METHod is set to ABSolute. Sending this command is equivalent to setting the **Level Setup** in the Measure menu.

Group: Measurement

Syntax 1: MEASUrement:REFLevel:ABSolute:MID <NR3>

Syntax 2: MEASUrement:REFLevel:ABSolute:MID?

Argument: <NR3> is the mid reference level, in volts. The default is 0.0 V.

Example: `MEASUREMENT:REFLEVEL:ABSOLUTE:MID .71`
sets the mid reference level to .71 volts.

MEASUrement:REFLevel:ABSolute:MID2

Description: Sets or queries the mid reference level for the "to" waveform when taking a delay measurement, and is the 50% reference level when MEASUrement:REFLevel:METHod is set to ABSolute. Sending this command is equivalent to setting the **Level Setup** in the Measure menu.

Group: Measurement

Syntax 1: MEASUrement:REFLevel:ABSolute:MID2 <NR3>

Syntax 2: MEASUrement:REFLevel:ABSolute:MID2?

Argument: <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 volts.

MEASUrement:REFLevel:METHod

Description: Specifies which reference levels are used for measurement calculations. Sending this command is equivalent to setting the levels in the **Level Setup** side menu.

Group: Measurement

Syntax 1: MEASUrement:REFLevel:METHod { ABSolute **|** PERCent }

Syntax 2: MEASUrement:REFLevel:METHod?

Argument 1: ABSolute specifies that the reference levels are set explicitly using the MEASUrement:REFLevel:ABSolute commands. This method is useful when precise values are required. For instance, when designing to published interface specifications such as RS-232-C.

Argument 2: PERCent specifies that the reference levels are calculated as a percent relative to HIGH and LOW. The percentages are defined using the MEASUrement:REFLevel:PERCent commands.

Example 1: `MEASUREMENT:REFLEVEL:METHOD ABSolute`
specifies that explicit user-defined values are used for the reference levels.

Example 2: `MEASUREMENT:REFLEVEL:METHOD?`
returns either `ABSOLUTE` or `PERCENT`, indicating the reference levels used.

MEASUrement:REFLevel:PERCent:HIGH

Description: Sets or queries the percent, where 100% is equal to HIGH, that is used to calculate the high reference level when MEASUrement:REFLevel:METHod is set to PERCent. Sending this command is equivalent to setting the **Level Setup** in the Measure menu.

Group: Measurement

Syntax 1: MEASUrement:REFLevel:PERCent:HIGH <NR3>

Syntax 2: MEASUrement:REFLevel:PERCent:HIGH?

Argument 1: <NR3> ranges from 0 to 100 percent, and is the high reference level. The default is 90%.

Example: `MEASUREMENT:REFLEVEL:PERCENT:HIGH 95`
specifies that the high reference level is set to 95% of HIGH.

MEASUrement:REFLevel:PERCent:LOW

Description: Sets or queries the percent, where 100% is equal to HIGH, that is used to calculate the low reference level when MEASUrement:REFLevel:METHod is set to PERCent. Sending this command is equivalent to setting the **Level Setup** in the Measure menu.

Group: Measurement

Syntax 1: MEASUrement:REFLevel:PERCent:LOW <NR3>

Syntax 2: MEASUrement:REFLevel:PERCent:LOW?

Argument: <NR3> ranges from 0 to 100 percent, and is the low reference level. The default is 10%.

Example: `MEASUREMENT:REFLEVEL:PERCENT:LOW?`
might return 15, meaning that the low reference level is 15% of HIGH.

MEASUrement:REFLevel:PERCent:MID

Description: Sets or queries the percent, where 100% is equal to HIGH, that is used to calculate the mid reference level when MEASUrement:REFLevel:METHod is set to PERCent. Sending this command is equivalent to setting the **Level Setup** in the Measure menu.

Group: Measurement

Syntax 1: MEASUrement:REFLevel:PERCent:MID <NR3>

Syntax 2: MEASUrement:REFLevel:PERCent:MID?

Argument: <NR3> ranges from 0 to 100 percent, and is the mid reference level. The default is 50%.

Example: `MEASUREMENT:REFLEVEL:PERCENT:MID 60`
specifies that the mid reference level is set to 60% of HIGH.

MEASUrement:REFLevel:PERCent:MID2

Description: Sets or queries the percent, where 100% is equal to HIGH, that is used to calculate the mid reference level for the second waveform specified when taking a delay measurement. Sending this command is equivalent to setting the **Level Setup** in the Measure menu.

Group: Measurement

Syntax 1: MEASUrement:REFLevel:PERCent:MID2 <NR3>

Syntax 2: MEASUrement:REFLevel:PERCent:MID2?

Argument: <NR3> ranges from 0 to 100 percent, and is the mid reference level. The default is 50%.

Example: `MEASUREMENT:REFLEVEL:PERCENT:MID2 40`
specifies that the mid reference level is set to 40% of HIGH.

MEASUrement:SNAPShot

Description: Displays the measurement snapshot.

Group: Measurement

Syntax: MEASUrement:SNAPShot

Example: `MEASUREMENT:SNAPSHOT`

MEASUrement:STATIstics:MODE

Description: Controls the operation and display of measurement statistics.

Group: Measurement

Syntax 1: MEASUrement:STATIstics:MODE { OFF | MINMax | MEANSTDdev }

Syntax 2: MEASUrement:STATIstics:MODE?

Argument 1: OFF turns measurements off. This is the default value.

Argument 2: MINMax turns on statistics and displays the min and max statistics.

Argument 3: MEANSTDdev turns on statistics and displays the mean and standard deviation statistics.

Example: `MEASUREMENT:STATISTICS:MODE MINMAX`
turns on statistics and displays the min and max statistics.

MEASUrement:STATIstics:WEIghting

Description: Controls the responsiveness of mean and standard deviation to waveform changes.

Group: Measurement

Syntax 1: MEASUrement:STATIstics:WEIghting <NR1>

Syntax 2: MEASUrement:STATIstics:WEIghting?

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

Example: `MEASUREMENT:STATISTICS:WEIGHTING 4`
sets the weighting to 4.

MESSage

Description: Clears the message window and the MESSage? query returns the current message parameters.

Group: Display

Syntax 1: MESSage CLEar

Syntax 2: MESSage?

Argument: CLEar removes the message from the message window. This is equivalent to sending MESSage SHOW "".

Example: `MESSAGE CLEAR`
clears the message from the window.

MESSage:BOX

Description: Defines the size and position of the message window. This command does not display the window unless MESSage:STATE is ON.

Group: Display

Syntax 1: MESSage:BOX <X1>,<Y1>,<X2>,<Y2>

Syntax 2: MESSage:BOX?

Argument 1: <X1> and <X2> = 0 to 640, and are pixel positions along the horizontal axis. <X1> defines the left and <X2> defines the right side of the window.

Argument 2: <Y1> and <Y2> = 0 to 480, and are pixel positions along the vertical axis. <Y1> defines the top and <Y2> defines the bottom of the window. The reserved height of all characters is 15 pixels so the window must be at least that high to fully display characters. Shorter windows clip characters.

MESSage:SHOW

Description: Clears the contents of the message window and displays the new message in the window.

Group: Display

Syntax 1: MESSage:SHOW <QString>

Syntax 2: MESSage:SHOW?

Return Value: <QString> is the message and can include any legal TDS character. The maximum length of the message is 1000 characters.

The message is left-justified, and is displayed on a single line starting with the top most line in the window. A line feed character can be embedded in the string to position the message on multiple lines. You can also use white space and tab characters to position the message within a line.

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

The ESC character followed by the @ character turns inverse video on or off and can be embedded in the message string. The first ESC character followed by a @ character displays all the text that follows in inverse video until another ESC character followed by a @ character is found in the string.

NOTE: The use of any escape codes other than those described above may produce unpredictable results.

The label area is the height and width you have set using the MESSage:Box command. The length of the label that fits in the label area depends on the contents of the label because the width of characters varies.

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

Example 1: `MESSAGE:SHOW "Hello world"`
displays "Hello world" in the upper left corner of the box (you can define the box size with the MESSAGE BOX command).

Example 2: `MESSAGE:SHOW "Hello >@world>@ ... hello`
displays "Hello world ... hello" in the upper left corner of the box and the word "world" is displayed in inverse video. In this example, > stands for the escape character. The escape character may appear differently for you depending on your GPIB talker-listener program.

MESSage:STATE

Description: Controls the display of the message window.

Group: Display

Syntax 1: MESSage:STATE { OFF | ON | <NR1> }

Syntax 2: MESSage:STATE?

Argument 1: <OFF> or <NR1> = 0 removes the message window from the screen.

Argument 2: <ON> or <NR1> NOT = 0 displays the message window and its contents on the screen. The size of the window is defined by MESSage:BOX.

NEWpass

(No Query Form)

Description: Changes the password that enables access to password protected data. The PASSWord command must be successfully executed before using this command or an execution error will be generated.

Group: Miscellaneous

Related Commands: PASSWord, *PUD

Syntax: NEWpass <QString>

Argument: <QString> is the new password. The password can include up to 10 characters.

Example: `NEWPASS "mypassword"`
creates a new password for accessing the user protected data.

*OPC

Description: Generates the operation complete message in the Standard Event Status Register (SESR) when all pending operations finish. The *OPC? query places the ASCII character "1" into the output queue when all pending operations are finished. The *OPC? response is not available to read until all pending operations finish. For a complete discussion of the use of these registers and the output queue, see page .

Group: Status and Error

Related Commands: BUSY?, *WAI

Syntax 1: *OPC

Syntax 2: *OPC?

The *OPC command allows you to synchronize the operation of the oscilloscope with your application program. For more information, refer to the section on Synchronization Methods.

Table 2-31: Commands that Generate an Operation Complete Message

Operation	Command
Automatic scope adjustment	AUTOSet EXECute
Internal self-calibration	*CAL
Single sequence acquisition	ACQuire:STATE ON or ACQuire:STATE RUN (when ACQuire:STOPAfter is set to SEQuence)
Hardcopy output	HARDCopy STARt

*OPT?

(TDS 510A, 500D, 600B, & 700D) (Query Only)

Description: Returns a list of options installed in your oscilloscope.

Group: Status and Error

Syntax: *OPT?

Example: `OPT?`
Might return: `13:Rs232/cent,1M:extended record length, 2F:math pack,05:video trigger,0,CD:color display.`

PASSWord

(No Query Form)

Description: Enables the *PUD and NEWpass set commands. Sending PASSWord without any arguments disables these same commands. Once the password is successfully entered, the *PUD and NEWpass commands are enabled until the oscilloscope is powered off, or until the FACtory command, the PASSWord command with no arguments, or the *RST command is issued.

To change the password, you must first enter the valid password with the PASSWord command and then change to your new password with the NEWpass command. Remember that the password is case sensitive.

Group: Miscellaneous

Related Commands: NEWpass, *PUD

Syntax 1: PASSWord

Syntax 2: PASSWord <QString>

Argument: <QString> is the password and can include up to 10 characters. The factory default password is "XYZZY"and is always valid.

Example 1: `PASSWORD "XYZZY"`
Enables the *PUB and NEWpass set commands.

Example 2: `PASSWORD`
Disables the *PUB and NEWpass set commands. You can still use the query version of *PUB.

*PSC

Description: Sets and queries the power-on status flag that controls the automatic power-on handling of the DESER, SRER, and ESER registers. When *PSC is true, the DESER register is set to 255 and the SRER and ESER registers are set to 0 at power-on. When *PSC is false, the current values in the DESER, SRER, and ESER registers are preserved in non-volatile memory when power is shut off and are restored at power-on. For more information, refer to the section on Registers.

Group: Status and Error

Related Commands: DESE, *ESE, FACtory, *RST, *SRE

Syntax 1: *PSC <NR1>

Syntax 2: *PSC?

Argument 1: <NR1> = 0 sets the power-on status clear flag to false, disables the power-on clear and allows the oscilloscope to possibly assert SRQ after power-on.

Argument 2: <NR1> NOT = 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. Using an out-of-range value causes an execution warning.

Example 1: `*PSC 0`
sets the power-on status clear flag to false.

Example 2: `*PSC?`
might return the value 1, showing that the power-on status clear flag is set to true.

*PUD

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

Group: Miscellaneous

Related Commands: PASSWord

Syntax 1: *PUD <Block>

Syntax 2: *PUD?

Argument: <Block> is a string containing up to 100 characters.

Example 1: `*PUD #229This instrument belongs to me`
stores the string "This instrument belongs to me" in the user protected data area.

Example 2: `*PUD?`
might return `#221Property of Company X`.

*RCL

(No Query Form)

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

Group: Save and Recall

Related Commands: DELEte:SETUp, FACtory, *LRN?, RECAll:SETUp, *RST, *SAV, SAVe:SETUp

Syntax: *RCL <NR1>

Argument: <NR1> is a value in the range from 1 to 10, and specifies a setup storage location. Using an out-of-range value causes an execution error (222, "Data out of range").

Example: `*RCL 3`
restores the oscilloscope from a copy of the settings stored in memory location 3.

RECAll:ACQDATA

(TDS 500D & 700D) (No Query Form)

Description: Replaces the indicated channel's live acquisition data with that saved in the indicated file.

Group: Save and Recall

Syntax: RECAll:ACQDATA { <file path>, CH<x> }

Argument 1: <file path> (available on instruments with the Option 1F File System) is the location in mass storage memory where the setup will be recalled from.

<file path> is a quoted string that defines the file name and path. Input the file path using the form <drive>/<dir>/<filename>. <drive> and one or more <dir>s are optional. If you do not specify them, the TDS will read the file from the current directory. <filename> stands for a filename of up to 8 characters followed by a period (".") and any 3-char extension. Do not use wild card characters.

The current directory refers to the name of a directory as returned by the FILESystem:CWD command.

Argument 2: CH <x> is an input channel

Example 1: `RECALL:ACQDATA “hd0:/MYFILE.WF1”,CH2`
replaces CH2's data with that stored in `hd0:/MYFILE.WF1`.

RECAll:IMAGEHistogram

(TDS 500D & 700D) (No Query Form) (File System Only)

Description: Recalls a stored image histogram from disk. DPO mode must be selected, not Normal mode. Sending this command is equivalent to selecting **Recall to Image Histogram** in the Save/Recall Waveform menu. Recalling an image histogram stops acquisitions in process.

Group: Save and Recall

Syntax: RECAll:IMAGEHistogram <file path>

Argument 1: <file path> is a quoted string that defines the file name and path. Input the file path using the form <drive>/<dir>/<filename>. <drive> and one or more <dir>s are optional. If you do not specify them, the TDS will recall the waveform from the default directory. <filename> stands for a filename of up to 8 characters followed by a period (".") and any 3-char extension. Do not use wild card characters.

The default directory refers to the name of a directory as returned by the FILESystem:CWD command.

Example: `RECALL:IMAGEHISTOGRAM "TEK00000.IMG"`
recalls the image histogram stored in the file named TEK00000.IMG.

RECAll:SETUp

(No Query Form)

Description: Restores a stored or factory front-panel setup of the oscilloscope. Sending this command is equivalent to selecting **Recall Saved Setup** or **Recall Factory Setup** or **Recall Current Setup** in the Save/Recall Setup menu.

Group: Save and Recall

Related Commands: DELEte:SETUp, FACtory, *RCL, *RST, *SAV, SAVe:SETUp

Syntax: RECAll:SETUp { FACtory **|** <NR1> **|** <file path> }

Argument 1: FACtory selects the factory setup.

Argument 2: <NR1> is a value in the range from 1 to 10 and specifies a setup storage location. Using an out-of-range value causes an execution error (222, "Data out of range").

Argument 3: <file path> (available on instruments with the Option 1F File System) is the location in mass storage memory where the setup will be recalled from.

<file path> is a quoted string that defines the file name and path. Input the file path using the form <drive>/<dir>/<filename>. <drive> and one or more <dir>s are optional. If you do not specify them, the TDS will read the file from the default directory. <filename> stands for a filename of up to 8 characters followed by a period (".") and any 3-char extension. Do not use wild card characters.

The default directory refers to the name of a directory as returned by the FILESystem:CWD command.

Example 1: `RECALL:SETUP FACTORY`
recalls (and makes current) the front panel setup to its factory defaults.

Example 2: `RECALL:SETUP 1`
recalls the front panel setup from setup1.

Example 3: `RECALL:SETUP "TEK00000.SET"`
recalls the front panel setup from the file TEK00000.SET in the default directory and on the default drive.

RECAll:WAVEform

(No Query Form) (File System Only)

Description: Recalls a stored waveform into a reference location. Sending this command is equivalent to selecting **Recall File** in the Save/Recall Waveform menu.

Group: Save and Recall

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

Argument 1: REF<x> is the location in internal reference memory where the waveform is recalled from.

Argument 2: <file path> is a quoted string that defines the file name and path. Input the file path using the form <drive>/<dir>/<filename>. <drive> and one or more <dir>s are optional. If you do not specify them, the TDS will recall the waveform from the default directory. <filename> stands for a filename of up to 8 characters followed by a period (".") and any 3-char extension. Do not use wild card characters.

The default directory refers to the name of a directory as returned by the FILESystem:CWD command.

Example: `RECALL:WAVEFORM "TEK00000.WFM",REF1`
recalls the waveform stored in the file named TEK00000.WFM to reference location 1.

REM

(No Query Form)

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

Group: Miscellaneous

Syntax: REM <QString>

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

Example: `REM "This is a comment"`
is ignored by the instrument.

*RST

(No Query Form)

Description: (Reset) returns the oscilloscope to a known set of instrument settings, but does not purge any aliases or stored settings.

Group: Status and Error

Related Commands: FACtory, *PSC, *RCL, RECAll:SETUp, *SAV, SAVe:SETUp

Syntax: *RST

*RST does the following:

o Returns the instrument settings to the factory defaults (see Appendix C: Factory Initialization Settings)

The *RST command does not alter the following:

- The state of the IEEE Std 488.1-1987 interface
- The selected IEEE Std 488.1-1987 address of the oscilloscope
- Calibration data that affect device specifications
- The Output Queue
- The Service Request Enable Register setting
- The Standard Event Status Enable Register setting
- The Power-on status clear flag setting
- Alias definitions
- Stored settings
- The *PUD? response

RS232?

(RS-232/Centronics Hardcopy Interface Only) (Query Only)

Description: Queries the RS232 settings.

Group: RS232

Syntax: RS232?

Example: `RS232?`
might return:
`RS232 BAUD: 9600, SOFTFLAGGING: OFF, HARDFLAGGING: ON, PARITY: NONE, STOPBITS: 1`

RS232:BAUd

(RS-232/Centronics Hardcopy Interface Only)

Description: Sets or queries RS-232-C interface transmission speed.

Group: RS232

Syntax 1: RS232:BAUd <NR1>

Syntax 2: RS232:BAUd?

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

Example: `RS232:BAUD 9600`
sets the transmission rate to 9600 baud.

RS232:HARDFlagging

(RS-232/Centronics Hardcopy Interface Only)

Description: Sets or queries the input and output hard flagging over the RS-232 port. It uses the RFR (Ready For Receive) and CTS (Clear To Send) lines to control data transmission. On output, the oscilloscope transmits data only when CTS is asserted. When CTS is not asserted, the oscilloscope stops transmitting data. On input, it asserts RFR until the receive queue is full. Then it unasserts RFR to stop transmission from an external printer. CTS remains unasserted until the receive queue is not full. At that time, CTS is asserted again to restart transmission.

Group: RS232

Syntax 1: RS232:HARDFlagging { ON | OFF | <NR1> }

Syntax 2: RS232:HARDFlagging?

Argument 1: <ON> or <NR1> NOT = 0 turns on hardflagging.

Argument 2: <OFF> or <NR1> = 0 turns off hardflagging.

Example: `RS232:HARDFLAGGING ON`
turns on hard flagging.

RS232:PARity

(RS-232/Centronics Hardcopy Interface Only)

Description: Sets or queries the parity used for all RS-232-C data transfers. Parity adds a bit to the character sequence. When parity is odd or even, the oscilloscope generates the selected parity on output and checks all input against the selected parity. When parity is none, the oscilloscope performs no input parity error checks and generates no output parity.

Group: RS232

Syntax 1: RS232:PARity { EVEN | ODD | NONe }

Syntax 2: RS232:PARity?

Argument 1: EVEN indicates the parity bit is sent with even parity and bytes received are expected to have even parity.

Argument 2: ODD indicates the parity bit is sent with odd parity and bytes received are expected to have odd parity.

Argument 3: NONe indicates that no parity bit is sent and none are expected.

Example: `RS232:PARITY EVEN`
sets the parity to even.

RS232:SOFTFlagging

(RS-232/Centronics Hardcopy Interface Only)

Description: Sets or queries the input and output soft flagging over the RS-232 port. It stops transmitting data any time it receives an XOFF (DC3) character. It sends an XOFF character when its 512 byte input buffer has 80 free bytes. The oscilloscope begins transmitting data again when it receives an XON (DC1) character. It sends XON when its input buffer has 100 free bytes.

Group: RS232

Syntax 1: RS232:SOFTFlagging { ON **|** OFF **|** <NR1> }

Syntax 2: RS232:SOFTFlagging?

Argument 1: ON or <NR1> NOT = 0 turns on softflagging.

Argument 2: OFF or <NR1> = 0 turns off softflagging.

Example: `RS232:SOFTFLAGGING ON`
turns on soft flagging.

RS232:STOPBits

(RS-232/Centronics Hardcopy Interface Only)

Description: Sets or queries the number of transmission stop bits sent with each character to identify the end of data for that character.

Group: RS232

Syntax 1: RS232:STOPBits <NR1>

Syntax 2: RS232:STOPBits?

Argument: <NR1> is 1 or 2.

Example: `RS232:STOPBITS 1`
sets the number of stop bits to 1.

*SAV

(No Query Form)

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

Group: Save and Recall

Related Commands: DELEte:SETUp, FACtory, *RCL, RECAll:SETUp, SAVe:SETUp

Syntax: *SAV <NR1>

Argument: <NR1> is a value in the range from 1 to 10 and specifies a location. Using an out-of-range value causes an execution error. Any settings that have been stored previously at this location will be overwritten.

Example: `*SAV 2`
saves the current settings in memory location 2.

SAVe:ACQDATA

(TDS 500D & 700D) (No Query Form)

Description: Writes out the acquisition data and its characteristics to the indicated file.

When extended-acquisition-length mode is on, this command will only write in the internal data format.

Group: Save and Recall

Syntax: SAVe:ACQDATA { CH<x> , <file path> }

Argument 1: CH <x> is an input channel

Argument 2: <file path> (available on instruments with the Option 1F File System) is the location in mass storage memory where the setup will be saved to.

<file path> is a quoted string that defines the file name and path. Input the file path using the form <drive>/<dir>/<filename>. <drive> and one or more <dir>s are optional. If you do not specify them, the TDS will read the file from the current directory. <filename> stands for a filename of up to 8 characters followed by a period (".") and any 3-char extension. Do not use wild card characters.

The current directory refers to the name of a directory as returned by the FILESystem:CWD command.

Example 1: `SAVE:ACQDATA CH3, “hd0:/MYFILE.WF1”`
saves the CH3 acquisition data to the MYFILE.WF1 file on device hd0.

SAVe:IMAGEHistogram

(TDS 500D & 700D) (No Query Form) (File System Only)

Description: Stores an image histogram in a mass storage file. Sending this command is equivalent to selecting either the **Save Image Histogram** item in the Save/Recall Waveform menu. Saving an image histogram stops acquisitions in process. The file will contain over 400,000 bytes in shallow mode and over 800,000 bytes in deep mode.

Group: Save and Recall

Related Command: RECAll:IMAGEHistogram

Syntax: SAVe:IMAGEHistogram <file path> }

Argument 1: <file path> is a quoted string that defines the file name and path. Input the file path using the form <drive>/<dir>/<filename>. <drive> and one or more <dir>s are optional. If you do not specify them, the oscilloscope will write the file to the current directory. <filename> stands for a filename of up to 8 characters followed by a period (".") and the 3-char extension "IMH".

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

Example 1: `SAVE:IMAGEHISTOGRAM "TEK00000.IMH"`
saves an image histogram to the file TEK00000.IMH in the default directory and on the default drive.

SAVe:SETUp

(No Query Form)

Description: Saves the current front-panel setup into the specified memory location. Sending this is equivalent to selecting the **Save Current Setup** in the Save/Recall Setup menu.

Group: Save and Recall

Related Commands: DELEte:SETUp, FACtory, *RCL, RECAll:SETUp, *SAV

Syntax: SAVe:SETUp { <NR1> **|** <file path> }

Argument 1: <NR1> is a value in the range from 1 to 10 and specifies a location. Using an out-of-range value causes an execution error. Any settings that have been stored previously at this location will be overwritten.

Argument 2: <file path> (available on instruments with the Option 1F File System) is a quoted string that defines the file name and path. Input the file path using the form <drive>/<dir>/<filename>. <drive> and one or more <dir>s are optional. If you do not specify them, the TDS will write the file to the current directory. <filename> stands for a filename of up to 8 characters followed by a period (".") and the 3-char extension "SET". We recommend you use “SET” for the extension to identify files that store setup data.

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

Settings saved in one TDS oscilloscope may or may not work on a different model TDS or on the same model TDS with a different version of firmware.

Example 1: `SAVE:SETUP 5`
saves the current front-panel setup in memory location 5.

Example 2: `SAVE:SETUP "TEK00000.SET"`
saves the current front-panel setup to the file TEK00000.SET in the current directory and on the current drive.

SAVe:WAVEform

(No Query Form)

Description: Stores a waveform in one of four reference memory locations or a mass storage file. Sending this command is equivalent to selecting either the **Save Waveform** or the **Save to File Waveform** item in the Save/Recall Waveform menu.

Group: Save and Recall

Related Command: DELEte:WAVEform

Syntax: SAVe:WAVEform <wfm>,{ REF<x> **|** <file path> }

Argument 1: <wfm> is CH<x>, MATH<x>, or REF<x>, and is the waveform that will be saved.

Argument 2: REF<x> is the location where the waveform will be stored.

Argument 3: <file path> (on instruments with the Option 1F File System) is a quoted string that defines the file name and path. Input the file path using the form <drive>/<dir>/<filename>. <drive> and one or more <dir>s are optional. If you do not specify them, the oscilloscope will write the file to the current directory. <filename> stands for a filename of up to 8 characters followed by a period (".") and the 3-char extension "WFM". The TDS 400A and 700D can also use a "CSV" extension for spreadsheet format files or a "DAT" extension for MathCAD format files.

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

Example 1: `SAVE:WAVEFORM MATH2,REF1`
saves the math 2 waveform in reference memory location 2.

Example 2: `SAVE:WAVEFORM MATH1,"TEK00000.WFM"`
saves the math1 waveform to the file TEK00000.WFM in the default directory and on the default drive.

SAVe:WAVEform:FILEFormat

(Not on TDS 510A)

Description: Specifies the file format for saved waveforms.

Group: Save and Recall

Related Command: SAVE:WAVEform

Syntax 1: SAVe:WAVEform:FILEFormat (INTERNal **|** SPREADSheet **|** MATHCad }

Syntax 2: SAVe:WAVEform:FILEFormat?

Argument 1: INTERNal specifies the internal format. Internal format files have a .wfm extension.

Argument 2: SPREADSheet specifies the spreadsheet format. Spreadsheet format files have a .CSV extension.

Argument 3: MATHCad specifies the MathCad format. MathCad format files have a .DAT extension.

If you are writing a MathCad program, you should be aware that the TDS-MathCad file has the following features:

- ASCII format
- first four values contain header information
- first header value holds the TDS waveform record length
- second header value holds time, in seconds, between samples
- third header value holds the trigger position (expressed as an index in the data position. For extended-acquisition-length records, the index reported can be outside the waveform because the trigger can be outside the waveform. If before the waveform, it’s negative. If after, it’s a large positive)
- fourth header value refers to the fractional trigger position
- delimiters are carriage returns

Example: `SAVE:WAVEFORM:FILEFORMAT SPREADSHEET`
specifies the waveform, when saved, will be stored in a spreadsheet-compatible format.

SELect?

(Query Only)

Description: Returns the selected waveform and the display status of all waveforms.

Group: Vertical

Syntax: SELect?

Example: `SELECT?`
might return `:SELECT:CH1 1;CH2 0;CH3 0;CH4 0;MATH1 0;MATH2 0;MATH3 0;REF1 0;REF2 0;REF3 0;REF4 0`

SELect:<wfm>

Description: Controls the display and selection of waveforms. There can be up to eleven waveforms displayed at one time, but only one waveform can be selected at a time. The selected waveform is the waveform that was most recently turned on. This command is equivalent to pressing a front-panel **CH** or **MORE** button. <wfm> can be CH<x>, MATH<x>, or REF<x> except in extended-acquisition-length and DPO modes where MATH<x> is no used.

Group: Vertical

Syntax 1: SELect:<wfm> { OFF | ON | <NR1> }

Syntax 2: SELect:<wfm>?

Argument 1: OFF or <NR1> = 0 turns off the display of the specified waveform.

Argument 2: ON or <NR1> NOT = 0 turns on the display of the specified waveform. The waveform also becomes the selected waveform.

Example 1: `SELECT:CH2 ON`
turns the channel 2 display on and selects channel 2.

Example 2: `SELECT:REF1?`
returns either `0` or `1`, indicating whether the REF1 waveform is selected.

SELect:CONTROl

Description: Sets or queries the waveform that is currently affected by the cursor and vertical commands.

<wfm> can be CH<x>, MATH<x>, or REF<x> except in extended-acquisition-length and DPO modes where MATH<x> is not used.

Group: Vertical

Syntax 1: SELect:CONTROl <wfm>

Syntax 2: SELect:CONTROl?

Argument: <wfm> is CH<x>, MATH<x>, or REF<x>, and is the selected waveform.

Example: `SELECT:CONTROL?`
might return CH1 as the selected waveform.

SET?

(Query Only)

Description: Returns a string listing the oscilloscope settings, except for configuration information for the calibration values. You can use this string to return the oscilloscope to the state it was in when you made the SET? query. This command is identical to the *LRN? command.

Group: Miscellaneous

Related Commands: HEADer, *LRN?, VERBose

Syntax: SET?

NOTE: The SET? query always returns a string with command headers, regardless of the setting of the HEADer command. This is because the returned string is intended to be able to be sent back to the oscilloscope as a command string. The VERBose command can still be used to specify whether the returned headers should be abbreviated or full length.

Example: SET?
a partial return string may look like this:

```
:ACQUIRE:STOPAFTER RUNSTOP;STATE 1;MODE SAMPLE;NUMENV 10;NUMAVG 16;REPET
1;:APPMENU:TITLE "Application Menu";LABEL:BOTTOM1 "";BOTTOM2 "";BOTTOM3
"";BOTTOM4 ""; BOTTOM5 "";BOTTOM6 "";BOTTOM7 "";RIGHT1 "";RIGHT2 "";
RIGHT3 "";RIGHT4 "";RIGHT5 "";:HEADER 1;:VERBOSE 1; :ALIAS:STATE
0;:DISPLAY:FORMAT YT;STYLE VECTORS;FILTER SINX;PERSISTENCE 500.0E-
3;GRATICULE FULL;TRIGT 1;INTENSITY:OVERALL 85;WAVEFORM 75;TEXT
60;CONTRAST 150;:MESSAGE:SHOW "hello";STATE 1;BOX 74,84,475,135;:LOCK
NONE; :HARDCOPY:FORMAT EPSIMAGE;PORT GPIB;LAYOUT PORTRAIT;
```

*SRE

Description: (Service Request Enable) sets and queries the bits in the Service Request Enable Register (SRER). For more information, refer to the section on Registers.

Group: Status and Error

Related Commands: *CLS, DESE, *ESE, *ESR?, EVENT?, EVMSg?, FACtory, *PSC, *STB?

Syntax 1: *SRE <NR1>

Syntax 2: *SRE?

Argument: <NR1> is a value in the range from 0 to 255. The binary bits of the SRER are set according to this value. Using an out-of-range value causes an execution error. The power-on default for SRER is 0 if *PSC is 1. If *PSC is 0, the SRER maintains its value through a power cycle.

Example 1: `*SRE 48`
sets the bits in the SRER to 00110000 binary.

Example 2: `*SRE?`
might return a value of 32, showing that the bits in the SRER have the binary value 00100000.

*STB?

(Query Only)

Description: (Read Status Byte) query returns the contents of the Status Byte Register (SBR) using the Master Summary Status (MSS) bit. For more information, refer to the section on Registers.

Group: Status and Error

Related Commands: *CLS, DESE, *ESE, *ESR?, EVENT?, EVMSg?, FACtory, *SRE

Syntax: *STB?

Return Value: <NR1>

Example: *STB?
might return the value 96, showing that the SBR contains the binary value 01100000.

TEKSecure

Description: Initializes both waveform and setup memories. This overwrites any previously stored data.

TEKSecure writes zeros in all waveform reference memory, regardless of selected record length, and puts all setups in the factory init state.

TEKSecure then verifies that the waveform and setup memory are in the desired state. It displays a pass or a fail notifier on completion.

Group: Miscellaneous

Syntax: TEKSecure

TIMe

Description: Sets or queries the time that the oscilloscope can display.

Group: Miscellaneous

Related Commands: DATE, DISplay:CLOCk

Syntax 1: TIMe <QString>

Syntax 2: TIMe?

Argument: <QString> is a date in the form "hh:mm:ss".
hh refers to the hour number from 1 to 24.
mm refers to the minute number in the hour from 0 to 59.
ss refers to the seconds number in the minute from 0 to 59.
There must be a colon after the hh and after the mm.

Example: `TIME "01:24:00"`
specifies that the time is set to 01:24 AM.

TRIGger

Description: Forces a trigger event to occur, and the TRIGger query returns the current trigger parameters.

Group: Trigger

Syntax 1: TRIGger FORCe

Syntax 2: TRIGger?

Argument: FORCe creates a trigger event. If TRIGger:STATE is READy, the acquisition will complete, otherwise this command will be ignored. This is equivalent to pressing the front-panel **FORCE TRIGGER** button.

Example 1: `TRIGGER FORCe`
forces a trigger event to occur.

Example 2: `TRIGGER?`
might return
```
:TRIGGER:MAIN:MODE AUTO;TYPE EDGE;LEVEL -480.0E-3;HOLDOFF:VALUE
0;:TRIGGER:MAIN:EDGE:SOURCE CH1; COUPLING DC;SLOPE
RISE;:TRIGGER:MAIN:LOGIC:CLASS PATTERN;FUNCTION AND;WHEN TRUE;
THRESHOLD:CH1 1.40E+0; CH2 1.200E+0;CH3 1.200E+0;CH4
1.200E+0;:TRIGGER:MAIN:LOGIC:INPUT:CH1 HIGH;CH2 X;CH3
X;:TRIGGER:MAIN:LOGIC:PATTERN:INPUT:CH4
X;:TRIGGER:MAIN:LOGIC:STATE:INPUT:CH4 RISE;:TRIGGER:MAIN:PULSE:CLASS
GLITCH;SOURCE CH1;GLITCH:WIDTH 2.0E-9;FILTER ACCEPT;POLARITY POSITIVE;
:TRIGGER:MAIN:PULSE:RUNT:POLARITY POSITIVE;THRESHOLD:HIGH 2.00E+0;LOW
800.0E-3;:TRIGGER:MAIN:PULSE:WIDTH:LOWLIMIT 2.0E-9;HIGHLIMIT 2.0E-9;WHEN
WITHIN;POLARITY POSITIVE;:TRIGGER:DELAY:TYPE EDGE;LEVEL -480.0E-3;BY
TIME; EDGE:SOURCE CH1;SLOPE RISE;COUPLING DC; :TRIGGER:DELAY:TIME 16.0E-
9;EVENTS:COUNT 2
```

TRIGger:DELay

Description: Sets the delayed trigger level and returns the current delayed trigger parameters.

Group: Trigger

Syntax 1: TRIGger:DELay SETLevel

Syntax 2: TRIGger:DELay?

Argument: SETLevel sets the delayed trigger level to half way between the MIN and MAX amplitudes of the trigger source input. This is equivalent to selecting **Set to 50%** in the Delayed Edge Level side menu.

Example 1: `TRIGGER:DELAY SETLEVEL`
sets the delayed trigger level to 50% of MAX and MIN.

Example 2: `TRIGGER:DELAY?`
might return
```
:TRIGGER:DELAY:TYPE EDGE;LEVEL 0.0E+0;BY TIME;EDGE:SOURCE CH1;SLOPE
RISE;COUPLING DC;:TRIGGER:DELAY:TIME 16.0E-9;EVENTS:COUNT 2
```

TRIGger:DELay:BY

Description: Selects whether the delayed trigger occurs after a specified number of events or a specified period of time after the main trigger. This is equivalent to setting **Delay by** in the Delayed Trig menu.

Group: Trigger

Related Commands: TRIGger:DELay:EVENTS:COUNt, TRIGger:DELay:TIMe

Syntax 1: TRIGger:DELay:BY { EVENTS **|** TIMe **|** EVENTSTime (TDS 510A, 500D, 600B, & 700D only) **|** RUNSAfter }

Syntax 2: TRIGger:DELay:BY?

Argument 1: EVENTS
sets the delayed trigger to occur after a set number of trigger events after the main trigger. The number of events is specified by
TRIGger:DELay:EVENTS:COUNt.

Argument 2: TIMe
sets the delayed trigger to occur a set time after the main trigger event. The time period is specified by TRIGger:DELay:TIMe.

Argument 3: EVENTSTime (TDS 510A, 500D, 600B, & 700D only) sets a specified time after a specified number of delay trigger trigger events - after the main trigger event. For example in examining a pulse train, you might use the main trigger to detect the start of the train, then use the delay by events to go to the position of interest within the pulse train, and then use the time delay to wait a specified time period before starting the data acquisition.

Argument 4: RUNSAfter looks for a main trigger, then waits a user-specified time, then starts acquiring data.

Example: `TRIGGER:DELAY:BY?`
might return `EVENTS.`

TRIGger:DELay:EDGE?

(Query Only)

Description: Returns the coupling, slope, and source for the delayed trigger.

Group: Trigger

Syntax: TRIGger:DELay:EDGE?

Example: `TRIGGER:DELAY:EDGE?`
might return `:TRIGGER:EDGE:SOURCE CH1;SLOPE RISE;COUPLING DC`

TRIGger:DELay:EDGE:COUPling

Description: Selects the type of coupling for the delayed trigger. This command is equivalent to selecting **Coupling** in the Delayed Trig menu.

Group: Trigger

Syntax 1:
TRIGger:DELay:EDGE:COUPling { AC (TDS 400A & 510A) **|** DC **|** HFRej (TDS 400A & 510A) **|** LFRej (TDS 400A & 510A) **|** MAINTrigger (TDS 500D, 600B, & 700D only) **|** NOISErej }

Syntax 2: TRIGger:DELay:EDGE:COUPling?

Argument 1: AC selects AC trigger coupling (TDS 400A & 510A).

Argument 2: DC selects DC trigger coupling.

Argument 3: HFRej coupling removes the high frequency components of the DC signal (TDS 400A & 510A).

Argument 4: LFRej coupling removes the low frequency components of the AC signal (TDS 400A & 510A).

Argument 5: MAINTrigger coupling sets the delayed trigger coupling to match the setting on the main trigger (TDS 400A & 510A).

Argument 6: NOISErej selects DC low sensitivity.

Example 1: `TRIGGER:DELAY:EDGE:COUPLING DC`
sets the delay trigger to DC coupling.

Example 2: `TRIGGER:DELAY:EDGE:COUPLING?`
might return `LFREJ` for the delayed trigger coupling.

TRIGger:DELay:EDGE:SLOpe

Description: Selects either a rising or falling edge for the delayed trigger. This command is equivalent to selecting **Slope** in the Delayed Trig menu.

Group: Trigger

Syntax 1: TRIGger:DELay:EDGE:SLOpe { RISe **|** FALL }

Syntax 2: TRIGger:DELay:EDGE:SLOpe?

Argument 1: FALL specifies to trigger on the falling or negative edge of a signal.

Argument 2: RISe specifies to trigger on the rising or positive edge of a signal.

Example: `TRIGGER:DELAY:EDGE:SLOPE?`
might return `RISE`, indicating that the delayed trigger occurs on the rising edge.

TRIGger:DELay:EDGE:SOUrce

Description: Selects the source for the delayed trigger. This command is equivalent to selecting **Source** in the Delayed Trig menu.

Group: Trigger

Syntax 1:
TRIGger:DELay:EDGE:SOUrce { AUXiliary
(not available on TDS 520D & 724D) | CH<x> }

Syntax 2: TRIGger:DELay:EDGE:SOUrce?

Argument 1: AUXiliary specifies an external trigger using the Auxiliary Trigger Input connector that is located on the rear panel of the instrument. The TDS 520D & 724D do not have an Auxiliary Trigger input and so do not support this argument.

Argument 2: CH<x> specifies one of the input channels.

Example: `TRIGGER:DELAY:EDGE:SOURCE CH1`
selects channel 1 as the input source for the delayed trigger.

TRIGger:DELay:EVENTS?

(Query Only)

Description: Returns the current delayed trigger event parameter.

Group: Trigger

Syntax: TRIGger:DELay:EVENTS?

Example: `TRIGGER:DELAY:EVENTS?`
might return `:TRIGGER:DELAY:EVENTS:COUNT 2`

TRIGger:DELay:EVENTS:COUNt

Description: Sets or queries the number of events that must occur before the delayed trigger occurs when TRIGger:DELay:BY is set to EVENTS. This is equivalent to setting the **Delay by Events** count in the Delayed Edge Delay side menu.

Group: Trigger

Syntax 1: TRIGger:DELay:EVENTS:COUNt <NR1>

Syntax 2: TRIGger:DELay:EVENTS:COUNt?

Argument: <NR1> is the number of delayed edge trigger events. The TDS 400A range is 1 to (10E7-1). The TDS 510A, 500D, 600B, and 700D range is 2 to 10E7.

Example 1: `TRIGGER:DELAY:EVENTS:COUNT 4`
specifies that the delayed trigger will occur four trigger events after the main trigger.

Example 2: `TRIGGER:DELAY:EVENTS:COUNT?`
might return 2, indicating that two events must occur after the main trigger before the delayed trigger can occur.

TRIGger:DELay:LEVel

Description: Selects the level of the delayed trigger. This command is equivalent to setting **LEVel** in the Delayed Trig menu.

Group: Trigger

Syntax 1: TRIGger:DELay:LEVel { ECL **|** TTL **|** <NR3> }

Syntax 2: TRIGger:DELay:LEVel?

Argument 1: ECL specifies a preset ECL level of -1.3 V.

Argument 2: TTL specifies a preset TTL level of 1.4 V.

Argument 3: <NR3> is the delayed trigger level, in volts.

Example: `TRIGGER:DELAY:LEVEL 2E-3`
sets the delayed trigger level to 2 mV.

TRIGger:DELay:TIMe

Description: Sets or queries the delay time when HORizontal:DELay:MODe is set to TRIGAfter. This command is identical to the HORizontal:DELay:TIMe:TRIGAfter command, and is equivalent to setting the **Delay by Time** value in the Delayed Edge Delay side menu.

When HORizontal:DELay:MODe is set to RUNSAfter, the delay time is set by the HORizontal:DELay:TIMe:RUNSAfter command.

Group: Trigger

Related Commands:
HORizontal:DELay:MODe, HORizontal:DELay:TIMe:RUNSAfter, HORizontal:DELay:TIMe:TRIGAfter

Syntax 1: TRIGger:DELay:TIMe <NR3>

Syntax 2: TRIGger:DELay:TIMe?

Argument: <NR3> is the delay time, in seconds.

Example: `TRIGGER:DELAY:TIME 4E-6`
sets the delay time to 4 us.

TRIGger:DELay:TYPe

Description: Sets or queries the type of delayed trigger.

Group: Trigger

Syntax 1: TRIGger:DELay:TYPe EDGE

Syntax 2: TRIGger:DELay:TYPe?

Argument: EDGE is a normal trigger. A trigger event occurs when a signal passes through a specified voltage level in a specified direction. Use the TRIGger:DELay:LEVel and TRIGger:DELay:EDGE:SLOpe commands to set the voltage level and direction respectively.

Example: `TRIGGER:DELAY:TYPE?`
always returns `EDGE` as the type of delayed trigger.

TRIGger:MAIn

Description: Sets the main trigger level and returns the current main trigger parameters.

Group: Trigger

Syntax 1: TRIGger:MAIn SETLevel

Syntax 2: TRIGger:MAIn?

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

This argument works differently with AMI Communications Triggering settings. Specifically, when AMI is chosen, this command measures the peak-to-peak level and sets an upper threshold value at 75% and a lower threshold value at 25% of the peak-to-peak value. If the pulse form is CMI, NRZ, or an Eye Diagram, the level is set to 50%.

Example: `TRIGGER:MAIN SETLEVEL`
sets the main trigger level mid way between MAX and MIN.

TRIGger:MAIn:COMMunication:AMI:PULSEForm

(TDS 500D & 700D)

Description: Sets or queries the communication trigger AMI pulse form to one of three possibilities.

Group: Trigger

Syntax 1: TRIGger:MAIn:COMMunication:AMI:PULSEForm { PLUSOne **|** MINUSOne **|** EYEdiagram }

Syntax 2: TRIGger:MAIn:COMMunication:AMI:PULSEForm?

Argument 1: PLUSOne corresponds to the Isolated +1 on the front panel menu. This is the default value.

Argument 2: MINUSOne corresponds to the Isolated -1.

Argument 3: EYEdiagram corresponds to Eye Diagram.

Example: `TRIGGER:MAIN:COMM:AMI:PULSEFORM PLUSONE`
sets the AMI pulseform to the Isolated +1.

TRIGger:MAIn:COMMunication:AMI:THReshold:HIGH

(TDS 500D & 700D)

Description: Sets or queries the AMI communication trigger's high threshold value in volts. The threshold works identically to the threshold in Pulse Slewrate.

Group: Trigger

Syntax 1: TRIGger:MAIn:COMMunication:AMI:THReshold:HIGH <NR3>

Syntax 2: TRIGger:MAIn:COMMunication:AMI:THReshold:HIGH?

Argument: <NR3> is the high value of the threshold. The unit of measure is volts. The default is +0.5 volts.

Example: `TRIGGER:MAIN:COMM:AMI:THRESHOLD:HIGH 2.8 E-2`
sets the high threshold to 28 milliVolts.

TRIGger:MAIn:COMMunication:AMI:THReshold:LOW

(TDS 500D & 700D)

Description: Sets or queries the AMI communication trigger's low threshold value in volts. The threshold works identically to the threshold in Pulse Slewrate.

Group: Trigger

Syntax 1: TRIGger:MAIn:COMMunication:AMI:THReshold:LOW <NR3>

Syntax 2: TRIGger:MAIn:COMMunication:AMI:THReshold:LOW?

Argument: <NR3> is the low value of the threshold. The unit of measure is volts. The default is -0.5 volts.

Example: `TRIGGER:MAIN:COMM:AMI:THRESHOLD:LOW -2.8 E-2`
sets the low threshold to -28 milliVolts.

TRIGger:MAIn:COMMunication:BITRate

(TDS 500D & 700D)

Description: Sets or queries the communication trigger signal bit rate. If this command changes the bit rate, the command then also causes the communication standard to set to “custom”. If both the TRIGger:MAIn:COMMunication:STANDard command and the TRIGger:MAIn:COMMunication:BITRate command are used, the last one executed takes precedence.

Group: Trigger

Syntax 1: TRIGger:MAIn:COMMunication:BITRate { <NR3> }

Syntax 2: TRIGger:MAIn:COMMunication:BITRate?

Argument: <NR3> is the bit rate in bits per second. The default is 1.5444E+6.

Example: `TRIGGER:MAIN:COMM:BITRATE 1.053 E+8`
sets the bit rate to 105.3 Mb/s.

TRIGger:MAIn:COMMunication:CMI:PULSEForm

(TDS 500D & 700D)

Description: Sets or queries the communication trigger CMI pulse form to one of three possibilities.

Group: Trigger

Syntax 1: TRIGger:MAIn:COMMunication:CMI:PULSEForm { PLUSOne **|** MINUSOne **|** EYEdiagram **|** ZERO }

Syntax 2: TRIGger:MAIn:COMMunication:CMI:PULSEForm?

Argument 1: PLUSOne corresponds to triggering on a positive mark. This is the default value.

Argument 2: MINUSOne corresponds to triggering on a negative mark.

Argument 3: EYEdiagram corresponds to Eye Diagram.

Argument 4: ZERO corresponds to triggering on a Zero bit.

Example: `TRIGGER:MAIN:COMM:CMI:PULSEFORM PLUSONE`
selects triggering on a positive mark.

TRIGger:MAIn:COMMunication:CODe

(TDS 500D & 700D)

Description: Sets or queries the communication trigger signal code. If this command changes the code, this command also sets the standard to custom.

Group: Trigger

Syntax 1: TRIGger:MAIn:COMMunication:CODe { AMI | CMI | NRZ }

Syntax 2: TRIGger:MAIn:COMMunication:CODe?

Argument 1: AMI refers to the Alternate Mark Inversion encoding scheme. This is the default.

Argument 2: CMI refers to the Coded Mark Inversion encoding scheme.

Argument 3: NRZ refers to Non-Return to Zero codes.

Example: `TRIGGER:MAIN:COMM:CODE AMI`
selects the AMI communication code

TRIGger:MAIn:COMMunication:NRZ:PULSEForm

(TDS 500D & 700D)

Description: Sets or queries the communication trigger NRZ pulse form to one of 19 possibilities.

Group: Trigger

Syntax 1: TRIGger:MAIn:COMMunication:NRZ:PULSEForm { EYEdiagram | RISE | FALL | PATTERN0 | PATTERN1 | PATTERN2 | PATTERN3 | PATTERN4 | PATTERN5 | PATTERN6 | PATTERN7 | P0 | P1 | P2 | P3 | P4 | P5 | P6 | P7 }

Syntax 2: TRIGger:MAIn:COMMunication:NRZ:PULSEForm?

Argument 1: Eyediagram selects an Eye Diagram image. This is the default value.

Argument 2: RISE selects a positive edge trigger.

Argument 3: FALL selects a negative edge trigger.

Argument 4: PATTERN0 causes a trigger on Pattern0. Intersymbol interference can be identified with the Pattern Violations Trigger. The leading symbol (0, 1, x) and trailing symbol (0, 1, x) are bit values that precede or follow the three symbols of interest (in the middle). The underlying trigger that implements Pattern Violation Triggering is pulse width triggering.

Argument 5: PATTERN1 causes a trigger on Pattern1 1 001 x.

Argument 6: PATTERN2 causes a trigger on Pattern2 x 010 x.

Argument 7: PATTERN3 causes a trigger on Pattern3 x 011 0.

Argument 8: PATTERN4 causes a trigger on Pattern4 1 100 1.

Argument 9: PATTERN5 causes a trigger on Pattern5 x 101 x.

Argument 10: PATTERN6 causes a trigger on Pattern6 0 110 x.

Argument 11: PATTERN7 causes a trigger on Pattern7 0 111 0.

Argument 12: P0 is a synonym for PATTERN0

Argument 13: P1 is a synonym for PATTERN1

Argument 14: P2 is a synonym for PATTERN2

Argument 15: P3 is a synonym for PATTERN3

Argument 16: P4 is a synonym for PATTERN4

Argument 17: P5 is a synonym for PATTERN5

Argument 18: P6 is a synonym for PATTERN6

Argument 19: P7 is a synonym for PATTERN7

Example: `TRIGGER:MAIN:COMM:CODE:NRZ:PULSEFORM EYEDIAGRAM`
selects a trigger on an Eye Diagram image.

TRIGger:MAIn:COMMunication:SOUrce

(TDS 500D & 700D)

Description: Sets or queries the source for the main communication trigger. This is equivalent to selecting the source in the Communication Source side menu.

Group: Trigger

Syntax 1: TRIGger:MAIn:COMMunication:SOUrce CH<x>

Syntax 2: TRIGger:MAIn:COMMunication:SOUrce?

Argument: CH<x> specifies one of the input channels.

Example: `TRIGGER:MAIN:COMMUNICATION:SOURCE CH2`
selects Channel 2 as the source for the main communication trigger.

TRIGger:MAIn:COMMunication:STANdard

(TDS 500D & 700D)

Description: Sets or queries the communication trigger standard which identifies the code and bit rate. If this command changes the code, then the pulse form is changed to the last pulse form that was set for that code.

Group: Trigger

Syntax 1: TRIGger:MAIn:COMMunication:STANdard { DS1 | DS1A | DS1C | DS2 | DS3 | DS3 Rate | DS4NA | E1 | E2 | E3 | E4 | E5 | STS1 | STS3 | OC1 | OC3 | OC12 | STM1E | FC133 | FC266 | FC531 | FC1063 | FDDI | D1 | D2 | ENET1250 | ENET10 |ENET100 | VIDEO360 | Custom }

Syntax 2: TRIGger:MAIn:COMMunication:STANdard?

Argument 1: DS1 refers to DS1 (1.544 Mb/s) AMI standard.

Argument 2: DS1A refers to DS1A (2.048 Mbs/s) AMI standard.

Argument 3: DS1C refers to DS1C (3.152 Mb/s) AMI standard.

Argument 4: DS2 refers to DS2 (6.312 Mb/s) AMI standard.

Argument 5: DS3 refers to DS3 (44.736 Mb/s) AMI standard.

Argument 6: DS3 Rate refers to the ITU G.703 (44.736 Mb/s) standard.

Argument 7: DS4Na refers to DS4NA (139.26 Mb/s) CMI standard.

Argument 8: E1 refers to E1 (2.048 Mb/s) AMI standard.

Argument 9: E2 refers to E2 (8.44 Mb/s) AMI standard.

Argument 10: E3 refers to E3 (34.368 Mb/s) AMI standard.

Argument 11: E4 refers to E4 (139.26 Mb/s) CMI standard.

Argument 12: E5 refers to E5 or CEPT (565 Mb/s) NRZ standard.

Argument 13: STS1 refers to STS-1 (51.84 Mb/s) AMI standard.

Argument 14: STS3 refers to STS-3 (155.52 Mb/s) CMI standard.

Argument 15: OC1 refers to OC1/STM0 (51.84 Mb/s) CMI standard.

Argument 16: OC3 refers to OC3/STM1 ((155.52 Mb/s) NRZ standard.

Argument 17: OC12 refers to OC12/STM4 (622.08 Mb/s) NRZ standard.

Argument 18: STM1E refers to STM1E (155.52 Mb/s) CMI standard.

Argument 19: FC133 refers to FC133 (132.7 Mb/s) NRZ standard.

Argument 20: FC266 refers to FC266 (265.6 Mb/s) NRZ standard.

Argument 21: FC531 refers to FC531 (531.2 Mb/s) NRZ standard.

Argument 22: FC1063 refers to FC1063 (1.063 Gb/s) NRZ standard.

Argument 23: FDDI refers to FDDI (125 Mb/s) NRZ standard.

Argument 24: D1 refers to 4:2:2 or D1 (270 Mb/s) NRZ standard.

Argument 25: D2 refers to 4fsc NTSC or D2 (143.18 Mb/s) NRZ standard.

Argument 26: ENET1250 refers to the IEEE Draft P802.3z/D3 (1.25 Gb/s) standard.

Argument 27: ENET10 refers to the ANSI/IEEE 802.3 Fifth Edition 1996 - Information Technology - Telecommunications and Information Exchange (10 Mb/s) standard.

Argument 28: ENET100 refers to the ANSI X3.263-1995 for Information Technology (125 Mb/s) standard.

Argument 29: VIDEO360 refers to the ANSI/SMPTE 259M 4:2:2 component serial video (360 Mb/s) standard.

Argument 30: CUSTom can take a non-standard, floating-point bit rate. <NR3>.

Example: `TRIGGER:MAIN:COMMUNICATION:STANDARD E4`
selects the E4 communication standard.

TRIGger:MAIn:EDGE?

(Query Only)

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

Group: Trigger

Syntax: TRIGger:MAIn:EDGE?

Example: `TRIGGER:MAIN:EDGE?`
might return SOURCE CH1;COUPLING DC;SLOPE RISE

TRIGger:MAIn:EDGE:COUPling

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

Group: Trigger

Syntax 1: TRIGger:MAIn:EDGE:COUPling { AC **|** DC **|** HFRej **|** LFRej **|** NOISErej }

Syntax 2: TRIGger:MAIn:EDGE:COUPling?

Argument 1: AC selects AC trigger coupling.

Argument 2: DC selects DC trigger coupling.

Argument 3: HFRej coupling removes the high frequency components of the DC signal.

Argument 4: LFRej coupling removes the low frequency components of the AC signal.

Argument 5: NOISErej selects DC low sensitivity. It requires added signal amplitude for more stable, less false triggering.

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

TRIGger:MAIn:EDGE:SLOpe

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

Group: Trigger

Syntax 1: TRIGger:MAIn:EDGE:SLOpe { FALL | RISe }

Syntax 2: TRIGger:MAIn:EDGE:SLOpe?

Argument 1: FALL specifies to trigger on the falling or negative edge of a signal.

Argument 2: RISe specifies to trigger on the rising or positive edge of a signal.

Example: `TRIGGER:MAIN:EDGE:SLOPE RISE`
sets the main edge trigger to occur on the rising slope.

TRIGger:MAIn:EDGE:SOUrce

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

Group: Trigger

Syntax 1: TRIGger:MAIn:EDGE:SOUrce { AUXiliary (not available on TDS 520D & 724D) **|** CH<x> **|** LINE }

Syntax 2: TRIGger:MAIn:EDGE:SOUrce?

Argument 1: AUXiliary specifies an external trigger using the Auxiliary Trigger Input connector that is located on the rear panel of the instrument. The TDS 520D & 724D do not have an Auxiliary Trigger input and so do not support this argument.

Argument 2: CH<x> specifies one of the input channels.

Argument 3: LINE specifies AC line voltage.

Example 1: `TRIGGER:MAIN:EDGE:SOURCE LINE`
specifies the AC line voltage as the main edge trigger source.

Example 2: `TRIGGER:MAIN:EDGE:SOURCE?`
might return CH2 for the main edge trigger source.

TRIGger:MAIn:HOLDOff?

(Query Only)

Description: For the TDS 500D, 600B, & 700D, returns the main trigger holdoff default (TIMe or DEFAult) and main trigger holdoff time. For the TDS 400A & 510A, returns the main trigger holdoff value.

Group: Trigger

Syntax: TRIGger:MAIn:HOLDOff?

Example: `TRIGGER:MAIN:HOLDOFF?`
for the TDS 500D, 600B, and 700D, might return
`:TRIGGER:MAIN:HOLDOFF:TIME 250.0E-9;BY DEFAULT`

or, for the TDS 400A and 510A, might return:
`:TRIGGER:MAIN:HOLDOFF:VALUE 0.`

TRIGger:MAIn:HOLDOff:ACTUal?

(TDS 500D, 600B, & 700D) (Query Only)

Description: Returns the main trigger holdoff value in seconds. This is equivalent to selecting **Mode & Holdoff** from the main Trigger menu and viewing the value in the side menu **Holdoff Default** or **Holdoff Time** items (whichever is highlighted).

Group: Trigger

Syntax: TRIGger:MAIn:HOLdoff:ACTUal?

Example: `TRIGGER:MAIN:HOLDOFF:ACTUAL?`
might return `4E-6` showing the holdoff time is set to 4 us.

TRIGger:MAIn:HOLDOff:BY

(TDS 500D, 600B, & 700D)

Description: Sets or queries the main trigger holdoff default. This is equivalent to selecting **Mode & Holdoff** from the main Trigger menu, then setting **Default Holdoff** or **Holdoff (Time)** in the resulting side menu.

Group: Trigger

Syntax: TRIGger:MAIn:HOLDOff:BY { TIMe **|** DEFAult }

Argument 1: TIMe enables the user to set the holdoff time.

Argument 2: DEFault automatically calculates a holdoff time to use. This time is typically equivalent to the greater of 1/2 screen (5 divisions) or time of 250 ns. The maximum value is 12 seconds. For example, if the oscilloscope is set to 1 msec/division then the default holdoff will be 1 msec/division x 25 divs = 25 msec..

Example: `TRIGGER:MAIN:HOLDOFF:BY TIME`
sets the holdoff to the by time setting. This enables the user to set the holdoff time.

TRIGger:MAIn:HOLDOff:TIMe

(TDS 500D, 600B, & 700D)

Description: Sets or queries the main trigger holdoff time. This is equivalent to setting **Holdoff Time** in the Mode & Holdoff side menu.

Group: Trigger

Syntax 1: TRIGger:MAIn:HOLDOff:TIMe <NR3>

Syntax 2: TRIGger:MAIn:HOLDOff:TIMe?

Argument: <NR3> holdoff time in seconds. The range is 250 ns to 12.0 seconds.

Example: `TRIGGER:MAIN:HOLDOFF:TIME 10`
sets the holdoff time to be 10 seconds.

TRIGger:MAIn:HOLDOff:VALue

(TDS 400A & 510A)

Description: Sets or queries the main trigger holdoff value. This is equivalent to setting **Holdoff** in the Mode & Holdoff side menu.

Group: Trigger

Syntax 1: TRIGger:MAIn:HOLDOff:VALue <NR1>

Syntax 2: TRIGger:MAIn:HOLDOff:VALue?

Argument: <NR1> is from 0 to 100, and is a percent of the holdoff range.

Example: `TRIGGER:MAIN:HOLDOFF:VALUE 10`
sets the holdoff value to be 10% of the holdoff range.

TRIGger:MAIn:LEVel

Description: Sets the main trigger level. This command is equivalent to adjusting the front-panel **TRIGGER MAIN LEVEL** knob.

With Communications triggers, this command makes no change to AMI thresholds. It does change CMI and NRZ levels. It makes no change to value ranges.

Group: Trigger

Syntax 1: TRIGger:MAIn:LEVel { ECL **|** TTL **|** <NR3> }

Syntax 2: TRIGger:MAIn:LEVel?

Argument 1: ECL specifies a preset ECL level of -1.3 V.

Argument 2: TTL specifies a preset TTL level of 1.4 V.

Argument 3: <NR3> is the main trigger level, in volts.

Example 1: `TRIGGER:MAIN:LEVEL?`
might return `TTL`, indicating that the main edge trigger is set to 1.4 V.

Example 2: `TRIGGER:MAIN:LEVEL 0.5`
sets the main trigger level to 0.5 V.

TRIGger:MAIn:LOGIc?

(TDS 510A, 500D, 600B, & 700D) (Query Only)

Description: Returns all main logic trigger parameters.

Group: Trigger

Syntax: TRIGger:MAIn:LOGIc?

Example: `TRIGGER:MAIN:LOGIC?`
might return `:TRIGGER:MAIN:LOGIC:CLASS PATTERN;FUNCTION AND;WHEN TRUE;THRESHOLD:CH1 0;CH2 0;CH3 0;CH4 0;:TRIGGER:MAIN:LOGIC:INPUT:CH1 HIGH;CH2 X;CH3 X;:TRIGGER:MAIN:LOGIC:PATTERN:INPUT:CH4 X;:TRIGGER:MAIN:LOGIC:STATE:INPUT:CH4 RISE`

TRIGger:MAIn:LOGIc:CLAss

(TDS 510A, 500D, 600B, & 700D)

Description: Sets or queries the type of main logic trigger. This command is equivalent to selecting **Class** in the Trigger menu when the **Type** is set to Logic.

Group: Trigger

Syntax 1: TRIGger:MAIn:LOGIc:CLAss { PATtern **|** STATE **|** SETHold (TDS 500D, 600B, & 700D) }

Syntax 2: TRIGger:MAIn:LOGIc:CLAss?

Argument 1: PATtern means that the instrument triggers when the specified logical combinations of channels 1, 2, 3, and 4 are met.

Argument 2: STATE means that the instrument triggers when the specified conditions of channels 1, 2, and 3 are met after the channel 4 condition is met.

Argument 3: SETHold means the oscilloscope will trigger on the setup and hold violations between a data source and a clock source (TDS 500D, 600B, & 700D).

Example: `TRIGGER:MAIN:LOGIC:CLASS?`
might return `STATE`.

TRIGger:MAIn:LOGIc:FUNCtion

(TDS 510A, 500D, 600B, & 700D)

Description: Sets or queries the logical combination of the input channels for the main logic trigger.

When TRIGger:MAIn:LOGIc:CLAss is PATtern, this command applies to channels 1, 2, 3, and 4. When TRIGger:MAIn:LOGIc:CLAss is STATE, only channels 1, 2, and 3 are logically combined. This command is equivalent to selecting the function in the Logic Pattern Function side menu.

Group: Trigger

Syntax 1: TRIGger:MAIn:LOGIc:FUNCtion { AND **|** NANd **|** NOR **|** OR }

Syntax 2: TRIGger:MAIn:LOGIc:FUNCtion?

Argument 1: AND specifies that the instrument will trigger if all the conditions are true.

Argument 2: NANd specifies that the instrument will trigger if any of the conditions are false.

Argument 3: NOR specifies that the instrument will trigger if all of the conditions are false.

Argument 4: OR specifies that the instrument will trigger if any of the conditions are true.

Example 1: `TRIGGER:MAIN:LOGIC:FUNCTION NOR`
sets the logical combination of channels to be true when none of the conditions are true.

Example 2: `TRIGGER:MAIN:LOGIC:FUNCTION?`
might return `NAND`.

TRIGger:MAIn:LOGIc:INPut?

(TDS 510A, 500D, 600B, & 700D) (Query Only)

Description: Returns the main logic trigger input for all channels.

Group: Trigger

Syntax: TRIGger:MAIn:LOGIc:INPut?

Example: `TRIGGER:MAIN:LOGIC:INPUT?`
might return `:TRIGGER:MAIN:LOGIC:INPUT:CH1 HIGH;CH2 X;CH3 X`

TRIGger:MAIn:LOGIc:INPut:CH<x>

(TDS 510A, 500D, 600B, & 700D)

Description: Sets or queries the main logic trigger input for the specified channel. The channel is specified by <x> and is 1, 2, or 3. This is equivalent to setting the inputs in the Logic Pattern Inputs side menu.

Group: Trigger

Syntax 1: TRIGger:MAIn:LOGIc:CLAss

Syntax 2: TRIGger:MAIn:LOGIc:INPut:CH<x> { HIGH | LOW | X }

Syntax 3: TRIGger:MAIn:LOGIc:INPut:CH<x>?

Argument 1: HIGH specifies logic high.

Argument 2: LOW specifies logic low.

Argument 3: X specifies a do not care state.

Example: `TRIGGER:MAIN:LOGIC:INPUT:CH2 LOW`
sets the main logic trigger input to logic low for channel 2.

TRIGger:MAIn:LOGIc:PATtern:INPut:CH4

(TDS 510A, 500D, 600B, & 700D)

Description: Sets or queries the main logic trigger input for channel 4. These are the inputs used when TRIGger:MAIn:LOGIc:CLAss is set to PATtern. This is equivalent to setting the channel 4 input in the Logic Pattern Inputs side menu.

Group: Trigger

Related Command: TRIGger:MAIn:LOGIc:CLAss

Syntax 1: TRIGger:MAIn:LOGIc:PATtern:INPut:CH4 { HIGH | LOW | X }

Syntax 2: TRIGger:MAIn:LOGIc:PATtern:INPut:CH4?

Argument 1: HIGH specifies logic high.

Argument 2: LOW specifies logic low.

Argument 3: X specifies a do not care state.

Example 1: `TRIGGER:MAIN:LOGIC:PATTERN:INPUT:CH4 LOW`
sets the main logic trigger input to logic low for channel 4 when the logic class is set to PATtern.

Example 2: `TRIGGER:MAIN:LOGIC:PATTERN:INPUT:CH4?`
might return X, indicating that the logic input for channel 4 is do not care.

TRIGger:MAIn:LOGIc:PATtern:WHEn

(TDS 510A, 500D, 600B, & 700D)

Description: Sets or queries a condition for generating a main logic pattern trigger.

Group: Trigger

Syntax 1:
TRIGger:MAIn:LOGIc:PATtern:WHEn { TRUe | FALSe | LESSThan | MOREThan }

Syntax 2: TRIGger:MAIn:LOGIc: PATtern: WHEn?

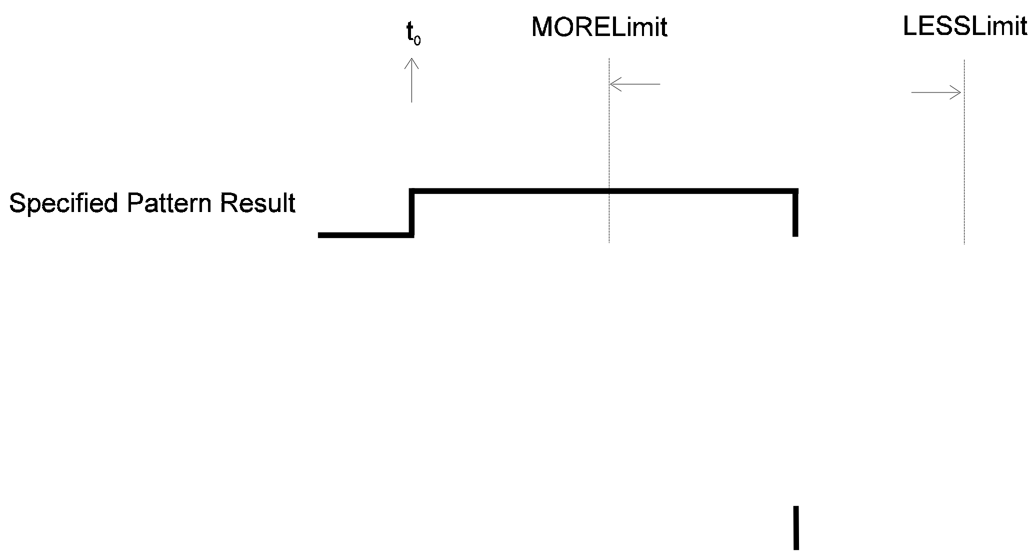
Argument 1: TRUe specifies the trigger to occur when the pattern becomes true.

Argument 2: FALSe specifies the trigger to occur when the pattern becomes false.

Argument 3: LESSThan specifies trigger to occur if the specific pattern is true less than the LESSLimit. (see Figure 2-5 and TRIGger:MAIn:LOGIc:PATtern:WHEn:LESSLimit Trigger is evaluated at the true-false transition.

Argument 4: MOREThan specifies trigger to occur if the specific pattern is true longer than the more limit. (see Figure 2-5 and TRIGger:MAIn:LOGIc:PATtern:WHEn:MORELimit Trigger is evaluated at the true-false transition.

Figure 2-5: LESSThan and MOREThan Arguments



TRIGger:MAIn:LOGIc:PATtern:WHEn:LESSLimit

(TDS 510A, 500D, 600B, & 700D)

Description: Sets or queries the maximum time the selected pattern may be true and still generate a main logic pattern trigger.

Group: Trigger

Syntax 1: TRIGger:MAIn:LOGIc:PATtern:WHEn:LESSLimit <NR3>

Syntax 2: TRIGger:MAIn:LOGIc: PATtern: WHEn: LESSLimit?

Argument: <NR3> time to hold pattern true.

TRIGger:MAIn:LOGIc:PATtern:WHEn:MORELimit

(TDS 510A, 500D, 600B, & 700D)

Description: Sets or queries the minimum time the selected pattern may be true and still generate a main logic pattern trigger.

Group: Trigger

Syntax 1: TRIGger:MAIn:LOGIc:PATtern:WHEn:MORELimit <NR3>

Syntax 2: TRIGger:MAIn:LOGIc: PATtern: WHEn: MORELimit?

Argument: <NR3> time to hold pattern true.

TRIGger:MAIn:LOGIc:SETHold:CLOCk:EDGE

(TDS 500D, 600B, & 700D)

Description: Sets or queries the clock edge polarity for setup and hold violation triggering. This is equivalent to selecting **Define Clock** from the main Trigger menu and **Polarity** in the resulting side menu.

Group: Trigger

Syntax 1: TRIGger:MAIn:LOGIc:SETHold:CLOCk:EDGE {FALL **|** RISe }

Syntax 2: TRIGger:MAIn:LOGIc:SETHold:CLOCk:EDGE?

Argument 1: FALL specifies falling edge.

Argument 2: RISe specifies rising edge.

Example: `TRIGGER:MAIN:LOGIC:SETHOLD:CLOCK:EDGE RISE`
specifies the polarity as the rising edge.

TRIGger:MAIn:LOGIc:SETHold:CLOCk:LEVel

(TDS 500D, 600B, & 700D)

Description: Sets or queries the main logic setup/hold clock voltage trigger level. This is equivalent to selecting **Levels** from the main Trigger menu and **Clock Level** in the resulting side menu.

Group: Trigger

Syntax 1: TRIGger:MAIn:LOGIc:SETHold:CLOCk:LEVel { ECL **|** TTL **|** <NR3> }

Syntax 2: TRIGger:MAIn:LOGIc:SETHold:CLOCk:LEVel?

Argument 1: ECL specifies a preset ECL level of -1.3 V.

Argument 2: TTL specifies a preset TTL level of 1.4 V.

Argument 3: <NR3> is the main trigger level, in volts.

Example: `TRIGGER:MAIN:LOGIC:SETHOLD:CLOCK:LEVEL 1.4`
sets the main logic trigger setup/hold clock level to 1.4 volts.

TRIGger:MAIn:LOGIc:SETHold:CLOCk:SOUrce

(TDS 500D, 600B, & 700D)

Description: Sets or queries the source for the clock for the main logic trigger setup/hold input. The channel is specified by the <x> and is 1, 2, 3, or 4. This is equivalent to selecting **Define Clock** from the main Trigger menu and **CH1**, **CH2**, **CH3**, or **CH4** in the resulting side menu.

Group: Trigger

Syntax 1: TRIGger:MAIn:LOGIc:SETHold:CLOCk:SOUrce CH<x>

Syntax 2: TRIGger:MAIn:LOGIc:SETHold:CLOCk:SOUrce?

Argument: CH<x> specifies one of the input channels (CH1, CH2, CH3, or CH4).

Example: `TRIGGER:MAIN:LOGIC:SETHOLD:CLOCK:SOURCE CH2`
selects Channel 2 as the clock source for the main logic trigger setup/hold.

TRIGger:MAIn:LOGIc:SETHold:DATa:LEVel

(TDS 500D, 600B, & 700D)

Description: Sets or queries the main logic set/hold data level. This is equivalent to selecting **Levels** from the main Trigger menu and **Data Level** in the resulting side menu.

Group: Trigger

Syntax 1: TRIGger:MAIn:LOGIc:SETHold:DATa:LEVel { ECL | TTL | <NR3> }

Syntax 2: TRIGger:MAIn:LOGIc:SETHold:DATa:LEVel?

Argument 1: ECL specifies a preset ECL level of -1.3 V.

Argument 2: TTL specifies a preset TTL level of 1.4 V.

Argument 3: <NR3> is the main trigger level, in volts.

Example: `TRIGGER:MAIN:LOGIC:SETHOLD:DATA:LEVEL 1.4`
specifies the main logic setup/hold data level to 1.4 volts.

TRIGger:MAIn:LOGIc:SETHold:DATa:SOUrce

(TDS 500D, 600B, & 700D)

Description: Sets or queries the data channel for the main logic trigger set/hold input. The channel is specified by <x> and is 1, 2, 3, or 4. This is equivalent to selecting **Data Source** from the main Trigger menu and **CH1**, **CH2**, **CH3**, or **CH4** in the resulting side menu.

Group: Trigger

Syntax 1: TRIGger:MAIn:LOGIc:SETHold:DATa:SOUrce CH<x>

Syntax 2: TRIGger:MAIn:LOGIc:SETHold:DATa:SOUrce?

Argument: CH<x> specifies one of the input channels (CH1, CH2, CH3, or CH4).

Example: `TRIGGER:MAIN:LOGIC:SETHOLD:DATA:SOURCE CH2`
selects Channel 2 as the source for the main logic trigger set/hold.

TRIGger:MAIn:LOGIc:SETHold:HOLDTime

(TDS 500D, 600B, & 700D)

Description: Sets or queries the main logic trigger hold time. This is equivalent to selecting **Setup/Hold Times** from the main Trigger menu and **Hold Time** in the resulting side menu.

Group: Trigger

Syntax 1: TRIGger:MAIn:LOGIc:SETHold:HOLDTime <NR3>

Syntax 2: TRIGger:MAIn:LOGIc:SETHold:HOLDTime?

Argument: <NR3> specifies the hold time setting in seconds. Positive values for hold time occur after the clock edge. Negative values occur before the clock edge.

Example: `TRIGGER:MAIN:LOGIC:SETHOLD:HOLDTime 200 E-12`
sets the main logic trigger sethold holdtime to 200 nanoseconds.

TRIGger:MAIn:LOGIc:SETHold:SETTime

(TDS 500D, 600B, & 700D)

Description: Sets or queries the main logic trigger set time. This is equivalent to selecting **Setup/Hold Times** from the main Trigger menu and **Setup Time** in the resulting side menu.

Group: Trigger

Syntax 1: TRIGger:MAIn:LOGIc:SETHold:SETTime <NR3>

Syntax 2: TRIGger:MAIn:LOGIc:SETHold:SETTime?

Argument: <NR3> specifies the setup time setting in seconds. Positive values occur before the clock edge. Negative values occur after the clock edge.

Example: `TRIGGER:MAIN:LOGIC:SETHOLD:SETTIME 600 E-12`
sets the main logic trigger sethold time to 600 nanoseconds.

TRIGger:MAIn:LOGIc:STATE:INPut:CH4

(TDS 510A, 500D, 600B, & 700D)

Description: Sets or queries the main logic trigger input for channel 4. This input is used when TRIGger:MAIn:LOGIc:CLAss is set to STATE. This is equivalent to setting the channel 4 input in the Logic Pattern Inputs side menu.

Group: Trigger

Syntax 1: TRIGger:MAIn:LOGIc:STATE:INPut:CH4 { FALL | RISe }

Syntax 2: TRIGger:MAIn:LOGIc:STATE:INPut:CH4?

Argument 1: FALL specifies falling edge.

Argument 2: RISe specifies rising edge.

Example: `TRIGGER:MAIN:LOGIC:STATE:INPUT:CH4 RISE`
specifies that the main logic trigger input for channel 4 is the rising edge when the logic class is set to STATE.

TRIGger:MAIn:LOGIc:STATE:WHEn

(TDS 510A, 500D, 600B, & 700D)

Description: Sets or queries the main logic state trigger.

Group: Trigger

Syntax 1: TRIGger:MAIn:STATE:WHEn { TRUe | FALSe }

Syntax 2: TRIGger:MAIn:LOGIc: STATE: WHEn?

Argument 1: TRUe specifies the trigger to occur when the condition is met on the fourth channel and the pattern of the first three channels are at the desired states.

Argument 2: FALSe

TRIGger:MAIn:LOGIc:THReshold?

(TDS 510A, 500D, 600B, & 700D) (Query Only)

Description: Returns the main logic trigger threshold voltage for all channels.

Group: Trigger

Syntax: TRIGger:MAIn:LOGIc:THReshold?

Example: `TRIGGER:MAIN:LOGIC:THRESHOLD?`
might return `:TRIGGER:MAIN:LOGIC:THRESHOLD:CH1 0;CH2 0;CH3 0;CH4 0`

TRIGger:MAIn:LOGIc:THReshold:CH<x>

(TDS 510A, 500D, 600B, & 700D)

Description: Sets or queries the main logic trigger threshold voltage for the channel specified by <x>. This is equivalent to setting the thresholds in the Logic State Threshold and Logic Pattern Threshold side menus.

Group: Trigger

Syntax 1: TRIGger:MAIn:LOGIc:THReshold:CH<x> <NR3>

Syntax 2: TRIGger:MAIn:LOGIc:THReshold:CH<x>?

Argument: <NR3> specifies the threshold voltage.

Example: `TRIGGER:MAIN:LOGIC:THRESHOLD:CH1 .5`
sets the main logic trigger threshold for channel 1 to .5 volts.

TRIGger:MAIn:LOGIc:WHEn

(TDS 510A, 500D, 600B, & 700D)

Description: Specifies whether the main logic trigger occurs when the specified state goes true or false when TRIGger:MAIn:LOGIc:CLAss is set to PATtern. This is equivalent to selecting **Trigger When** in the Trigger menu.

Group: Trigger

Syntax 1: TRIGger:MAIn:LOGIc:WHEn { FALSe **|** TRUe }

Syntax 2: TRIGger:MAIn:LOGIc:WHEn?

Example: `TRIGGER:MAIN:LOGIC:WHEN TRUE`
specifies that the main logic trigger when the logic pattern is true.

TRIGger:MAIn:MODe

Description: Sets or queries the main trigger mode. This command is equivalent to selecting **Mode & Holdoff** in the Trigger menu.

Group: Trigger

Syntax 1: TRIGger:MAIn:MODe { AUTO | NORMal }

Syntax 2: TRIGger:MAIn:MODe?

Argument 1: AUTO generates a trigger if a trigger is not detected within a specific time period.

Argument 2: NORMal waits for a valid trigger event.

Example: `TRIGGER:MAIN:MODE AUTO`
specifies that a trigger event is automatically generated.

TRIGger:MAIn:PULse?

(TDS 510A, 500D, 600B, & 700D) (Query Only)

Description: Returns the main pulse trigger parameters.

Group: Trigger

Syntax: TRIGger:MAIn:PULse?

Example: `TRIGGER:MAIN:PULSE?`
might return
```
:TRIGGER:MAIN:PULSE:CLASS GLITCH;SOURCE CH1; GLITCH:WIDTH 2.0E-9;FILTER
ACCEPT;POLARITY POSITIVE;:TRIGGER:MAIN:PULSE:RUNT:POLARITY
POSITIVE;THRESHOLD:HIGH 2.00E+0;LOW 800.0E-
3;:TRIGGER:MAIN:PULSE:WIDTH:LOWLIMIT 2.0E-9;HIGHLIMIT 2.0E-9;WHEN
WITHIN;POLARITY POSITIVE
```
as the current main pulse trigger parameters.

TRIGger:MAIn:PULse:CLAss

(TDS 510A, 500D, 600B, & 700D)

Description: Sets or queries the type of pulse to trigger on. This command is equivalent to selecting **Class** in the Trigger menu.

Group: Trigger

Syntax 1: TRIGger:MAIn:PULse:CLAss { GLItch **|** RUNT **|** WIDth **|** SLEWRate (TDS 500D, 600B, & 700D) **|** TIMEOut (TDS 500D, 600B, 700D) }

Syntax 2: TRIGger:MAIn:PULse:CLAss?

Argument 1: GLItch triggers when a pulse is found that is of the specified polarity and width. These are set with the commands TRIGger:MAIn:PULse:GLItch:POLarity and TRIGger:MAIn:PULse:GLItch:WIDth.

Argument 2: RUNT triggers when a pulse crosses the first preset voltage threshold but does not cross the second preset threshold before recrossing the first. The thresholds are set with the
TRIGger:MAIn:PULse:RUNT:THReshold:LOW
and
TRIGger:MAIn:PULse:RUNT:THReshold:HIGH
commands. The crossing can be either positive or negative as specified by
TRIGger:MAIn:PULse:RUNT:POLarity.

Argument 3: WIDth triggers when a pulse is found that has the specified polarity and is either inside or outside the limits as specified by
TRIGger:MAIn:PULse:WIDth:LOWLimit
and
TRIGger:MAIn:PULse:WIDth:HIGHLimit.
The polarity is selected using the TRIGger:MAIn:PULse:WIDth:POLarity command.

Argument 4: SLEWRate triggers when the slew rate of the source violates the specified conditions (TDS 500D, 600B, & 700D).

Argument 5: TIMEOut triggers when the pulse train stops in the selected state for longer than the specified time (TDS 500D, 600B, & 700D).

Example: `TRIGGER:MAIN:PULSE:CLASS WIDTH`
specifies a width pulse for the main trigger.

TRIGger:MAIn:PULse:GLItch?

(TDS 510A, 500D, 600B, & 700D) (Query Only)

Description: Returns the current main glitch pulse trigger parameters.

Group: Trigger

Syntax: TRIGger:MAIn:PULse:GLItch?

Example: `TRIGGER:MAIN:PULSE:GLITCH?`
might return
`:TRIGGER:MAIN:PULSE:CLASS GLITCH;SOURCE CH1; GLITCH:WIDTH 2.0E-9;FILTER ACCEPT;POLARITY POSITIVE.`

TRIGger:MAIn:PULse:GLItch:FILTer

(TDS 510A, 500D, 600B, & 700D)

Description: Controls glitch detection. This command is equivalent to selecting **Filter** in the Trigger menu.

Group: Trigger

Syntax 1: TRIGger:MAIn:PULse:GLItch:FILTer { ACCept **|** REJect }

Syntax 2: TRIGger:MAIn:PULse:GLItch:FILTer?

Argument 1: ACCept specifies that the oscilloscope will trigger only on pulses that are narrower than the specified width when the main trigger type is set to pulse glitch. The width is specified using the TRIGger:MAIn:PULse:GLItch:WIDth command.

Argument 2: REJect specifies that the oscilloscope will trigger only on pulses that are wider than the specified width when the main trigger type is set to pulse glitch. The width is specified using the TRIGger:MAIn:PULse:GLItch:WIDth command.

Example: `TRIGGER:MAIN:PULSE:GLITCH:FILTER?`
returns either `ACCEPT` or `REJECT`, indicating whether glitches are filtered.

TRIGger:MAIn:PULse:GLItch:POLarity

(TDS 510A, 500D, 600B, & 700D)

Description: Sets or queries the polarity for the main pulse glitch trigger. This command is equivalent to selecting **Polarity & Width** in the Trigger menu.

Group: Trigger

Syntax 1:
TRIGger:MAIn:PULse:GLItch:POLarity { EITher **|** NEGAtive **|** POSITIVe }

Syntax 2: TRIGger:MAIn:PULse:GLItch:POLarity?

Example: `TRIGGER:MAIN:PULSE:GLITCH:POLARITY EITHER`
specifies that the polarity of the glitch can be either positive or negative.

TRIGger:MAIn:PULse:GLItch:WIDth

(TDS 510A, 500D, 600B, & 700D)

Description: Sets or queries the width for the main pulse glitch trigger. This command is equivalent to selecting **Polarity & Width** in the Trigger menu.

Group: Trigger

Syntax 1: TRIGger:MAIn:PULse:GLItch:WIDth <NR3>

Syntax 2: TRIGger:MAIn:PULse:GLItch:WIDth?

Argument: <NR3> is the width of the glitch, in seconds.

Example: `TRIGGER:MAIN:PULSE:GLITCH:WIDTH 15E-6`
sets the width of the glitch to 15 us.

TRIGger:MAIn:PULse:RUNT?

(TDS 510A, 500D, 600B, & 700D) (Query Only)

Description: Returns the current parameters for the main pulse runt trigger.

Group: Trigger

Syntax: TRIGger:MAIn:PULse:RUNT?

Example: `TRIGGER:MAIN:PULSE:RUNT?`
might return
`:TRIGGER:MAIN:PULSE:RUNT:POLARITY POSITIVE;THRESHOLD:HIGH 2.00E+0;LOW 800.0E-3.`

TRIGger:MAIn:PULse:RUNT:POLarity

(TDS 510A, 500D, 600B, & 700D)

Description: Sets or queries the polarity for the main pulse runt trigger. This command is equivalent to selecting **Polarity** in the Trigger menu.

Group: Trigger

Syntax 1: TRIGger:MAIn:PULse:RUNT:POLarity { EITher **|** NEGAtive **|** POSITIVe }

Syntax 2: TRIGger:MAIn:PULse:RUNT:POLarity?

Argument 1: 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.

Argument 2: 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.

Argument 3: EITher indicates either NEGAtive or POSITIVe polarity.

Example: `TRIGGER:MAIN:PULSE:RUNT:POLARITY NEGATIVE`
specifies that the polarity of the main pulse runt trigger is negative.

TRIGger:MAIn:PULse:RUNT:THReshold?

(TDS 510A, 500D, 600B, & 700D) (Query Only)

Description: Returns the upper and lower thresholds for the main pulse runt trigger.

Group: Trigger

Syntax: TRIGger:MAIn:PULse:RUNT:THReshold?

Example: `TRIGGER:MAIN:PULSE:RUNT:THRESHOLD?`
might return
`:TRIGGER:MAIN:PULSE:RUNT:THRESHOLD:HIGH 2.00E+0;LOW 800.0E-3.`

TRIGger:MAIn:PULse:RUNT:THReshold:BOTh

(TDS 500D, 600B, & 700D)

Description: Sets or queries the trigger level switching thresholds for the main pulse runt trigger. This command is equivalent to setting **Set to TTL** or **Set to ECL** in the Main Pulse Runt Trigger menu's **Thresholds** side menu item.

Group: Trigger

Syntax 1: TRIGger:MAIn:PULse:RUNT:THReshold:BOTh { ECL | TTL }

Syntax 2: TRIGger:MAIn:PULse:RUNT:THReshold:BOTh?

Argument 1: ECL sets the upper threshold to -1.1 V and the lower threshold to -1.5 V.

Argument 2: TTL sets the upper threshold to 1.8 V and the lower threshold to 0.8 V.

Example: `TRIGGER:MAIN:PULSE:RUNT:THRESHOLD:BOTH TTL`
sets the threshold of the pulse runt trigger to the nominal TTL voltage levels.

TRIGger:MAIn:PULse:RUNT:THReshold:HIGH

(TDS 510A, 500D, 600B, & 700D)

Description: Sets or queries the upper limit for the main pulse runt trigger. This command is equivalent to setting the threshold in the Pulse Runt Threshold side menu.

Group: Trigger

Syntax 1: TRIGger:MAIn:PULse:RUNT:THReshold:HIGH <NR3>

Syntax 2: TRIGger:MAIn:PULse:RUNT:THReshold:HIGH?

Argument: <NR3> is the threshold, in volts.

Example: `TRIGGER:MAIN:PULSE:RUNT:THRESHOLD:HIGH 120E-3`
sets the upper limit of the pulse runt trigger to 120 mV.

TRIGger:MAIn:PULse:RUNT:THReshold:LOW

(TDS 510A, 500D, 600B, & 700D)

Description: Sets or queries the lower limit for the main pulse runt trigger. This command is equivalent to setting the threshold in the Pulse Runt Threshold side menu.

Group: Trigger

Syntax 1: TRIGger:MAIn:PULse:RUNT:THReshold:LOW <NR3>

Syntax 2: TRIGger:MAIn:PULse:RUNT:THReshold:LOW?

Argument: <NR3> is the threshold, in volts.

Example: `TRIGGER:MAIN:PULSE:RUNT:THRESHOLD:LOW 50E-3`
sets the lower limit of the pulse runt trigger to 50 mV.

TRIGger:MAIn:PULse:RUNT:WHEn

(TDS 500D, 600B, & 700D)

Description: Sets or queries the type of pulse width the trigger checks for when it uncovers a runt. This is equivalent to selecting **Trigger When** from the main Trigger's Pulse, Runt menu and **Occurs** or **Wider Than** in the resulting side menu.

Group: Trigger

Syntax 1: TRIGger:MAIn:PULse:RUNT:WHEn { OCCurs **|** WIDERthan }

Syntax 2: TRIGger:MAIn:PULse:RUNT:WHEn?

Argument 1: OCCurs specifies a trigger if a runt of any detectable width occurs.

Argument 2: WIDERthan specifies a trigger if a runt of greater than the specified width occurs.

Example: `TRIGGER:MAIN:PULSE:RUNT:WHEN WIDERTHAN`
sets the runt trigger to occur when the oscilloscope detects a runt in a pulse wider than the specified width.

TRIGger:MAIn:PULse:RUNT:WIDth

(TDS 500D, 600B, & 700D)

Description: Sets or queries the minimum width for a valid main pulse runt trigger. This command is equivalent to entering a value in the Trigger menu's **Wider Than** side menu.

Group: Trigger

Syntax 1: TRIGger:MAIn:PULse:RUNT:WIDth <NR3>

Syntax 2: TRIGger:MAIn:PULse:RUNT:WIDth?

Argument: <NR3> is the minimum width in seconds.

Example: `TRIGGER:MAIN:PULSE:RUNT:WIDTH 15E-6`
sets the minimum width of the pulse runt trigger to 15 us.

TRIGger:MAIn:PULse:SLEWRate:DELTATime

(TDS 500D, 600B, & 700D)

Description: Sets or queries the delta time used in calculating the slew rate trigger. This is equivalent to selecting **Trigger When** from the main Trigger's Slew Rate menu and **Delta Time** in the resulting side menu.

Group: Trigger

Syntax 1: TRIGger:MAIn:PULse:RUNT:DELTATime <NR3>

Syntax 2: TRIGger:MAIn:PULse:RUNT:DELTATime?

Argument: <NR3> is the delta time in seconds.

Example: `TRIGGER:MAIN:PULSE:SLEWRATE:DELTATIME 15E-6`
sets the slew rate trigger's delta time to 15 us.

TRIGger:MAIn:PULse:SLEWRate:POLarity

(TDS 500D, 600B, & 700D)

Description: Sets or queries the polarity for the main pulse slew rate trigger. This command is equivalent to selecting **Polarity** in the Trigger menu with **Type** set to **SlewRate**.

Group: Trigger

Syntax 1: TRIGger:MAIn:PULse:RUNT:POLarity { EITher **|** NEGAtive **|** POSITIVe }

Syntax 2: TRIGger:MAIn:PULse:RUNT:POLarity?

Argument 1: NEGAtive indicates that a pulse edge must traverse from the upper (most positive) to lower (most negative) level for slew rate triggering to occur.

Argument 2: POSITIVe indicates that a pulse edge must traverse from the lower (most negative) to higher (most positive) level for slew rate triggering to occur.

Argument 3: EIThe r indicates either NEGAtive or POSITIVe polarity.

Example: `TRIGGER:MAIN:PULSE:SLEWRATE:POLARITY EITHER`
specifies that the polarity of the slew rate can be either positive or negative.

TRIGger:MAIn:PULse:SLEWRate:SLEWRate?

(TDS 500D, 600B, & 700D) (Query Only)

Description: Returns the slew rate value. This is the:

((Upper Threshold - Lower Threshold) / Delta Time)

The value is limited to the three most significant digits.

Group: Trigger

Syntax 1: TRIGger:MAIn:PULse:SLEWRate:SLEWRate?

Example: `TRIGGER:MAIN:PULSE:SLEWRATE:SLEWRATE?`
returns the slew rate. It is given as an appropriate amount of volts per second. For example, 1.2E+6 would represent a slew rate of 1.2 megavolts/second.

TRIGger:MAIn:PULse:SLEWRate:THReshold:BOTh

(TDS 500D, 600B, & 700D)

Description: Sets the upper and lower slew rate trigger thresholds. This is equivalent to selecting **Thresholds** from the main Trigger's Slew Rate menu and entering a value in the resulting side menu's **High Threshold** or **Low Threshold** items.

Group: Trigger

Syntax: TRIGger:MAIn:PULse:SLEWRate:THReshold:BOTh { TTL **|** ECL }

Argument 1: TTL sets the upper threshold to 1.8 V and the lower threshold to 0.8 V.

Argument 2: ECL sets the upper threshold to -1.1 V and the lower threshold to -1.5 V.

Example: `TRIGGER:MAIN:PULSE:SLEWRATE:THRESHOLD:BOTH TTL`
sets the trigger threshold to TTL.

TRIGger:MAIn:PULse:SLEWRate:THReshold:HIGH

(TDS 500D, 600B, & 700D)

Description: Sets or queries the upper (most positive) limit of the two threshold levels that a pulse edge must traverse for the slew rate trigger to occur. This command is equivalent to setting the higher threshold in the Pulse Slew Rate Trigger's **Thresholds** side menu item.

Group: Trigger

Syntax 1: TRIGger:MAIn:PULse:SLEWRate:THReshold:HIGH <NR3>

Syntax 2: TRIGger:MAIn:PULse:SLEWRate:THReshold:HIGH?

Argument: <NR3> is the threshold, in volts

Example: `TRIGGER:MAIN:PULSE:SLEWRATE:THRESHOLD:HIGH 120E-3`
sets the upper limit of the pulse slew rate trigger to 120 mV.

TRIGger:MAIn:PULse:SLEWRate:THReshold:LOW

(TDS 500D, 600B, & 700D)

Description: Sets or queries the lower (most negative) limit of the two threshold levels that a pulse edge must traverse for the slew rate trigger to occur. This command is equivalent to setting the lower threshold in the Pulse Slew Rate Trigger's **Thresholds** side menu item.

Group: Trigger

Syntax 1: TRIGger:MAIn:PULse:SLEWRate:THReshold:LOW <NR3>

Syntax 2: TRIGger:MAIn:PULse:SLEWRate:THReshold:LOW?

Argument: <NR3> is the threshold, in volts

Example: `TRIGGER:MAIN:PULSE:SLEWRATE:THRESHOLD:LOW 50E-3`
sets the lower limit of the pulse slew rate trigger to 50 mV.

TRIGger:MAIn:PULse:SLEWRate:WHEn

(TDS 500D, 600B, & 700D)

Description: Sets or queries whether to check for a slewing signal that is faster or slower than the specified delta time. This is equivalent to selecting **Trigger When** from the main Trigger's Slew Rate menu and entering a value in the resulting side menu's **Trig if Less Than** or **Trig if Greater Than** items.

Group: Trigger

Syntax 1: TRIGger:MAIn:PULse:SLEWRate:THReshold:WHEn { FASTERthan | SLOWERthan }

Syntax 2: TRIGger:MAIn:PULse:SLEWRate:THReshold:WHEn?

Argument 1: FASTERthan sets the trigger to occur when the slew is faster than the set volts/second rate.

Argument 2: SLOWERthan sets the trigger to occur when the slew is slower than the set volts/second rate.

Example: `TRIGGER:MAIN:PULSE:SLEWRATE:WHEN FASTERTHAN`
sets the slew rate trigger to work when the slew is faster than the set volts/second rate.

TRIGger:MAIn:PULse:SOUrce

(TDS 510A, 500D, 600B, & 700D)

Description: Sets or queries the source for the main pulse trigger. This is equivalent to selecting the source in the Pulse Runt Source side menu.

Group: Trigger

Syntax 1: TRIGger:MAIn:PULse:SOUrce CH<x>

Syntax 2: TRIGger:MAIn:PULse:SOUrce?

Argument: CH<x> specifies one of the input channels.

Example: `TRIGGER:MAIN:PULSE:SOURCE CH2`
selects Channel 2 as the source for the main pulse trigger.

TRIGger:MAIn:PULse:TIMEOut:POLarity

(TDS 500D, 600B, & 700D Only)

Description: Sets or queries the polarity for the main pulse timeout trigger. This command is equivalent to selecting **Polarity** in the Trigger menu with **Type** set to **Time out**.

Group: Trigger

Syntax 1: TRIGger:MAIn:PULse:TIMEOut:POLarity { STAYSHigh (or POSITIVe) **|** STAYSLow (or NEGAtive) **|** EITHer }

Syntax 2: TRIGger:MAIn:PULse:TIMEOut:POLarity?

Argument 1: STAYSLow (or NEGAtive) indicates that a pulse edge must stay low the required time period for timeout triggering to occur.

Argument 2: STAYSHigh (or POSITIVe) indicates that a pulse edge must stay high the required time period for timeout triggering to occur.

Argument 3: EITher indicates either STAYSHigh or STAYSLow polarity.

Example: `TRIGGER:MAIN:PULSE:TIMEOUT:POLARITY EITHER`
specifies that the polarity of the timeout trigger can be either positive or negative.

TRIGger:MAIn:PULse:TIMEOut:TIMe

(TDS 500D, 600B, & 700D Only)

Description: Sets or queries the pulse timeout trigger time, in seconds. This command is equivalent to selecting **Time** in the Trigger menu and entering a value with the keypad or general purpose knob.

Group: Trigger

Syntax 1: TRIGger:MAIn:PULse:TIMEOut:TIMe <NR3>

Syntax 2: TRIGger:MAIn:PULse:TIMEOut:TIMe?

Argument: <NR3> is the timeout time period, in seconds

Example: `TRIGGER:MAIN:PULSE:TIMEOUT:TIME 3.134E-6`
sets the timeout time to 3.134 us.

TRIGger:MAIn:PULse:WIDth?

(TDS 510A, 500D, 600B, & 700D) (Query Only)

Description: Returns the width parameters for the main pulse width trigger.

Group: Trigger

Syntax: TRIGger:MAIn:PULse:WIDth?

Example: `TRIGGER:MAIN:PULSE:WIDTH?`
might return
`:TRIGGER:MAIN:PULSE:WIDTH:LOWLIMIT 2.0E-9;HIGHLIMIT 2.0E-9;WHEN WITHIN;POLARITY POSITIVE`
as the current main pulse trigger parameters.

TRIGger:MAIn:PULse:WIDth:HIGHLimit

(TDS 510A, 500D, 600B, & 700D)

Description: Sets or queries the upper limit for the main pulse width trigger. This is equivalent to setting **Upper Limit** in the Pulse Width Trig When side menu.

Group: Trigger

Syntax 1: TRIGger:MAIn:PULse:WIDth:HIGHLimit <NR3>

Syntax 2: TRIGger:MAIn:PULse:WIDth:HIGHLimit?

Argument: <NR3> is the upper limit, in seconds.

TRIGger:MAIn:PULse:WIDth:LOWLimit

(TDS 510A, 500D, 600B, & 700D)

Description: Sets or queries the lower limit for the main pulse width trigger. This is equivalent to setting **Lower Limit** in the Pulse Width Trig When side menu.

Group: Trigger

Syntax 1: TRIGger:MAIn:PULse:WIDth:LOWLimit <NR3>

Syntax 2: TRIGger:MAIn:PULse:WIDth:LOWLimit?

Argument: <NR3> is the lower limit, in seconds.

TRIGger:MAIn:PULse:WIDth:POLarity

(TDS 510A, 500D, 600B, & 700D)

Description: Sets or queries the polarity for the main pulse width trigger. This is equivalent to selecting the polarity in the Pulse Width Polarity side menu.

Group: Trigger

Syntax 1: TRIGger:MAIn:PULse:WIDth:POLarity { NEGAtive **|** POSITIVe }

Syntax 2: TRIGger:MAIn:PULse:WIDth:POLarity?

Argument 1: NEGAtive specifies a negative pulse.

Argument 2: POSITIVe specifies a positive pulse.

TRIGger:MAIn:PULse:WIDth:WHEn

(TDS 510A, 500D, 600B, & 700D)

Description: Selects the condition when the trigger occurs. This is equivalent to selecting the condition in the Pulse Width Trig When side menu.

Group: Trigger

Syntax 1: TRIGger:MAIn:PULse:WIDth:WHEn { OUTside **|** WIThin }

Syntax 2: TRIGger:MAIn:PULse:WIDth:WHEn?

Argument 1: OUTside specifies a trigger when the duration of the pulse is greater than the high limit or less than the low limit specified. The high and low limits are specified with the TRIGger:MAIn:PULse:WIDth:HIGHLimit and TRIGger:MAIn:PULse:WIDth:LOWLimit commands respectively.

Argument 2: WIThin specifies a trigger when the duration of the pulse is within the high and low limits. The high and low limits are specified with the TRIGger:MAIn:PULse:WIDth:HIGHLimit and TRIGger:MAIn:PULse:WIDth:LOWLimit commands respectively.

Example: `TRIGGER:MAIN:PULSE:WIDTH:WHEN?`
returns either OUTSIDE or WITHIN, indicating the conditions for generating a pulse trigger.

TRIGger:MAIn:TYPe

Description: Sets or queries the type of main trigger. This is equivalent to setting **Type** in the Trigger menu.

Group: Trigger

Syntax 1: TRIGger:MAIn:TYPe { EDGE **|** LOGIc **|** PULse **|** COMMunication (TDS 500D & 700D)**|** VIDeo }
(Only the TDS 510A, 500D, 600B, & 700D use the LOGIc and PULse arguments. Only the TDS 500D and 700D use the COMMunication argument. Only oscilloscopes with option 05 use the VIDeo argument.)

Syntax 2: TRIGger:MAIn:TYPe { EDGE **|** LOGIc **|** PULse **|** COMMunication (TDS 500D & 700D)**|** VIDeo }?

Argument 1: EDGE is a normal trigger. A trigger event occurs when a signal passes through a specified voltage level in a specified direction and is controlled by the TRIGger:MAIn:EDGE commands.

Argument 2: LOGIc (TDS 510A, 500D, 600B, & 700D only) specifies that a trigger occurs when specified conditions are met and is controlled by the TRIGger:MAIn:LOGIc commands.

Argument 3: PULse (TDS 510A, 500D, 600B, & 700D only) specifies that a trigger occurs when a specified pulse is found and is controlled by the TRIGger:MAIn:PULse commands.

Argument 4: COMMunication (TDS 500D & 700D only) specifies that a trigger occurs when a specified signal is found and is controlled by the TRIGger:MAIn:COMMunication commands.

Argument 5: VIDeo (option 05 only) specifies that a trigger occurs when a specified signal is found and is controlled by the TRIGger:MAIn:VIDeo commands.

Example: `TRIGGER:MAIN:TYPE?`
might return `PULSE` indicating that the main trigger type is a pulse trigger.

TRIGger:MAIn:VIDeo?

(Option 05) (Query Only)

Description: Returns the main video trigger parameters.

Group: Trigger

Syntax: TRIGger:MAIn:VIDeo?

Example: `TRIGGER:MAIN:VIDEO?`
might return:
`NTS;CH1;NEGA;NUMER;2;1;COLO;COLO;787;59.94E+0;1050;2;890.0E-9;3.56E-6;15.00E-6;11.56E-6;15.89E-6`
as the current main video trigger parameters.

TRIGger:MAIn:VIDeo:BY

(TDS 400A) (Option 05)

Description: Sets or queries the video trigger delay mode. This is equivalent to using the Video TV Delay Mode side menu.

Group: Trigger

Syntax 1: TRIGger:MAIn:VIDeo:BY { TIMe **|** LINES **|** LINE }

Syntax 2: TRIGger:MAIn:VIDeo:BY?

Argument 1: TIMe specifies a delay by time.

Argument 2: LINES specifies a delay by a number of video lines. For the TDS 400A, this argument is available only for backward compatibility. If the TDS 400A receives this argument, it will convert it to LINE. The TDS 400A will not output this argument in response to a query.

Argument 3: LINE specifies a delay by a number of video lines.

Example: `TRIGGER:MAIN:VIDEO:BY TIME`
specifies a delay by time.

TRIGger:MAIn:VIDeo:FIELD

(Option 05)

Description: Sets or queries the field the video trigger acts on. For the TDS 400A, this is equivalent to using the Video **Scan** side menu when **Class** is NOT set to **Custom**.

For the TDS 510A, 500D, 600B, & 700D, this is equivalent to pressing **Field** in the video main menu, **Field** in the side menu, and entering a value with the keypad or general purpose knob.

Group: Trigger

Syntax 1 (TDS 400A)**:** TRIGger:MAIn:VIDeo:FIELD { ODD **|** EVEN **|** ALL **|** FIELD1 **|** FIELD2 **|** FIELDEither }

Syntax 2 (TDS 510A, 500D, 600B, & 700D)**:** TRIGger:MAIn:VIDeo:FIELD <NR1>

Syntax 3 (TDS 400A)**:** TRIGger:MAIn:VIDeo:FIELD?

Argument 1(TDS 400A)**:** ODD specifies interlaced video field 1.

Argument 2(TDS 400A)**:** EVEN specifies interlaced video field 2.

Argument 3(TDS 400A)**:** All specifies alternating both video field 1 and video field 2.

Argument 4(TDS 400A)**:** FIELD1 specifies interlaced video field 1. For the TDS 400A, this argument is available only for backward compatibility. If the TDS 400A receives this argument, it will convert it to ODD. The TDS 400A will not output this argument in response to a query.

Argument 5(TDS 400A)**:** FIELD2 specifies interlaced video field 2. For the TDS 400A, this argument is available only for backward compatibility. If the TDS 400A receives this argument, it will convert it to EVEN. The TDS 400A will not output this argument in response to a query.

Argument 6(TDS 400A)**:** FIELDEither specifies alternating both video field 1 and video field 2. For the TDS 400A, this argument is available only for backward compatibility. If the TDS 400A receives this argument, it will convert it to ALL. The TDS 400A will not output this argument in response to a query.

Argument 7 (TDS 510A, 500D, 600B, & 700D)**:** <NR1> specifies the video (color) field. For example, 1 specifies interlaced video field one and 2 specifies interlaced video field two.

PAL signals have 1 to 8 fields, HDTV signals have 1 or 2, and FlexFormat supports 1 or 2.

Example 1 (TDS 400A): `TRIGGER:MAIN:VIDEO:FIELD FIELD1` selects odd fields.

Example 2 (TDS 510A, 500D, 600B, & 700D):
`TRIGGER:MAIN:VIDEO:FIELD 1` selects field 1.

TRIGger:MAIn:VIDeo:FIELDType

(TDS 510A, 500D, 600B, & 700D) (Option 05)

Description: Sets or queries the field the video trigger acts on. This is equivalent to pressing **Field** in the video main menu and then **Field**, **Odd**, **Even** or **All** in the side menu.

Group: Trigger

Syntax 1: TRIGger:MAIn:VIDeo:FIELDType { NUMERic **|** ALL **|** EVEN **|** ODD }

Syntax 2: TRIGger:MAIn:VIDeo:FIELDType?

Argument 1: NUMERic specifies a selected line in the selected field. If you send this command when the mode is MONO or SECAM, the oscilloscope will generate an SRQ.

Argument 2: ALL specifies a selected line in all fields.

Argument 3: EVEN specifies a selected line in even fields.

Argument 4: ODD specifies a selected line in odd fields.

Example: `TRIGGER:MAIN:VIDEO:FIELDTYPE ALL` selects a selected line in all fields.

TRIGger:MAIn:VIDeo:FLEXformat?

(TDS 510A, 500D, 600B, & 700D) (Option 05) (Query Only)

Description: Returns the main flexible-format video trigger parameters.

Group: Trigger

Syntax: TRIGger:MAIn:VIDeo:FLEXformat?

Example: `TRIGGER:MAIN:VIDEO:FLEXFORMAT?`
might return:
`59.94E+0;1050;2;890.0E-9;3.56E-6;15.00E-6;11.56E-6;15.89E-6`
as the flexible-format video trigger parameters.

TRIGger:MAIn:VIDeo:FLEXformat:FIELDRATE

(TDS 510A, 500D, 600B, & 700D) (Option 05)

Description: Sets or queries the flexible-format video frames per second (e.g. 59.94 frames per second for 1050 and 50 for 1250). This is equivalent to selecting **Setup** from the video main menu (with **FlexFmt** as the **Standard**), **Field Rate** from the side menu, and entering a value with the keypad or the general purpose knob.

Group: Trigger

Syntax 1: TRIGger:MAIn:VIDeo:FLEXformat:FIELDrate { <NR3> }

Syntax 2: TRIGger:MAIn:VIDeo:FLEXformat:FIELDrate?

Argument: <NR3> the field rate.

Example: `TRIGGER:MAIN:VIDEO:FLEXFORMAT:FIELDRATE?`
returns the specified field rate.

TRIGger:MAIn:VIDeo:FLEXformat:FIELDS

(TDS 510A, 500D, 600B, & 700D) (Option 05)

Description: Sets or queries the flexible-format video fields. This is equivalent to pressing **Setup** from the video main menu (with **FlexFmt** as the **Standard**), **Fields** from the side menu, and entering the value with the keypad or the general purpose knob.

Group: Trigger

Syntax 1: TRIGger:MAIn:VIDeo:FLEXformat:FIELDS <NR1>

Syntax 2: TRIGger:MAIn:VIDeo:FLEXformat:FIELDS?

Argument: <NR1> the number of fields in the standard.

Example: `TRIGGER:MAIN:VIDEO:FLEXformat:FIELDS?`
returns the number of fields in the format.

TRIGger:MAIn:VIDeo:FLEXformat:LINES

(TDS 510A, 500D, 600B, & 700D) (Option 05)

Description: Sets or queries the flexible-format video lines in a frame. This is equivalent to pressing **Setup** from the video main menu (with **FlexFmt** as the **Standard**), **Lines** from the side menu, and entering the value with the keypad or the general purpose knob.

Group: Trigger

Syntax 1: TRIGger:MAIn:VIDeo:FLEXformat:LINES { <NR3> }

Syntax 2: TRIGger:MAIn:VIDeo:FLEXformat:LINES?

Argument: <NR3> the frame lines.

Example: `TRIGGER:MAIN:VIDEO:FLEXFORMAT:LINES?`
returns the specified number of lines.

TRIGger:MAIn:VIDeo:FLEXformat:NEGSynchwidth

(TDS 510A, 500D, 600B, & 700D) (Option 05)

Description: Sets or queries the flexible-format negative sync width. The HDTV horizontal sync is a tri-level sync. The first of the two consecutive sync pulses used is negative and the second is positive. The positive sync pulse starts on the rising edge of the negative sync. The two pulses have the same width such that specifying the negative pulse is only required. Setting the width is equivalent to pressing **Setup** from the video main menu (with **FlexFmt** as the **Standard**), **Sync Width** from the side menu, and entering the value with the keypad or the general purpose knob.

Group: Trigger

Syntax 1: TRIGger:MAIn:VIDeo:FLEXformat:NEGSyncwidth <NR3>

Syntax 2: TRIGger:MAIn:VIDeo:FLEXformat:NEGSyncwidth?

Argument: <NR3> the negative sync width.

Example: `TRIGGER:MAIN:VIDEO:FLEXFORMAT:NEGSYNCWIDTH?`
returns the specified flexible-format negative sync width.

TRIGger:MAIn:VIDeo:FLEXformat:V1STArttime

(TDS 510A, 500D, 600B, & 700D) (Option 05)

Description: Sets or queries the time from the positive edge of the tri-sync pulse for the last line in the selected field (t0) to the leading edge (negative) of the first negative vertical sync pulse. This is equivalent to selecting **Setup** from the video main menu (with **FlexFmt** as the **Standard**), pressing **V1 Start Time** in the side menu, and entering a value with the keypad or the general purpose knob.

Group: Trigger

Syntax 1: TRIGger:MAIn:VIDeo:FLEXformat:V1STArttime <NR3>

Syntax 2: TRIGger:MAIn:VIDeo:FLEXformat:V1STArttime?

Argument: <NR3> the v1 starttime.

Example: `TRIGGER:MAIN:VIDEO:FLEXFORMAT:V1STARTTIME?`
returns the specified HDTV v1starttime.

TRIGger:MAIn:VIDeo:FLEXformat:V1STOptime

(TDS 510A, 500D, 600B, & 700D) (Option 05)

Description: Sets or queries the time from t0 to the trailing edge (positive) of the first negative vertical sync pulse. This is equivalent to selecting **Setup** from the video main menu (with **FlexFmt** as the **Standard**), pressing **V1 Stop Time** in the side menu, and entering a value with the keypad or the general purpose knob.

Group: Trigger

Syntax 1: TRIGger:MAIn:VIDeo:FLEXformat:V1STOptime <NR3>

Syntax 2: TRIGger:MAIn:VIDeo:FLEXformat:V1STOptime?

Argument: <NR3> the v1 stoptime.

Example: `TRIGGER:MAIN:VIDEO:FLEXFORMAT:V1STOPTIME?`
returns the specified v1stoptime.

TRIGger:MAIn:VIDeo:FLEXformat:V2STArttime

(TDS 510A, 500D, 600B, & 700D) (Option 05)

Description: Sets or queries the time from the positive edge of the tri-sync pulse for the last line in the selected field (t0) to the leading edge (positive) of the second vertical sync pulse. Note: the second pulse may be a negative pulse or the negative portion of a tri-sync pulse that is within the last line (usually located at the 1/2 line point). This is equivalent to selecting **Setup** from the video main menu (with **FlexFmt** as the **Standard**), pressing **V2 Start Time** in the side menu, and entering a value with the keypad or the general purpose knob.

Group: Trigger

Syntax 1: TRIGger:MAIn:VIDeo:FLEXformat:V2STArttime <NR3>

Syntax 2: TRIGger:MAIn:VIDeo:FLEXformat:V2STArttime?

Argument: <NR3> the v2 starttime.

Example: `TRIGGER:MAIN:VIDEO:FLEXFORMAT:V2STARTTIME?`
returns the specified v2 starttime.

TRIGger:MAIn:VIDeo:FLEXformat:V2STOptime

(TDS 510A, 500D, 600B, & 700D) (Option 05)

Description: Sets or queries the time from t0 to the trailing edge (positive) of the second negative vertical sync pulse. This is equivalent to selecting **Setup** from the video main menu (with **FlexFmt** as the **Standard**), pressing **V2 Stop Time** in the side menu, and entering a value with the keypad or the general purpose knob.

Group: Trigger

Syntax 1: TRIGger:MAIn:VIDeo:FLEXformat:V2STOptime <NR3>

Syntax 2: TRIGger:MAIn:VIDeo:FLEXformat:V2STOptime?

Argument: <NR3> the v2 stoptime.

Example: `TRIGGER:MAIN:VIDEO:FLEXFORMAT:V2STOPTIME?`
returns the specified v2 stoptime.

TRIGger:MAIn:VIDeo:HDTv

(TDS 510A, 500D, 600B, & 700D) (Option 05)

Description: Sets or queries the high definition TV frame rate. This is equivalent to toggling **HDTV** from the Video main-menu **Standard** pop-up, pressing **Format**, and then selecting a frame rate from the side menu.

Group: Trigger

Syntax 1: TRIGger:MAIn:VIDeo:HDTv <NR1>

Syntax 2: TRIGger:MAIn:VIDeo:HDTv?

Argument: <NR1> specifies the frame rate. More precisely, 787 selects a 787/59.94/2:1 format. 1050 selects a 1050/59.94/2:1 format. 1125 selects a 1125/60/2:1 format. 1250 selects a 1250/50/2:1 format.

Example: `TRIGGER:MAIN:VIDEO:HDTV 787`
specifies 787/59.94/2:1 format.

TRIGger:MAIn:VIDeo:HOLdoff?

(TDS 400A) (Option 05) (Query Only)

Description: Returns the video trigger holdoff value.

Group: Trigger

Syntax: TRIGger:MAIn:VIDeo:HOLdoff?

Example: `TRIGGER:MAIN:VIDEO:HOLDOFF?`
might return :TRIGGER:MAIN:VIDEO:HOLDOFF:VALUE 0.

TRIGger:MAIn:VIDeo:HOLdoff:VALue

(TDS 400A) (Option 05)

Description: Sets or queries the video trigger holdoff value. This is equivalent to setting **Holdoff** in the Mode & Holdoff side menu of the video trigger menu.

Group: Trigger

Syntax 1: TRIGger:MAIn:VIDeo:HOLdoff:VALue <NR1>

Syntax 2: TRIGger:MAIn:VIDeo:HOLdoff:VALue?

Argument: <NR1> is from 0 to 100, and is a percent of the holdoff range.

Example: `TRIGGER:MAIN:HOLDOFF:VALUE 10`
set the holdoff value to be 10% of the holdoff range.

TRIGger:MAIn:VIDeo:INTERLAce

(TDS 400A) (Option 05)

Description: Sets or queries the video trigger interlace format. This is equivalent to setting **Interlace** in the **Scan Rate and Interlace** main menu of the video trigger menu when **Class** is set to **Custom**.

Group: Trigger

Syntax 1:
TRIGger:MAIn:VIDeo:INTERLAce { FIELD1 | FIELD2 | FIELDEither | OFF }

Syntax 2: TRIGger:MAIN:VIDeo:INTERLAce?

TRIGger:MAIn:VIDeo:LINE

(Option 05)

Description: Sets or queries the video trigger in terms of a number of lines. This is equivalent to pressing the **Line** (**TV Delay Mode** in the TDS 400A) item in the video main menu, pressing **Line** in the side menu (if needed), and entering a value with the keypad or the general purpose knob.

The minimum line number is the starting line in the field. For PAL, field 1 (odd fields in the TDS 400A) has line 2 (1 in the TDS 400A) as the minimum, and field 2 (even fields in the TDS 400A) has line 315 (314 in the TDS 400A).

Group: Trigger

Syntax 1: TRIGger:MAIn:VIDeo:LINE <NR1>

Syntax 2: TRIGger:MAIn:VIDeo:LINE?

Argument: <NR1> specifies a number of lines to delay by.

Example: `TRIGGER:MAIN:VIDEO:LINE 5`
selects 5 lines for the desired delay period.

TRIGger:MAIn:VIDeo:LINES

(TDS 400A) (Option 05)

Description: Sets or queries the video trigger delay in terms of a number of lines. This is equivalent to entering data in the **Line** item in the Video **TV Delay Mode** side menu. This command is available for backwards compatibility.

Group: Trigger

Syntax 1: TRIGger:MAIn:VIDeo:LINES { NR3 }

Syntax 2: TRIGger:MAIn:VIDeo:LINES?

Argument: <NR3> specifies a number of lines to delay by.

Example: `TRIGGER:MAIN:VIDEO:LINES 5`
selects 5 lines for the desired delay period.

TRIGger:MAIn:VIDeo:NTSc

(TDS 510A, 500D, 600B, & 700D) (Option 05)

Description: Sets or queries the NTSC mode selection. This is equivalent to choosing **525/NTSC** in the video main-menu pop-up, **Mode** in the main menu, and a side menu item (**NTSC** or **Mono**).

Group: Trigger

Syntax 1: TRIGger:MAIn:VIDeo:NTSc { MONo **|** COLOr }

Syntax 2: TRIGger:MAIn:VIDeo:NTSc?

Argument 1: MONo specifies mode for non-color signals.

Argument 2: COLOr specifies mode for color field triggering and enables numeric triggering.

Example: `TRIGGER:MAIN:VIDEO:NTSC MONO`
specifies numeric fields are invalid.

TRIGger:MAIn:VIDeo:PAL

(TDS 510A, 500D, 600B, & 700D) (Option 05)

Description: Sets or queries the PAL video trigger mode selection. This is equivalent to toggling **625/PAL** in the video main-menu pop-up, **Mode** in the main menu, and a side menu item (**PAL**, **Mono**, or **SECAM**).

Group: Trigger

Syntax 1: TRIGger:MAIn:VIDeo:PAL { MONo **|** COLOr **|** SECAm }

Syntax 2: TRIGger:MAIn:VIDeo:PAL?

Argument 1: MONo specifies mode for noncolor signals.

Argument 2: COLOr specifies mode for color field triggering and enables numeric triggering.

Argument 3: SECAm specifies mode for SECAM signals.

Example: `TRIGGER:MAIN:VIDEO:PAL MONO`
specifies non-color PAL signals.

TRIGger:MAIn:VIDeo:SCAN

(TDS 400A) (Option 05)

Description: Sets or queries the video trigger scan parameters. This is equivalent to using the Video **Scan Parameters** side menu.

Group: Trigger

Syntax 1: TRIGger:MAIn:VIDeo:SCAN { RATE1 **|** RATE2 **|** RATE3 **|** RATE4 }

Syntax 2: TRIGger:MAIn:VIDeo:SCAN?

Argument 1: RATE1 specifies a 15 to 20 kHz video line rate.

Argument 2: RATE2 specifies a 20 to 25 kHz video line rate.

Argument 3: RATE3 specifies a 25 to 35 kHz video line rate.

Argument 4: RATE4 specifies a 35 to 64 kHz video line rate.

Example: `TRIGGER:MAIN:VIDEO:SCAN RATE1`
selects rate 1.

TRIGger:MAIn:VIDeo:SCANPeriod

(TDS 400A) (Option 05)

Description: Sets or queries the video trigger scan period.

Group: Trigger

Syntax 1: TRIGger:MAIn:VIDeo:SCANPeriod <NR3>

Syntax 2: TRIGger:MAIn:VIDeo:SCANPeriod?

TRIGger:MAIn:VIDeo:SOUrce

(Option 05)

Description: Sets or queries the source for the main video trigger. This is equivalent to selecting **Source** in the video main menu and a desired channel from the side menu.

Group: Trigger

Syntax 1: TRIGger:MAIn:VIDeo:SOUrce { CH<x> }

Syntax 2: TRIGger:MAIn:VIDeo:SOUrce?

Argument:
CH<x> specifies one of the input channels (CH1, CH2, CH3, or CH4; or CH1, CH2, AUX1, or AUX2 on the TDS 520D, 620B, 680B, & 724D).

Example: `TRIGGER:MAIN:VIDEO:SOURCE CH1`
selects channel 1 as the source for the main video trigger.

TRIGger:MAIn:VIDeo:STANdard

(TDS 510A, 500D, 600B, & 700D) (Option 05)

Description: Sets or queries the video trigger standard. This is equivalent to selecting the standard in the video **Standard** pop-up (**525/NTSC**, **625/PAL**, **HDTV**, or **FlexFmt**).

Group: Trigger

Syntax 1: TRIGger:MAIn:VIDeo:STANdard { NTSC **|** PAL **|** HDTv **|** FLEXformat}

Syntax 2: TRIGger:MAIn:VIDeo:STANdard?

Argument 1: NTSC specifies the NTSC 525/59.94/2:1 standard.

Argument 2: PAL specifies the PAL 625/59.94/2:1 standard.

Argument 3: HDTv allows selection of the following HDTV formats:
787/59.94/1:1, 1050/59.94/2:1, 1050/59.94/2:1, 1125/60/2:1, 1250/50/2:1.

Argument 4: FLEXformat allows the user to specify the video parameters. The default table is 1050/59.94/2:1.

Example: `TRIGGER:MAIN:VIDEO:STANDARD NTSC`
selects the NTSC video format.

TRIGger:MAIn:VIDeo:SYNc

(Option 05)

Description: Sets or queries the video trigger sync polarity. This is equivalent to selecting **Sync Polarity** from the video main menu and a side-menu item (**Neg Sync** or **Pos Sync**).

Group: Trigger

Syntax 1: TRIGger:MAIn:VIDeo:SYNc { POSITIVe **|** NEGAtive }

Syntax 2: TRIGger:MAIn:VIDeo:SYNc?

Argument 1: POSITIVe specifies a positive going voltage.

Argument 2: NEGAtive specifies a negative going voltage.

Example: `TRIGGER:MAIN:VIDEO:SYNC POSITIVE`
selects a positive going voltage for the desired synchronization pulse.

TRIGger:MAIn:VIDeo:SYStem

(TDS 400A) (Option 05)

Description: Sets or queries the video trigger class. This is equivalent to selecting the class in the **Video Class** side menu of the Video menu.

Group: Trigger

Syntax 1: TRIGger:MAIn:VIDeo:SYStem { NTSc | PAL | SECAM | CUSTom }

Syntax 2: TRIGger:MAIn:VIDeo:SYStem?

Argument 1: NTSc selects a condition that adheres to the National Television System Committee standards. Specifically, it assumes a line rate of 525 lines per frame and a frame rate of 30 Hz.

Argument 2: PAL selects a condition that adheres to the Phase Alternate Line standard. Specifically, it assumes a line rate of 625 lines per frame and a frame rate of 25 Hz.

Argument 3: SECAM selects a condition that adheres to the SECAM standard.

Argument 4: CUSTom selects a condition that adheres to the frequency range of the video signal as you have defined them from the available ranges.

Example: `TRIGGER:MAIN:SYSTEM NTSC`
selects triggering to occur on an NTSC compatible signal.

TRIGger:MAIn:VIDeo:TIMe

(TDS 400A) (Option 05)

Description: Sets or queries the video trigger delay time. This is equivalent to entering the time in the **Delay by Time** item of the Video **TV Delay Mode** side menu.

Group: Trigger

Syntax 1: TRIGger:MAIn:VIDeo:TIMe { <NR3> }

Syntax 2: TRIGger:MAIn:VIDeo:TIMe?

Argument: <NR3> specifies a delay time.

Example: `TRIGGER:MAIN:VIDEO:TIME 5E-6`
selects 5 us for the desired delay time.

TRIGger STATE?

(Query Only)

Description: Returns the current state of the triggering system.

Group: Trigger

Syntax: TRIGger:STATE?

Return Value 1: ARMed
indicates that the instrument is acquiring pretrigger information. All triggers are ignored when TRIGger:STATE is ARMING.

Return Value 2: AUTO
indicates that the oscilloscope is in auto mode and acquires data even in the absence of a trigger.

Return Value 3: FASTframe (TDS 500D and 700D)
indicates that the instrument is in FastFrame mode. This means normal trigger status monitoring is turned off. The oscilloscope will not return armed, partial, ready, trigger, or auto while in this state.

Return Value 4: INStavu (TDS 500D and 700D)
indicates that the instrument is in DPO mode.

Return Value 5: PARTial
indicates that the main trigger has occurred and the oscilloscope is waiting for trigger(s) for the delay by events.

Return Value 6: REAdy
indicates that all pretrigger information has been acquired and the oscilloscope is ready to accept a trigger.

Return Value 7: SAVe
indicates that the oscilloscope is in save mode and is not acquiring data.

Return Value 8: TRIGger
indicates that the oscilloscope has seen a trigger and is acquiring the posttrigger information.

Example: `TRIGGER:STATE?`
might return `ARMED`, indicating that pretrigger data is being acquired.

*TRG

(No Query Form)

Description: (Trigger) executes commands that are defined by *DDT.

The Group Execute Trigger (GET) interface message has the same effect as the *TRG command.

Group: Miscellaneous

Related Commands: Alias commands, *DDT

Syntax: *TRG

Example: *TRG
immediately executes all commands that have been defined by *DDT.

*TST?

(Query Only)

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

Group: Miscellaneous

Syntax: *TST?

Return Value: <NR1> and is always 0.

UNLock

(No Query Form)

Description: Unlocks the front panel. This command is equivalent to LOCk NONe.

NOTE: If the oscilloscope is in the Remote With Lockout State (RWLS), the UNLOCk command has no effect. For more information see the ANSI-IEEE Std. 488.1-1987 Standard Digital Interface for Programmable Instrumentation, section 2.8.3 on RL State Descriptions.

Group: Miscellaneous

Related Command: LOCk

Syntax: UNLock ALL

Argument: ALL specifies all front-panel buttons and knobs.

VERBose

Description: Sets and queries the Verbose State that controls the length of keywords on query responses. Keywords can be both headers and arguments. This command does not affect IEEE Std 488.2-1987 Common Commands (those starting with an asterisk).

Group: Miscellaneous

Related Commands: HEADer, *LRN, SET?

Syntax 1: VERBose { OFF **|** ON **|** <NR1> }

Syntax 2: VERBose?

Argument 1: ON or <NR1> NOT = 0 sets the Verbose State true, which returns full-length keywords for applicable setting queries.

Argument 2: OFF or <NR1> = 0 sets the Verbose State false, which returns minimum-length keywords for applicable setting queries.

Example 1: VERBOSE ON
sets the Verbose State true.

Example 2: VERBOSE?
might return the value 1, showing that the Verbose State is true.

*WAI

(No Query Form)

Description: (Wait) Prevents the oscilloscope from executing further commands or queries until all pending operations finish. This command allows you to synchronize the operation of the oscilloscope with your application program. For more information, refer to the section on synchronization methods.

Group: Status and Error

Related Commands: BUSY?, *OPC

Syntax: *WAI

WAVFrm?

(Query Only)

Description: Returns WFMPre? and CURVe? data for the waveform or waveforms as specified by the DATa:SOUrce command. This command is equivalent to sending WFMPre? and CURVe?.

TDS 400A: when the TDS 400A is in external clock mode, the time-per-div field will contain 50 clks/div.

Group: Waveform

Related Commands: CURVe?, DATa:SOURce, WFMPre?

Syntax: WAVFrm?

WFMPre?

(Query Only)

Description: Returns the waveform formatting data for the first ordered waveform as specified by the DATa:SOUrce command. The channel and math waveforms selected by the DATa:SOUrce command must be displayed.

TDS 400A: when the TDS 400A is in external clock mode, the time-per-div field will contain 50 clks/div.

Group: Waveform

Related Command: WAVFrm?

Syntax: WFMPre?

Return Value: The format of the response is:
BYT_Nr <NR1>;BIT_Nr <NR1>;ENCdg { ASC | BIN };BN_Fmt { RI | RP };BYT_Or { LSB | MSB }; <wfm>:WFID <Qstring>;NR_PT <NR1>;PT_FMT { ENV | Y };XUNit <QString>;XINcr <NR3>;PT_Off <NR1>;YUNit <QString>;YMUlt <NR3>; YOFf <NR3>;YZEro<NR3>[;<wfm>:WFID <Qstring>;NR_PT <NR1>;PT_FMT{ ENV | Y };XUNit<QString>;XINcr <NR3>;PT_Off <NR1>;YUNit <QString>;YMUlt <NR3>; YOFf <NR3>;YZEro <NR3>...]

WFMPre:BIT_Nr

Description: Returns the number of bits per binary waveform point for the first ordered waveform as specified by the DATa:SOUrce command. The WFMPre:BIT_Nr command is ignored on input.

Group: Waveform

Related Commands: DATa:WIDth, WFMPre:BYT_Nr

Syntax 1: WFMPre:BIT_Nr <NR1>

Syntax 2: WFMPre:BIT_Nr?

Argument: <NR1> is either 8 or 16, and is equivalent to WFMPre:BYT_Nr * 8.

Example: `WFMPRE:BIT_NR?`
might return 8, indicating that there are 8 bits per waveform point.

WFMPre:BN_Fmt

Description: Sets or queries the format of binary data for the first ordered waveform as specified by the DATa:SOUrce command.

Group: Waveform

Related Commands: DATa:ENCdg, WFMPre:BYT_Or, WFMPre:ENCdg

Syntax 1: WFMPre:BN_Fmt { RI **|** RP }

Syntax 2: WFMPre:BN_Fmt?

Argument 1: RI specifies signed integer data-point representation.

Argument 2: RP specifies positive integer data-point representation.

Example 1: `WFMPRE:BN_FMT RP`
specifies that the binary waveform data are positive integer data-points.

Example 2: `WFMPRE:BN_FMT?`
returns either `RI` or `RP` as the current waveform data format.

WFMPre:BYT_Nr

Description: Sets or queries the binary field data width for the first ordered waveform as specified by the DATa:SOUrce command. This command is equivalent to the DATa:WIDth command.

Group: Waveform

Related Commands: DATa:WIDth, WFMPre:BIT_Nr

Syntax 1: WFMPre:BYT_Nr <NR1>

Syntax 2: WFMPre:BYT_Nr?

Argument: <NR1> is the number of bytes per point and can be 1 or 2.

Example: `WFMPRE:BYT_NR 2`
specifies that there are 2 bytes per waveform data point.

WFMPre:BYT_Or

Description: Selects which byte of binary waveform data is transmitted first during a waveform data transfer when DATa:WIDth (or WFMPre:BYT_Nr) is set to 2.

Group: Waveform

Related Commands: DATa:ENCdg, WFMPre:BN_Fmt, WFMPre:ENCdg

Syntax 1: WFMPre:BYT_Or { LSB | MSB }

Syntax 2: WFMPre:BYT_Or?

Argument 1: LSB selects the least significant byte to be transmitted first.

Argument 2: MSB selects the most significant byte to be transmitted first.

Example 1: `WFMPRE:BYT_OR MSB`
specifies that the most significant byte in the waveform data will be transferred first.

Example 2: `WFMPRE:BYT_OR?`
returns either MSB or LSB depending on which data byte is transferred first.

WFMPre:ENCdg

Description: Sets or queries the type of encoding for waveform data transferred with the CURVe command.

Group: Waveform

Related Commands: DATa:ENCdg, WFMPre:BYT_Or, WFMPre:BN_Fmt

Syntax 1: WFMPre:ENCdg { ASC | BIN }

Syntax 2: WFMPre:ENCdg?

Argument 1: ASC specifies ASCII curve data.

Argument 2: BIN specifies binary curve data.

Example 1: `WFMPRE:ENCDG ASC`
specifies that the waveform data is in ASCII format.

Example 2: `WFMPRE:ENCDG?`
might return BIN, indicating that the waveform data is in binary format.

WFMPre:PT_Fmt

(No Query Form)

Description: Selects the point format of the waveform data for the first ordered waveform as specified by the DATa:SOUrce command.

Group: Waveform

Syntax: WFMPre:PT_Fmt { ENV **|** Y }

Argument 1: ENV specifies that the waveform is transmitted as maximum and minimum point pairs. Only y values are explicitly transmitted. Absolute coordinates are given by:

X(n) = 0 + XINcr (n - PT_Off)
Y(n(max) = YZEro + YMUlt (y(n(max)) - YOFf)
Y(n(min)) = YZEro + YMUlt (y(n(min)) - YOFf)

Argument 2: Y specifies a normal waveform where one ASCII or binary data point is transmitted for each point in the waveform record. Only y values are explicitly transmitted. Absolute coordinates are given by:

X(n) = 0 + XINcr (n - PT_Off)
Y(n) = YZEro + YMUlt (y(n) - YOFf)

Example: `WFMPRE:PT_Fmt ENV`
sets the waveform data point format to enveloped.

WFMPre:PT_Off

(No Query Form)

Description: Specifies the trigger point within the waveform record for the reference waveform specified by the DATa:DESTination command.

Group: Waveform

Related Command: HORizontal:TRIGger:POsition

Syntax: WFMPre:PT_Off <NR1>

Argument: <NR1> = 0 to the record length, and is the position of the trigger point relative to DATa:STARt.

Example: `WFMPRE:PT_OFF 1`
specifies that the trigger point is the first point in the waveform record.

WFMPre:XINcr

(No Query Form)

Description: Specifies the horizontal sampling interval for the reference waveform specified by the DATa:DESTination command.

Group: Waveform

Syntax: WFMPre:XINcr <NR3>

Argument: <NR3> is the sampling interval in seconds per point.

WFMPre:YMUlt

(No Query Form)

Description: Specifies the vertical scale factor for the reference waveform specified by the DATa:DESTination command.

Group: Waveform

Syntax: WFMPre:YMUlt <NR3>

Argument: <NR3> is the vertical scale factor in YUNits (usually volts) per division.

WFMPre:YOFf

(No Query Form)

Description: Specifies the offset of the vertical component for the reference waveform specified by the DATa:DESTination command.

Group: Waveform

Syntax: WFMPre:YOFf <NR3>

Argument: <NR3> is the vertical offset in digitizing levels.

WFMPre:YZEro

(No Query Form)

Description: Specifies the offset voltage for the reference waveform specified by the DATa:DESTination command.

Group: Waveform

Syntax: WFMPre:YZEro <NR3>

Argument: <NR3> is of the offset in YUNits (usually volts).

Table 2-32 lists additional WFMPre commands that are included for compatibility purposes.

NOTE: These commands do not support a query form and all information is ignored.

Table 2-32: Additional WFMPre Commands

Command	Argument	Description
WFMPre:CRVchk	{CHKSM0|NONe}	Binary curve error check
WFMPre:NR_PT	<NR1>	Number of waveform points
WFMPre:WFId	<QString>	Waveform identifier
WFMPre:XUNit	<QString>	Horizontal units
WFMPre:XMUlt	<NR3>	Horizontal (X-axis) scale factor
WFMPre:XOFf	<NR3>	Horizontal (X-axis) offset
WFMPre:XZEro	<NR3>	Horizontal (X-axis) origin offset
WFMPre:YUNit	<QString>	Vertical units
WFMPre:ZMUlt	<NR3	Z-axis scale factor
WFMPre:ZOFf	<NR3>	Z-axis offset
WFMPre:ZUNit	<QString>	Z-axis units
WFMPre:ZZEro	<NR3>	Z-axis origin offset

NOTE: When returning WFMPRE:<wfm> information from the oscilloscope, <wfm> specifies the waveform source (CH<x>, MATH<x>, or REF<x>). The source must also be set using the DAta:SOUrce command. When sending WFMPRE:<wfm> information to the scope, the <wfm> specification is ignored and the reference location specified by DATa:DESTination is used instead.

WFMPre:<wfm>?

(Query Only)

Description: Returns the waveform formatting data for first ordered waveform as specified by the DATa:SOUrce command. Channel and math waveforms must be displayed before they can be queried. Querying an invalid reference waveform generates an execution error.

In extended-acquisition-length mode, it will return the acquisition record information as opposed to the waveform record information.

Group: Waveform

Syntax: WFMPre:<wfm>?

Return Value: The format of the response is:
<wfm>:WFID <Qstring>;NR_PT <NR1>;PT_FMT { ENV | Y };XUNit <QString>;XINcr <NR3>;PT_Off <NR1>;YUNit <QString>;YMUlt <NR3>;YOFf <NR3>;YZEro <NR3> [;<wfm>:WFID <Qstring>;NR_PT <NR1>;PT_FMT { ENV | Y };XUNit <QString>;XINcr <NR3>;PT_Off <NR1>;YUNit <QString>;YMUlt <NR3>;YOFf <NR3>;YYZEro <NR3>...

WFMPre:<wfm>:NR_Pt

Description: Sets or queries the number of points that are in the transmitted waveform record. This value is ignored on input.

In extended-acquisition-length mode, it will return the acquisition record information as opposed to the waveform record information.

Related Command: DATa:DESTination

Group: Waveform

Syntax 1: WFMPre:<wfm>:NR_Pt <NR1>

Syntax 2: WFMPre:<wfm>:NR_Pt?

Argument: <NR1> is the number of data points. If DATa:WIDth is 2 then there are twice as many bytes.

<NR1> = 0 means that the waveform record is of an unspecified length.

Example: `WFMPRE:CH1:NR_Pt?`
might return 5000 as the number of data points in the waveform record transferred from channel 1.

WFMPre:<wfm>:PT_Fmt

Description: Selects the data point format for the first ordered waveform as selected by the DATa:SOUrce command. On input <wfm> always defaults to the reference location specified by DATa:DESTination regardless of what is sent.

Group: Waveform

Related Command: DATa:DESTination

Syntax 1: WFMPre:<wfm>:PT_Fmt { ENV **|** Y }

Syntax 2: WFMPre:<wfm>:PT_Fmt?

Argument 1: ENV specifies that the waveform is transmitted as minimum and maximum point pairs. Only y values are explicitly transmitted. Absolute coordinates are given by:
...
X(n) = 0 + XINcr (n - PT_Off)
Y(n(min) = YZEro + YMUlt (y(n(min)) - YOFf)
Y(n(max)) = YZEro + YMUlt (y(n(max)) - YOFf)

....
Argument 2: Y specifies a normal waveform where one ASCII or binary data point is transmitted for each point in the waveform record. Only y values are explicitly transmitted. Absolute coordinates are given by:

X(n) = 0 + XINcr (n - PT_Off)
Y(n) = YZEro + YMUlt (y(n) - YOFf)

Example: `WFMPRE:MATH1:PT_FMT?`
might return ENV, indicating that the MATH1 waveform data format is enveloped.

WFMPre:<wfm>:PT_Off

Description: Returns the trigger point within the waveform record. On input <wfm> always defaults to the reference location specified by DATa:DESTination regardless of what is sent.

Group: Waveform

Syntax 1: WFMPre:<wfm>:PT_Off <NR1>

Syntax 2: WFMPre:<wfm>:PT_Off?

Argument: <NR1> = 0 to the recordlength, and is the position of the trigger point relative to DATa:STARt when queried.

In extended-acquisition-length mode, <NR1> refers to the acquisition length.

Example: WFMPRE:CH1:PT_OFF?
returns 0 indicating the trigger position within the waveform record.

WFMPre:<wfm>:WFId

Description: Returns information about the waveform such as input coupling, volts per division, time per division, acquisition mode, and record length.

The WFMPre:<wfm>:WFId command is ignored on input.

TDS 400A: when the TDS 400A is in external clock mode, the time-per-div field will contain "50 clks/div".

Group: Waveform

Syntax 1: WFMPre:<wfm>:WFId <QString>

Syntax 2: WFMPre:<wfm>:WFId?

Argument: <QString> is the waveform identifier string.

WFMPre:<wfm>:XINcr

Description: Sets or queries the horizontal sampling interval. On input <wfm> always defaults to the reference location specified by DATa:DESTination regardless of what is sent.

TDS 400A: when the TDS 400A is in external clock mode, the time-per-div field will contain "50 clks/div".

Group: Waveform

Syntax 1: WFMPre:<wfm>:XINcr <NR3>

Syntax 2: WFMPre:<wfm>:XINcr?

Argument: <NR3> is the sampling interval.

WFMPre:<wfm>:XUNit

Description: Returns the horizontal (X-axis) units of the waveform data at the time of creation.

The WFMPre:<wfm>:XUNit command is ignored on input.

TDS 400A: when the TDS 400A is in external clock mode, the time-per-div field will contain "50 clks/div".

Group: Waveform

Syntax 1: WFMPre:<wfm>:XUNit <QString>

Syntax 2: WFMPre:<wfm>:XUNit?

Argument: <QString> is "s" for seconds and specifies the units.

Example: WFMPRE:CH1:XUNIT?
might return "s", indicating that the horizontal units for channel 1 are seconds.

WFMPre:<wfm>:XZEro

(TDS 500D, 600B, & 700D)

Description: Sets or queries the horizontal (X-axis) origin offset. On input <wfm> always defaults to the reference location specified by DATa:DESTination regardless of what is sent.

Group: Waveform

Syntax 1: WFMPre:<wfm>:XZEro <NR3>

Syntax 2: WFMPre:<wfm>:XZEro?

Argument: <NR3> is the offset in XUNits (usually time).

WFMPre:<wfm>:YMUlt

Description: Sets or queries the vertical scale factor, in YUNit per unscaled data point value. On input <wfm> always defaults to the reference location specified by DATa:DESTination regardless of what is sent.

Group: Waveform

Syntax 1: WFMPre:<wfm>:YMUlt <NR3>

Syntax 2: WFMPre:<wfm>:YMUlt?

Argument: <NR3> is the scale factor, in YUNits (usually volts), per digitizing level.

WFMPre:<wfm>:YOFf

Description: Sets or queries the vertical position of the waveform. On input <wfm> always defaults to the reference location specified by DATa:DESTination regardless of what is sent.

Group: Waveform

Syntax 1: WFMPre:<wfm>:YOFf <NR3>

Syntax 2: WFMPre:<wfm>:YOFf?

Argument: <NR3> is the position in digitizing levels.

WFMPre:<wfm>:YUNit

Description: Returns the vertical (Y-axis) units of the waveform data at the time of creation. The WFMPre:<wfm>:YUNit command is ignored on input.

Group: Waveform

Syntax 1: WFMPre:<wfm>:YUNit <QString>

Syntax 2: WFMPre:<wfm>:YUNit?

Argument: <QString> is "V" for volts or "VV" for volts**2, and specifies the units.

Example: `WFMPRE:CH2:YUNIT?`
might return "V", meaning that the units for the vertical component of the channel 2 waveform data are volts.

WFMPre:<wfm>:YZEro

Description: Sets or queries the vertical (Y-axis) offset voltage. On input <wfm> always defaults to the reference location specified by DATa:DESTination regardless of what is sent.

Group: Waveform

Syntax 1: WFMPre:<wfm>:YZEro <NR3>

Syntax 2: WFMPre:<wfm>:YZEro?

Argument: <NR3> is the offset in YUNits (usually volts).

ZOOm

Description: Resets the zoom transforms to default values for all traces or live traces. The ZOOm query returns the current vertical and horizontal positioning and scaling of the display. This command is equivalent to selecting **Reset All Factors** or **Reset Live Factors** in the TDS 400A, 500D, 600B, or 700D Zoom Reset menu or selecting **Reset Zoom Factors** in the Zoom menu of the TDS 510A.

Group: Zoom

Syntax 1: ZOOm { RESET **|** RESETLive (TDS 400A, 500D, 600B, & 700D) }

Syntax 2: ZOOm?

Argument: RESET resets the zoom transforms to default values for all traces.

Argument: RESet (TDS 400A, 500D, 600B, & 700D) resets the zoom transforms to default values for all live traces.

Example: `ZOOM`?
might return
`:ZOOM:STATE 0;HORIZONTAL:SCALE 1.00E+0;POSITION 500.0E-3;LOCK LIVE;:ZOOM:VERTICAL:SCALE 1.0E+0;POSITION 0.0E+0.`

ZOOm:DUAl

(TDS 400A, 500D, 600B, & 700D)

Description: Turns Dual Zoom mode on and off

Group: Zoom

Syntax 1: ZOOm DUAl { OFF **|** ON **|** <NR1> }

Syntax 2: ZOOm:DUAl?

Argument 1: OFF or <NR1> = 0 turns Dual Zoom mode off.

Argument 2: ON or <NR1> NOT = 0 turns Dual Zoom mode on.

Example 1: `ZOOM:DUAL ON` enables the Dual Zoom features.

Example 2: `ZOOM:DUAL?` returns either 0 or 1 depending on the state of the Dual Zoom mode.

ZOOm:DUAl:OFFSet

(TDS 400A, 500D, 600B, & 700D)

Description: Adjusts the requested horizontal offset between the centers of the main and second zoom boxes.

Group: Zoom

Syntax 1: ZOOm DUAl:OFFSet <NR3>

Syntax 2: ZOOm:DUAl:OFFSet?

Argument: <NR3> is the offset time in seconds.

Example: `ZOOM:DUAL:OFFSET 100.0E-6`
adjusts the offset time in seconds between the centers of the main and second zoom boxes.

ZOOm:GRAticule

(TDS 400A, 500D, 600B, & 700D)

Description: Selects between the upper and lower graticule for use by the zoom preview state. If you select the lower graticule, horizontal and vertical knob changes will affect the underlying acquisition system. If you select the upper graticule, horizontal and vertical knob changes will affect the zoom factors.

Group: Zoom

Syntax 1: ZOOm GRAticule { LOWEr **|** UPPer }

Syntax 2: ZOOm:GRAticule?

Argument 1: LOWEr selects the lower graticule.

Argument 2: UPPer selects the upper graticule.

Example: `ZOOM:GRATICULE?`
might return `UPPER`, indicating that the upper graticule is selected.

ZOOm:HORizontal:LOCk

Description: Specifies the waveforms that the horizontal zoom parameters affect. This is equivalent to setting **Horizontal Lock** in the Zoom side menu.

Group: Zoom

Syntax 1: ZOOm:HORizontal:LOCk { ALL **|** LIVe **|** NONe }

Syntax 2: ZOOm:HORizontal:LOCk?

Argument 1: LIVe specifies that all live (CH<x>) waveforms will be horizontally positioned and scaled together.

In extended-acquisition-length mode, LIVe is the only valid argument.

Argument 2: NONe specifies that only the selected waveform is positioned and scaled using the horizontal zoom parameters.

Argument 3: ALL specifies that all (CH<x>, Ref<x>, Math<x>) waveforms will be horizontally positioned and scaled together.

Example 1: `ZOOM:HORIZONTAL:LOCK LIVE`
specifies that all live waveforms are positioned and scaled together.

Example 2: `ZOOM:HORIZONTAL:LOCK?`
returns either `LOCK` or `NONE`.

ZOOm:HORizontal:POSition

Description: Sets or queries the horizontal position of zoomed waveforms. The setting of the ZOOm:HORizontal:LOCk command determines the waveforms affected. For example, if ZOOm:HORizontal:LOCk is set to LIVe then only live (as opposed to reference and math) waveforms are affected.

In extended-acquisition-length mode, sets or queries the horizontal position of the zoomed acquisition record.

Group: Zoom

Syntax 1: ZOOm:HORizontal:POSition <NR3>

Syntax 2: ZOOm:HORizontal:POSition?

Argument: <NR3> is from 0 to 100, and is the percent of the waveform that is to the left of the graticule.

Example: `ZOOM:HORIZONTAL:POSITION 50`
centers the waveform on the display.

ZOOm:HORizontal:SCAle

Description: Sets or queries the horizontal expansion factor. This command is equivalent to using the front-panel **Horizontal Scale** knob when Zoom is on. It is also equivalent to using **Zoom Preview** with the upper graticule selected in the TDS 500D, 600B, or 700D.

Group: Zoom

Syntax 1: ZOOm:HORizontal:SCAle <NR3>

Syntax 2: ZOOm:HORizontal:SCAle?

Argument: <NR3> is the amount of expansion in the horizontal direction.

It is less than 1 if there is horizontal compression.

It is more than 1 if there is horizontal expansion.

Example: `ZOOM:HORIZONTAL:SCALE?`
might return `1.00E+0` as the horizontal scale factor.

ZOOm:STATE

Description: Turns Zoom mode on and off. When Zoom mode is on, the horizontal and vertical position and scale commands affect the waveform display, not the acquisition. This is the only way to position and scale math and reference waveforms. This command is equivalent to turning **Zoom** on and off in the Zoom side menu.

Group: Zoom

Syntax 1: ZOOm:STATE { <NR1> **|** OFF **|** ON **|** PREView (not on TDS 510A) }

Syntax 2: ZOOm:STATE?

Argument 1: OFF or <NR1> = 0 turns Zoom mode off.

Argument 2: ON or <NR1> NOT = 0 turns Zoom mode on. When DPO mode is in use, the state value becomes a requested value, to be restored when DPO mode is switched off. Do not use this condition if DPO mode is on. DPO mode overrides the zoom on state.

Argument 3: PREView (not on TDS 510A) sets ZOOM:STATE to preview. Also, it causes the oscilloscope to display both the ZOOM:STATE OFF and ZOOM:STATE ON traces simultaneously in dual, half-height graticules. Do not use this condition if DPO mode is on. DPO mode overrides the zoom preview state.

In extended-acquisition-length mode, entering the zoom preview state causes fit-to-screen to be on. If the oscilloscope exits the extended-acquisition-length mode or the zoom preview state, the fit-to-screen mode returns back to its prior setting.

Example 1: `ZOOM:STATE ON`
enables the Zoom feature.

Example 2: `ZOOM:STATE?`
returns either `OFF`, `PREVIEW`, or `ON` depending on the state of Zoom mode.

ZOOm:VERTical:POSition

Description: Sets or queries the vertical position of waveforms.

Group: Zoom

Syntax 1: ZOOm:VERTical:POSition <NR3>

Syntax 2: ZOOm:VERTical:POSition?

Argument: <NR3> is the vertical position in divisions.

Example: `ZOOM:VERTICAL:POSITION?`
might return `:ZOOM:VERTICAL:POSITION 0`

ZOOm:VERTical:SCAle

Description: Sets or queries the vertical expansion and compression factor.

Group: Zoom

Related Commands: ACQuire:MODe

Syntax 1: ZOOm:VERTical:SCAle <NR3>

Syntax 2: ZOOm:VERTical:SCAle?

Argument: <NR3> is the amount of vertical expansion or compression.

Example: `ZOOM:VERTICAL:SCALE?`
might return `:ZOOM::VERTICAL:SCALE 1.0E+0`

Example Programs

The example programs illustrate methods you can use to control the oscilloscope from the GPIB interface. The diskettes that contain with this help file also contain listings for these programs written in Microsoft QuickBASIC 4.5 and Microsoft QuickC 2.5.

The programs run on a PC compatible system equipped with a Tektronix (National Instruments) GPIB board and associated drivers. For example, the programs will work with a Tektronix S3FG210 (National Instruments GPIB-PCII/IIA) GPIB package.

All the example programs assume that the GPIB system recognizes the oscilloscope as DEV1 and the PC (controller) as GPIB0. You can use the IBCONF.EXE program to assign these names.

Example Software Overview
Compiling the Example Software
Compiling and Linking QuickC Programs
Compiling and Linking QuickBASIC Programs

Example Software Overview

The example software includes:

- MEAS: automatically measures waveform parameters.
- COMM: shows communication between controller and oscilloscope.
- GETWFM: reads a waveform from an oscilloscope and stores it in a file.
- CURSOR: uses cursors to measure waveform parameters.
- TL: a talker-listener program.

Compiling the Example Software

The example programs diskette contains programs written in Microsoft QuickBASIC 4.5 and Microsoft QuickC 2.5.

Executable versions of the programs are in the PROGRAMS directory. Source versions are in the SOURCES directory. Within this directory, the QuickBASIC programs are in the Q-BASIC subdirectory and the QuickC programs are in the QUICK-C subdirectory.

A README file in each directory explains how to build executable code from the source files provided.

The QuickC directory also comes with sample MAKE files and sample executable files. These have the suffix .MAK.

If you wish to develop code, you will need to use files that come with the GPIB system. Specifically, the QuickBASIC programs use QBDECL.BAS and QBIB.OBJ. The QuickC programs use DECL.H and MCIB.OBJ.

NOTE: The programs you compile in the Sources directory work with the Tektronix S3FG210 (National Instruments GPIB-PCII/IIA) GPIB system. It may take extra steps or changes to get them to work with older Tektronix GURU and other GPIB systems.

Compiling and Linking QuickC Programs

To make an executable for any example, perform the following:

1. Install QuickC. Select the SMALL memory model. Be sure to set up your path so DOS can access the QuickC directory.

2. Install the Tektronix S3FG210 (National Instruments GPIB-PCII/IIA) GPIB board and drivers. Remember to identify the GPIB device as DEV1. You can use the IBCONF.EXE program to do this.

3. Copy the files from the examples diskette to your hard disk. You might also create a special directory to store them. For example, if the current drive is hard disk C, you want to store the examples in drive C, and the examples diskette is in drive B, you might type:

   mkdir examples

   cd examples

   copy B:\quick-c*.* .

4. For this installation, you will also want to copy DECL.H and MCIB.OBJ from your Tektronix S3FG210 (National Instruments GPIB-PCII/IIA) GPIB drivers directory to this directory. For example, if the GPIB drivers are in the gpib-pc directory and you are in the example programs directory, you would type:

   copy \gpib-pc\decl.h .

   copy \gpib-pc\mcib.obj .

5. To compile and link your TDS sample C programs, simply type:

   nmake <file name>.mak
   where <file name> refers to the name of the example program you wish to compile and link. Specifically:

   To compile and link MEAS.C, type: nmake meas.mak

   To compile and link COMM.C, type: nmake comm.mak

   To compile and link GETWFM.C, type: nmake getwfm.mak

   To compile and link CURSOR.C, type: nmake cursor.mak

   To compile and link TL.C, type: nmake tl.mak

6. Run the program by simply typing the program name.

   To run meas, type: meas

   To run comm, type: comm

   To run getwfm, type: getwfm

   To run cursor, type: cursor

To run tl, type: tl

Compiling and Linking QuickBASIC Programs

To make an executable for any of the following files, perform the following:

1. Install QuickBASIC.

2. Install the Tektronix S3FG210 (National Instruments GPIB-PCII/IIA) GPIB board and drivers. Remember to reboot your PC to initialize the GPIB drivers.

3. Copy the files from the examples diskette to your hard disk. You might also create a special directory to store them. For example, if the current drive is hard disk C, you want to store the examples in drive C, and the examples diskette is in drive B, you might type:

   mkdir examples

   cd examples

   copy b:\q-basic*.* .

4. For this installation, you will also want to copy QBDECL.BAS and QBIB.OBJ from your Tektronix S3FG210 (National Instruments GPIB-PCII/IIA) GPIB drivers directory to the directory your example programs are in. For example, if the GPIB drivers are in the gpib-pc directory and you are in the example programs directory, you would type:

   copy \gpib-pc\qbdecl.bas .

   copy \gpib-pc\qbib.obj .

5. Perform the following two steps for example programs:

   a. Compile the program by using the following command:

      bc /o <file>.bas;

      where <file> is one of the example program names.

      To compile MEAS.BAS, type: bc /o meas.bas;

      To compile COMM.BAS, type: bc /o comm.bas;

      To compile GETWFM.BAS, type: bc /o getwfm.bas;

      To compile CURSOR.BAS, type: bc /o cursor.bas;

      To compile TL.BAS, type: bc /o tl.bas;

   b. Link the compiled program with the qbib.obj module to create the executable program (file.EXE) by using the following command:

      link <file>.obj+qbib.obj;

      where <file> is one of the above program names.

      To link MEAS.OBJ, type: link meas.obj+qbib.obj;

      To link COMM.OBJ, type: link comm.obj+qbib.obj;

To link GETWFM.OBJ, type: link getwfm.obj+qbib.obj;

To link CURSOR.OBJ, type: link cursor.obj+qbib.obj;

To link TL.OBJ, type: link tl.obj+qbib.obj;

GPIBIO.BAS is a collection of input/output routines used by the other programs and is included for proper file compilation.

6. Run the program by simply typing the program name.

   To run meas, type: meas

   To run comm, type: comm

   To run getwfm, type: getwfm

   To run cursor, type: cursor

   To run tl, type: tl

NOTE: The example programs disable front-panel operation while they are running and reenable it when they terminate. If your program terminates prematurely, front-panel operation may remain disabled. To reenable front-panel operation, do one of the following: cycle power on the oscilloscope or send the GPIB command UNLOCK ALL to unlock the front panel. You can send the UNLOCK ALL command with the TL program included in your sample programs disk.

Status and Events

The oscilloscope provides a status and event reporting system for the GPIB interface. This system informs you of certain significant events that occur within the oscilloscope.

The oscilloscope status handling system consists of five 8-bit registers and two queues. This section describes these registers and components. It also explains how the event handling system operates.

Registers
Queues
Event Handling Sequence
Messages
Synchronization Methods

Registers

The registers in the event handling system fall into two functional groups:

- Status Registers contain information about the status of the oscilloscope. They include the Standard Event Status Register (SESR) and the Status Byte Register (SBR).
- Enable Registers determine whether selected types of events are reported to the Status Registers and the Event Queue. They include the Device Event Status Enable Register (DESER), the Event Status Enable Register (ESER), and the Service Request Enable Register (SRER).

Status Registers

The Standard Event Status Register (SESR) and the Status Byte Register (SBR) record certain types of events that may occur while the oscilloscope is in use. IEEE Std 488.2-1987 defines these registers.

Each bit in a Status Register records a particular type of event, such as an execution error or service request. When an event of a given type occurs, the oscilloscope sets the bit that represents that type of event to a value of one. (You can disable bits so that they ignore events and remain at zero. See the Enable Registers section on a following screen.) Reading the status registers tells you what types of events have occurred.

The Standard Event Status Register (SESR) - The SESR, shown in Figure 3-1, records eight types of events that can occur within the oscilloscope. Use the *ESR? query to read the SESR register. Reading the register clears the bits of the register so that the register can accumulate information about new events.

Figure 3-1: The Standard Event Status Register (SESR)

7	6	5	4	3	2	1	0
PON	URQ	CME	EXE	DDE	QYE	RQC	OPC

Table 3-1: SESR Bit Functions

Bit	Function
7 (MSB)	**PON** (Power On). Shows that the oscilloscope was powered on. The completion of the diagnostic tests also sets this bit.
6	**URQ** (User Request). Shows that an Application menu button was pressed.
5	**CME** (Command Error). Shows that an error occurred while the oscilloscope was parsing a command or query. Command error messages are listed in Table 3-4 below.
4	**EXE** (Execution Error). Shows that an error occurred while the oscilloscope was executing a command or query. Execution error messages are listed in Table 3-5 below.
3	**DDE** (Device Error). Shows that a device error occurred. Device error messages are listed in Table 3-6 below.
2	**QY**E (Query Error).

	Shows that either an attempt was made to read the Output Queue when no data was present or pending, or that data in the Output Queue was lost.
1	**RQC** (Request Control).
	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) - shown in Figure 3-2, records whether output is available in the Output Queue, whether the oscilloscope requests service, and whether the SESR has recorded any events.

Use a Serial Poll or the *STB? query to read the contents of the SBR. The bits in the SBR are set and cleared depending on the contents of the SESR, the Event Status Enable Register (ESER), and the Output Queue. When you use a Serial Poll to obtain the SBR, bit 6 is the RQS bit. When you use the *STB? query to obtain the SBR, bit 6 is the MSS bit. Reading the SBR does not clear the bits.

Figure 3-2: The Status Byte Register (SBR)

7	6	5	4	3	2	1	0
_	RQS or MSS	ESB	MAV	_	_	_	_

Table 3-2: SBR Bit Functions

Bit	Function
7 (MSB)	Not used.
6	RQS (Request Service),
	obtained from a serial poll. Shows that the oscilloscope requests service from the GPIB controller.
6	MSS (Master Status Summary),
	obtained from *STB? query. Summarizes the ESB and MAV bits in the SBR.
5	ESB (Event Status Bit).
	Shows that status is enabled and present in the SESR.
4	MAV (Message Available).
	Shows that output is available in the Output Queue.
3 - 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 its bit in the Status Register, the corresponding bit in the Enable Register must be set to one. If the bit in the Enable Register is set to zero, the event is not recorded.

Various commands set the bits in the Enable Registers. The Enable Registers and the commands used to set them are described below.

The Device Event Status Enable Register (DESER) - is shown in Figure 3-3. This register controls which types of events are reported to the SESR and the Event Queue. The bits in the DESER correspond to those in the SESR, as described earlier.

Use the DESE command to enable and disable the bits in the DESER. Use the DESE? query to read the DESER.

Figure 3-3: The Device Event Status Enable Register (DESER)

7	6	5	4	3	2	1	0
PON	URQ	CME	EXE	DDE	QYE	RQC	OPC

The Event Status Enable Register (ESER) - is shown in Figure 3-4. It controls which types of events are summarized by the Event Status Bit (ESB) in the SBR. Use the *ESE command to set the bits in the ESER. Use the *ESE? query to read it.

Figure 3-4: The Event Status Enable Register (ESER)

7	6	5	4	3	2	1	0
PON	URQ	CME	EXE	DDE	QYE	RQC	OPC

The Service Request Enable Register (SRER) - is shown in Figure 3-5. It controls which bits in the SBR generate a Service Request and are summarized by the Master Status Summary (MSS) bit.

Use the *SRE command to set the SRER. Use the *SRE? query to read it. The RQS bit remains set to one until either the Status Byte Register is read with a Serial Poll or the MSS bit changes back to a zero.

Figure 3-5: The Service Request Enable Register (SRER)

7	6	5	4	3	2	1	0
-	-	ESB	MAV	-	-	-	-

The Enable Registers and the *PSC Command

The *PSC command controls the Enable Registers contents at power-on. Sending *PSC 1 sets the Enable Registers at power on as follows:

- DESER 255 (equivalent to a DESe 255 command)
- ESER 0 (equivalent to an *ESE 0 command)
- SRER 0 (equivalent to an *SRE 0 command)

Sending *PSC 0 lets the Enable Registers maintain their values in non-volatile memory through a power cycle.

NOTE: To enable the PON (Power On) event to generate a Service Request, send *PSC 0, use the DESe and *ESE commands to enable PON in the DESER and ESER, and use the *SRE command to enable bit 5 in the SRER. Subsequent power-on cycles will generate a Service Request.

Queues

The oscilloscope status and event reporting system contains two queues: the Output Queue and the Event Queue.

The Output Queue

The oscilloscope stores query responses in the Output Queue. It empties this queue each time it receives a new command or query message after an <EOM>. The controller must read a query response before it sends the next command (or query) or it will lose responses to earlier queries.

Warning. When a controller sends a query, an <EOM>, and a second query, the oscilloscope normally clears the first response and outputs the second while reporting a Query Error (QYE bit in the ESER) to indicate the lost response. A fast controller, however, may receive a part or all of the first response as well. To avoid this situation, the controller should always read the response immediately after sending any terminated query message or send a DCL (Device Clear) before sending the second query.

The Event Queue

The Event Queue stores detailed information on up to 20 events. If more than 20 events stack up in the Event Queue, the 20th event is replaced by event code 350, "Too many events."

Read the Event Queue with the EVENT? query (which returns only the event number), with the EVMSG? query (which returns the event number and a text description of the event), or with the ALLEV? query (which returns all the event numbers along with a description of the event). Reading an event removes it from the queue.

Before reading an event from the Event Queue, you must use the *ESR? query to read the summary of the event from the SESR. This makes the events summarized by the *ESR? read available to the EVENT? and EVMSG? queries, and empties the SESR.

Reading the SESR erases any events that were summarized by previous *ESR? reads but not read from the Event Queue. Events that follow an *ESR? read are put in the Event Queue but are not available until *ESR? is used again.

Messages

Tables 3-3 through 3-9 shown below list all the programming interface messages the oscilloscope generates in response to commands and queries.

For most messages, a secondary message from the oscilloscope gives more detail about the cause of the error or the meaning of the message. This message is part of the message string, and is separated from the main message by a semicolon.

Each message is the result of an event. Each type of event sets a specific bit in the SESR and is controlled by the equivalent bit in the DESER. Thus, each message is associated with a specific SESR bit. In the message tables that follow, the associated SESR bit is specified in the table title, with exceptions noted with the error message text.

Table 3-3 given below shows the messages when the system has no events or status to report. These have no associated SESR bit.

Table 3-3: No Event Messages

Code	Message
0	No events to report - queue empty
1	No events to report - new events pending *ESR?

Table 3-4 below shows the error messages generated by improper command syntax. Check that the command is properly formed and that it follows the rules in the section on command Syntax.

Table 3-4: Command Error Messages - CME Bit 5

Code	Message
100	Command error
101	Invalid character
102	Syntax error
103	Invalid separator
104	Data type error
105	GET not allowed
106	Invalid program data separator
108	Parameter not allowed
109	Missing parameter
110	Command header error
111	Header separator error
112	Program mnemonic too long
113	Undefined header
118	Query not allowed
120	Numeric data error
121	Invalid character in number
123	Exponent too large
124	Too many digits
128	Numeric data not allowed
130	Suffix error

131	Invalid suffix
134	Suffix too long
138	Suffix not allowed
140	Character data error
141	Invalid character data
144	Character data too long
148	Character data not allowed
150	String data error
151	Invalid string data
152	String data too long
158	String data not allowed
160	Block data error
161	Invalid block data
168	Block data not allowed
170	Expression error
171	Invalid expression
178	Expression data not allowed
180	Alias error
181	Invalid outside alias definition
183	Invalid inside alias definition
184	Command in alias requires more/fewer parameters

Table 3-5 lists the execution errors that are detected during execution of a command. In these error messages, you should read "macro" as "alias."

Table 3-5: Execution Error Messages - EXE Bit 4

Code	Message
200	Execution error
201	Invalid while in local
202	Settings lost due to rtl
210	Trigger error
211	Trigger ignored
212	Arm ignored
220	Parameter error
221	Settings conflict
222	Data out of range
223	Too much data
224	Illegal parameter value
230	Data corrupt or stale
240	Hardware error
241	Hardware missing
242	Hardware configuration error
243	Hardware I/O device error
250	Mass storage error
251	Missing mass storage
252	Missing media
253	Corrupt media

254	Media full
255	Directory full
256	File name not found
257	File name error
258	Media protected
260	Expression error
261	Math error in expression
2200	Measurement error, Measurement system error
2201	Measurement error, Zero period
2202	Measurement error, No period found
2203	Measurement error, No period, second waveform
2204	Measurement error, Low signal amplitude
2205	Measurement error, Low amplitude, second waveform
2206	Measurement error, Invalid gate
2207	Measurement error, Measurement overflow
2208	Measurement error, Waveform does not cross Mid Ref
2209	Measurement error, No second Mid Ref crossing
2210	Measurement error, No Mid Ref crossing, second waveform
2211	Measurement error, No backwards Mid Ref crossing
2212	Measurement error, No negative crossing
2213	Measurement error, No positive crossing
2214	Measurement error, No crossing
2215	Measurement error, No crossing, second waveform
2216	Measurement error, No crossing, target waveform
2217	Measurement error, Constant waveform
2218	Measurement error, Unused
2219	Measurement error, No valid edge - No arm sample
2220	Measurement error, No valid edge - No arm cross
2221	Measurement error, No valid edge - No trigger cross
2222	Measurement error, No valid edge - No second cross
2223	Measurement error, Waveform mismatch
2224	Measurement error, WAIT calculating
2225	Measurement error, No waveform to measure
2226	Null Waveform
2227	Positive and Negative Clipping
2228	Measurement error, Positive Clipping
2229	Measurement error, Negative Clipping
2230	Measurement error, High Ref < Low Ref
2235	Math error, Invalid math description
2236	Math error, Reference waveform is invalid
2237	Math error, Out of acquisition memory
2238	Too many channels need to be acquired
2239	EAL turned off (EAL = extended-acquisition-length mode)
2240	Invalid password
2241	Waveform request is invalid
2242	Data start and stop > record length
2243	Waveform requested is not a data source
2244	Waveform requested is not turned on

2245	Saveref error, Selected channel is turned off
2246	Saveref error, Selected channel data invalid
2247	Saveref error, Out of reference memory
2248	Saveref error, Source reference data invalid
2249	Reference deletion error, Waveform in use for math
2258	Acq must be stopped
2259	File too big
2260	Calibration error
2270	Alias error
2271	Alias syntax error
2272	Alias execution error
2273	Illegal alias label
2274	Alias parameter error
2275	Alias definition too long
2276	Alias expansion error
2277	Alias redefinition not allowed
2278	Alias header not found
2279	Alias label too long
2280	Alias table full
2281	Wrong configuration
2285	TekSecure (R) Pass
2286	TekSecure (R) Fail
2290	Limit error, Reference in use
2291	Limit error, Reference data invalid
2292	Limit error, Out of reference memory
2293	Limit error, Selected channel is turned off
2301	Cursor error, Off-screen
2302	Cursor error, cursors in different frames
2311	Group requested has not been selected or has been deleted

Table 3-6 below lists the device errors that can occur during oscilloscope operation. These errors may indicate that the oscilloscope needs repair.

Table 3-6: Device Error Messages - DDE Bit 3

Code	Message
300	Device-specific error
310	System error
311	Memory error
312	PUD memory lost
313	Calibration memory lost
314	Save/recall memory lost
315	Configuration memory lost
350	Queue overflow (does not set DDE bit)

Table 3-7 below lists the system event messages. These messages are generated whenever certain system conditions occur.

Table 3-7: System Event Messages

Code	Message
400	Query event
401	Power on (PON bit 7 set)
402	Operation complete (OPC bit 0 set)
403	User request (URQ bit 6 set)
404	Power fail (DDE bit 3 set)
405	Request control
410	Query INTERRUPTED (QYE bit 2 set)
420	Query UNTERMINATED (QYE bit 2 set)
430	Query DEADLOCKED (QYE bit 2 set)
440	Query UNTERMINATED after indefinite response (QYE bit 2 set)
450	Right menu button #1 pushed (URQ bit 6 set)
451	Right menu button #2 pushed (URQ bit 6 set)
452	Right menu button #3 pushed (URQ bit 6 set)
453	Right menu button #4 pushed (URQ bit 6 set)
454	Right menu button #5 pushed (URQ bit 6 set)
460	Bottom menu button #1 pushed (URQ bit 6 set)
461	Bottom menu button #2 pushed (URQ bit 6 set)
462	Bottom menu button #3 pushed (URQ bit 6 set)
463	Bottom menu button #4 pushed (URQ bit 6 set)
464	Bottom menu button #5 pushed (URQ bit 6 set)
465	Bottom menu button #6 pushed (URQ bit 6 set)
466	Bottom menu button #7 pushed (URQ bit 6 set)

Table 3-8 below lists warning messages that do not interrupt the flow of command execution. These notify you that you may get unexpected results.

Table 3-8: Execution Warning Messages - EXE Bit 4

Code	Message
500	Execution warning
510	String data too long, truncated
525	Parameter underrange
526	Parameter overrange
527	Parameter rounded
528	Parameter out of range
530	Data stop > stop. Values swapped internally
531	Data stop > record length, Curve truncated
532	Curve data too long, Curve truncated
540	Measurement warning
541	Measurement warning, Low signal amplitude

542	Measurement warning, Unstable histogram
543	Measurement warning, Low resolution
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
550	DPO mode is active - deactivate to see change
551	DPO mode is active - deactivate to use math
552	EAL on - turn off to see change (EAL = extended-acquisition-mode)
553	EAL on - turn off to use math (EAL = extended-acquisition-mode)
570	Saveref warning, decimated 500k waveform to 250k Ref (TDS 700D)
571	Option 05 not present
572	Option 2F not present

Table 3-9 below shows internal errors that indicate an internal fault in the oscilloscope.

Table 3-9: Internal Warning Messages

Code	Message
600	Internal warning
620	Internal warning, Bad thermistor
630	Internal warning, 50 W overload

Event Handling Sequence

Figure 3-6, shown below, shows how to use the status and event handling system. In the explanation that follows, numbers in parentheses refer to numbers in Figure 3-6.

Figure 3-6: Status and Event Handling Process
==
{bmc PRJ.WMF}

When an event occurs, a signal is sent to the DESER (1). If that type of event is enabled in the DESER (that is, if the bit for that event type is set to 1), the appropriate bit in the SESR is set to one, and the event is recorded in the Event Queue (2). If the corresponding bit in the ESER is also enabled (3), then the ESB bit in the SBR is set to one (4).

When output is sent to the Output Queue, the MAV bit in the SBR is set to one (5).

When a bit in the SBR is set to one and the corresponding bit in the SRER is enabled (6), the MSS bit in the SBR is set to one and a service request is generated (7).

Synchronization Methods

Although most GPIB commands are completed almost immediately after being received by the oscilloscope, some commands start a process that requires more time. For example, once a HARDCOPY START command is executed it may be a few seconds before the hardcopy operation is complete. Rather than remain idle while the operation is in process, the oscilloscope will continue processing other commands. This means that some operations will not be completed in the order that they were sent.

Sometimes the result of an operation depends on the result of an earlier operation. A first operation must complete before the next one gets processed. The oscilloscope status and event reporting system provides ways to do this.

For example, a typical application might involve acquiring a single-sequence waveform and then taking a measurement on the acquired waveform. You could use the following command sequence to do this:

```
/** Set up single-sequence acquisition **/
SELECT:CH1 ON
HORIZONTAL:RECORDLENGTH 500
ACQUIRE:MODE SAMPLE
ACQUIRE:STOPAFTER SEQUENCE

/** Acquire waveform data **/
ACQUIRE:STATE ON

/** Set up the measurement parameters **/
MEASUREMENT:IMMED:TYPE AMPLITUDE
MEASUREMENT:IMMED:SOURCE CH1

/** Take amplitude measurement on acquired data **/
MEASUREMENT:IMMED:VALUE?
```

The acquisition of the waveform requires extended processing time. It may not finish before the oscilloscope takes an amplitude measurement (See Figure 3-7). This can result in an incorrect amplitude value.

Figure 3-7: Command Processing Without Using Synchronization

```
============================================================

  ___________________
  |ACQUIRE:STATE ON |
  +-----------------+
                      _______________________________
                      |Acquiring Waveform Data       |
                      +------------------------------+

                       ___________________________
                       |MEASUREMENT:IMMED:VALUE? |
                       +-------------------------+
|<----------------------------------------------------->|
                     Processing Time
```

```
============================================================
```

To ensure the oscilloscope completes waveform acquisition before taking the measurement on the acquired data, you can synchronize the program. Figure 3-8 shows the desired processing sequence.

Figure 3-8: Processing Sequence With Synchronization

```
============================================================
 __________________
|ACQUIRE:STATE ON |
+-----------------+_____________
                   |Acquiring  |
                   |Waveform   |
                   |Data       |
                   +-----------+___________________________
                               |MEASUREMENT:IMMED:VALUE? |
                               +-------------------------+
|<------------------------------------------------------->|
                      Processing Time

============================================================
```

You can use four commands to synchronize the operation of the oscilloscope with your application program: *WAI, BUSY?, *OPC, and *OPC?
.

Using the *WAI Command

You can force commands to execute sequentially by using the *WAI command. This command forces completion of the previous commands before processing new ones.

The same command sequence using the *WAI command for synchronization looks like this:

```
/* Set up single-sequence acquisition */
SELECT:CH1 ON
HORIZONTAL:RECORDLENGTH 500
ACQUIRE:MODE SAMPLE
ACQUIRE:STOPAFTER SEQUENCE

/* Acquire waveform data */
ACQUIRE:STATE ON

/* Set up the measurement parameters */
MEASUREMENT:IMMED:TYPE AMPLITUDE
MEASUREMENT:IMMED:SOURCE CH1

/* Wait until the acquisition is complete before taking the measurement
*/
*WAI

/* Take amplitude measurement on acquired data */
MEASUREMENT:IMMED:VALUE?
```

Though *WAI is one of the easiest ways to achieve synchronization, it is also the most costly. The processing time of the oscilloscope is slowed since it is processing a single command at a time. This time could be spent doing other tasks.

The controller can continue to write commands to the input buffer of the oscilloscope, but the commands will not be processed by the oscilloscope until all operations in process are complete. If the input buffer becomes full, the controller will be unable to write more commands to the buffer. This can cause a time-out.

Using the BUSY Query

The BUSY? query allows you to find out whether the oscilloscope is busy processing a command that has an extended processing time such as single-sequence acquisition.

The same command sequence using the BUSY? query for synchronization looks like this:

```
/* Set up single-sequence acquisition */
SELECT:CH1 ON
HORIZONTAL:RECORDLENGTH 500
ACQUIRE:MODE SAMPLE
ACQUIRE:STOPAFTER SEQUENCE

/* Acquire waveform data */
ACQUIRE:STATE ON

/* Set up the measurement parameters */
MEASUREMENT:IMMED:TYPE AMPLITUDE
MEASUREMENT:IMMED:SOURCE CH1

/* Wait until the acquisition is complete before taking the measurement
*/
While BUSY? keep looping

/* Take amplitude measurement on acquired data */
MEASUREMENT:IMMED:VALUE?
```

This sequence lets you create your own wait loop rather than using the *WAI command. The BUSY? query helps you avoid time-outs caused by writing too many commands to the input buffer. The controller is still tied up, though, and the repeated BUSY? query will result in more bus traffic.

Using the *OPC Command

If the corresponding status registers are enabled, the *OPC command sets the OPC bit in the Standard Event Status Register (SESR) when an operation is complete. You achieve synchronization by using this command with either a serial poll or service request handler.

Serial Poll Method: Enable the OPC bit in the Device Event Status Enable Register (DESER) and the Event Status Enable Register (ESER) using the DESE and *ESE commands.

When the operation is complete, the OPC bit in the Standard Event Status Register (SESR) will be enabled and the Event Status Bit (ESB) in the Status Byte Register will be enabled.

The same command sequence using the *OPC command for synchronization with serial polling looks like this:

```
/* Set up single-sequence acquisition */
SELECT:CH1 ON
HORIZONTAL:RECORDLENGTH 500
ACQUIRE:MODE SAMPLE
ACQUIRE:STOPAFTER SEQUENCE

/* Enable the status registers */
DESE 1
*ESE 1
*SRE 0

/* Acquire waveform data */
ACQUIRE:STATE ON

/* Set up the measurement parameters */
MEASUREMENT:IMMED:TYPE AMPLITUDE
MEASUREMENT:IMMED:SOURCE CH1

/* Wait until the acquisition is complete before taking the measurement.
*/
*OPC
While serial poll = 0, keep looping

/* Take amplitude measurement on acquired data */
MEASUREMENT:IMMED:VALUE?
```

This technique requires less bus traffic than did looping on BUSY?.

Service Request Method: Enable the OPC bit in the Device Event Status Enable Register (DESER) and the Event Status Enable Register (ESER) using the DESE and *ESE commands.

You can also enable service requests by setting the ESB bit in the Service Request Enable Register (SRER) using the *SRE command. When the operation is complete, a Service Request will be generated.

The same command sequence using the *OPC command for synchronization looks like this:

```
/* Set up single-sequence acquisition */
SELECT:CH1 ON
HORIZONTAL:RECORDLENGTH 500
ACQUIRE:MODE SAMPLE
ACQUIRE:STOPAFTER SEQUENCE

/* Enable the status registers */
DESE 1
*ESE 1
*SRE 32

/* Acquire waveform data */
ACQUIRE:STATE ON

/* Set up the measurement parameters */
MEASUREMENT:IMMED:TYPE AMPLITUDE
```

MEASUREMENT:IMMED:SOURCE CH1

/* Wait until the acquisition is complete before taking the measurement
*/
*OPC
Program can now do different tasks such as talk to other devices. The
SRQ, when it comes, interrupts those tasks and returns control to this
task.

/* Take amplitude measurement on acquired data */
MEASUREMENT:IMMED:VALUE?
This technique is more efficient but requires more sophisticated
programming.

Using the *OPC? Query

The *OPC? query places a 1 in the Output Queue once an operation is complete. A timeout could occur if you try to read the output queue before there is any data in it.

The same command sequence using the *OPC? query for synchronization looks like this:

/* Set up single-sequence acquisition */
SELECT:CH1 ON
HORIZONTAL:RECORDLENGTH 500
ACQUIRE:MODE SAMPLE
ACQUIRE:STOPAFTER SEQUENCE

/* Acquire waveform data */
ACQUIRE:STATE ON

/* Set up the measurement parameters */
MEASUREMENT:IMMED:TYPE AMPLITUDE
MEASUREMENT:IMMED:SOURCE CH1

/* Wait until the acquisition is complete before taking the measurement
*/
*OPC?
Wait for read from Output Queue.

/* Take amplitude measurement on acquired data */
MEASUREMENT:IMMED:VALUE?

This is the simplest approach. It requires no status handling or loops. However, you must set the controller time-out for longer than the acquisition operation.

Appendices

This section contains the following appendices:

Appendix A: Reserved Words

The following is a list of reserved words of the oscilloscope. Do not use these words for aliases. Capital letters identify the required minimum spelling. Hint: Use the full spelling for the most robust code as the minimum spelling rules may change over time and from model to model.

*CAL
*CLS
*DDT
*ESE
*ESR
*IDN
*LRN
*OPC
*PSC
*PUD
*RCL
*RST
*SAV
*SRE
*STB
*TRG
*TST
*WAI
ABOrt
ABSolute
AC
ACCept
ACQDAta
ACQDUration
ACQLength
ACQuire
ACQUISition
ACTivate
ACTUal
ALIas
ALL
ALLEv
ALLOcate
AMPlitude
AND
APPMenu
AREa
ARMed
ASC
ASCIi
AUTO
AUTOSet
AUXiliary
AVErage
BACKGround
BACKWards
BANdwidth
BASe
BAUd
BELl

BIN
BIT_Nr
BLAckman
BMP
BMPCOLOR
BN_Fmt
BOLd
BOTh
BOTTOM1
BOTTOM2
BOTTOM3
BOTTOM4
BOTTOM5
BOTTOM6
BOTTOM7
BOX
BOXPcnt
BURst
BUSY
BY
BYCONTents
BYT_Nr
BYT_Or
CALibrate
CATalog
CARea
CENtronics
CH1
CH2
CH3
CH4
CHKsm0
CLAss
CLEar
CLEARMenu
CLEARSNapshot
CLEARSpool
CLOCk
CMEan
COLLision
COLOr
COMPARE
CONTRast
CONTROl
COPy
COUNt
COUPling
CPU
CRMs
CROSSHair
CRVchk
CURRent
CURSor
CURSOR1
CURSOR2
CURVe

CUSTom
CWD
DATa
DATE
DC
DEFAult
DEFINE
DELay
DELAYEd
DELEte
DELTa
DELTATime
DELWarn
DESE
DESKJet
DESTination
DIAg
DIR
DIREction
DISAbled
DISplay
DOTs
DPO
DPU411
DPU412
DUAl
ECL
EDGE
EDGE1
EDGE2
EITher
ENAbled
ENCdg
ENV
ENVElope
EPSColor
EPSCOLImg
EPSImage
EPSMono
EPSOn
EVEN
EVENT
EVENTS
EVENTSTime
EVMsg
EVQty
EXECute
EXTAtten
EXTDACQ
EXTDBatten
EXTINCTPCT
EXTINCTDB
FACtory
FALL
FALSe
FASTERthan

FASTframe
FFT
FIELD
FIELD1
FIELD2
FIELDEither
FIELDRATE
FIELDS
FIELDType
FIFtyFILEName
FILESystem
FILTer
FIRst
FITtoscreen
FLAg
FLEXformat
FORCe
FORMat
FORWards
FPAnel
FRAme
FREE
FREESpace
FREQuency
FULl
FUNCtion
GATing
GLItch
GND
GPIb
GRAticule
GRAYscale
GRId
HALt
HAMming
HARDCopy
HARDFlagging
HBArs
HDELTA
HDR
HDTv
HEADer
HERtz
HFRej
HIGH
HIGHLimit
HIRes
HIStogram
HISTOMASK
HITs
HOLdoff
HOLDTime
HORizontal
HPGl
HPOS1
HPOS2

HUNdred
ID
IMMed
IMPedance
INDependent
INFInite
INFPersist
INIT
INPut
INStavu
INTENSIFied
INTENSITy
INTERLAce
INTERLeaf
INVert
IRE
LABel
LANdscape
LASERJet
LAYout
LENgth
LESSLimit
LESSThan
LEVel
LFRej
LIMit
LINE
LINES
LINEAr
LIVe
LOCk
LOG
LOGIc
LONG
LOW
LOWLimit
LSB
MAIn
MAP
MATH
MATH1
MATH2
MATH3
MAXimum
MEAN
MEANDBM
MEANSTDdev
MEAS1
MEAS2
MEAS3
MEAS4
MEASUrement
MEDian
MEG
MESSage
METHod

MID
MID2
MINImum
MINMax
MINUSOne
MKDir
MODe
MONo
MORELimit
MOREThan
MSB
NAMe
NANd
NDUty
NEGAtive
NEGSynchwidth
NEWpass
NOISErej
NONe
NOR
NORMal
NOVershoot
NR_Pt
NTSc
NUMACq
NUMAVg
NUMEnv
NUMERic
NWIdth
ODD
OFF
OFFSET
OFFSETAdj
ON
ONCe
OR
OPTion
OUTside
OVERAll
OVERWrite
PAIred
PAL
PALEtte
PARity
PARTial
PASSWord
PATtern
PCX
PCXCOLOR
PDUty
PEAKDetect
PEAKHits
PERCent
PERIod
PERSistence
PHAse

PK2pk
PLUSOne
POInts
POLarity
PORT
PORTRait
POSition
POSITION1
POSITION2
POSITIVe
POVershoot
PREView
PRInt
PRObe
PROCessing
PT_Fmt
PT_Off
PULse
PWIdth
RATE1
RATE2
RATE3
RATE4
READFile
READY
RECAll
RECORDLength
RECORDSNap
RECORDSTart
RECTangular
REF
REF1
REF2
REF3
REF4
REFLevel
REGular
REJect
REM
REName
REPEt
RESET
RESETAll
RESUlt
RI
RIBinary
RIGHT1
RIGHT2
RIGHT3
RIGHT4
RIGHT5
RISe
RLE
RMDir
RMS
RP

RPBinary
RS232
RUN
RUNSAfter
RUNSTop
RUNT
SAMple
SAVe
SCAle
SCAN
SCROLLBAR
SCROLLTEXT
SECAm
SECdiv
SECOnds
SELect
SEQuence
SET
SETHold
SETLevel
SETTime
SETUp
SHORt
SHOW
SINX
SLEWRate
SLOpe
SLOWERthan
SNAp
SNAPShot
SOFTFlagging
SOUrce
SOURCE1
SOURCE2
SPECTral
SRIbinary
SRPbinary
STANdard
STARt
STATE
STATIstics
STDdev
STOP
STOPAfter
STOPBits
STORe
STYle
SYNc
SYStem
TARget
TBposition
TEKSecure
TEMPErature
TEMPLate
TEXt
THInkjet

THReshold
TIFf
TIMe
TITLe
TO
TOLerance
TRACk
TRIGAfter
TRIGBar
TRIGger
TRIGT
TRUe
TTL
TWEnty
TWOfifty
TYPe
UNITS
UNITString
UNLock
VALue
VARpersist
VBArs
VDELTA
VECtors
VERBose
VERTical
VIDeo
VOLts
V1STArttime
V1STOptime
V2STArttime
V2STOptime
WAVEform
WAVEFORMS
WAVFrm
WFId
WFMPre
WHEn
WIDth
WIThin
WRITEFile
X
XINcr
XMUlt
XOFf
XUNit
XY
XZEro
Y
YMUlt
YOFf
YT
YUNit
YZEro
ZERO
ZMUlt

ZOFf
ZONe
ZOOm
ZUNit
ZZEro

Appendix B: Interface Specifications

This appendix describes details of the GPIB remote interface of the oscilloscope. Normally, you will not need this information to use the oscilloscope, but the information is useful when connecting to controllers of unusual configuration.

GPIB Function Subsets

The oscilloscope supports many GPIB function subsets, as listed below. Some of the listings describe subsets that the oscilloscope does not support.

- SH1 (Source Handshake)
  The oscilloscope can transmit multiline messages across the GPIB.

- AH1 (Acceptor Handshake)
  The oscilloscope can receive multiline messages across the GPIB.

- T5 (Talker)
  The oscilloscope becomes a talker when its talk address is sent with the ATN (Attention) line asserted. It can send both response data and status information when responding to a serial poll. It ceases to be a talker when the talk address if another device is sent with ATN asserted. The oscilloscope has talk-only capability for hardcopy operation.

- L4 (Listener)
  The oscilloscope becomes a listener when its listen address is sent with the ATN (Attention) line asserted. The oscilloscope does not have listen-only capability.

- SR1 (Service Request)
  The oscilloscope asserts an SRQ (Service Request) line to notify the controller when it requires service.

- RL1 (Remote/Local)
  The oscilloscope responds to both the GTL (Go To Local) and LLO (Local Lock Out) interface messages.

- PP0 (Parallel Poll)
  The oscilloscope has no parallel poll capability. It does not respond to the following interface messages: PPC, PPD, PPE, and PPU. The oscilloscope does not send out a status message when the ATN (Attention) and EOI (End or Identify) lines are asserted simultaneously.

- DC1 (Device Clear)
  The oscilloscope responds to the DCL (Device Clear) and, when made a listener, the SDC (Selected Device Clear) interface messages.

- DT1 (Device Trigger)
  When acting as a listener, the oscilloscope responds to the GET (Group Execute Trigger) interface message.

- C0 (Controller)
  The oscilloscope cannot control other devices.

- E2 (Electrical)
  The oscilloscope uses tristate buffers to provide optimal high-speed data transfer.

Interface Messages

Table B-1 shows the standard interface messages that are supported by the oscilloscope.

Table B-1: TDS Family Oscilloscope Standard Interface Message

Message	GPIB
DCL	Yes
GET	Yes
GTL	Yes
LLO	Yes
PPC	NO
PPD	NO
PPE	No
PPU	No
SDC	Yes
SPD	Yes
SPE	Yes
TCT	No
UNL	Yes
UNT	Yes
Listen Address	Yes
Talk Address	Yes

Appendix C: Factory Initialization Settings

The factory initialization settings provide you a known state for the oscilloscope.

Factory initialization sets values as shown in Table C-1.

Table C-1: Factory Initialization Settings

Control	Changed by Factory Init to
Acquire mode	Sample
Acquire repetitive signal	TDS 400A, 510A, 500D, & 700D: ON (Enable ET)
Acquire stop after	RUN/STOP button only
Acquire # of averages	16
Acquire # of envelopes	10
Channel selection	Channel 1 on, all others off
Cursor H Bar 1 position	10% of graticule height (-3.2 divs from the center)
Cursor H Bar 2 position	90% of the graticule height (+3.2 divs from the center)
Cursor V Bar 1 position	10% of the record length
Cursor V Bar 2 position	90% of the record length
Cursor amplitude units	Base
Cursor function	Off
Cursor mode	Independent
Cursor time units	Seconds
Date and time	No change
Delay events, triggerable after main	TDS 400A: 1 TDS 510A, 500D, 600B, 700D: 2
Delay time, delayed runs after main	TDS 400A: 10 ns TDS 510A, 500D, 600B, 700D: 16 ns
Delay time, delayed triggerable after main	TDS 400A: 60 ns TDS 510A, 500D, 600B, 700D: 16 ns
Delay trigger average #	16
Delay trigger envelope #	10
Delayed, delay by	Delay by Time
Delayed edge trigger coupling	TDS 400A: DC, Main Trigger
Delayed edge trigger level	0 V
Delayed edge trigger slope	Rising
Delayed edge trigger source	Channel 1
Delayed, time base mode	Delayed Runs After Main
Deskew, Channel/Probe	0 seconds
Display clock	No Change
Display color - collision contrast	TDS 644B, 684B, 700D: Off
Display color - map math colors	TDS 644B, 684B, 700D: Color `Math'
Display color - map reference colors	TDS 644B, 684B, & 700D: Color `Ref'

Display color - palette	TDS 644B, 684B, & 700D: Normal
Display color - palette colors	TDS 644B, 684B, & 700D: The colors of each palette are reset to factory hue, saturation, and lightness (HLS) values
Display color - persistence palette	TDS 644B, 684B, & 700D: Temperature
Display format	YT
Display graticule type	Full
Display DPO persistence	TDS 500D & 700D: Varpersist
Display DPO style	TDS 500D & 700D: Vectors
Display DPO varpersist	TDS 500D & 700D: 500 e-3
Display intensity - contrast	TDS 400A, 500D, 620B, & 680B: 175%
	TDS 510A: 150%
Display intensity - overall	TDS 400A, 510A, 500D, 620B, & 680B: 85%
Display intensity - text	TDS 644B, 684B, & 700D: 100%
	TDS 400A, 510A, 500D, 620B, & 680B: 60%
Display intensity - waveform	TDS 644B, 684B, & 700D: 100%
	TDS 400A, 510A, 500D, 620B, & 680B: 75%
Display interpolation filter	Sin(x)/x
Display mode	Normal
Display style	Vectors
Display trigger bar style	Short
Display trigger "T"	On
Display variable persistence	500 ms
Edge trigger coupling	DC
Edge trigger level	0.0 V
Edge trigger slope	Rising
Edge trigger source	Channel 1
GPIB parameters	No change
Hardcopy Format	Unchanged
Hardcopy Layout	Unchanged
Hardcopy Palette	Unchanged
Hardcopy Port	Unchanged
Horizontal - delay time/division	50 us
Horizontal - delay trigger position	50%
Horizontal - delay trigger record length	500 points (10 divs)
Horizontal - FastFrame	TDS 500D & 700D: Off
Horizontal - FastFrame, frame count	TDS 500D & 700D: 2
Horizontal - FastFrame, frame length	500
Horizontal fit to screen	Off
Horizontal main time/division	500 us
Horizontal main trigger position	50%
Horizontal main trigger record length	500 points (10 divs)
Horizontal time base	Main only

Limit template	
+V Limit	40 mdiv
+H Limit	40 mdiv
Limit template destination	Ref1
Limit template source	Ch1
Limit test sources	Ch1 compared to Ref1; all others compared to none.
Limit Testing	Off
Limit Testing - hardcopy if condition met	Off
Limit Testing - ring bell if condition met	Off
Logic pattern trigger Ch4 (Ax2) input	TDS 510A, 500D, 600B, & 700D: X (do not care)
Logic state trigger Ch4 (Ax2) input	TDS 510A, 500D, 600B, & 700D: Rising edge
Logic trigger class	TDS 500D, 600B, & 700D: Pattern
Logic trigger input (pattern and state)	TDS 510A, 500D, 600B, & 700D: Channel 1 = H (high), Channels 2 & 3 (Ax1) = X (do not care)
Logic trigger logic (pattern and state)	TDS 510A, 500D, 600B, & 700D: AND
Logic trigger pattern time qualification	TDS 510A, 500D, 600B, & 700D:
Lower limit	5 ns
Upper limit	5 ns
Logic trigger sources and levels (Setup/Hold)	TDS 500D, 600B, & 700D: Data Source = Channel 1 = 1.4 V Clock Source = Channel 2 = 1.4 V (Source levels are clipped to 1.2 V at the default volts/division setting established by Factory Init) Clock Edge = Rising
Logic trigger threshold (all channels) (pattern and state)	TDS 510A, 500D, 600B, & 700D: 1.4 V (clipped to 1.2 V at the default volts/division setting when no 10X probe attached)
Logic trigger triggers when ... (pattern and state)	TDS 510A, 500D, 600B, & 700D: Goes TRUE
Main trigger mode	Auto
Main trigger type	Edge
Math1 definition	Ch 1 + Ch 2
Math1 extended processing	TDS 510A, 500D, 600B, & 700D: No extended processing
Math2 definition	Ch 1 - Ch 2 (FFT of Ch 1 for instruments with Option

	2F Advanced DSP Math)
Math2 extended processing	TDS 510A, 500D, 600B, & 700D: No extended processing
Math3 definition	Inv of Ch 1
Math3 extended processing	TDS 510A, 500D, 600B, & 700D: No extended processing
Measure Delay edges	Both rising and forward searching
Measure Delay to	Channel 1 (Ch1)
Measure Gating	Off
Measure High Ref	90% and 0 V (units)
Measure High-Low Setup	Histogram
Measure Low Ref	10% and 0 V (units)
Measure Mid Ref	50% and 0 V (units)
Measure Mid2 Ref	50% and 0 V (units)
Pulse glitch filter state	TDS 510A, 500D, 600B, & 700D: On (Accept glitch)
Pulse glitch trigger polarity	TDS 510A, 500D, 600B, & 700D: Positive
Pulse glitch width	TDS 510A, 500D, 600B, & 700D: 2.0 ns
Pulse runt high threshold	TDS 510A, 500D, 600B, & 700D: 1.2 V
Pulse runt low threshold	TDS 510A, 500D, 600B, & 700D: 0.8 V
Pulse runt trigger polarity	TDS 510A, 500D, 600B, & 700D: Positive
Pulse runt triggers when ...	TDS 510A, 500D, 600B, & 700D: Occurs
Pulse slew rate delta time	2.0 ns
Pulse slew rate polarity	Positive
Pulse slew rate thresholds	Trig if faster than
Upper	1.80 V
Lower	800 mV
Pulse slew rate triggers when	Trig if faster than
Pulse timeout polarity	TDS 500D & 600B: Either
Pulse timeout time	TDS 500D & 600B: 2.0 ns
Pulse trigger class	TDS 510A, 500D, 600B, & 700D: Glitch
Pulse trigger level	TDS 510A, 500D, 600B, & 700D: 0.0V
Pulse trigger source (Glitch, runt, and width)	TDS 500D, 600B, & 700D: Channel 1 (Ch1)
Pulse width lower limit	TDS 510A, 500D, 600B, & 700D: 2.0 ns
Pulse width trigger polarity	TDS 510A, 500D, 600B, & 700D: Positive
Pulse width trigger when	TDS 510A, 500D, 600B, & 700D: Within limits
Pulse width upper limit	TDS 510A, 500D, 600B, & 700D: 2.0 ns
Repetitive signal	TDS 500D & 700D: On
RS-232 parameters	No change
Saved setups	No change
Saved waveforms	No change
Stop after	R/S button
Vertical bandwidth (all channels)	Full
Vertical coupling (all channels)	DC
Vertical impedance	1 M-ohm

(termination) (all channels)	
Vertical offset (all channels)	0 V
Vertical position (all channels)	0 divs.
Vertical volts per division (all channels)	100 mV per division
Zoom dual	TDS 400A, 500D, 600B, & 700D: Off
Zoom dual offset	TDS 400A, 500D, 600B, & 700D: Minimum available
Zoom, dual window, selected graticle	TDS 400A, 500D, 600B, & 700D: Upper
Zoom graticule	Upper
Zoom horizontal (all channels)	TDS 510A: 1.0X TDS 400A, 500D, 600B, 700D: 2.0X
Zoom horizontal lock	All
Zoom horizontal position (all channels)	50% = 0.5 (the middle of the display)
Zoom mode	Off
Zoom vertical (all channels)	TDS 510A: 1.0X TDS 400A, 500D, 600B, 700D: 2.0X
Zoom vertical position (all channels)	0 divisions