

Gould Classic Series Programming Manual

**Copyright © 1996
Gould Instrument Systems**



Contents	
Introduction	
GPIB Communication	1
Commands	2
ACQuisition Subsystem	3
ANALysis Subsystem	4
CALibrate Subsystem	5
CHANnel Subsystem	6
Common & Root Commands	7
CURSor Subsystem	8
DATE Subsystem	9
DATum Subsystem	10
DISPlay Subsystem	11
LIMits Subsystem	12
MassMEMory Subsystem	13
MEASurement Subsystem	14
MEMory Subsystem	15
MENU Subsystem	16
PLOT& PRINT Subsystems	17
RS423 Subsystem	18
SEQuence Subsystem	19
SETup Subsystem	20
SYStem Subsystem	21
TRace Subsystem	22
TRANsfer Subsystem	23
TRIGger & TRIGger[AB] Subsystems	24
TRIGgerTOOLS and TRIGgerTOOLS[AB] Subsystem	25
TRIGgerTV Subsystem	26
Service Facilities	27



Introduction	1	:HISTogram	24
		:BINS	24
1 About RS423 & GPIB interfaces	2	:HORIZontalRange	25
1.1 Communication Via GPIB & RS423	2	:LOWERresult	25
1.2 GPIB (IEEE-488)	2	:MAXreading	25
1.3 RS423	3	:MEASurement	25
1.4 About GPIB and RS423 Commands	4	:MINreading	26
1.5 About Command Commands	5	:OVERflow	26
1.6 Status Reporting	5	:UNDERflow	26
1.7 Bulk Transfers	5	:UpdateRATE	26
		:UPPERresult	26
2 Commands	10	:YSCaling	27
2.1 RS423/GPIB Command Summary	10	:INTEgrateSCaling	27
		:MULTSCaling	27
3 ACQuisition Subsystem	15	:SOURCE1	27
:ACQuisition	15	:SOURCE2	28
:AVerAge	15	:STOP	28
:ENable	15	:TYPE	28
:FACTor	15	:STOP	28
:FASTreARM	16		
:GLitchDETECT	16	5 CALibrate Subsystem	29
:MEMoryLENGth	16	:CALibrate	29
:MultiSHOT	16	:ENable	29
:NoSEGments	17	:FORCE	29
:RUN	17		
:STATUS	17	6 CHANnel Subsystem	30
:TimeBASE	18	:CHANnel[1234]	30
		:COUPling	30
4 ANALysis Subsystem	19	:HOId	30
:ANALysis	20	:OFFSet 31	
:TRace[12345678]	20	:POSition	31
:DIFFerentiateSCaling	20	:PRObe	31
:DIVideSCaling	20	:RANGe	32
:EXECute	20	:TERM50	32
:FFT	21	:UNcal	33
:HorizSCaling	21	:VARIABLE	33
:LINearSCaling	21		
:OFFSet	21	6a CHANnelSCaling Subsystem	34
:POInts	22	:CHANnelSCaling	34
:VertMode	22	:ENable	34
:VertSCaling	22	:CHANnel[1234]	34
:WINDow	22	:ENable	34
:FILTerFACTor	22	:OFFSet	35
:Function	23	:SCaling	35
:GRAPH	23	:UNITs	35
:LEFTedge	23	:TIME	36
:MAXreading	23	:DatumtoCURsor	36
:MEASurement	23	:ENable	36
:MINreading	24	:SCaling	37
:RIGHTedge	24	:UNITs	37
:UpdateRATE	24		

7 Common and Root Commands

*CAL	38
*CLS	38
*ESE	39
*ESR	39
*IDN	39
*OPC	39
*OPT	40
*RST	40
*SRE	41
*STB	42
*TRG	42
*TST	42
*WAI	42
:ALL?	43
:AUTosetup	43
:BandwidthLimit	43
:BEEP	43
:CANCEL	43
:DATE	44
:HELP?	44
:KEYclick	44
:LER?	44
:LTER?	45
:MENU	45
:MODE	45
:PLOT	46
:SER?	46
:TEXT	46
:TIME	46

8 CURSor Subsystem

:CURSor	47
:ScreenXpos	47
:ScreenYpos	47
:TRAce	47
:Xpos	48
:Ypos	48

9 DATE Subsystem

:DATE	49
:FORMAT	49

10 DATum Subsystem

:DATum	50
:ScreenXpos	50
:ScreenYpos	50
:Xpos	50
:Ypos	50

11 DISPlay Subsystem

:DISPlay	52
:ALLColors	52
:BACKground	52
:COLor	52
:CURSors	52
:COLor	52
:GRATicule	52
:BORder	52
:COLor	53
:MAJorAxes	53
:MINorAxes	53
:STAt	53
:GRaticuleiNTERsection	54
:COLor	54
:HOLDAll	54
:MainTRace[12345678]	54
:COLor	54
:MAXmin	55
:PERSDotjoin	55
:PERSISTence	55
:COLor	55
:STAt	55
:PERSMode	56
:PERSSWEeps	56
:PERSTime	56
:SWEep	56
:TEXT	57
:COLor	57
:TRace{ 1M ...8Z }	57
:STAt	57
:TRACEDotjoin	57
:TRaceINTERsection	57
:COLor	57
TruTRACE	58
:XSHift	58
:XY	58
:ZOOM	59
:ZoomPOSition	5
:ZoomTRace[12345678]	59
:COLor	59
:YZoomFACtor	60
:YZoomPOSition	60

12 LIMits Subsystem

:LIMits	61
:BANDsize	61
:LOWer	61
:MODE	61
:PlotonFAIL	62
:SetBand	62
:UPper	62

13 MassMEMory Subsystem**63**

:MassMEMory	63
:COpy	63
:DESTination	63
:DEVICE	63
:RuNnAMe	64
:RuNnUMber	64
:SEQuence	64
:SETup	65
:TRACeNAME	65
:USER	66
:EXECute	66
:SOurce	63
:DEVICE	63
:RuNnAMe	64
:RuNnUMber	64
:SEQuence	64
:SETup	65
:TRACeNAME	65
:USER	65
:TYPE	63
:RECall	63
:EXECute	66
:RuNnUMber	64
:TRACe	65
:TRACeNAME	65
:TYPE	65
:SAVE	63
:EXECute	66
:RuNnUMber	64
:TYPE	65
:SAVeRECall	63
:DEVICE	63
:RuNnAMe	64
:SEQuence	64
:SEQueNcenUMber	64
:SETup	65
:USER	66
:UTILities	66
:CREate	66
:DEVICE	63
:DELeTe	66
:FORMat	67
:LIST	67
:NEwNAME	68
:NEwNUMber	68
:REName	67
:RuNnAMe	64
:RuNnUMber	64
:SEQuence	64
:SETup	65
:TRACeNAME	65
:USER	66

14 MEASurements Subsystem**69**

:MEASurements	69
:FULLRESolution	69
:STATE	70
:YT	70
:FUnction	70
:MEASurement	70
:OPERator	70
:VALue	70
:ANNOtation	70
:BOUnds	70
:CLearlist	71
:CRossing	71
:CRossLEVel	71
:DATNOisefl	71
:DATum	71
:DATumVALue	71
:DELeTe	71
:DISPlay	72
:FACtor	7
:FIXedVALue	72
:FUNDamental	72
:HARMonic	72
:HIGHtrans	72
:KNEE	72
:LEFTtrans	73
:LOWtrans	73
:MARKers	73
:NAME	73
:OFFset	73
:REFeRence	73
:ReSeT	73
:RESult	74
:RESultNUMeric	74
:RIGHTtrans	74
:SCAling	74
:SEGment	74
:SOUrce1YT	74
:SOUrce2YT	75
:STanDardreference	75
:TimeDATumREFeRence	75
:TOPbase	75
:TRANSDIRection	75
:TRANSNUMber	75
:UNITS	75
:WEIGHTing	75
:XY	76
:FUnction	77
:MEASurement	76
:OPERator	76
:ANNOtation	76
:CLearlist	76
:DatumLOCKedtorelease	76

:DELeTe	76	19 SEQuence Subsystem	87
:DispAREAfIl	77	:SEQuence	87
:DISPlay	77	:CLEAR[12345678]	87
:FACTOR	77	:CONTINUE	87
:FIXedVALue	77	:CYCles[12345678]	87
:NAME	77	:NAME	88
:OFFset	77	:RUN[12345678]	88
:REFeRence	77	:SOFTkey	88
:ReSeT	77		
:RESult	78	20 SETUp Subsystem	89
:RESultNUMeric	78	:SETUp	89
:SCAling	78	:RECALL	89
:SOURce1YT	78	:SAVE	89
:SOURce2YT	78		
:STANdardreference	78	21 SYStem Subsystem	90
:UNITS	78	:SYStem	90
:WEIGHTing	78	:ERRor	90
		:ERRorCLear	92
15 MEMory Subsystem	79	:HEADer	92
:MEMory	79	:LONGform	92
:CLear	79		
:RECall	79	22 TRace Subsystem	93
:SAVE	79	:TRace	93
		:DATEandtime[12345678]	93
16 MENU Subsystem	80	:SEGmentTime[12345678]	93
:MENU	80	:TRaceSCAling[12345678]	94
:TRANSpArency	80	:X	94
		:Y	94
17 PLOT Subsystem	81		
:PLOT	81	23 TRANSfer Subsystem	95
:DESt	81	:TRANSfer	96
:DRiVer	81	:ACQuisition	96
:EXTeRnaldest	81	:ALL	96
:GRaticule	82	:DATAonly	96
:LABELsdatecursor	82	:WindowSTArT	100
:LENGth	82	:WindowSTOp	101
:LIMitstype	82	:FILE	97
:PERStype	83	:EXECute	97
		:DEViCe	97
17a PRINT Subsystem	84	:RuNnAMe	99
:PRINT	84	:RuNnUMber	99
:DENSity	84	:SEQuence	99
		:SETUp	99
18 RS423 Subsystem	85	:TRACeNAME	100
:RS423	85	:TYPE	100
:BAUD	85	:USER	100
:BPC	85	:FORMat	97
:ECHOandprompt	85	:GRAPH	98
:HAnd	86	:ALL	96
:PARity	86	:DATAonly	96
:STOpBits	86	:WindowSTArT	100
		:WindowSTOp	101

:HISTOgram	98	:DIVidebyN	108
:ALL	96	:COUNT	108
:DATAonly	96	:CONTrol	108
:WindowSTArt	100	:PHASEslip	108
:WindowSTOp	101	:FREQuency	109
:MAIN	98	:FREQuency	109
:DATAonly	96	:PERiod	109
:WindowSTArt	100	:WHENFrequency	109
:WindowSTOp	101	:WHENPeriod	109
:RADIX	99	:MISSing	110
:SEQuence	99	:PERiod	110
:WORDsize	101	:MODE	110
:ZOOM	101	:PulseWidth	110
:DATAonly	96	:WIDth	110
:WindowSTArt	100	:WHEN	110
:WindowSTOp	101	:RUNT	111
24 TRIGger Subsystem	102	:POLarity	111
:TRIGger	102	:SKEW	111
:SElect	102	:PERiod	111
:SYStem	102	:WHEN	111
24a TRIGger[AB] Subsystem	103	:SLEWrate	111
:TRIGger[AB]	103	:PERiod	111
:AUTotrig	103	:WHEN	112
:COUpling	103	25a TRIGgerTOOLS[AB] Subsystem	113
:DElay	104	:TRIGgerTOOLS[AB]	113
:LEVEL{1,2,3,4,EXT}	104	:COMBination	113
:PREPost	104	:AnyChangeQUALification	113
:PREtrig	104	:HiLowQUALification	113
:SLOPE{1,2,3,4,EXT}	105	:Source{1,2,3,4}	114
:Source	105	:TRIGger	114
25 TRIGgerTOOLS Subsystem	106	:FREQuency	114
:TRIGgerTOOLS	106	:EDGE	114
:ADElayNGatesB	106	:PulseWidth	114
:COUNT	106	:POLarity	114
:ADElayTimeGatesB	107	:SLEWrate	115
:PERiod	107	:EDGE	115
:BAND	107	26 TRIGgerTV Subsystem	116
:WHEN	107	:TRIGgerTV	116
:BGatesaDelayN	107	:MODE	116
:COUNT	107	:Source	116
:COMBination	107	:syncPOLarity[1234]	116
:PERiod	107	27 Service Facilities	117
:DElaybyN	108		
:COUNT	108		



Introduction

WARNING Read section 1.1 Safety and Power requirements of the Classic series operator manual before installing or operating this instrument.

About the Classic Series of Digital Storage Oscilloscopes

Gould has used innovative state-of-the-art electronics and the latest manufacturing techniques to achieve both high performance and low cost in the Classic range of Digital Storage Oscilloscopes. All the essential, time tested features which users of premium performance DSOs have come to expect, are available and more: up to 400 MHz bandwidth; 2 GS/s sampling rate; variable input offset; 10 k, 50 k[†], 200 k[†] memory lengths, all segmentable; full color display[†] with color-matched trace annotation and ground markers; floppy and hard[†] disks; and a variety of I/O[†] and plotting interfaces. In addition, a wide range of specialized functions can be provided, including FFT[†], Trigger Tools[†], Advanced Measurements[†] (YT and XY), Sequences with decision making macro capability[†], built-in thermal plotter[†], and limits testing with template generation capability. Evolution and refinement of our well accepted user interface ensures that this comprehensive feature set remains well focused.

The Classic 6000 has a bandwidth of 200 MHz and a maximum sampling rate of 100 MS/s on all four channels.

The Classic 6500 has a bandwidth of 100 MHz and a maximum sampling rate of 100 MS/s on all four channels.

The Classic 9500 has a bandwidth of 400 MHz and a maximum sample rate of 2 GS/s.

Operating philosophy

Fast display update of the acquired waveforms on a high brightness, color screen creates a very responsive instrument with clear differentiation among traces. The wide choice of display modes — Refresh, Persistence, Roll, X-Y, with pre- or post-trigger viewing and live zoom — together with advanced trigger facilities and TruTrace[®], make for quick and precise acquisition and display of the signal characteristics.

Dedicated front panel controls to position and scale traces quickly coupled with user-defined soft keys for regularly used functions ensure that the Classic range is easy to use. Analyze signals further, make specific measurements or archive waveforms for later reference: Classic DSOs provide complete application solutions.

TruTrace[®]

The Gould Classic range not only provides today's solution oriented features in a very cost effective instrument, it also addresses the display of acquired signals and derived traces in a new and innovative way: *TruTrace*. For the first time, the effect of all the data points within a complex waveform — even 200 k in length — can be seen in a compressed raster-scan overview. *TruTrace* accomplishes this by displaying compressed traces as grey-scale images; traces very similar to those which an analog, real-time oscilloscope would produce from the same signals. This means that anomalies in complex, switching power supply start-up waveforms, for example can be recognized readily, without time-consuming windowing and zooming. Alternatively, because the acquired data is always accessible, any displayed trace can be examined in more detail should it be necessary. In this case, the independent, per-channel control of Y-zoom and Y-position provided in the Classic series allows personalised display formatting to suit the application.

TruTrace can be used to display any trace or combination of up to eight traces, — transients, zoom traces, derived traces which result from analysis operations or recalled traces.

TruTrace performs an intelligent compression of acquired data in which each data point contributes to the displayed waveform and intensity variations are produced, allowing the display of far more information about an acquired signal than available with conventional DSOs. Intensity variation is also used to highlight areas where traces overlap making it impossible to "lose" a trace entirely behind another.

About this Manual

This manual contains information on the GPIB and RS423 interfaces and their commands.

Section 1.1, Safety and Power requirements of the operator manual, must be read and note taken of its contents before using the oscilloscope.

The manual is split into various sections.

Section 1 contains general information about the interfaces and RS423 and GPIB commands.

Section 2 contains a summary of all the commands in alphabetical order so that the subsystem to which they belong can be found.

Sections 3 to 26 contain detailed explanations of all the commands grouped by subsystem.

For instrument operating instructions see the Classic series operator manual.

1.1 Communication via GPIB/RS423

This section of the operators manual introduces the concepts of GPIB and RS423 communication and explains in detail the command set.

There are four basic operations that can be performed with a controller and the DSO through the control port:

- Set up the instrument and make waveform measurements
- Retrieve setup information and measurement results
- Send captured waveform data to the controller
- Return previously transferred waveforms to the scope

Other more complicated tasks can be accomplished by combining these four building blocks.

Both the GPIB and the RS423 ports are very versatile allowing fast communication between a host computer and the DSO. Only one can be used as a control port at any one time.

The two ports are electrically different, but the high level communication format is essentially the same in either case.

Except where indicated this section applies to both GPIB and RS423 operation.

1.2 GPIB (IEEE-488)

This is a parallel bus interface allowing communication between a host computer and up to 30 peripherals. It is based on the controller/talker/listener structure whereby the controller – usually the host computer – commands various peripherals to ‘talk’ or ‘listen’ according to the desired data path. Each peripheral on the bus is assigned a unique address number and on the DSO this is set via the GPIB interface menu –section 3.16.

There are three main modes of operation of the GPIB:

Local: The operator has full control of the instrument via the front panel. The DSO may be interrogated but not manipulated in any way by the host computer.

Remote: All the DSO’s GPIB commands can be used in this mode. The only front panel control that operates in this mode is the abort button (47) which when pressed, returns the instrument to local mode.

Remote with Local Lock Out: This is an extension of the Remote mode. None of the instrument’s front panel controls operate except the power on/off switch. This mode can normally only be released by the controller.

The connections to the GPIB are listed below. For further details of the connector see Appendix 5.

Pin	Signal	Description	
1	DIO1	Data input output line 1	
2	DIO2	Data input output line 2	
3	DIO3	Data input output line 3	
4	DIO4	Data input output line 4	
13	DIO5	Data input output line 5	
14	DIO6	Data input output line 6	
15	DIO7	Data input output line 7	
16	DIO8	Data input output line 8	
5	EOI	End Or Identify	Interface Management Lines
17	REN	Remote Enable	
9	IFC	Interface Clear	
10	SRQ	Service Request	
11	ATN	Attention)	
6	DAV	Data Valid	Handshake Lines
7	NRFD	Not ready for data	
8	NDAC	Data not accepted	
18	(DAV)		Return lines Twisted With same name line
19	(NRFD)		
20	(NDAC)		
21	(IFC)		
22	(SRQ)		
23	(ATN)		
12	Shield		
24	Logic Ground		

Table 1.2 GPIB IEEE-488 pin connections

Protocols

The protocols of IEEE 488.2 define the overall structure of the communication process, when devices should talk or listen and what happens when protocols are not followed.

Elements

Input Buffer The input buffer of the DSO is the memory area where commands and queries are stored prior to being parsed and actioned. This allows the controller to send a string of commands to the DSO which can then be parsed and actioned by the instrument while the controller is performing other tasks such as talking to another instrument. The DSO’s input buffer will hold 512 bytes of data, although the DSO is capable of handling block data greater than this where appropriate.

Output Queue The output queue of the DSO is the memory area where all output data is stored until read by the controller. The DSO’s output queue will hold 512 bytes of data, although the DSO is capable of handling block data greater than this where appropriate.

Parser The parser is the name given to the software which interprets commands sent to the DSO and determines the appropriate actions. Parsing means the action taken by the parser to interpret a command.

Protocol operation

Each time the instrument is powered up or when it receives a device clear command, the input and output buffers are cleared, and the parser is reset to the root level of the command tree see figure 5.3.

The DSO and the controller communicate using program messages and response messages. Program messages are sent by the controller to the DSO and response messages are sent to the controller from the DSO in response to a query message. A query message is a program message that contains one or more queries.

The DSO can only talk in response to a valid query message, i.e. when it has something to say. The controller should always read a response to a previous query before sending another program message.

Protocol errors

Addressed to talk with nothing to say If the DSO is addressed to talk before it has received a query, it will indicate a query error and will not send any bytes over the bus. If the DSO has nothing to say because of an error during a query execution the instrument will not indicate a query error but simply wait to receive the next transmission from the controller.

Addressed to talk with no listeners on the bus If the DSO is addressed to talk while there are no listening devices on the bus the DSO will wait for a device to listen or for the controller to take control.

Command error A command error will be reported if the instrument detects a syntax error or an unrecognized command.

Execution Error An execution error will be reported if a parameter is found to be out of range, or if the current instrument settings are not suitable for the requested command or query.

Device Specific Error A device specific error will be reported if the instrument is unable to execute a device dependent command.

Query Error A query error will be reported if the proper protocol for reading a query is not followed. A query error will also be generated by an unterminated and interrupted message as detailed below.

Unterminated Condition A query error will be generated if the controller attempts to read a response message before terminating the program message. If this occurs, the parser will be reset and the response will be cleared from the DSO's output queue. The response will not be sent over the bus.

Interrupted condition A query error will be generated if the controller attempts to send a program message before reading the entire response message to a previous query. The unread part of the response is lost but the new program message will not be affected.

Buffer Deadlock If the DSO's input and output queues both become full a buffer deadlock may occur. This is only likely to happen if a very long program message containing queries that result in a lot of response data is sent to the DSO. The input buffer will become full so that the entire programme message cannot be sent but the controller cannot read any of the response data until it has finished sending the program message; a deadlock.

If this situation occurs, the DSO will clear the output queue, thus breaking the deadlock, and it will continue to clear the responses until the end of the current program message. The query error bit will be set by this condition.

1.3 RS423

The RS423 interface is an updated form of the RS232 standard and is a serial connection requiring a dedicated input and output on the host computer to each peripheral.

RS423 uses lower voltages and lower impedances than RS232 but is electrically compatible

Occasionally problems can be encountered with RS423 cabling. These difficulties are almost always resolved by correct connection of the data and handshake lines to ensure the correct direction of signal travel.

Some of the commands on the DSO are special IEEE488.2 commands and as such are meaningless on RS423. For compatibility however, assertive commands are always accepted by the DSO but those that are non RS423 will have no action. Non RS423 queries will return 0.

The RS423 connections on the interface for the DSO are listed below. For further details of the connectors see Appendix 5.

RS423 Interface

Pin	Signal	Description
2	TXD	Transmit Data (out)
3	RXD	Receive Data (in)
5	0 V	Signal Ground
8	RTS	Request to Send (out)
7	CTS	Clear to Send (in)

Table 1.3 RS423 Pin Connections

1.4 About GPIB and RS423 Commands

The commands of the DSO are simple and English-like. Where necessary, mnemonics have been used to replace large words or text.

Each instrument-control header or key word has both a long form and a short form. In this manual the long form is always used and the short form is indicated by the uppercase characters in the long form. When using commands, case is unimportant. Both upper and lower case are acceptable.

Programs written using the long form of commands are more easily read and understood. The shortform uses less memory storage and is faster as less data has to be transmitted.

As an example of commands, the following all set channel 1 to an input sensitivity of 50 mV/Div.

```
:CHANNEL1:RANGE 50E-3
```

```
:CHANNE1:RANG 50E-3
```

```
:CHAN1:RANG 50E-3
```

Commands are based on an hierarchical structure. This allows the same instrument-control header (keyword) to be used several times for different purposes.

All commands, unless otherwise noted, have a query form. A query is usually a command header followed by a question mark.

Some commands have only a query form, that is they do not modify the instrument status, merely examine a particular feature.

Multiple commands and queries can be sent on the same line

by separating each command with a semi-colon.

```
:CHAN1:RANG 50E-3;;DISP:HOLDALL ON
```

Some commands have various options associated with them e.g. the Channel command is followed by the channel number that it refers to. Where these various options are compulsory they are enclosed in square brackets: []. Each option is separated by a vertical line: | This means that one and only one of the items within the curly brackets must be included in the command. Some commands may have more than one set of brackets, in which case one item from each set must be used.

Where the command is a simple on off function, numeric data <NRf> may be used instead of OFF or ON. Zero stands for OFF and any other number stands for ON. In these cases the individual command descriptions detail the meaning of Zero and non zero values.

In this manual the following conventions and abbreviations are used:

- <cr> is used to represent the carriage return control code.
- <lf> is used to represent the line feed control code.
- [] denotes an item that must be included in the command.
- | separates options within []. where the options are single character, the | is omitted.
- EOI stands for: End Or Identify. (GPIB signal line).
- <numeric_value> means that a number is taken as the argument of the command.
- <data> means that alphanumeric data is used or expected as the argument of the command.
- * Means either the whole set e.g. all channels, or it can be replaced by a single member of the set e.g. CHAN1.
- <Boolean> is used throughout this manual as a shorthand for the form ON | OFF | <NRf>. Boolean parameters have a value of 0 or 1 and are unitless.
- On input, <NRf> is rounded to an integer. A non-zero result is interpreted as 1.
- ON and OFF are accepted as inputs to the DSO.
- ON corresponds to 1 and OFF corresponds to 0.
- Queries always return 1 or 0, never ON or OFF.

Interface Capabilities

The interface capabilities of the DSO, as defined by IEEE 488.1 are SH1, AH1, T5, L4, SR1, RL1, DC1, C0, E2, PP0 and DT1.

Concepts

The GPIB bus has two modes of operation: Command mode and Data mode.

The bus is in Command mode when the ATN line is high and in Data mode when ATN is low.

The Data mode is used to send device dependant messages which include all the commands and queries detailed in section 2

The command mode is used to send talk and listen addresses and the following bus commands

DCL Device Clear; Clears input and output buffers, resets the parser and clears any outstanding commands

SDC Selected Device Clear; Clears input and output buffers, resets the parser and clears any outstanding commands

GET Group Execute Trigger command: causes the DSO to arm.

IFC Interface Clear; Halts all bus activity, un addresses all devices and returns control to the controller.

1.5 About Common Commands

The common commands are defined in the IEEE 488.2 standard and are common to all instruments complying with this standard. The common commands are used to control some of the DSO's basic functions such as instrument identification and reset and the status reporting and clearing functions.

Numbers

Numbers that appear in commands must conform to certain conditions:

- The number must contain less than 20 characters,
- The mantissa may contain a sign (+/-), + will be assumed if none is specified,
- The exponent is optional, if included it must be preceded by an 'E' or 'e',
- The exponent may contain a sign (+ or -), if omitted + is assumed.

Numbers transmitted by the DSO will, in addition to the above constraints, use engineering notation.

The DSO will accept and buffer characters from the bus until a correctly terminated record is completed or until the buffer overflows.

Command Types

There are two types of command:

Query: This is a request for the current setting of a control or the current contents of a memory to be transmitted. The instrument responds by sending an appropriate record. e.g. sending TIME? will result in a record of the form

TIME<numeric_value>,<numeric_value>,<numeric_value>

being sent back.

Command: This requests that a setting or the contents of a memory be changed. Commands are always in the form 'parameter setting'. e.g. TIME 12,36,59 will set the time to 36 minutes 59 seconds past 12 o'clock.

Note: *Commands are only valid when the instrument is in Remote mode. If an command is received when the instrument is in local, the DSO will enter remote mode.*

1.6 Status Reporting

Status registers keep a note of the current instrument status. By examining the status registers, the user can find out whether an operation has been completed, whether a command has caused an error and other information about the instrument. See figure 1.6 for the contents and names of the status bytes on the DSO.

1.7 Bulk Transfers

Bulk transfers were devised to allow a remote user to send or receive either trace or instrument set-up data. In particular, transfers out of the instrument allow useful or important traces and instrument set-ups to be saved on an external storage device, such as a PC. Transfers into the scope allow this saved information to be recalled, or, in the case of trace transfers, custom traces to be created.

This latter point highlights an important difference between trace and set-up transfers. Instrument set-up transfers are provided only as a means of backing up and restoring useful set-ups. It is not possible to make up custom, transferable

set-ups on, say, a PC, then transmit them to the instrument since the meaning of the values transmitted and stored as part of a set-up is only understandable by the DSO.

The Commands used for data transfers are shown below. For details of the command syntax and usage see the appropriate entry in the command detail section.

TRACE TRANSFERS SET-UP TRANSFERS

```
:TRANsfer:ACQuisition      :TRANsfer:SETup
:TRANsfer:MAIN
:TRANsfer:ZOOM
:TRANsfer:LIMIT:UPper
                        :LOWer
```

The data transmitted between the DSO and controller consists of two distinct parts. A header giving information about the destination or source of the following trace or setup data, and the trace or set-up data itself.

In the case of transfers from the DSO to a controller the header is optional and can be disabled or enabled using the :SYS:HEADER<Boolean> command. If a set-up is received by a controller without a header, a suitable header can be created prior to sending the set-up data back to the DSO.

Trace Transfers

Trace transfers fall into 5 categories based on the source or destination of the data.

ACQuisition transfers send and receive data that has been acquired by the DSO

MAIN and ZOOM transfers send and receive data that is being displayed on the screen

It is important to understand the difference between ACQuisition and MAIN & ZOOM transfers. ACQuisition transfer send or receive data acquired by the DSO, MAIN and DISPLAY transfers send and receive data currently being displayed by the DSO.

When a memory length of 0.5 k is in use, ACQuisition and MAIN are one and the same, as the DSO normally displays all 500 points of the data it is acquiring. However, with a longer memory length the DSO only displays a portion of the data being acquired. In this case, ACQuisition transfers will transfer the entire acquired trace data as before, but DISPlay transfers will send or receive the *displayed trace data*, i.e. the 500 point trace that appears on the screen.

A ZOOM transfer will similarly always be a 500 point transfer of the displayed zoomed portion of the main trace.

Furthermore, the ACQuisition and MEMory transfers can be in one of two forms; ALL or DATAonly.

ALL type transfers.

The 'trace data' transmitted by the DSO consists of 3 parts. Firstly 500, 5000 or 50000 data bytes that represent the points acquired by the DSO. Secondly, this is followed immediately by a series of data bytes which contain information about the instrument settings at the time the trace was acquired. The third and final part is data used to check the integrity of these instrument settings. Any errors in this data will cause the scope to reject the entire transfer.

ALL type transfers are always longer than DATAonly transfers.

The TRANsfer:WindowSTart and TRANsfer:WindowSTop functions described below have no effect on ALL type transfers.

ALL type transfers are designed to provide a way of archiving and restoring all information about the requested trace.

DATAonly transfers

Contain only bytes of data representing the trace as displayed on the screen. In this mode the TRANsfer:WindowSTart and TRANsfer:WindowSTop functions described below come into effect to define the portion of the traces to transfer.

DATAonly transfers are designed to provide a quick and flexible way of transferring acquired trace data.

However, the state of the DSO is not included in this transfer. Thus, when a DATAonly transfer is used to send trace data to a destination trace, the DSO will use the information associated with the original destination trace in order to make measurements.

LIMIT:UPPER and LIMIT:LOWER transfers are always DATAonly type transfers.

Default Transfers

By default as transfers are DATAonly. This means it is not necessary to specify 'DATAonly' as part of the command, i.e. the two queries

```
TRANSFER:ACQ?1 and
TRANSFER:ACQ:DATAONLY?1
```

are identical.

:TRANsfer:LIMIT transfers are always DATAonly and therefore 'DATAonly' should not be specified as part of the corresponding command.

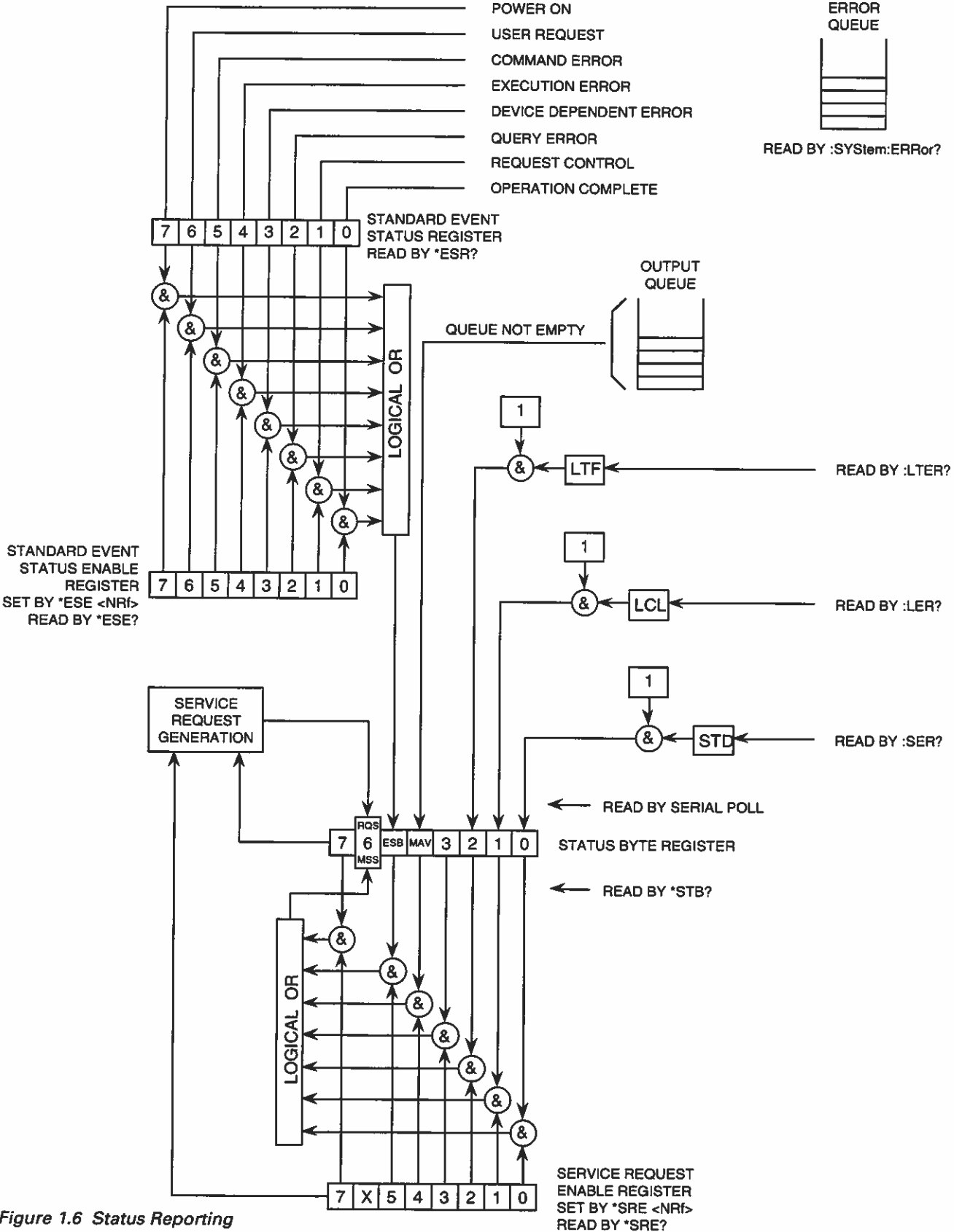


Figure 1.6 Status Reporting

DSO To Controller Transfer Failures

Under normal circumstances the DSO will send out exactly the data requested by a controller. However, with REFERENCETrace and SETUp transfers the corresponding requested data may not be valid.

For example, if reference trace 2 has never been used, the command TRANSFER:REFERENCETRACE2M? is invalid. The same is true of unused SETups and TRaces. If such a request is made of the DSO it will a) report an error in the usual way, and b) send to the controller the expected number of bytes of data, all of which will be zeros.

DATAonly Transfer Windows

With DATAonly transfers the DSO provides the facility to transfer any subsection of any trace, to and from the instrument. This can be useful if, say, only the center 4 divisions of a trace are of interest. Clearly transfers will complete faster if less trace data is requested or sent.

The section of the trace to be transferred is defined using the TRANSfer:WindowSTArt and TRANSfer:Window STOP commands. Remember that for a 0.5 k memory length there are 50 sample points per division, so the center 4 divisions would be selected as follows.

TRANSfer:WindowSTArt 150; WindowSTOP 349

Transfers to and from the DSO will now be restricted to the 200 points in the center of the trace.

Sending the command:

TRANSfer:WindowSTArt 250; WindowSTOP 449

will select divisions 5 to 9 inclusive, and data transmitted to the DSO in DATAonly mode will appear in that position on the screen.

One of the most common errors encountered when transferring data to and from the DSO is an unexpectedly short transfer. This is often the case when the transfer window required has not explicitly set. In general, the transfer window should be set to the required value prior to performing transfers rather than assuming that the DSO was last used with the required window.

Textual and Binary (raw) Transfer Formats

Having discussed 'what' data can be transmitted to and from the DSO it is now necessary to define the form of the transfers. Trace or set-up data can be transmitted by the instrument in two forms.

Textual transfers consist of ASCII letters and numbers alone, that is they are in a human readable form. For example, a trace acquired by the instrument may start low and rise within the first few samples. In this case the data output by the instrument would be of the form,

12,100,175,180,185,

and so on. As can be seen this is easily readable, but has the disadvantage that up to 4 ASCII characters are sent for each sample point in the source trace. Naturally this makes the transfer time longer.

If a transfer header is produced by the DSO or controller it will always be in an ASCII (human readable) form. Any user generated traces must also have the header in ASCII form.

Transfer Radix

For textual transfers from DSO to controller it is possible to select the number base used by the DSO instrument when converting the source trace data into ASCII. The factory set default radix is DECIMAL as shown above. HEXADECIMAL or OCTAL can also be selected, which would produce the following output for the example above,

#HC,#H64,#HAF,#HB4,#HB9

#Q14,#Q144,#Q257,#Q264,#Q271

In general, hexadecimal transfers are longer than decimal transfers and octal transfers are longer than hexadecimal transfers. The ability to generate values in different number bases allows the recipient of the data greater flexibility when attempting to decode the trace transfer into a usable form.

For textual transfers from a controller to the DSO, a mixture of DECIMAL, HEXADECIMAL and OCTAL is acceptable. Thus,

12,#H64,175,#Q264,#HB9

will be accepted by the DSO and will be translated into the same trace data as the three previous examples.

During textual transfers the data transmitted must conform to the IEEE standards for <DECIMAL PROGRAM DATA> or <NON-DECIMAL PROGRAM DATA>.

Binary Transfers

The advantage of binary transfers over textual transfers is their speed. Each item of data transferred is sent as an 8 bit value. Apart from the overhead of the header, there is a one to one relationship between the length of data requested (see TRANSFER WINDOWS above) and the number of bytes sent. Effectively, trace or set-up data from the DSO is transferred to a controller without translation to ASCII text format first.

This type of transfer is incompatible with RS423 communications with XON/XOFF handshaking because the special control characters used to implement XON/XOFF can appear as part of the trace or set-up data, causing the transfer to fail. Either RTS/CTS handshaking or textual transfers should be used instead.

The data part of a binary/raw transfer consists of four items.

- 1) a single '#' character
- 2) a single character selected from '1', '2' ... '9' which specifies the length of part 3)
- 3) 1 to 9 characters which make define the number of bytes of data to follow
- 4) up to 999,999,999 bytes of data

In practice the DSO will only transmit and receive between 1 and approximately 200800 bytes of data as there is no transfer command that would generate or require amounts of data outside this range. Excessive data sent to the DSO beyond that required by the command in operation will be accepted but discarded.

During binary (raw) transfers the data transmitted must conform to the IEEE standards for <DEFINITE LENGTH ARBITRARY BLOCK RESPONSE DATA>.

2 RS423/GPIB Commands

This section contains information about the remote commands used and recognized by the DSO. Following the summary in section 2.1 below, the commands in each subsystem are described in alphabetical order of subsystem.

2.1 RS423/GPIB Command Summary

This alphabetical list of the interface commands can be used to find the subsystem to which a command belongs. To find detailed explanations of a command see the appropriate subsystem section (section 3 to 26).

Command	Subsystem
---------	-----------

*CAL	Common
*CLS	Common
*ESE	Common
*ESR	Common
*IDN	Common
*OPC	Common
*OPT	Common
*RST	Common
*SRE	Common
*STB	Common
*TRG	Common
*TST	Common
*WAI	Common
:ACQuisition	:TRANsfer
:ADELayNGatesB	:TRIGgerTOOLS
:ADELayTimeGatesB	:TRIGgerTOOLS
:ALL	:TRANsfer:ACQuisition
:ALL	:TRANsfer:GRAPH
:ALL	:TRANsfer:HISTOgram
:ALL?	Common
:ALLColors	:DISPlay
:ANNotation	:MEASurements:XY
:ANNotation	:MEASurements:YT
:AnyChangeQUALification	:TRIGgerTOOLS[AB]:COMBination
:AUTosetup	Root
:AUTotrig	:TRIGger[AB]
:AVeraGe	:ACQuisition
:BACKground	:DISPlay
:BAND	:TRIGgerTOOLS
:BANDsize	:LIMits
:BandwidthLimit	Root
:BAUD	:RS423
:BEEP	Root
:BGatesaDelayN	:TRIGgerTOOLS
:BINS	:ANALysis:TRace[12345678] :HISTOgram

Command	Subsystem
---------	-----------

:BORDER	:DISPlay:GRATicule
:BOUNds	:MEASurements:YT
:BPC	:RS423
:CANCEL	Root
:CLear	:MEMory
:CLEARlist	:MEASurements:XY
:CLEARlist	:MEASurements:YT
:CLEAR[12345678]	:SEQuence
:COLor	:DISPlay:BACKground
:COLor	:DISPlay:CURSors
:COLor	:DISPlay:GRATicule
:COLor	:DISPlay:GRaticuleINtersection
:COLor	:DISPlay:MainTRace[12345678]
:COLor	:DISPlay:PERSISTence
:COLor	:DISPlay:TEXT
:COLor	:DISPlay:TRaceINtersection
:COLor	:DISPlay:ZoomTR[12345678]
:COMBination	:TRIGgerTOOLS
:COMBination	:TRIGgerTOOLS[AB]
:CONTinue	:SEQuence
:CONTrol	:TRIGgerTOOLS:DIVidebyN
:COPY	:MassMEMory
:COUNT	:TRIGTOOLS:ADELayNGatesB
:COUNT	:TRIGTOOLS :BGatesaDelayN
:COUNT	:TRIGgerTOOLS:DELaybyN
:COUNT	:TRIGgerTOOLS:DIVidebyN
:COUpling	:CHANnel[1234]
:COUpling	:TRIGger[AB]
:CREate	:MassMEMory:UTILities
:CROSSing	:MEASurements:YT
:CROSSLEVel	:MEASurements:YT
:CURSors	:DISPlay
:CYCles[12345678]	:SEQuence
:DATAonly	:TRANsfer:ACQuisition
:DATAonly	:TRANsfer:GRAPH
:DATAonly	:TRANsfer:HISTOgram
:DATAonly	:TRANsfer:MAIN
:DATAonly	:TRANsfer:ZOOM
:DATE	Root
:DATEandtime[12345678]	:TRace
:DATNOisefl	:MEASurements:YT
:DATUM	:MEASurements:YT
:DatumLOCKedtotrace	:MEASurements:XY
:DATumVALue	:MEASurements:YT
:DELay	:TRIGger[AB]
:DELaybyN	:TRIGgerTOOLS
:DELeTe	:MassMEMory:UTILities
:DELeTe	:MEASurements:XY

Command	Subsystem	Command	Subsystem
:DELEte	:MEASurements:YT	:FUNDamental	:MEASurements:YT
:DEst	:PLOT	:GLitchDEtect	:ACQuisition
:DESTination	:MassMEMory:COpy	:GRAPH	:ANALysis:TRace[12345678]
:DEvice	:MMEMory:COpy:DESTination	:GRAPH	:TRANsfer
:DEvice	:MassMEMory:COpy:SOurce	:GRATicule	:DISPlay
:DEvice	:MassMEMory:SAVeRECall	:GRaticule	:PLOT
:DEvice	:MassMEMory:UTILities	:GRaticuleiNtersection	:DISPlay
:DEvice	:TRANsfer:FILE	:HAnd	:RS423
:DIFFerentiateSCaling	:ANALysis:TRace[12345678]	:HARMonic	:MEASurements:YT
:DispAREAfll	:MEASurements:XY	:HEADer	:SYStem
:DISplay	:MEASurements:XY	:HELP?	Root
:DISplay	:MEASurements:YT	:HIGHtrans	:MEASurements:YT
:DIVidebyN	:TRIGgerTOOLS	:HiLowQUALification	:TRIGgerTOOLS[AB]
:DIVideSCaling	:ANALysis:TRace[12345678]		:COMBination
:DRiVer	:PLOT	:HISTOgram	:TRANsfer
:ECHOandprompt	:RS423	:HISTOgram	:ANALysis:TRace[12345678]
:EDGE	:TRIGTOOLS[AB]:FREQuency	:Hold	:CHANnel[1234]
:EDGE	:TRIGTOOLS[AB]:SLEWrate	:HOLDAll	:DISPlay
:ENable	:ACQuisition:AVeRaGe	:HORIZontalRange	:ANALysis
:ENable	:CALibrate		:TRace[12345678]:HISTogram
:ERRor	:SYStem	:HorizSCaling	:ANALysis:TR[12345678]:FFT
:ERRorClear	:SYStem	:INtegrateSCaling	:ANALysis:TRace[12345678]
:EXECute	:ANALysis:TRace[12345678]	:KEYclick	Root
:EXECute	:MassMEMory:COpy	:KNEE	:MEASurements:YT
:EXECute	:MassMEMory:RECall	:LABELsdatecursor	:PLOT
:EXECute	:MassMEMory:SAVe	:LEFTedge	:ANALysis
:EXECute	:TRANsfer:FILE		:TRace[12345678]:GRAPH
:EXtErnaldest	:PLOT	:LEFTtrans	:MEASurements:YT
:FACTOR	:ACQuisition:AVeRaGe	:LENGth	:PLOT
:FACTOR	:MEASurements:XY	:LER?	Root
:FACTOR	:MEASurements:YT	:LEVEL{1,2,3,4,EXT}	:TRIGger[AB]
:FASTreARM	:ACQuisition	:LIMitstype	:PLOT
:FFT	:ANALysis:TRace[12345678]	:LINearSCaling	:ANALysis:TR[12345678]:FFT
:FILE	:TRANsfer	:LIST	:MassMEMory:UTILities
:FILTerFACTOR	:ANALysis:TR[12345678]:FFT	:LONGform	:SYStem
:FIXedVALue	:MEASurements:YT	:LOWer	:LIMits
:FIXedVLue	:MEASurements:XY	:LOWERresult	:ANALysis
:FORce	:CALibrate		:TRace[12345678]:HISTogram
:FORMAT	:DATE	:LOWtrans	:MEASurements:YT
:FORMat	:MassMEMory:UTILities	:LTER?	Root
:FORMat	:TRANsfer	:MAIN	:TRANsfer
:FREQuency	:TRIGgerTOOLS	:MainTRace[12345678]	:DISPlay
:FREQuency	:TRIGgerTOOLS:FREQuency	:MAJorAxes	:DISPlay:GRATicule
:FREQuency	:TRIGgerTOOLS[AB]	:MARKers	:MEASurements:YT
:FULLRESolution	:MEASurements	:MAXmin	:DISPlay
:FUNction	:ANALysis:TRace[12345678]	:MAXreading	:ANALysis
:FUNction	:MEASurements:XY		:TRace[12345678]:GRAPH
:FUNction	:MEASurements:YT	:MAXreading	:ANALysis
			:TRace[12345678]:HISTogram

Command	Subsystem	Command	Subsystem
:MEASurement	:ANALysis	:PLOT	Root
	:TRace[12345678]:GRAPH	:PlotonFAIL	:LIMits
:MEASurement	:ANALysis	:POInts	:ANALysis:TR[12345678]:FFT
	:TRace[12345678]:HISTogram	:POLarity	:TRIGgerTOOLS:RUNT
:MEASurement	:MEASurements:XY:Function	:POLarity	:TRIGgerTOOLS[AB]
:MEASurement	:MEASurements:YT:Function		:PulseWidth
:MEMoryLENGth	:ACQuisition	:POSition	:CHANnel[1234]
:MENU	Root	:PREPost	:TRIGger[AB]
:MINorAxes	:DISPlay:GRATicule	:PREtrig	:TRIGger[AB]
:MINreading	:ANALysis	:PRObe	:CHANnel[1234]
	:TRace[12345678]:GRAPH	:PulseWidth	:TRIGgerTOOLS
:MINreading	:ANALysis	:PulseWidth	:TRIGgerTOOLS[AB]
	:TRace[12345678]:HISTogram	:RADIX	:TRANSfer
:MISSing	:TRIGgerTOOLS	:RANGe	:CHANnel[1234]
:MODE	:LIMits	:RECALL	:SETup
:MODE	:TRIGgerTOOLS	:RECall	:MassMEMory
:MODE	:TRIGgerTV	:RECall	:MEMory
:MultiSHOT	:ACQuisition	:REFerence	:MEASurements:XY
:MULtiSCaling	:ANALysis:TRace[12345678]	:REFerence	:MEASurements:YT
:NAME	:MEASurements:XY	:REName	:MassMEMory:UTILities
:NAME	:MEASurements:YT	:RESet	:MEASurements:XY
:NAME	:SEQuence	:ReSeT	:MEASurements:YT
:NEWNAME	:MassMEMory:UTILities	:RESult	:MEASurements:XY
:NEWNUMBER	:MassMEMory:UTILities	:RESult	:MEASurements:YT
:NoSEGments	:ACQuisition	:RESultNUMeric	:MEASurements:XY
:OFFSet	:ANALysis:TR[12345678]:FFT	:RESultNUMeric	:MEASurements:YT
:OFFSet	:CHANnel[1234]	:RIGHTedge	:ANALysis
:OFFSet	:MEASurements:XY		:TRace[12345678]:GRAPH
:OFFSet	:MEASurements:YT	:RIGHTtrans	:MEASurements:YT
:OPERator	:MEASurements:XY:Function	:RUN	:ACQuisition
:OPERator	:MEASurements:YT:Function	:RuNnAMe	:MassMEM:COPY:DESTination
:OVERflow	:ANALysis	:RuNnAMe	:MassMEMory:COPY:Source
	:TRace[12345678]:HISTogram	:RuNnAMe	:MassMEMory:SAVeRECall
:PARity	:RS423	:RuNnAMe	:MassMEMory:UTILities
:PERiod	:TRIGgerTOOLS	:RuNnAMe	:TRANSfer:FILE
	:ADELayTimeGatesB	:RuNnUMber	:MassMEM:COPY:DESTination
:PERiod	:TRIGgerTOOLS:COMBination	:RuNnUMber	:MassMEMory:COPY:Source
:PERiod	:TRIGgerTOOLS:FREQuency	:RuNnUMber	:MassMEMory:RECall
:PERiod	:TRIGgerTOOLS:MISSing	:RuNnUMber	:MassMEMory:SAVE
:PERiod	:TRIGgerTOOLS:SKEW	:RuNnUMber	:MassMEMory:UTILities
:PERiod	:TRIGgerTOOLS:SLEWrate	:RuNnUMber	:TRANSfer:FILE
:PERSDotjoin	:DISPlay	:RUNT	:TRIGgerTOOLS
:PERSISTence	:DISPlay	:RUN[12345678]	:SEQuence
:PERSMode	:DISPlay	:SAVE	:MassMEMory
:PERSSWEeps	:DISPlay	:SAVE	:MEMory
:PERSTime	:DISPlay	:SAVE	:SETup
:PERStype	:PLOT	:SAVeRECall	:MassMEMory
:PHASEslip	:TRIGgerTOOLS:DIVidebyN	:SCAling	:MEASurements:XY

Command	Subsystem	Command	Subsystem
:SCAling	:MEASurements:YT	:STopBits	:RS423
:ScreenXpos	:CURSor	:SWEep	:DISPlay
:ScreenXpos	:DATum	:syncPOLarity	:TRIGgerTV
:ScreenYpos	:CURSor	:SYStem	:TRIGger
:ScreenYpos	:DATum	:TERM50	:CHANnel[1234]
:SEGment	:MEASurements:YT	:TEXT	:DISPlay
:SEGmenTtime[12345678] :TRace		:TEXT	Root
:SELEct	:TRIGger	:TIME	Root
:SEQuence	:MassMEM:COPY:DESTination	:TimeBASE	:ACQuisition
:SEQuence	:MassMEMory:COPY:Source	:TimeDATumREFerence	:MEASurements:YT
:SEQuence	:MassMEMory:SAVeRECall	:TOPbase	:MEASurements:YT
:SEQuence	:MassMEMory:UTILities	:TRACe	:MassMEMory:RECall
:SEQuence	:TRANsfer:FILE	:TRACe	:CURSor
:SEQueNcenUMber	:MassMEMory:SAVeRECall	:TRACEDotjoin	:DISPlay
:SER?	Root	:TRaceINTersection	:DISPlay
:SetBand	:LIMits	:TRACeNAME	:MassMEM:COPY:DESTination
:SETup	:MassMEM:COPY:DESTination	:TRACeNAME	:MassMEMory:COPY:Source
:SETup	:MassMEMory:COPY:Source	:TRACeNAME	:MassMEMory:RECall
:SETup	:MassMEMory:SAVeRECall	:TRACeNAME	:MassMEMory:UTILities
:SETup	:MassMEMory:UTILities	:TRACeNAME	:TRANsfer:FILE
:SETup	:TRANsfer:FILE	:TRace[12345678]	:ANALysis
:SKEW	:TRIGgerTOOLS	:TRace{1M,2M,3M,4M,5M,6M,7M,8M,1Z,2Z,3Z,4Z,5Z,6Z,7Z,8Z}	:DISPlay
:SLEWrate	:TRIGgerTOOLS	:TRaceSCAling[12345678] :TRace	
:SLEWrate	:TRIGgerTOOLS[AB]	:TRANSDIREction	:MEASurements:YT
:SLOPE{1,2,3,4,EXT}	:TRIGger[AB]	:TRANSNUMber	:MEASurements:YT
:SOFTkey	:SEQuence	:TRANSparency	:MENU
:Source	:MassMEMory:COPY	:TRIGger	:TRIGgerTOOLS[AB]
:Source	:TRIGgerTOOLS[AB]		:COMBination
	:COMBination		
:Source	:TRIGgerTV	:TruTRACE	:DISPlay
:Source	:TRIGger[AB]	:TYPE	:ANALysis:TRace[12345678]
:Source1	:ANALysis:TRace[12345678]	:TYPE	:MassMEMory:COPY
:SOUrce1YT	:MEASurements:XY	:TYPE	:MassMEMory:RECall
:SOUrce1YT	:MEASurements:YT	:TYPE	:MassMEMory:SAVE
:Source2	:ANALysis:TRace[12345678]	:TYPE	:TRANsfer:FILE
:SOUrce2YT	:MEASurements:XY	:UNcal	:CHANnel[1234]
:SOUrce2YT	:MEASurements:YT	:UNDERflow	:ANALysis
:STanDardreference	:MEASurements:XY		:TRace[12345678]:HISTogram
:STanDardreference	:MEASurements:YT		
:STATE	:MEASurements	:UNITS	:MEASurements:XY
:StAte	:DISPlay:GRATicule	:UNITS	:MEASurements:YT
:StAte	:DISPlay:PERSISTence	:UpdateRATE	:ANALysis
:StAte	:DISPlay		:TRace[12345678]:GRAPH
	:TRace{1M,2M,3M,4M,5M,6M,7M,8M,1Z,2Z,3Z,4Z,5Z,6Z,7Z,8Z}	:UpdateRATE	:ANALysis
			:TRace[12345678]:HISTogram
:Status	:ACQuisition	:UPper	:LIMits
:STOP	:ANALysis	:UPPERresult	:ANALysis
:STOP	:ANALysis:TRace[12345678]		:TRace[12345678]:HISTogram

Command	Subsystem	Command	Subsystem
:USER	:MassMEMor:COpy:Source	:WindowSTOp	:TRANsfer:ACQuisition
:USER	:MassMEM:COpy:DESTination	:WindowSTOp	:TRANsfer:GRAPH
:USER	:MassMEMory:SAVeRECall	:WindowSTOp	:TRANsfer:HISTOgram
:USER	:MassMEMory:UTILities	:WindowSTOp	:TRANsfer:MAIN
:USER	:TRANsfer:FILE	:WindowSTOp	:TRANsfer:ZOOM
:UTILities	:MassMEMory	:WORDsize	:TRANsfer
:VALue	:MEASurements:YT:FUnction	:X	:TRace:TRSCaling[12345678]
:VARIABLE	:CHANnel[1234]	:Xpos	:CURSor
:VertSCaling	:ANALysis:TR[12345678]:FFT	:Xpos	:DATum
:WEIGHTing	:MEASurements:XY	:XSHift	:DISPlay
:WEIGHTing	:MEASurements:YT	:XY	:DISPlay
:WHEN	:TRIGgerTOOLS:BAND	:XY	:MEASurements
:WHEN	:TRIGgerTOOLS:PulseWidth	:Y	:TRace:TRSCaling[12345678]
:WHEN	:TRIGgerTOOLS:SKEW	:Ypos	:CURSor
:WHEN	:TRIGgerTOOLS:SLEWrate	:Ypos	:DATum
:WHENFrequency	:TRIGgerTOOLS:FREQuency	:YSCaling	:ANALysis
:WHENPeriod	:TRIGgerTOOLS:FREQuency		:TRace[12345678]:HISTogram
:WIDth	:TRIGgerTOOLS:PulseWidth	:YT	:MEASurements
:WINDow	:ANALysis:TR[12345678]:FFT	:YZoomFACtor	:DISP:ZoomTRace[12345678]
:WindowSTArt	:TRANsfer:ACQuisition	:YZoomPOSition	:DISP:ZoomTRace[12345678]
:WindowSTArt	:TRANsfer:GRAPH	:ZOOM	:DISPlay
:WindowSTArt	:TRANsfer:HISTOgram	:ZOOM	:TRANsfer
:WindowSTArt	:TRANsfer:MAIN	:ZoomPOSition	:DISPlay
:WindowSTArt	:TRANsfer:ZOOM	:ZoomTRace[12345678]	:DISPlay

3 The ACQuisition Subsystem is used for controlling all aspects of the instrument's acquisition performance.

COMMANDS

```
:ACQuisition
  :AVeraGe
    :ENable
    :FACTOR
  :FASTreARM
  :GLitchDETECT
  :MEMoryLENGth
  :MultiSHOT
  :NoSEGments
  :RUN
  :SAMPLEmode
  :Status
  :TimeBASE
```

Name:	AVeraGe:ENable	Command, Query
Function:	<p>This command turns the averaging function on or off. Numeric values may be sent, 0 is interpreted as OFF and any other number is interpreted as ON.</p> <p>This query returns the current state of the averaging function.</p>	
Command Syntax:	:ACQuisition:AVeraGe:ENable <Boolean>	
Example:	:ACQuisition:AVeraGe:ENable ON	
Query syntax:	:ACQuisition:AVeraGe:ENable?	
Returned Format:	:ACQuisition:AVeraGe:ENable <Boolean>	

Name:	AVeraGe:FACTOR	Command, Query
Function:	<p>This command sets the averaging factor. The possible factors are 2, 4, 8, 16, 32, 64, 128, 256, 512 and 1024 although any number may be sent as it will be rounded to the nearest allowable factor. 512 and 1024 are not valid for 50 k or 200 k acquisitions.</p> <p>This query returns the current averaging factor.</p>	
Command Syntax:	:ACQuisition:AVeraGe:FACTOR <numeric_value>	
Example:	:ACQuisition:AVeraGe:FACTOR 64	
Query syntax:	:ACQuisition:AVeraGe:FACTOR?	
Returned Format:	:ACQuisition:AVeraGe:FACTOR <numeric_value>	

Name: **FASTreARM** Command, Query

Function: This command is used to set the type of multishot acquisitions and works by turning the fast rearm feature ON or OFF

This query shows the current state of FASTreARM.

Command Syntax: :ACQuisition:FASTreARM <boolean>

Example: :ACQuisition:FASTreARM ON

Query syntax: :ACQuisition:FASTreARM?

Returned Format: :ACQuisition:FASTreARM <boolean>

Name: **GLitchDETECT** Command, Query

Function: This command turns the acquisition glitch detect function on and off. Numeric values may be sent, 0 is interpreted as OFF and any other number is interpreted as ON.

When set to ON, DISPLAY :MAXmin is also set to on.

This query returns the current state of the glitch detect function.

Command Syntax: :ACQuisition:GLitchDETECT <Boolean>

Example: :ACQuisition:GLitchDETECT ON

Query syntax: :ACQuisition:GLitchDETECT?

Returned Format: :ACQuisition:GLitchDETECT <Boolean>

Name: **MEMoryLENgth** Command, Query

Function: This command is used to set the acquisition memory length Valid values are 500, 5000, 50000 and 200000 for 0.5 k, 5 k, 50 k and 200 k respectively.

This query returns the current acquisition memory length.

Command Syntax: :ACQuisition:MEMoryLENgth <numeric_value>

Example: :ACQuisition:MEMoryLENgth 500

Query syntax: :ACQuisition:MEMoryLENgth?

Returned Format: :ACQuisition:MEMoryLENgth <numeric_value>

Name: **MultiSHOT** Command, Query

Function: This command is used to control the acquisition multishot function.

This query returns the current state of the acquisition multishot function.

Command Syntax: :ACQuisition:MultiSHOT <Boolean>

Example: :ACQuisition:MultiSHOT ON

Query syntax: :ACQuisition:MultiSHOT ?

Returned Format: :ACQuisition:MultiSHOT <Boolean>

ACQuisition Subsystem

ACQuisition Subsystem

Name: NoSEGments Command, Query

Function: This command is used to set the number of segments when the instrument is in multishot mode. The number can be either 2, 10 or 20.
This query returns the number of segments selected for multishot acquisitions.

Command Syntax: :ACQuisition:NoSEGments <numeric_data>

Example: :ACQuisition:NoSEGments 10

Query syntax: :ACQuisition:NoSEGments?

Returned Format: :ACQuisition:NoSEGments <numeric_data>

Name: RUN Command, Query

Function: This command has the same effect as pressing the front panel run button (17) or the single shot button (15). <data> can be either Single or Continuous.
This query returns the current run status.

Command Syntax: :ACQuisition:RUN <data>

Example: :ACQuisition:RUN Continuous

Query syntax: :ACQuisition:RUN?

Returned Format: :ACQuisition:RUN <data>

Name: SAMPLEmode Command, Query (6500 and 9500 only)

Function: This command changes the sample mode from repetitive sampling (ets) to transient (single shot) on those timebases which are capable of supporting both modes. <data> can be either REPetitive or TRANsient.
This query returns the current samplemode status.

Command Syntax: :ACQuisition:SAMPLEmode <data>

Example: :ACQuisition:SAMPLEmode REPetitive

Query syntax: :ACQuisition:SAMPLEmode?

Returned Format: :ACQuisition:SAMPLEmode <data>

3

Name:	STatus?	Query
Function:	This query returns the current acquisition status of the instrument. A Stands for 'Armed' F Indicates that the acquisition store has been 'filled' with enough pre-trigger information to be ready for a fresh acquisition. T Stands for 'Triggered': the instrument has received or generated a valid trigger; acquisition has begun. S 'Stored': acquisition is complete. <data> will be A, AF, AFT, AFS or AFTS	
Query syntax:	:ACquisition:STatus?	
Returned Format:	:ACquisition:STatus <data>	

Name:	TimeBASE	Command, Query
Function:	This command is used to set the timebase speed in seconds per division. The valid values depend on the store length. 200 k stores 200 to 100E-6 50 k stores 200 to 50E-6 5 k stores 200 to 5E-6 0.5 k stores 200 to 20E-9 The complete range of values is: 10E-9, 20E-9, 25E-9, 50E-9, 100E-9, 200E-9, 250E-9, 500E-9, 1E-6, 2E-6, 2.5E-6, 5E-6, 10E-6, 20E-6, 25E-6, 50E-6, 100E-6, 200E-6, 250E-6, 500E-6, 1E-3, 2E-3, 2.5E-3, 5E-3, 10E-3, 20E-3, 25E-3, 50E-3, 100E-3, 200E-3, 250E-3, 500E-3, 1, 2, 2.5, 5, 10, 20, 25, 50, 100, 200 and 250. Although the above values are the most obvious choices, numbers such as 5E-1 or 0.0005E3 are just as valid. Any other number may be sent and it will be rounded. The values that are multiples of 2.5 are only available on the 6500 and 9500. If EXTERNAL is sent, the DSO will use a TTL signal connected to the rear panel connector as the clock signal. This query returns the current timebase speed.	
Command Syntax	:ACquisition:TimeBASE <numeric_value> EXTERNAL	
Example:	:ACquisition:TimeBASE 200E-6	
Query syntax:	:ACquisition:TimeBASE?	
Returned Format:	:ACquisition:TimeBASE <numeric_value> EXTERNAL	

4 The ANALysis Subsystem is used for controlling all aspects of the instrument's acquisition performance.

COMMANDS

:ANALysis

:STOP (command only)

:TRace[12345678]

:DIFFerentiateSCaling

:DIVideSCaling

:EXECute (command only)

:FFT

:HorizSCaling

:LINearSCaling

:OFFSet

:POInts

:VertSCaling

:WINDow

:FILTerFActor

:FUnction

:GRAPH

:LEFTedge

:MAXreading

:MEASurement

:MINreading

:RIGHTedge

:UpdateRATE

:HISTogram

:BINS

:HORIZontalRange

:LOWERresult

:MAXreading

:MEASurement

:MINreading

:OVERflow

:UNDERflow

:UpdateRATE

:UPPERresult

:YSCaling

:INTegrateSCaling

:MULtSCaling

:SOurce1

:SOurce2

:STOP (command only)

:TYPE

Name: DIFFerentiateSCaling Command, query

Function: This command is used to set the scaling factor applied to a differentiated trace when the DIFFerentiate function is used.

The scaling factors for <data> are: x0.5, x1.0, x2.0, x5.0, x10.0, x20.0 and x50.0.

This query returns the current scaling factor to be used by the DIFFerentiate function.

4

Command Syntax: :ANALYSIS:TRace[12345678]:DIFFerentiateSCaling <data>

Example: :ANALYSIS:TRace[12345678]:DIFFerentiateSCaling x5.0

Query Syntax: :ANALYSIS:TRace[12345678]:DIFFerentiateSCaling?

Returned format: :ANALYSIS:TRace[12345678]:DIFFerentiateSCaling <data>

Name: DIVideSCaling Command, query

Function: This command is used to set the scaling factor applied to a divided trace when the DIVide function is used.

The scaling factors for <data> are: x0.1, x0.2, x0.5 and x1.0.

This query returns the current scaling factor to be used by the DIVide function.

Command Syntax: :ANALYSIS:TRace[12345678]:DIVideSCaling <data>

Example: :ANALYSIS:TRace6:DIVideSCaling x0.5

Query Syntax: :ANALYSIS:TRace[12345678]:DIVideSCaling?

Returned format: :ANALYSIS:TRace[12345678]:DIVideSCaling <data>

Name: EXEcute Command

Function: The operation of this command depends upon the :ANALYSIS:TRACeCALCulate mode.

When the :ANALYSIS :TRACeCALCulate mode is set to manual, and the instrument is single-shot stored, this command will cause the set arithmetic operation to be performed.

When the :ANALYSIS :TRACeCALCulate mode is set to auto, this command will cause the set arithmetic operation to be performed at the end of every acquisition. Before issuing this command, the destination, function and operands should have been set up.

Command Syntax: :ANALYSIS:TRace[12345678]:EXEcute

Example :ANALYSIS:TRace2:EXEcute

ANALysis Subsystem

ANALysis Subsystem

Name: **FFT:HorizSCaling** Command, query

Function: This command selects the horizontal scaling type to be used for an FFT on a trace. The scaling can be either LINear or LOG.

This query returns the currently set horizontal scaling type for FFTs.

Command Syntax: :ANALysis:TRace[12345678]:FFT:HorizSCaling <data>

Example: :ANALysis:TRace4:FFT:HorizSCaling LOG

Query Syntax: :ANALysis:TRace[12345678]:FFT:HorizSCaling?

Returned format: :ANALysis:TRace[12345678]:FFT:HorizSCaling <data>

Name: **FFT:LINearSCaling** Command, query

Function: This command sets the scaling factor for the FFT results. The range for <numeric_data> is from 1 to 1000. This scaling factor is a divisor so that the area of interest can be viewed. The scaling factor is applied when the FFT is acquired.

This query returns the currently set scaling factor for FFT results.

Command Syntax: :ANALysis:TRace[12345678]:FFT:LINearSCaling <numeric_data>

Example: :ANALysis:TRace2:FFT:LINearSCaling 200

Query Syntax: :ANALysis:TRace[12345678]:FFT:LINearSCaling?

Returned Format: :ANALysis:TRace[12345678]:FFT:LINearSCaling <numeric_data>

Name: **FFT:OFFSet** Command, query

Function: This command sets the offset in samples to be used for FFT results. The offset range for <numeric_data> is up to the number of samples and varies according to the current memory length and the value set by the POInts command. The offset and the points must total less than the memory length.

This query returns the offset to be used for FFT results.

Command Syntax: :ANALysis:TRace[12345678]:FFT:OFFSet <numeric_data>

Example: :ANALysis:TRace2:FFT:OFFSet 120

Query Syntax: :ANALysis:TRace[12345678]:FFT:OFFSet?

Returned Format: :ANALysis:TRace[12345678]:FFT:OFFSet <numeric_data>

Name: **FFT:POInts** Command, query

Function: This command sets the number of points of the acquired trace that are to be used for the FFT. The number of permitted points for <numeric_data> varies with memory length:

For 500 point traces the only value is 256. For 5000 point traces the values are 256 512, 1024, 2048 and 4096. For 50,000 and 200,000 point traces the values are 256, 512, 1024, 2048, 4096, 8182 and 16384.

This query returns the currently set number of points for an FFT

Command Syntax: :ANALysis:TRace[12345678]:FFT:POInts <numeric_data>

Example: :ANALysis:TRace7:FFT:POInts 256

Query Syntax: :ANALysis:TRace[12345678]:FFT:POInts?

Returned Format: :ANALysis:TRace[12345678]:FFT:POInts<numeric_data>

ANALysis Subsystem

ANALysis Subsystem

Name: **FFT:VertMode** Command, query

Function: This command selects the vertical mode to be used for an FFT on a trace. The mode can be either AUTO or FIXED.
This query returns the currently set vertical mode for FFTs.

Command Syntax: :ANALysis:TRace[12345678]:FFT:VertMode <data>

Example: :ANALysis:TRace5:FFT:VertMode AUTO

Query Syntax: :ANALysis:TRace[12345678]:FFT:VertMode?

Returned format: :ANALysis:TRace[12345678]:FFT:VertMode <data>

Name: **FFT:VertSCaling** Command, query

Function: This command selects the vertical scaling type to be used for an FFT on a trace. The scaling can be either LINEar or LOG.
This query returns the currently set vertical scaling type for FFTs.

Command Syntax: :ANALysis:TRace[12345678]:FFT:VertSCaling <data>

Example: :ANALysis:TRace5:FFT:VertSCaling LINEar

Query Syntax: :ANALysis:TRace[12345678]:FFT:VertSCaling?

Returned format: :ANALysis:TRace[12345678]:FFT:VertSCaling <data>

Name: **FFT:WINDow** Command, query

Function: This command selects the windowing function to be used for an FFT on a trace. The windowing can be either HANNing or RECTangular.
The query returns the current window setting.

Command Syntax: :ANALysis:TRace[12345678]:FFT:WINDow <data>

Example: :ANALysis:TRace3:FFT:WINDow RECTangular

Query Syntax: :ANALysis:TRace[12345678]:FFT:WINDow?

Returned format: :ANALysis:TRace[12345678]:FFT:WINDow <data>

Name: **FILterFACtor** Command, Query

Function: This command is used to set the filter factor used by the FILter command. One of 6 filter factors; 1 – 6 can be selected. The actual filter frequency depends on the timebase rate.
This query returns the filter factor set to be used by the FILter command.

Command Syntax: :ANALysis:TRace[12345678]:FILterFACtor <numeric_value>

Example: :ANALysis:TRace1: FILterFACtor 5

Query syntax: :ANALysis:TRace[12345678]:FILterFACtor?

Returned Format: :ANALysis:TRace[12345678]:FILterFACtor <numeric_value>

ANALysis Subsystem

ANALysis Subsystem

Name:

Function

Command, Query

Function:

This command is used to specify which analysis function is to be performed on the specified, stored trace. Scaling factors for relevant commands are set using the appropriate scaling commands. When using the DISPLAYchannel function, first set the required channel using the SOURCE1 command. Remember that traces 1 to 4 can only be mapped to channels 1 to 4 respectively. Traces 5 to 8 can be mapped to any of the input channels.

The available functions are:

NONE	FILter	INTEgrate	DIFFerentiate
DISPLAYchannel	INVert	FFT	ADD
SUBtract	MULTiply	DIVide	GRAPH
HISTogram			

This query returns the currently set analysis function for the specified trace.

Command Syntax:

:ANALysis:TRace[12345678]:FUnction <data>

Example:

:ANALysis:TRace3:FUnction:INTEgrate

Query Syntax:

:ANALysis:TRace[12345678]:FUnction?

Returned format:

:ANALysis:TRace[12345678]:FUnction: <data>

Name:	GRAPH:LEFTedge	Query
Function:	This command reads the sample number of the left most point on the graph. If the graph has not scrolled, this value will be zero.	
Query Syntax:	:ANALYSIS:TRACE[12345678]:GRAPH:LEFTedge?	
Returned format:	:ANALYSIS:TRACE[12345678]:GRAPH:LEFTedge <numeric_value>	

Name:	GRAPH:MAXreading	Command, Query
Function:	<p>This command sets the value of the measurement that corresponds to the top of the screen.</p> <p>This query returns the value that corresponds with the top of the screen.</p>	
Command Syntax:	:ANALYSIS:TRACE[12345678]:GRAPH:MAXreading <numeric_value>	
Example:	:ANALYSIS:TRACE1:GRAPH:MAXreading 300	
Query Syntax:	:ANALYSIS:TRACE[12345678]:GRAPH:MAXreading?	
Returned format:	:ANALYSIS:TRACE[12345678]:GRAPH:MAXreading <numeric_value>	

Name:	GRAPH:MEASurement	Command, Query
Function:	<p>This command sets the sub measurement to be graphed. The sub measurement will be 1, 2 or 3. The measurement that is being graphed can then be worked out by looking at YT cursor measurements, calculation1.</p> <p>This query returns the sub measurement that is to be graphed.</p>	
Command Syntax:	:ANALYSIS:TRACE[12345678]:GRAPH:MEASurement <numeric_value>	
Example:	:ANALYSIS:TRACE7:GRAPH:MEASurement 1	
Query Syntax:	:ANALYSIS:TRACE[12345678]:GRAPH:MEASurement?	
Returned format:	:ANALYSIS:TRACE[12345678]:GRAPH:MEASurement <numeric_value>	

Name: **GRAPH:MINreading** Command, Query

Function: This command sets the value of the measurement that corresponds to the bottom of the screen.
This query returns the value of the measurement that corresponds to the bottom of the screen.

Command Syntax: :ANALysis:TRace[12345678]:GRAPH:MINreading <numeric_value>

Example: :ANALysis:TRace4:GRAPH:MINreading 1

Query Syntax: :ANALysis:TRace[12345678]:GRAPH:MINreading?

Returned format: :ANALysis:TRace[12345678]:GRAPH:MINreading <numeric_value>

Name: **GRAPH:RIGHTedge** Query

Function: This command reads the sample number of the right most point on the graph.

Query Syntax: :ANALysis:TRace[12345678]:GRAPH:RIGHTedge?

Returned format: :ANALysis:TRace[12345678]:GRAPH:RIGHTedge <numeric_value>

Name: **GRAPH:UpdateRATE** Command, Query

Function: This command sets the update rate of the graphed measurement. The rate can either be the current acquisition rate or a number of seconds:
ACQ, 1, 2, 5, 10, 20, 50, 100
This query returns the currently set update rate.

Command Syntax: :ANALysis:TRace[12345678]:GRAPH:UpdateRATE <data>|<numeric_value>

Example: :ANALysis:TRace2:GRAPH:UpdateRATE ACQ

Query Syntax: :ANALysis:TRace[12345678]:GRAPH:UpdateRATE?

Returned format: :ANALysis:TRace[12345678]:GRAPH:UpdateRATE <data>|<numeric_value>

Name: **HISTograms:BINS** Command, Query

Function: This command sets the number of bins into which measurement results are put. The number of bins can be 10, 20, 50, 100, 250 or 500.
This query returns the currently set number of bins.

Command Syntax: :ANALysis:TRace[12345678]:HISTogram:BINS <numeric_value>

Example: :ANALysis:TRace3:HISTogram:BINS 250

Query Syntax: :ANALysis:TRace[12345678]:HISTogram:BINS?

Returned format: :ANALysis:TRace[12345678]:HISTogram:BINS <numeric_value>

ANALysis Subsystem

ANALysis Subsystem

4

Name: **HISTogram:HORIZontalRange** Command, Query

Function: This command is used to set the mode of the horizontal range for histograms. The two modes are MANUAL or AUTO.

In auto mode the range is automatically adjusted when values are picked up which lie outside the current range.

This query returns the current horizontal range mode.

Command Syntax: :ANALysis:TRace[12345678]:HISTogram:HORIZontalRange <data>

Example: :ANALysis:TRace4:HISTogram:HORIZontalRange AUTO

Query Syntax: :ANALysis:TRace[12345678]:HISTogram:HORIZontalRange?

Returned format: :ANALysis:TRace[12345678]:HISTogram:HORIZontalRange <data>

Name: **HISTogram:LOWERresult** Query

Function: Return the highest measurement result recorded in the underflow bin, or the minimum range of the histogram if the underflow bin is empty.

Query Syntax: :ANALysis:TRace[12345678]:HISTogram:LOWERresult?

Returned format: :ANALysis:TRace[12345678]:HISTogram:LOWERresult <numeric_value>

Name: **HISTogram:MAXreading** Command, Query

Function: This command sets the value of the measurement that corresponds to the right hand side of the screen.

This query returns the value of the measurement that corresponds to the right hand side of the screen.

Command Syntax: :ANALysis:TRace[12345678]:HISTogram:MAXreading <numeric_value>

Example: :ANALysis:TRace5:HISTogram:MAXreading 425

Query Syntax: :ANALysis:TRace[12345678]:HISTogram:MAXreading?

Returned format: :ANALysis:TRace[12345678]:HISTogram:MAXreading <numeric_value>

Name: **HISTogram:MEASurement** Command, Query

Function: This command sets the sub measurement to perform a histogram of. The sub measurement will be 1, 2 or 3. The measurement that the histogram is being produced from can then be worked out by looking at YT cursor measurements, calculation1.

This query returns the sub measurement that is to be used to produce a histogram.

Command Syntax: :ANALysis:TRace[12345678]:HISTogram:MEASurement <numeric_value>

Example: :ANALysis:TRace5:HISTogram:MEASurement <numeric_value>

Query Syntax: :ANALysis:TRace[12345678]:HISTogram:MEASurement?

Returned format: :ANALysis:TRace[12345678]:HISTogram:MEASurement <numeric_value>

Name: **HISTogram:MINreading** Command, Query

Function: This command sets the value of the measurement that corresponds to the left hand side of the screen.
This query returns the value of the measurement that corresponds to the left hand side of the screen.

Command Syntax: :ANALysis:TRace[12345678]:HISTogram:MINreading <numeric_value>

Example: :ANALysis:TRace4:HISTogram:MINreading 2

Query Syntax: :ANALysis:TRace[12345678]:HISTogram:MINreading?

Returned format: :ANALysis:TRace[12345678]:HISTogram:MINreading <numeric_value>

Name: **HISTogram:OVERflow** Query

Function: Returns the number of measurement results which overflowed the histogram range. If the horizontal range was automatic, the value returned should be zero.

Query Syntax: :ANALysis:TRace[12345678]:HISTogram:OVERflow?

Returned format: :ANALysis:TRace[12345678]:HISTogram:OVERflow <numeric_value>

Name: **HISTogram:UpdateRATE** Command, Query

Function: This command sets the update rate of the histogram measurement. The rate can either be the current acquisition rate or a number of seconds:
ACQ, 1, 2, 5, 10, 20, 50, 100
This query returns the currently set update rate.

Command Syntax: :ANALysis:TRace[12345678]:HISTogram:UpdateRATE <data>|<numeric_value>

Example: :ANALysis:TRace2:HISTogram:UpdateRATE ACQ

Query Syntax: :ANALysis:TRace[12345678]:HISTogram:UpdateRATE?

Returned format: :ANALysis:TRace[12345678]:HISTogram:UpdateRATE <data>|<numeric_value>

Name: **HISTogram:UNDERflow** Query

Function: Returns the number of measurement results which underflowed the histogram range. If the horizontal range was automatic, the value returned should be zero.

Query Syntax: :ANALysis:TRace[12345678]:HISTogram:UNDERflow?

Returned format: :ANALysis:TRace[12345678]:HISTogram:UNDERflow <numeric_value>

Name: **HISTogram:UPPERresult** Query

Function: Return the highest measurement result recorded in the overflow bin, or the maximum range of the histogram if the overflow bin is empty.

Query Syntax: :ANALysis:TRace[12345678]:HISTogram:UPPERresult?

Returned format: :ANALysis:TRace[12345678]:HISTogram:UPPERresult <numeric_value>

ANALysis Subsystem

ANALysis Subsystem

Name: HISTogram:YSCaling Query

Function: This command returns the vertical scaling which has been automatically applied to the histogram.

Query Syntax: :ANALysis:TRace[12345678]:HISTogram:YSCaling?

Returned format: :ANALysis:TRace[12345678]:HISTogram:YSCaling <numeric_value>

Name: INTegrateSCaling Command, query

Function: This command is used to set the scaling factor applied to a integrated trace when the INTegrate function is used.

The scaling factors for <data> are: x0.1,x0.2,x0.5 and x1.0.

This query returns the current scaling factor to be used with the INTegrate function.

Command Syntax: :ANALysis:TRace[12345678]:INTegrateSCaling <data>

Example: :ANALysis:TRace2:INTegrateSCaling x5.0

Query Syntax: :ANALysis:TRace[12345678]:INTegrateSCaling?

Returned format: :ANALysis:TRace[12345678]:INTegrateSCaling <data>

Name: MULtSCaling Command, query

Function: This command is used to set the scaling factor applied to a multiplied trace when the MULtiply function is used.

The scaling factors for <data> are: x0.1,x0.2,x0.5 and x1.0.

This query returns the current scaling factor to be used with the MULtiply function.

Command Syntax: :ANALysis:TRace[12345678]:MULtSCaling <data>

Example: :ANALysis:TRace4:MULtSCaling x5.0

Query Syntax: :ANALysis:TRace[12345678]:MULtSCaling?

Returned format: :ANALysis:TRace[12345678]:MULtSCaling <data>

Name: SORuce1 Command, Query

Function: This command defines the first source trace to use for trace arithmetic. The <trace_name> can be:

TRace1	TRace5	CH1
TRace2	TRace6	CH2
TRace3	TRace7	CH3
TRace4	TRace8	CH4

Command Syntax: :ANALysis:TRace[12345678]:SORuce1 <trace_name>

Example: :ANALysis:TRace2:SORuce1 TRace1

Query Syntax: :ANALysis:TRace[12345678]:SORuce1?

Returned format: :ANALysis:TRace[12345678]:SORuce1 <trace_name>

Name: **SOURCE2** Command, Query

Function: This command defines the second source trace to use for trace arithmetic. The <trace_name> can be:

TRace1	TRace5	CH1
TRace2	TRace6	CH2
TRace3	TRace7	CH3
TRace4	TRace8	CH4

Command Syntax: :ANALYSIS:TRace[12345678]:SOURCE2 <trace_name>

Example: :ANALYSIS:TRace2:SOURCE2 TRace2

Query Syntax: :ANALYSIS:TRace[12345678]:SOURCE2?

Returned format: :ANALYSIS:TRace[12345678]:SOURCE2 <trace_name>

Name: **TRace[12345678]:STOP** Command

Function: This command stops the live analysis (which can be maths, graphs or histograms) in progress for the given trace.

Command Syntax: :ANALYSIS:TRace[12345678]:STOP

Name: **STOP**

Type: Command

Function: This command stops all live analysis, graphs and histograms in progress.

Command Syntax: :ANALYSIS:STOP

Name: **TYPE** Query

Function: This command returns the type of the trace. The trace types are:

ACQuired
MATH
GRAPH
HISTogram

Query Syntax: :ANALYSIS:TRace[12345678]:TYPE?

Returned format: :ANALYSIS:TRace[12345678]:TYPE <data>

5 The CALibrate Subsystem is used for controlling the instrument's internal calibration routines.

COMMANDS

:CALibrate
 :ENable
 :FORce (command only)

Name:	ENable	Command, Query
Function:	This command is used to enable or disable all the instrument's automatic calibration routines. Numeric values may be sent, 0 is interpreted as OFF i.e calibrations disabled and any other number is interpreted as ON; calibrations enabled.	
	This query returns the state of the instrument's calibration routines, either enabled or disabled.	
Command Syntax:	:CALibrate:ENable <Boolean>	
Example:	:CALibrate:ENable ON	
Query syntax:	:CALibrate:ENable?	
Returned Format:	:CALibrate:ENable <Boolean>	

Name:	FORce	Command
Function:	This command forces a calibration to be carried out. The calibration will be the next one that would have due, i.e. if this command is sent during the first 15 minutes after power up a warm start calibration will be performed. At other times a temperature calibration will happen.	
Command Syntax:	:CALibrate:FORce	
Example:	:CALibrate:FORce	

6 The CHANnel Subsystem is used for controlling all aspects of the input channel performance.

COMMANDS

:CHANnel[1234]
:COUPling
:HOLD
:OFFSet
:POSition
:PRObe
:RANGe
:TERM50
:UNcal
:VARiable

Name:	COUPling	Command, Query
Function:	This command is used to set a channels input coupling to either AC, DC or Ground, by sending either AC, DC or GND as <data> This query returns the current channel coupling.	
Command Syntax:	:CHANnel[1234]:COUPling <data>	
Example:	:CHANnel2:COUPling DC	
Query syntax:	:CHANnel[1234]:COUPling?	
Returned Format:	:CHANnel[1234]:COUPling <data>	
Name:	HOLD	Command, Query
Function:	This command is used to hold a channel display. Numeric values may be sent, 0 is interpreted as OFF and any other number is interpreted as ON. This query returns the current hold status of a channel.	
Command Syntax:	:CHANnel[1234]:HOLD <Boolean>	
Example:	:CHANnel3:HOLD ON	
Query syntax:	:CHANnel[1234]:HOLD?	
Returned Format:	:CHANnel[1234]:HOLD <Boolean>	
See also:	DISPlay:HOLDAll	

CHANnel Subsystem

CHANnel Subsystem

Name: **OFFSet** Command, Query

Function: This command is used to set a channel's vertical amplifier offset. The numeric data is sent in volts. The offset limits are dependant on the vertical range selected as shown below:

Vertical Range	Max. Offset
2 mV to 50 mV	±500 mV
100 mV to 500 mV	±5 V
1 V to 5 V	±50 V

This query returns the channel's offset in volts.

Command Syntax: :CHANnel[1234]:OFFSet <numeric_value>

Example: :CHANnel2:OFFSet 4.50

Query syntax: :CHANnel[1234]:OFFSet?

Returned Format: :CHANnel[1234]:OFFSet <numeric_value>

Name: **POSition** Command, Query

Function: This command is used to position a channel's displayed trace vertically on the screen. The numeric data is sent in screen divisions with -4 being the bottom of the screen, 0 the center and 4 the top. The maximum range is ±10 divisions.

This query returns the vertical position of a channel's trace display.

Command Syntax: :CHANnel[1234]:POSition <numeric_value>

Example: :CHANnel1:POSition 4.5

Query syntax: :CHANnel[1234]:POSition?

Returned Format: :CHANnel[1234]:POSition <numeric_value>

Name: **PROBe** Command, Query

Function: This command is used to set the probe ratio for a channel. Ratios that may be matched are ×1, ×10, ×100 and ×1000 sent as 1, 10, 100 and 1000 respectively.

This query returns the current probe ratio for a channel.

Command Syntax: :CHANnel[1234]:PROBe <numeric_value>

Example: :CHANnel4:PROBe 1000

Query syntax: :CHANnel[1234]:PROBe?

Returned Format: :CHANnel[1234]:PROBe <numeric_value>

CHANnel Subsystem

CHANnel Subsystem

Name:	RANGe	Command, Query
Function:	<p>This command is used to set the vertical scaling of a channel. The set of values are: 2E-3, 5E-3, 10E-3, 20E-3, 50E-3, 100E-3, 200E-3, 500E-3, 1, 2 and 5. Although the above values are the most obvious choices, numbers such as 5E-1 or 0.0005E3 are just as valid. Any other number may be sent and it will be rounded.</p> <p>This query returns the current vertical scaling for a channel.</p>	
Command Syntax:	:CHANnel[1234]:RANGe <numeric_value>	
Example:	:CHANnel1:RANGe 50E-3	
Query syntax:	:CHANnel[1234]:RANGe?	
Returned Format:	:CHANnel[1234]:RANGe <numeric_value>	

Name:	TERM50	Command, Query
Function:	<p>This command is used to turn on or off the 50 Ω termination to a channel input. AC coupling is not allowed with 50 Ω termination. Numeric values may be sent, 0 is interpreted as OFF and any other number is interpreted as ON.</p> <p>CAUTION: While the instrument is set to 50 Ω termination, the maximum permissible channel input voltage is 5 V RMS.</p> <p>This query returns the current state of the 50 Ω termination to a channel.</p>	
Command Syntax:	:CHANnel[1234]:TERM50 <Boolean>	
Example:	:CHANnel2:TERM50 ON	
Query syntax:	:CHANnel[1234]:TERM50?	
Returned Format:	:CHANnel[1234]:TERM50 <Boolean>	

CHANnel Subsystem

CHANnel Subsystem

Name:	UNcal	Command, Query
Function:	<p>This command is used to turn the UNcal feature of a channel On and Off.</p> <p>Numeric values may be sent, 0 is interpreted as OFF and any other number is interpreted as ON.</p> <p>When UNcal is On, the gain of a channel is determined by the VARiable command.</p> <p>This query returns the channel's uncal status.</p>	
Command Syntax:	:CHANnel[1234]:UNcal <Boolean>	
Example:	:CHANnel3:UNcal ON	
Query syntax:	:CHANnel[1234]:UNcal?	
Returned Format:	:CHANnel[1234]:UNcal <Boolean>	

Name:	VARiable	Command, Query
Function:	<p>This command is used in conjunction with the UNcal command to apply a variable gain to a channel input. Once UNcal has been set to On, the channel gain is determined by the value sent by this command. The gain can be set from 20% to 100%.</p> <p>This query returns the current variable gain of a channel.</p>	
Command Syntax:	:CHANnel[1234]:VARiable <numeric_value>	
Example:	:CHANnel4:VARiable 25	
Query syntax:	:CHANnel[1234]:VARiable?	
Returned Format:	:CHANnel[1234]:VARiable <numeric_value>	

6a The CHANnelSCaling Subsystem Tcontains the commands for setting scaling factors to be applied to measurements.

```
:CHANnelSCaling
  :ENable
    :ENable
    :SCaling
    :OFFSet
    :UNITs
  :TIME
    :DatumtoCURsor
    :ENable
    :SCaling
    :UNITs
  :CHANnel[1234]
    :ENable
    :SCaling
    :OFFSet
    :UNITs
```

Name:	ENable
Type:	Command, query
Function:	This command is used to enable or disable user scaling for all traces. This query returns the current state of user scaling.
Command Syntax:	:CHANnelSCaling:ENable <Boolean>
Example:	:CHANnelSCaling:ENable ON
Query Syntax:	:CHANnelSCaling:ENable?
Returned format:	:CHANnelSCaling:ENable <Boolean>

Name:	CHANnel[1234]:ENable
Type:	Command, query
Function:	This command is used to enable or disable user scaling for a particular reference trace. This query returns the current state of user scaling for a particular reference trace.
Command Syntax:	:CHANnelSCaling:CHANnel[1234]:ENable <Boolean>
Example:	:CHANnelSCaling:CHANnel[1234]:ENable ON
Query Syntax:	:CHANnelSCaling:CHANnel[1234]:ENable?
Returned format:	:CHANnelSCaling:CHANnel[1234]:ENable <Boolean>

CHANnelSCaling Subsystem

CHANnelSCaling Subsystem

Name: CHANnel[1234]:OFFSet

Type: Command, query

Function: This command is used to set the voltage offset value for a particular reference trace. The range is $\pm 1\text{E}\pm 30$.
This query returns the current voltage offset value for a particular reference trace.

Command Syntax: :CHANnelSCaling:CHANnel[1234]:OFFSet <numeric_value>

Example: :CHANnelSCaling:CHANnel2:OFFSet 30

Query Syntax: :CHANnelSCaling:CHANnel[1234]:OFFSet?

Returned format: :CHANnelSCaling:CHANnel[1234]:OFFSet <numeric_value>

6

Name: CHANnel[1234]:SCaling

Type: Command, query

Function: This command is used to set the voltage scaling factor for a particular reference trace. The numeric range is $\pm 1\text{E}\pm 30$.
This query returns the current voltage scaling factor for a particular reference trace.

Command Syntax: :CHANnelSCaling:CHANnel[1234]:SCaling <numeric_value>

Example: :CHANnelSCaling:CHANnel2:SCaling 4E-3

Query Syntax: :CHANnelSCaling:CHANnel[1234]:SCaling?

Returned format: :CHANnelSCaling:CHANnel[1234]:SCaling <numeric_value>

Name: CHANnel[1234]:UNITs

Type: Command, query

Function: This command is used to set an four character string for the scaling units of a particular reference trace. The string should be in quotes.
This query returns the units string for a particular reference trace.

Command Syntax: :CHANnelSCaling:CHANnel[1234]:UNITs "<data>"

Example: :CHANnelSCaling:CHANnel4:UNITs "DEG"

Query Syntax: :CHANnelSCaling:CHANnel[1234]:UNITs?

Returned format: :CHANnelSCaling:CHANnel[1234]:UNITs "<data>"

Name: **TIME:DATumtoCURsor**

Type: Command, Query

Function: This command is used to set the datum to cursor value for time scaling. When this is set it forces the time scaling to be recalculated using this value as the difference between the position of the time datum and cursor.

This query returns the scaled number set for the time represented by the current difference between the vertical datum and the cursor

Command Syntax: :CHANnelSCaling:TIME:DATumtoCURsor <numeric_value>

Example: ::CHANnelSCaling:TIME:DATumtoCURsor 34

Query Syntax: ::CHANnelSCaling:TIME:DATumtoCURsor?

Returned format: ::CHANnelSCaling:TIME:DATumtoCURsor <numeric_value>

Name: **TIME:ENable**

Type: Command,query

Function: This command is used to enable or disable horizontal scaling.

This query returns the current state of the horizontal scaling function.

Command Syntax: :CHANnelSCaling:TIME :ENable <Boolean>

Example: :CHANnelSCaling:TIME :ENable ON

Query Syntax: :CHANnelSCaling:TIME :ENable?

Returned format: :CHANnelSCaling:TIME :ENable <Boolean>

CHANnelSCaling Subsystem

CHANnelSCaling Subsystem

Name:	TIME:SCaling
Type:	Command, query
Function:	This command is used to set the horizontal scaling factor. The numeric range is $\pm 1\text{E}\pm 30$. This query returns the current horizontal scaling factor.
Command Syntax:	:CHANnelSCaling:TIME:SCaling <numeric_value>
Example:	:CHANnelSCaling:TIME:SCaling 4E-3
Query Syntax:	:CHANnelSCaling:TIME:SCaling?
Returned format:	:CHANnelSCaling:TIME:SCaling <numeric_value>

Name:	TIME:UNITs
Type:	Command, query
Function:	This command is used to set an four character string for the horizontal scaling units. The string should be in quotes. This query returns the units string for the horizontal scaling.
Command Syntax:	:CHANnelSCaling:TIME:UNITs "<data>"
Example:	:CHANnelSCaling:TIME:UNITs "secs"
Query Syntax:	:CHANnelSCaling:TIME:UNITs?
Returned format:	:CHANnelSCaling:TIME:UNITs "<data>"

7 The Common commands are commands defined by the IEEE 488.2 standard. This section details the DSO's implementation of the relevant commands. **The ROOT commands** are various single level commands for controlling DSO functions.

COMMANDS

*CAL (query only)
 *CLS
 *ESE (not RS423)
 *ESR (query only) (not RS423)
 *IDN (query only)
 *OPC
 *OPT (query only)
 *RST
 *SRE (not RS423)
 *STB (query only) (not RS423)
 *TRG
 *TST (query only)
 *WAI
 :ALL?
 :AUTosetup (command only)
 :BandwidthLimit
 :BEEP (command only)
 :CANCEL (command only)
 :DATE
 :HELP?
 :KEYclick
 :LER?
 :LTER?
 :MENU
 :PLOT (command only)
 :SER?
 :TEXT
 :TIME

Name:	*CAL?	Query
Function:	This query forces the DSO to perform a self-test. The DSO actions the auto-calibration routine and returns a "0" if everything is correct, a "1" if the calibration is marginal or a "2" if the instrument has failed to calibrate.	
Query syntax:	:*CAL?	
Returned Format:	:*CAL <numeric_value>	

Name:	*CLS	Command
Function:	Clear Status. This command is used to clear the DSO's status registers. See figure 1.6.	
Command Syntax:	:*CLS	
Example:	:*CLS	

Common & Root Commands

Common & Root Commands

Name: *ESE Command, Query Not RS423

Function: Event Status Enable. This command is used to set the Standard Event Status Enable Register bits. See figure 1.6. The bits set by this command enable the matching bits in the Standard Event Status Register.

This query returns the contents of the Standard Event Status Enable Register. See figure 1.6.

The numeric value for both the command and query is in the range 0–255.

Command Syntax: :*ESE <numeric_value>

Example: :*ESE 189

Query syntax: :*ESE?

Returned Format: :*ESE <numeric_value>

Name: *ESR? Query Not RS423

Function: Event Status Register. This query returns the contents of the Standard Event Status Register. See figure 1.6.

The numeric value returned is in the range 0–255.

Performing this query clears the corresponding register.

Query syntax: :*ESR?

Returned Format: :*ESR <numeric_value>

Name: *IDN? Query

Function: This query returns the instrument type, serial number and software version in that order.

The <numeric_value> returned for the instrument types is 6000.

The serial number is an eight digit number.

The software version <data> field consists of the software version fitted to the instrument and the IVN (Internal Version Number) of the software which changes each time the software is updated. The first four characters are the version number and the last four characters are the IVN so in the example below, the software is version 1.00 and the IVN is 1407

Query syntax: :*IDN?

Returned Format: :*IDN GOULD, <String>, <numeric_value>,<data>

Example: :*IDN GOULD, 6000,12345678,1.001407

Name: *OPC Command, Query

Function: Operation Complete This command causes the DSO to set the operation complete bit in the Standard Event Status Register when all pending device operations are complete.

This query returns a “1” when when all pending selected device operations are complete.

Command Syntax: :*OPC

Example: :*OPC

Query syntax: :*OPC?

Returned Format: :<numeric_value>

Name:	*OPT?	Query
Function:	This query is used to find out which options are fitted to, and active on, the DSO. It returns a comma-separated sequence of one-character fields each indicating the availability of one option. ASCII '0' indicates unavailability, '1' availability. The options are reported in the following sequence: Thermal plotter, Memory Card, Hard disk, Floppy disk, RAM disk, GPIB, XY cursor measurements, Analysis, Sequences, Measurement scaling, Custom measurements, trigger tools enhanced, Trigger tools standard	
Query syntax:	:*OPT?	
Returned Format:	:*OPT? <numeric_value>.....<numeric_value> (13 fields total)	

Name:	*RST	Command
Function:	This command resets the instrument to its default state. The GPIB or RS423 interface setup will not change nor will the instrument's calibration settings. The table below shows the DSO's default state.	

Acquisition		Vertical	
Acquisition mode	Continuous Refresh	Coupling	Gnd
Maxmin	Off	Uncal	Off
Bandwidthlimit	Off	Y shift	Centered
Storelength	0.5 k	Probe Gain	×1
Display Modes		Termination	1 MΩ
Menu	Trace Display	AddOff	
Mode	Main only	Invert	Off
Zoom	×1	Range	5 V/div
XY	Off	Channel selection	All On
Traces	TR1M,TR2M, TR3M, TR4M	Trigger	
Hold	Off	Control	A Trigger
Dot join	On	Slope	Positive
X shift	Centered	Source	Channel 1
Persistence dot join	Off	Coupling	AC
Persistence	Off	Level	Centered
Persistence type	Time	Pre/post	Pretrigger
Persistence time	5 secs	Pretrigger	0%
Persistence cycles	50	Delay	0s
Averaging	Off	TVtrigger	Off
Averaging factor	2	TVsync	Negative
Limits test	Off	A Gate B	Off
Colors		Autotrigger	On
All	General purpose default	Cursor	
Intensity		Cursor trace	None
Trace	10	Cursor X position	1 div from right
Bright up	8	Time Datum X position	1 div from left
Alpha/cursor	10	Voltage Datum Y position	Centered
Graticule	3	Mode	Absolute
XY/pers	10	Pulsewidth points	50%
Horizontal		Risetime	10/90%
Range	50 μs/div	Calculations	Volt/time, none,none
Uncal	Off		
Sweeps	Main		
Zoom	Off		

Common & Root Commands

Common & Root Commands

Analysis

Trace Calculate	OFF
Source 1	TRC1M
Operator	Add
Source 2	TRC1M
Scaling factors	×1
Filter factor	1
FFT Window	Hanning
FFT Horizontal scale	Linear
FFT Vertical Scale	Linear

Graphs and Histograms

Minimum reading	0
Maximum reading	10.0
Update rate	1 s
Measurement	1 (Volts)

Sequences

Soft keys	none
Run Cycles	1
Default Names	SEQ1....SEQ8
Sequence content	All empty

Save/Recall

Device	RAM disk
User	Default
Run Name	Unnamed
Run Number	000
Setup file name	Unnamed
Sequence file name	Unnamed

Command Syntax: :*RST**Example:** :*RST**Measurement Scaling**

measurement Scaling	Off
Traces	Off
Units	Volt
Vertical Scaling	1.0
Vertical Offset	0.0
Time entry mode	Direct
Datum to cursor	1.0
Horizontal scaling	1.0

Plot

Plotter destination	Internal thermal plotter (if fitted) external plotter otherwise
Labels, date, cursor	On
Graticule	Solid
Mode	Single
Persistence type	Solid
Limits type	Line

I/O

Baud	9600
BPC	8
Parity	None
Stop Bits	1
Handshake	XONXOFF
GPIB Address	7

Misc

Autocals	Enabled
Plot/Save Key	Plot
Keyclick	On
Menu transparency	Off
Date format	MM-DD-YY

Name: *SRE

Command, Query Not RS423

Function: Service Request Enable Register. This command is used to set the Service Request Enable Register bits. See figure 1.6. The bits set by this command enable the matching bits in the Status Byte Register.

This query returns the contents of the Service Request Enable Register. See figure 1.6.

The numeric value for both the command and query is in the range 0–255.

Command Syntax: :*SRE <numeric_value>**Example:** :*SRE 183**Query syntax:** :*SRE?**Returned Format:** :*SRE <numeric_value>

Name: ***STB?** Query Not RS423

Function: This query returns the contents of the Status Byte Register. See figure 1.6.
 The numeric value returned is in the range 0–255.
 The MSS (Master Summary Status) bit reported on bit 6 of this register indicates whether or not the DSO had at least one reason for requesting service.
 To read the instrument's status byte with RQS on bit 6, use the serial poll facility.

Query syntax: **:*STB?**

Returned Format: **:*STB <numeric_value>**

7

Name: ***TRG** Command

Function: This command has the same function as the ACQ:RUN SINGLE command, or the front panel single shot key.

Command Syntax: **:*TRG**

Example: **:*TRG**

Name: ***TST?** Query

Function: This query forces the DSO to perform a self-test. The DSO actions the auto-calibration routine and returns a "0" if everything is correct, a "1" if the calibration is marginal or a "2" if the instrument has failed to calibrate.

Query syntax: **:*TST?**

Returned Format: **:*TST <numeric_value>**

Name: ***WAI** Command

Function: This command does not set a bit or return a value, it instructs the DSO to wait until all pending operations are complete when it will respond to further commands. While it is waiting, the instrument will not accept any further commands.
 This command should be used whenever it is required to be sure that a previous command or event has happened before actioning another command. e.g. between performing a single shot acquisition on a timebase of 50 s/div and transferring the acquisition data, a *WAI command should be issued to ensure that the acquisition is complete.

Command Syntax: **:*WAI**

Example: **:*WAI**

Common & Root Commands

Common & Root Commands

Name:	ALL?	Query
Function:	<p>This query is used to determine the complete instrument status and setup by forcing the instrument to output all query responses to most commands that have both a command and query form.</p> <p>The returned data can be saved and sent to the DSO as a command to return the instrument to a particular status. This query has a similar function to the Setup commands but has the advantage that the data can be easily examined and understood as the answers are textual and are as described under the relevant queries.</p>	
Query syntax:	:ALL?	
Returned Format:	:	

Name:	AUTosetup	Command
Function:	This command has the same function as the front panel Auto Setup key.	
Command syntax:	:AUTosetup	
Example:	:AUTosetup	

Name:	BandwidthLimit	Command, Query
Function:	<p>This command is used to set the bandwidth of all the input channels to either 200 MHz, 100 MHz, 20 MHz, 1MHz or their full range by sending either TWOHUNDred, ONEHUNDred, TWEnty, ONE or NOne.</p> <p>Note that the values available for a prticular instrument vary according to the instrument type. See the specification in the operator manual for full details.</p> <p>This query returns the bandwidth limit state of the input channels.</p>	
Command syntax:	:BandwidthLimit <data>	
Example:	:BandwidthLimit TWEnty	
Query syntax:	:BandwidthLimit?	
Returned Format:	:BandwidthLimit <data>	

Name:	BEEP	Command
Function:	<p>This command is used to sound the instrument's internal beeper.</p> <p>The length of the beep can be varied from approximately 1 to 15 seconds by varying the numeric field from 1 to 15.</p>	
Command syntax:	:BEEP numeric_value>	
Example:	:BEEP 7	

Name:	CANCEL	Command
Function:	This command has the same function as the front panel Cancel key except that it does not return the instrument to local control. It is used to stop an AUTOS SETUP or a PLOT.	
Command syntax:	:CANCEL	
Example:	:CANCEL	

Name: **DATE** Command, Query

Function: This command is used to set the instrument's internal clock with the correct date. by sending the day of the month, the month and the year.

The month <data> names are: JANuary, FEBruary, MARch, APRil, MAY, JUNE, JULy, AUGust, SEPtember, OCTober, NOVember and DECember

This query is used to examine the current setting of the instrument's internal clock.

Command Syntax: :DATE <numeric_value>,<data>, <numeric_value>

Example: :DATE 14,JULy,1993

Query syntax: :DATE?

Returned Format: :DATE <numeric_value>,<data>, <numeric_value>

Notes: *The DSO can accept dates in either MM-DD-YY or DD-MM-YY format but will always output dates in the format set by the DATE:FORMAT command.*

Name: **HELP?** Query

Function: This query returns a list of all commands with all their possible arguments.

This will consist of about 4 k of alphanumeric data.

Query syntax: :HELP?

Returned Format: :HELP <list of all commands>

Name: **KEYclick** Command, Query

Function: This command is used to turn the keyclick feature off and on.

This query returns the current state of the keyclick function.

Command syntax: :KEYclick <Boolean>

Example: :KEYclick ON

Query syntax: :KEYclick?

Returned Format: :KEYclick <Boolean>

Name: **LER?** Query

Function: This query returns the Status of the Local Event Register (LER). The value returned will be either 1 or 0. See figure 1.6.

1 means that the instrument has been switched from remote to local operation since this query was last sent, i.e. the front panel abort key has been pressed.

0 means that the instrument has not been switched from remote to local operation since this command was last sent.

Sending this query will reset the LER to 0 once the value has been read.

Query syntax: :LER?

Returned Format: :LER 1|0

Common & Root Commands

Common & Root Commands

Name:	LTER?	Query
Function:	<p>This query returns the status of the Limit Test Event Register (LTER). See figure 1.6. The value returned will be either 1 or 0. 1 means that Limits testing has failed since this query was last sent. 0 means that Limits testing has passed.</p> <p>The LTER is reset to 0 once the value has been read by this query.</p>	
Query syntax:	:LTER?	
Returned Format:	:LTER 0 1	

Name:	MENU	Command, Query
Function:	<p>This command is used to change the DSO display to any of the menus by sending one of the menu titles listed below. The display can be returned to the normal trace display by sending OFF.</p> <p>This query returns the current state of the display and will be either a menu title or OFF.</p>	

Menu Titles

50OHM select	OPTions CONFIguration
A DELay N gates b	PERsistence
A DELay Time gates b	PLOT configuration
AVeraGing	PROBe Gain selection
BAND trigger	PULSE Width trigger
BANDWidth and calibration	RATE trigger
B GATes A delay by n	RS423 BAUD
CHANnel Master	RS423 PORT
CHANnel OFFSets	RUNT trigger
CHANnel SCaling	SAVe RECall Master
COMBI Test	SCAled Time MEASurements
COMBination trigger	SCAled Volts MEASurements
CURrent Status	SEQuence
DELay by N	SEQuence STatus
DESTinaTion tRACe	SET Time AND Date
DISPlay COLor	SKEW trigger
DISPlay Master	SLEW rate trigger
DIVide by N	SPECial
FFT Parameters	STanDard trigger
GPIB	TEST Limits
GRAPH Parameters	TRace FUNCTions
GRATicule style	TRace MAPping
HISTogram Parameters	TRace VIEW
I/O Master	TRIGger SETup
MEASurements Master	Trigger Tool Mode
MEMory Length	TV Trigger
MISSing trigger	PREFerences
OFF	UTILity master

Command Syntax:	:MENU <data>
Example:	:MENU CAPtureMaster
Query syntax:	:MENU?
Returned Format:	:MENU <data>

Name:	MODE	Command
Function:	This command has the same function as the front panel Refresh/Roll button.	
Command syntax:	:MODE REfresh ROll	
Example:	:MODE REfresh	
Name:	PLOT	Command
Function:	This command has the same function as the front panel Plot key. It will initiate a plot of the current screen display using the parameters set using the :PLOT commands.	
Command syntax:	:PLOT	
Example:	:PLOT	
Name:	SER?	Query
Function:	This query returns the status of the Stored Event Register (SER). See figure 1.6. The value returned will be either 1 or 0. 1 means that acquisition is complete and a trace is stored on the display and 0 means that no trace is currently stored on the display. Do not confuse this with traces stored in the trace memories.	
Query syntax:	:SER?	
Returned Format:	:SER 1 0	
Name:	TEXT	Command, Query
Function:	<p>This command allows the user to add annotation to the screen prior to plotting. Text can be added to any line of the display and will overwrite any existing text. Each line is 41 characters long.</p> <p>Line 1 is the top line and line 34 is the bottom line. Line 5 contains numeric key 2 information and line 29 contains numeric key 8 information.</p> <p>The TEXT* version of this command can be used to put the same text on every line. If "" is sent (2 quote marks), this clears all text from the display.</p> <p>To restore the normal text display use the MENU command.</p> <p>This query returns the text on the specified display line.</p>	
Command syntax:	:TEXT <numeric_value>, "<text>"	
Example:	:TEXT4," Text on line four"	
Query syntax:	:TEXT1?	
Returned Format:	:TEXT <numeric_value>,"{string}"	
Name:	TIME	Command, Query
Function:	<p>This command is used to set the instrument's internal clock with the correct time. The time is sent in the hours, minutes, seconds form.</p> <p>This query is used to examine the current setting of the instrument's internal clock.</p>	
Command Syntax:	:TIME <numeric_value>, <numeric_value>, <numeric_value>	
Example:	:TIME 16,45,00	
Query syntax:	:TIME?	
Returned Format:	:TIME <numeric_value>, <numeric_value>, <numeric_value>	

8 The CURSor Subsystem is used for controlling all aspects of the cursor movement and waveform measurements.

COMMANDS

:CURSor

:ScreenXpos

:ScreenYpos

:TRAcE

:Xpos

:Ypos

Name:	ScreenXpos	Query
Function:	This query returns the current horizontal position of the cursor on the screen. 0 is the left hand edge of the screen and 10 the right hand edge of the screen.	
	If the cursor is not on the screen a negative number is returned	
Query syntax:	:CURSor:ScreenXpos?	
Returned Format:	:CURSor:ScreenXpos <numeric_value>	

Name:	ScreenYpos	Query
Function:	This query returns the current vertical position of the cursor on the screen. The numeric data is with respect to the center of the display in screen divisions with 0 being the center and 4 the top of the display.	
Query syntax:	:CURSor:ScreenYpos?	
Returned Format:	:CURSor:ScreenYpos <numeric_value>	

Name:	TRAcE	Command, Query
Function:	This command is used to select the trace that the cursor and datum lines are applied to. The possible trace <data> names are:	
	TRAcE1M, TRAcE1Z, TRAcE2M, TRAcE2Z, TRAcE3M, TRAcE3Z, TRAcE4M, TRAcE4Z, TRAcE5M, TRAcE5Z, TRAcE6M, TRAcE6Z, TRAcE7M, TRAcE7Z, TRAcE8M and TRAcE8Z. OFF can also be sent to turn the cursor off.	
	This query returns the trace to which the cursor is applied.	
Command Syntax:	:CURSor:TRAcE <data> OFF	
Example:	:CURSor:TRAcE TRAcE2Z	
Query syntax:	:CURSor:TRAcE?	
Returned Format:	:CURSor:TRAcE <data>OFF	

CURSor Subsystem

CURSor Subsystem

Name:	Xpos	Command, Query
Function:	<p>This command sets the cursor position along the trace. The numeric value is in screen divisions from the left hand side of the trace. 0 is the left hand edge and 10 is the right hand edge. The position can be set to any sample point of the acquisition. Therefore the numeric value can be set in 0.0002 steps for a 50 k memory length, 0.002 steps for 5 k and 0.02 steps for 0.5 k.</p> <p>This query returns the current horizontal position of the cursor along the trace.</p>	
Command Syntax:	:CURSor:Xpos <numeric_value>	
Example:	:CURSor:Xpos 5.7	
Query syntax:	:CURSor:Xpos?	
Returned Format:	:CURSor:Xpos <numeric_value>	

8

Name:	Ypos	Query
Function:	<p>This query returns the current vertical position of the cursor. The numeric data is with respect to the center of the display in screen divisions with 0 being the center and 4 the top of the display.</p>	
Query syntax:	:CURSor:Ypos?	
Returned Format:	:CURSor:Ypos <numeric_value>	

9 The DATE Subsystem is used to set the format of the date.

COMMANDS

:DATE

:FORMAT

Name:	FORMAT	Command, Query
Function:	<p>This command sets the date format to be either month–day–year (MM–DD–YY) or day–month–year (DD–MM–YY). This format is then used on all screen displays and plot outputs. The chosen format is also used for all data transfers from the instrument.</p> <p>This query returns the current date format.</p>	
Command syntax:	:DATE:FORMAT <data>	
Example:	:DATE:FORMAT MM-DD-YY	
Query syntax:	:DATE:FORMAT?	
Returned Format:	:DATE:FORMAT <data>	
Notes:	<p>The DSO can always accept dates in either MM–DD–YY or DD–MM–YY format but will output dates in the format set by this command.</p>	

10 The DATum Subsystem is used for controlling and examining the time and voltage datum positions.

COMMANDS

:DATum

:ScreenXpos

:ScreenYpos

:Xpos

:Ypos

Name:	ScreenXpos	Query
Function:	This query returns the X position of the vertical datum line on the screen.	
Query syntax:	:DATum:ScreenXpos?	
Returned Format:	:DATum:Xpos <numeric_value>	

Name:	ScreenYpos	Query
Function:	This query returns the Y position of the horizontal datum line on the screen. 0 corresponds to the center of the screen and +4 the top edge of the screen.	
Query syntax:	:DATum:ScreenYpos?	
Returned Format:	:DATum:ScreenYpos <numeric_value>	

Name:	Xpos	Command, Query
Function:	<p>This command is used to position the vertical datum with respect to the left hand edge of the trace. The numeric data is in screen divisions with 0 being the left hand edge and 10 the right hand edge of the trace. The position can be set to any sample point of the acquisition. Therefore the numeric value can be set in 0.0002 steps for a 50 k memory length, 0.002 steps for 5 k and 0.02 steps for 0.5 k.</p> <p>This query returns the X position of the vertical datum line along the trace.</p>	
Command Syntax:	:DATum:Xpos <numeric_value>	
Example:	:DATum:Xpos 4.7	
Query syntax:	:DATum:Xpos?	
Returned Format:	:DATum:Xpos <numeric_value>	

Name:	Ypos	Command, Query
Function:	<p>This command is used to position the horizontal datum with respect to the center of the display. The numeric data is in screen divisions with 0 being the center and 4 the top of the display. The permissible range is ± 4.8 divisions.</p> <p>If the instrument is in ABS mode, the horizontal datum cannot be moved as it is fixed to the selected trace ground.</p> <p>This query returns the Y position of the horizontal datum line.</p>	
Command Syntax:	:DATum:Ypos <numeric_value>	
Example:	:DATum:Ypos 1.4	
Query syntax:	:DATum:Ypos?	
Returned Format:	:DATum:Ypos <numeric_value>	

11 The DISPlay Subsystem is used for controlling all aspects of the instrument's display performance. To set the brightness of the various display elements on a monochrome instrument, use the appropriate color command and set the levels according to the brightness required. If all three colors are set to 100% the display will be the brightest possible and if all three colors are set to 0% the display will be as dark as possible.

COMMANDS

```
:DISPlay
:ALLColors          (command only)
:BACKground
:COLor
:CURSors
:COLor
:GRATicule
:BORDER
:COLor
:MAJORaxes
:MINORaxes
:STate
:GRaticuleiNtersection
:COLor
:HOLDAll
:MainTRace[12345678]
:COLor
:MAXmin
:MEasTRansparency
:PERSDotjoin
:PERSISTence
:COLor
:STate
:PERSMODE
:PERSSWEeps
:PERSTime
:SWEep
:TEXT
:COLor
:TRace{ 1M,2M,3M,4M,5M,6M,7M,8M,1Z,2Z,3Z,4Z,5Z,6Z,7Z,8Z}
:STate
:TRACEDotjoin
:TRaceINtersection
:COLor
:TruTRACE
:XSHift
:XY
:ZOOM
:ZoomPOSition
:ZoomTRace[12345678]
:COLor
:YZoomFACtor
:YZoomPOSition
```

Name:	ALLColors	Command
Function:	This command is used to set the colors of the color display to one of the default sets. DEFAULT1 has a black background and DEFAULT2 has a white background.	
Command Syntax:	:DISPlay:ALLColors <data>	
Example:	:DISPlay:ALLColors DEFAULT1	

Name:	BACKground:COLor	Command, Query
Function:	This command is used to set the color of the background on a color display. The color is created by specifying the percentage of the red, green and blue components. Each component color can be set from 0% to 100% in 5% steps.	
	This query returns the current color of the background.	
Command Syntax:	:DISPlay:BACKground:COLor <numeric_value>, <numeric_value>, <numeric_value>	
Example:	:DISPlay:BACKground:COLor 40,50,20	
Query syntax:	:DISPlay:BACKground:COLor?	
Returned Format:	:DISPlay:BACKground:COLor <numeric_value>, <numeric_value>, <numeric_value>	

Name:	CURSors:COLor	Command, Query
Function:	This command is used to set the color of the cursors on a color display. The color is created by specifying the percentage of the red, green and blue components. Each component color can be set from 0% to 100% in 5% steps.	
	This query returns the current color of the background.	
Command Syntax:	:DISPlay:CURSors:COLor <numeric_value>, <numeric_value>, <numeric_value>	
Example:	:DISPlay:CURSors:COLor 40,50,20	
Query syntax:	:DISPlay:CURSors:COLor?	
Returned Format:	:DISPlay:CURSors:COLor <numeric_value>, <numeric_value>, <numeric_value>	

Name:	GRATicule:BORder	Command, Query
Function:	This command is used to set the border on the graticule on the display. The border can be either BROKEN or SOLID. The border can also be turned OFF	
	This query returns the state of the graticule border.	
Command Syntax:	:DISPlay:GRATicule:BORder <data>	
Example:	:DISPlay:GRATicule:BORder SOLID	
Query syntax:	:DISPlay:GRATicule:BORder?	
Returned Format:	:DISPlay:GRATicule:BORder <data>	

DISPlay Subsystem

DISPlay Subsystem

Name:	GRATicule:COLor	Command, Query
Function:	<p>This command is used to set the color of the graticule on a color display. The color is created by specifying the percentage of the red, green and blue components. Each component color can be set from 0% to 100% in 5% steps.</p> <p>This query returns the current color of the graticule.</p>	
Command Syntax:	:DISPlay:GRATicule:COLor <numeric_value>, <numeric_value>, <numeric_value>	
Example:	:DISPlay:GRATicule:COLor 40,50,20	
Query syntax:	:DISPlay:GRATicule:COLor?	
Returned Format:	:DISPlay:GRATicule:COLor <numeric_value>, <numeric_value>, <numeric_value>	

Name:	GRATicule:MAJorAxes	Command, Query
Function:	<p>This command is used to turn the major axes on the display ON or OFF. Numeric values may be sent, 0 is interpreted as OFF and any other number is interpreted as ON.</p> <p>This query returns the state of the major axes.</p>	
Command Syntax:	:DISPlay:GRATicule:MAJorAxes <Boolean>	
Example:	:DISPlay:GRATicule:MAJorAxes ON	
Query syntax:	:DISPlay:GRATicule:MAJorAxes?	
Returned Format:	:DISPlay:GRATicule:MAJorAxes <Boolean>	

Name:	GRATicule:MINorAxes	Command, Query
Function:	<p>This command is used to set the style of the minor axes of the graticule on the display. The minor axes can be either BROKEN or SOLID. The minor axes can also be turned OFF</p> <p>This query returns the state of the minor axes.</p>	
Command Syntax:	:DISPlay:GRATicule:MINorAxes <data>	
Example:	:DISPlay:GRATicule:MINorAxes BROKEN	
Query syntax:	:DISPlay:GRATicule:MINorAxes?	
Returned Format:	:DISPlay:GRATicule:MINorAxes <data>	

Name:	GRATicule:STAt e	Command, Query
Function:	<p>This command is used to turn the graticule On and Off. Numeric values may be sent, 0 is interpreted as OFF and any other number is interpreted as ON.</p> <p>This query returns the current state of the graticule.</p>	
Command Syntax:	:DISPlay:GRATicule:STAt e <Boolean>	
Example:	:DISPlay:GRATicule:STAt e ON	
Query syntax:	:DISPlay:GRATicule:STAt e?	
Returned Format:	:DISPlay:GRATicule:STAt e <Boolean>	

DISPlay Subsystem

DISPlay Subsystem

Name:	GRatIculeiNterSection:COLor	Command, Query
Function:	<p>This command is used to set the color of any trace/graticule intersections. The color is created by specifying the percentage of the red, green and blue components. Each component color can be set from 0% to 100% in 5% steps.</p> <p>This query returns the current color of the trace/graticule intersections.</p>	
Command Syntax:	:DISPlay: GRatIculeiNterSection:COLor <numeric_value>,<numeric_value>, <numeric_value>	
Example:	:DISPlay: GRatIculeiNterSection:COLor 40,50,20	
Query syntax:	:DISPlay:GRatIculeiNterSection:COLor?	
Returned Format:	:DISPlay: GRatIculeiNterSection:COLor <numeric_value>,<numeric_value>, <numeric_value>	

Name:	HOLDAll	Command, Query
Function:	<p>This command is used to turn the hold all function On and Off to freeze all channel displays in the same way as the front panel control Hold all control (19). Numeric values may be sent, 0 is interpreted as OFF and any other number is interpreted as ON.</p> <p>This query returns the current state of the Hold All function.</p>	
Command Syntax:	:DISPlay:HOLDAll <Boolean>	
Example:	:DISPlay:HOLDAll ON	
Query syntax:	:DISPlay:HOLDAll?	
Returned Format:	:DISPlay:HOLDAll <boolean>	

Name:	MainTRace[12345678]:COLor	Command, Query
Function:	<p>This command is used to set the color used for a trace on a color display. The color is created by specifying the percentage of the red, green and blue components. Each component color can be set from 0% to 100% in 5% steps.</p> <p>This query returns the current color used a trace.</p>	
Command Syntax:	:DISPlay:MainTRace[12345678]:COLor <numeric_value>, <numeric_value>, <numeric_value>	
Example:	:DISPlay:MainTRace7:COLor 40,50,20	
Query syntax:	:DISPlay:MainTRace[12345678]:COLor?	
Returned Format:	:DISPlay:MainTRace[12345678]:COLor <numeric_value>, <numeric_value>, <numeric_value>	

Name:	MAXmin	Command, Query
Function:	<p>This command controls the display Max/min function. Numeric values may be sent, 0 is interpreted as OFF and any other number is interpreted as ON.</p> <p>This query returns the current state of the Display Max/min function.</p>	
Command Syntax:	:DISPlay:MAXmin <Boolean>	
Example:	:DISPlay:MAXmin ON	
Query syntax:	:DISPlay:MAXmin?	
Returned Format:	:DISPlay:MAXmin <boolean>	

DISPlay Subsystem

DISPlay Subsystem

Name: **MEasTRansparency** Command, Query

Function: This command is used to make the background of the measurment box at the bottom of the display either solid or transparent..

This query returns the current state of the measurement transparency mode.

Command Syntax: :DISPlay:MEasTRansparency <boolean>

Example: :DISPlay:MEasTRansparency ON

Query syntax: :DISPlay:MEasTRansparency?

Returned Format: :DISPlay:MEasTRansparency <boolean>

Name: **PERSDotjoin**

Type: Command, Query

Function: This command is used to turn the persistence dot join function On and Off. Numeric values may be sent, 0 is interpreted as OFF and any other number is interpreted as ON.

This query returns the current state of the persistence dot join function.

Command Syntax: :DISPlay:PERSDotjoin <boolean>

Example: :DISPlay:PERSDotjoin OFF

Query syntax: :DISPlay:PERSDotjoin?

Returned Format: :DISPlay:PERSDotjoin <boolean>

Name: **PERSISTence:COLor** Command, Query

Function: This command is used to set the color of persisted traces. The color is created by specifying the percentage of the red, green and blue components. Each component color can be set from 0% to 100% in 5% steps.

This query returns the current color of persisted traces.

Command Syntax: :DISPlay:PERSISTence:COLor <numeric_value>,<numeric_value>, <numeric_value>

Example: :DISPlay:PERSISTence:COLor 40,50,20

Query syntax: :DISPlay:PERSISTence:COLor?

Returned Format: :DISPlay:PERSISTence:COLor <numeric_value>,<numeric_value>, <numeric_value>

Name: **PERSISTence:STate** Command, Query

Function: This command is used to turn the pesistence function On and Off. Numeric values may be sent, 0 is interpreted as OFF and any other number is interpreted as ON.

This query returns the current state of the persistence function.

Command Syntax: :DISPlay:PERSISTence:STate <Boolean>

Example: :DISPlay:PERSISTence:STate ON

Query syntax: :DISPlay:PERSISTence:STate?

Returned Format: :DISPlay:PERSISTence:STate <boolean>

DISPlay Subsystem

DISPlay Subsystem

Name:	PERSMode	Command, Query
Function:	<p>This command is used to set the persistence mode to either time or cycles, i.e to build up a composite display for either a set time or a set number of acquisitions. Values for <data> are TIME or SWEeps.</p> <p>This query returns the current state of the persistence mode.</p>	
Command Syntax:	:DISPlay:PERSMode <data>	
Example:	:DISPlay:PERSMode TIME	
Query syntax:	:DISPlay:PERSMode?	
Returned Format:	:DISPlay:PERSMode <data>	

Name:	PERSSWEeps	Command, Query
Function:	<p>This command is used to set the number of acquisitions that will be used to make up a persisted display. The valid numbers are 10, 20, 50, 100, 200, 500 and CONTInuous. Other numbers will be rounded.</p> <p>This query returns the number of acquisitions selected to form a persisted display.</p>	
Command Syntax:	:DISPlay:PERSSWEeps <numeric_value> CONTInuous	
Example:	:DISPlay:PERSSWEeps 20	
Query syntax:	:DISPlay:PERSSWEeps?	
Returned Format:	:DISPlay:PERSSWEeps <numeric_value> CONTInuous	

Name:	PERSTime	Command, Query
Function:	<p>This command is used to set the time between screen clears in persistence mode. Times are in seconds. Valid times are: 0.5, 2, 5, 15, 30, 90 and CONTInuous other values sent will be rounded.</p> <p>This query returns the time between screen clears in persistence mode.</p>	
Command Syntax:	:DISPlay:PERSTime <numeric_value> CONTInuous	
Example:	:DISPlay:PERSTime 30	
Query syntax:	:DISPlay:PERSTime?	
Returned Format:	:DISPlay:PERSTime <numeric_value> CONTInuous	

Name:	SWEep	Command, Query
Function:	<p>This command is used to define which traces are to be displayed. Either Main traces only, Zoom traces only or Main and Zoom traces can be selected. Values for <data> are Main, Zoom or MainandZoom.</p> <p>Remember that there are three traces per channel pair available so setting this command to Main and Zoom may result in some expected traces not being displayed.</p> <p>This query is used to determine the display sweep mode.</p>	
Command Syntax:	:DISPlay:SWEep <data>	
Example:	:DISPlay:SWEep Zoom	
Query syntax:	:DISPlay:SWEep?	
Returned Format:	:DISPlay:TRace <data>	

DISPlay Subsystem

DISPlay Subsystem

Name:	TEXT:COLor	Command, Query
Function:	<p>This command is used to set the color used for the text on a color display. The color is created by specifying the percentage of the red, green and blue components. Each component color can be set from 0% to 100% in 5% steps.</p> <p>This query returns the current color used for the text.</p>	
Command Syntax:	:DISPlay:TEXT:COLor <numeric_value>, <numeric_value>, <numeric_value>	
Example:	:DISPlay:TEXT:COLor 40,50,20	
Query syntax:	:DISPlay:TEXT:COLor?	
Returned Format:	:DISPlay:TEXT:COLor <numeric_value>, <numeric_value>, <numeric_value>	

Name:	TRace{1M,2M,3M,4M,5M,6M,7M,8M,1Z,2Z,3Z,4Z,5Z,6Z,7Z,8Z}:STate	Command, Query
Function:	<p>This command is used to turn on and off the main and zoom views for traces on the display. Numeric values may be sent, 0 is interpreted as OFF and any other number is interpreted as ON.</p> <p>This query is used to determine whether a particular main or zoom trace is turned on or off.</p>	
Command Syntax:	:DISPlay:TRace{ 1M,2M,3M,4M,5M,6M,7M,8M,1Z,2Z,3Z,4Z,5Z,6Z,7Z,8Z}:STate <data>	
Example:	:DISPlay:TRace3M:STate MainZoom	
Query syntax:	:DISPlay:TRace{ 1M,2M,3M,4M,5M,6M,7M,8M,1Z,2Z,3Z,4Z,5Z,6Z,7Z,8Z}:STate?	
Returned Format:	:DISPlay:TRace{ 1M,2M,3M,4M,5M,6M,7M,8M,1Z,2Z,3Z,4Z,5Z,6Z,7Z,8Z}:STate <data>	

Name:	TRACEDotjoin	Command, Query
Function:	<p>This command is used to turn the dot join function On and Off. Numeric values may be sent, 0 is interpreted as OFF and any other number is interpreted as ON.</p> <p>This query returns the current state of the dot join function.</p>	
Command Syntax:	:DISPlay:TRACEDotjoin <Boolean>	
Example:	:DISPlay:TRACEDotjoin ON	
Query syntax:	:DISPlay:TRACEDotjoin?	
Returned Format:	:DISPlay:TRACEDotjoin <Boolean>	

Name:	TRaceINTersection:COLor	Command, Query
Function:	<p>This command sets the color used for trace intersections on a color display. The color is created by specifying the percentage of the red, green and blue components. Each component color can be set from 0% to 100% in 5% steps.</p> <p>This query returns the current color used for trace intersections.</p>	
Command Syntax:	:DISPlay:TRaceINTersection:COLor <numeric_value>, <numeric_value>, <numeric_value>	
Example:	:DISPlay:TRaceINTersection:COLor 90,90,90	
Query syntax:	:DISPlay:TRaceINTersection:COLor?	
Returned Format:	:DISPlay:TRaceINTersection:COLor <numeric_value>, <numeric_value>, <numeric_value>	

DISPlay Subsystem

DISPlay Subsystem

Name: **TruTRACE** Command, Query

Function: This command is used to turn TruTrace On and Off. Numeric values may be sent, 0 is interpreted as OFF and any other number is interpreted as ON.

This query returns the current state of TruTrace.

Command Syntax: :DISPlay:TruTRACE <Boolean>

Example: :DISPlay:TruTRACE ON

Query syntax: :DISPlay:TruTRACE ?

Returned Format: :DISPlay:TruTRACE <boolean>

Name: **XSHift** Command, Query

Function: This command is used to shift all displayed traces horizontally with respect to the graticule. The numeric value is screen divisions with the permissible range of 0 to 10 divisions. 5 means that the trace is centered.

This query returns the amount of X shift currently applied to the trace display.

Command Syntax: :DISPlay:XSHift <numeric_value>

Example: :DISPlay:XSHift 3.2

Query syntax: :DISPlay:XSHift?

Returned Format: :DISPlay:XSHift <numeric_value>

Name: **XY** Command, Query

Function: This command is used to turn the XY display function Off and On. Numeric values may be sent, 0 is interpreted as OFF and any other number is interpreted as ON.

This query returns the current state of the XY display function

Command Syntax: :DISPlay:XY <Boolean>

Example: :DISPlay:XY ON

Query syntax: :DISPlay:XY?

Returned Format: :DISPlay:XY <Boolean>

DISPlay Subsystem

DISPlay Subsystem

Name:	ZOOM	Command, Query						
Function:	<p>This command is used to control the zoom factor of the screen display. It is used to set a factor or to turn zoom off. Allowable zoom factors depend upon the current memory length. Off and a factor of 1 mean the same thing.</p> <p>Memory Max Zoom</p> <table><tr><td>0.5 k</td><td>×10</td></tr><tr><td>5 k</td><td>×100</td></tr><tr><td>50 k</td><td>×1000</td></tr></table> <p>The complete range of values is: OFF 1, 2, 5, 10, 20, 50, 100, 200 and 500.</p> <p>This query returns the current state of the zoom function.</p>		0.5 k	×10	5 k	×100	50 k	×1000
0.5 k	×10							
5 k	×100							
50 k	×1000							
Command Syntax:	:DISPlay:ZOOM <numeric_value> OFF							
Example:	:DISPlay:ZOOM 5							
Query syntax:	:DISPlay:ZOOM?							
Returned Format:	:DISPlay:ZOOM <numeric_value> OFF							

Name:	ZoomPOSition	Command, Query
Function:	<p>This command is used to control the horizontal position of the center of the zoom window relative to the main trace sample points. The position can be set to any sample point of the acquisition. Therefore the numeric value can be set in 0.0002 steps for a 50 k memory length, 0.002 steps for 5 k and 0.02 steps for 0.5 k.</p> <p>The numeric value is screen divisions with the permissible range of 0 to 10 divisions. 5 means that the center of the zoom window corresponds to the center of the main trace.</p> <p>This query returns the current position of the center of the zoom window.</p>	
Command Syntax:	:DISPlay:ZoomPOSition <numeric_value>	
Example:	:DISPlay:ZoomPOSition 4.78	
Query syntax:	:DISPlay:ZoomPOSition?	
Returned Format:	:DISPlay:ZoomPOSition <numeric_value>	

Name:	ZoomTRace[12345678]:COLor	Command, Query
Function:	<p>This command sets the color used for zoomed traces. The color is created by specifying the percentage of the red, green and blue components. Each component color can be set from 0% to 100% in 5% steps.</p> <p>This query returns the current color used for zoomed traces.</p>	
Command Syntax:	:DISPlay:ZoomTRace[12345678]:COLor <numeric_value>, <numeric_value>, <numeric_value>	
Example:	:DISPlay:ZoomTRace4:COLor 90,90,90	
Query syntax:	:DISPlay:ZoomTRace[12345678]:COLor?	
Returned Format:	:DISPlay:ZoomTRace[12345678]:COLor <numeric_value>, <numeric_value>, <numeric_value>	

Name: **ZoomTRace[12345678]:YZoomFActor** Command, Query

Function: This command is used in conjunction with the TRace[12345678] command to vertically zoom a channel's trace about the center of the screen. Once TRace[12345678] has been set to Zoom or MainZoom, the amount of zoom is determined by the value sent by this command. The available zoom factors are x0.125, x0.25, x0.5, x1, x2, x4, x8, x16, x32.

This query returns the current Y zoom factor of a channel.

Command Syntax: :DISPlay:ZoomTRace[12345678]:YZoomFActor <numeric_value>

Example: :DISPlay:ZoomTRace4:YZoomFActor 25

Query syntax: :DISPlay:ZoomTRace[12345678]:YZoomFActor?

Returned Format: :DISPlay:ZoomTRace[12345678]:YZoomFActor <numeric_value>

Name: **ZoomTRace[12345678]:YZoomPOSition** Command, Query

Function: This command is used in conjunction with the YZoom command to vertically position a trace's Y zoomed trace. The position can be set to any sample level for the current resolution so <numeric_value> can be set on 0.001 division steps, with -4.266 being the bottom of the screen and 4.266 the top. 0 is the Center.

This query returns the vertical position of a traces's Y zoomed trace display.

Command Syntax: :DISPlay:ZoomTRace[12345678]:YZoomPOSition <numeric_value>

Example: :DISPlay:ZoomTRace7:YZoomPOSition 2.56

Query syntax: :DISPlay:ZoomTRace[12345678]:YZoomPOSition?

Returned Format: :DISPlay:ZoomTRace[12345678]:YZoomPOSition <numeric_value>

LIMits Subsystem

LIMits Subsystem

12 The LIMits Subsystem is used for controlling all aspects of the instrument's limits testing features. To display limit traces, ensure that trace 8 is turned on using the DISPLAY:TRace8 command

COMMANDS**:LIMits**

:BANDsize
 :LOWer (command only)
 :MODE
 :PlotonFAIL
 :SetBand (command only)
 :UPper (command only)

Name:	BANDsize	Command, Query
Function:	This command is used to set the band size in screen divisions around an existing trace that will form the limit traces..	
	This query returns the size of the limits band.	
Command Syntax:	:LIMits:BANDsize <numeric_data>	
Example:	:LIMits:BANDsize 2	
Query syntax:	:LIMits:BANDsize?	
Returned Format:	:LIMits:BANDsize <numeric_data>	

Name:	LOWer	Command
Function:	This command is used to define a lower limit for the limits detection routines. When this command is asserted, the current cursor trace becomes the lower limit.	
Command Syntax:	:LIMits:LOWer	
Example:	:LIMits:LOWer	

Name:	MODE	Command, Query
Function:	This command is used to set the limits testing mode to either OFF, ON or FAIL (stop-on-fail).	
	This query returns the current limits testing mode	
Command Syntax:	:LIMits:MODE <data>	
Example:	:LIMits:MODE FAIL	
Query syntax:	:LIMits:MODE?	
Returned Format:	:LIMits:MODE <data>	

LIMits Subsystem

LIMits Subsystem

Name:	PlotonFAIL	Command, Query
Function:	This command is used to set the plot-on-fail mode of the limits test function. This query returns the current mode of the limits test plot on fail function.	
Command Syntax:	:LIMits:PlotonFAIL <Boolean>	
Example:	:LIMits:PlotonFAIL ON	
Query syntax:	:LIMits:PlotonFAIL?	
Returned Format:	:LIMits:PlotonFAIL <Boolean>	

Name:	SetBand	Command
Function:	This command is used to set both the upper and lower limits using the a band around the current cursor trace using the band size set by the BANDsize command.	
Command Syntax:	:LIMits:SetBand	
Example:	:LIMits:SetBand	

Name:	UPper	Command
Function:	This command is used to define an upper limit for the limits detection routines. When this command is asserted, the current cursor trace becomes the upper limit.	
Command Syntax:	:LIMits:UPper	
Example:	:LIMits:UPper	

MassMEMory Subsystem

MassMEMory Subsystem

13 The MassMEMory subsystem contains all the command required to manage the memory devices. Before issuing an EXECute command, all the parameters should be set using the appropriate commands from the branch

COMMANDS

:MassMEMory

:COPY

:DESTination

:DEvice

:RuNnAMe

:RuNnUMber

:SEquence

:SETup

:TRACeNAME

:USER

:EXECute (assertive only)

:Source

:DEvice

:RuNnAMe

:RuNnUMber

:SEquence

:SETup

:TRACeNAME

:USER

:TYPE

:RECall

:EXECute (assertive only)

:RuNnUMber

:TRACe

:TRACeNAME

:TYPE

:SAVE

:EXECute (assertive only)

:RuNnUMber

:TYPE

:MassMEMory

:SAVeRECall

:DEvice

:RuNnAMe

:SEquence

:SEQueNcenUMber

:SETup

:USER

:UTILities

:CREate

:DEvice

:DELeTe (assertive only)

:FORMat (assertive only)

:LIST (query only)

:REName (assertive only)

:NEwNAME

:NEwNUMber

:RuNnAMe

:RuNnUMber

:SEquence

:SETup

:TRACeNAME

:USER

The :DEvice, :EXECute, :RuNnAMe, :RuNnUMber, :SEquence, :SEQueNcenUMber, :SETup, :TRACe, :TRACeNAME, :TYPE and :USER commands perform a similar function in all the branches in which they appear so they are only described once.

13

Name:	DEvice	Command, query
Function:	This command sets the mass memory storage device to be used. The devices for <string> are FloppyDISK and RamDISK.	
	This query returns the current memory storage device for the command branch in use.	
Command Syntax:	:MassMEMory:COPY:DESTination:DEvice<string>	
Example:	:MassMEMory:COPY:DESTination:DEvice RamDISK	
Query Syntax:	:MassMEMory:COPY:DESTination:DEvice?	
Returned Format:	:MassMEMory:COPY:DESTination:DEvice<string>	

MassMEMory Subsystem

MassMEMory Subsystem

Name:	RuNnAMe	Command, query
Function:	<p>This command sets the run name to be used by the branch function. The run name can be up to 8 characters long.</p> <p>This query returns the current run name of the file.</p>	
Command Syntax:	:MassMEMory:COPY:DESTination:RuNnAMe <string>	
Example:	:MassMEMory:COPY:DESTination:RuNnAMe PULSE	
Query Syntax:	:MassMEMory:COPY:DESTination:RuNnAMe?	
Returned Format:	:MassMEMory:COPY:DESTination:RuNnAMe <string>	

Name:	RuNnUMber	Command, query
Function:	<p>This command sets the run number to be used by the branch function. The run number can be from 0 to 999.</p> <p>This query returns the current run number of the file.</p>	
Command Syntax:	:MassMEMory:COPY:DESTination:RuNnUMber <numeric_data>	
Example:	:MassMEMory:COPY:DESTination:RuNnUMber 14	
Query Syntax:	:MassMEMory:COPY:DESTination:RuNnUMber?	
Returned Format:	:MassMEMory:COPY:DESTination:RuNnUMber <numeric_data>	

13

Name:	SEquence	Command, query
Function:	<p>This command sets the sequence name to be used by the branch function.</p> <p>The name can be up to 8 characters long.</p> <p>This query returns the current sequence name of the file.</p>	
Command Syntax:	:MassMEMory:COPY:DESTination:SEquence <string>	
Example:	:MassMEMory:COPY:DESTination:SEquence test1	
Query Syntax:	:MassMEMory:COPY:DESTination:SEquence?	
Returned Format:	:MassMEMory:COPY:DESTination:SEquence <string>	

Name:	SEQueNcenUMber	Command, query
Function:	<p>This command sets the sequence number to be used by the SAVeRECall command.</p> <p>The sequence number can be from 1 to 8.</p> <p>This query returns the current sequence number of the file.</p>	
Command Syntax:	:MassMEMory:COPY:DESTination:SEQueNcenUMber <numeric_data>	
Example:	:MassMEMory:COPY:DESTination:SEQueNcenUMber 7	
Query Syntax:	:MassMEMory:COPY:DESTination:SEQueNcenUMber?	
Returned Format:	:MassMEMory:COPY:DESTination:SEQueNcenUMber <numeric_data>	

MassMEMory Subsystem

MassMEMory Subsystem

Name:	SETup	Command, query
Function:	<p>This command sets the setup name to be used by the branch function.</p> <p>The name can be up to 8 characters long.</p> <p>This query returns the current setup name of the file.</p>	
Command Syntax:	:MassMEMory:COPY:DESTination:SETup <string>	
Example:	:MassMEMory:COPY:DESTination:SETup <string>	
Query Syntax:	:MassMEMory:COPY:DESTination:SETup?	
Returned Format:	:MassMEMory:COPY:DESTination:SETup <string>	

Name:	TRACe	Command, query
Function:	<p>This command sets the trace to be used by the RECall function. The possible values for <string> are TRace1, TRace2, TRace3, TRace4, TRace5, TRace6, TRace7, TRace8, and ALL.</p> <p>This query returns the current trace to be used by the RECall command.</p>	
Command Syntax:	:MassMEMory:RECall:TRACe <string>	
Example:	:MassMEMory:RECall:TRACe TRace4	
Query Syntax:	:MassMEMory:RECall:TRACe?	
Returned Format:	:MassMEMory:RECall:TRACe <string>	

Name:	TRACeNAME	Command, query
Function:	<p>This command sets the trace name to be used by the branch function.</p> <p>The name can be up to 8 characters long.</p> <p>This query returns the current trace name of the file.</p>	
Command Syntax:	:MassMEMory:COPY:DESTination:TRACeNAME <string>	
Example:	:MassMEMory:COPY:DESTination:TRACeNAME MYTEST	
Query Syntax:	:MassMEMory:COPY:DESTination:TRACeNAME?	
Returned Format:	:MassMEMory:COPY:DESTination:TRACeNAME <string>	

Name:	TYPE	Command, query
Function:	<p>This command sets the type of file copy, recall, or save to be executed. The possible values for <string> depend on the command branch in which TYPE is used.</p> <p>For RECall and SAVE, <string> can be TRACES, SETup or TRACesANDSETup.</p> <p>For COPY, <string> can be USER, SETup, RuNnAMe, RuNnUMber or TRACe.</p> <p>This query returns the type of file copy, recall or save to be executed.</p>	
Command Syntax:	:MassMEMory:COPY:TYPE <string>	
Example:	:MassMEMory:COPY:TYPE TRACe	
Query Syntax:	:MassMEMory:COPY:TYPE?	
Returned Format:	:MassMEMory:COPY:TYPE <string>	

MassMEMory Subsystem

MassMEMory Subsystem

Name:	USER	Command, query
Function:	<p>This command sets the user name to be used by the branch function.</p> <p>The name can be up to 8 characters long.</p> <p>This query returns the current user name of the file.</p>	
Command Syntax:	:MassMEMory:COPY:DESTination:USER <string>	
Example:	:MassMEMory:COPY:DESTination:USER <string>	
Query Syntax:	:MassMEMory:COPY:DESTination:USER?	
Returned Format:	::MassMEMory:COPY:DESTination:USER <string>	

Name:	EXECute	Command
Function:	This command causes a file to be copied recalled or saved using the previously set source and destination parameters.	
Command Syntax:	MassMEMory:COPY:EXECute	
Example:	MassMEMory:COPY:EXECute	

MaSSMEMory:UTILities Subsystem

Name:	CREate	Command
Function:	<p>This command is used to create a new user, a new run name or a new run number on a mass memory device. The actual names are set using the appropriate :UTILities command. The possible values for <string> are USER, RuNnAMe or RuNnUMber.</p> <p>This query returns the type of new object to be created.</p>	
Command Syntax:	:MassMEMory:UTILities:CREate<string>	
Example:	:MassMEMory:UTILities:CREate USER	
Query Syntax:	:MassMEMory:UTILities:CREate?	
Returned Format:	:MassMEMory:UTILities:CREate <string>	

Name:	DELeTe	Command
Function:	<p>This command is used to delete an object from a mass memory device. The actual names of the item to be deleted are set using the appropriate :UTILities command. The possible values for <string> are USER, SETup, TRACe, RuNnAMe, or RuNnUMber.</p> <p>This query returns the type of item to be deleted.</p>	
Command Syntax:	:MassMEMory:UTILities:DELeTe <string>	
Example:	:MassMEMory:UTILities:DELeTe TRACe	
Query Syntax:	:MassMEMory:UTILities:DELeTe?	
Returned Format:	:MassMEMory:UTILities:DELeTe <string>	

MassMEMory Subsystem

MassMEMory Subsystem

Name:	FORMat	Command
Function:	This command is used to format the selected mass memory device as set by the UTILities:DEvIce command.	
	CAUTION: Formatting a device causes all previously stored data to be lost.	
Command Syntax:	:MassMEMory:UTILities:FORMat	
Example:	:MassMEMory:UTILities:FORMat	

Name:	LIST	Query
Function:	This query returns a list of all the items of <type> on the selected memory device in the previously specified directories. The possible values for <type> are USER, SETup, RuNnAMe, RuNnUMber or TRACeNAMe.	
	LIST? USER will list the users on the device set by the :MassMEMory:UTILities :DEvIce command	
	LIST? RuNnAMe will list the run names on the device set by the :MassMEMory:UTILities :DEvIce command and the user set by the :MassMEMory :UTILities:USER command.	
	LIST? RuNnUMber will list the run numbers on the device set by :MassMEMory:UTILities :DEvIce, the user set by :MassMEMory:UTILities:USER and the run name set by the :MassMEMory:UTILities :RuNnAMe commands	
	LIST? SETup will list the setups on the device set by :MassMEMory:UTILities :DEvIce and the user set by :MassMEMory:UTILities:USER commands.	
Query Syntax:	:MassMEMory:UTILities:LIST <type>?	
Returned Format:	A list of files or directories on the mass storage device, separated by carriage returns.	

Name:	NEWNAME	Command, query
Function:	This command sets the name to which the selected file will be renamed by the REName command. The name can be up to 8 characters long.	
	This query returns the name to which the selected file will be renamed.	
Command Syntax:	:MassMEMory:UTILities:NEWNAME <string>	
Example:	:MassMEMory:UTILities:NEWNAME PETER	
Query Syntax:	:MassMEMory:UTILities:NEWNAME?	
Returned Format:	:MassMEMory:UTILities:NEWNAME <string>	

MassMEMory Subsystem

MassMEMory Subsystem

Name:	NEWNUMBER	Command, query
Function:	This command sets the number to which the selected file will be renumbered by the RENAME command. The number can be from 0 to 999	
	This query returns the number to which the selected file will be renumbered.	
Command Syntax:	:MassMEMory:UTILities:NEWNUMBER <numeric_data>	
Example:	:MassMEMory:UTILities:NEWNUMBER 417	
Query Syntax:	:MassMEMory:UTILities:NEWNUMBER <numeric_data>?	
Returned Format:	:MassMEMory:UTILities:NEWNUMBER <numeric_data>	

Name:	RENAME	Command
Function:	This command renames or renumbers a file using the previously set parameters.	
	The possible values for <type> are USER, SETUP, RUNNAME, RUNNUMBER or TRACEName.	
	For example, on the hard disk, to rename the user currently named as "JOHN" to "PETER" the command sequence might be:	
	:MassMEMory:UTILities:DEV HDISK	
	:MassMEMory:UTILities:USER "JOHN"	
	:MassMEMory:UTILities:NEWNAME "PETER"	
	:MassMEMory:UTILities:REN USER	
	This query returns the type of item that is to be renamed or renumbered.	
Command Syntax:	:MassMEMory:UTILities:RENAME<type>	
Example:	:MassMEMory:UTILities:RENAME USER	

14 The MEASurements Subsystem contains the remote commands for controlling all aspects of the measurements system.

COMMANDS

:MEASurements	:TimeDATumREFerence
:STATE	:TOPbase
:YT	:TRANSDIRection
:Function	:TRANSNUMber
:MEASurement	:UNITS
:OPERator	:WEIGHTing
:VALue	:MEASurements
:ANNotation	:XY
:BOUnds	:Function
:CLEARlist (assertive only)	:MEASurement
:CROSSing	:OPERator
:CROSSLEVel	:ANNotation
:DATNOiseFl	:CLEARlist (assertive only)
:DATUM	:DatumLOCKedtorelease
:DATumVALue	:DELeTe (assertive only)
:DELeTe (assertive only)	:DispAREAfll
:DISplay	:DISplay
:FACTOR	:FACTOR
:FIXedVALue	:FIXedVLue
:FUNDamental	:NAME
:HARMonic	:OFFset
:HIGHtrans	:REFerence
:KNEE	:RESet (assertive only)
:LEFTtrans	:RESult (interrogative only)
:LOWtrans	:RESultNUMeric (interrogative only)
:MARKers	:SCAling
:NAME	:SOURce1YT
:OFFset	:SOURce2YT
:REFerence	:STanDardreference
:ReSeT	:UNITS
:RESult (interrogative only)	:WEIGHTing
:RESultNUMeric (interrogative only)	
:RIGHTtrans	
:SCAling	
:SEGment	
:SOURce1YT	
:SOURce2YT	
:STanDardreference	

Command detail

In this section, M_number is the number of the measurement as shown on the instrument's display. Where a command has <boolean> specified as a parameter, OFF, ON or a numeric value may be sent. Zero is interpreted as off and any other number as on.

MEASurements Subsystem

MEASurements Subsystem

Name: **:STATE** command, query

This command turns all the measurements on or off.

Command Syntax: **:MEASurements:STATE <boolean>**

Name: **:MEASurement** command, query

This command is used to specify the measurement to be performed.

<string_data> is one of: CURsor, TOP, BASE, AMPlitude, PEAKtopeak, MAX, MIN, RISEtime, FALLtime, OVERshoot, PREshoot, PULSEwidth, FREQuency, PERIOD, DUTYcycle, RMS, ACRMS, AREA, TRIGger, RiseCROSSing, FallCROSSing KNEE, MEAN, COUNT, FFTHARMonic, DATUM, DIMensionedCONSTant or KNEE.

<numeric_data> refers to the trace number and is from 1 to 8.

Command Syntax: **:MEASurements:YT:Function:MEASurement <M_number>,<string_data>,<numric_data>**

Name: **:OPERator** command, query

This command is used to perform an operation on previously obtained measurements. This command requires two M_numbers, even if only one is required for instance the LOG function only requires one M_number. The second number is ignored when it is not needed.

<string_data> is the operation to be performed and is one of SUM, DELta, RATIo, MULtiply, AVerage, LOGarithm, AntiLOG or COSine.

Command Syntax: **:MEAS:YT:Function:OPERator <M_number>, <string_data>,<M_number>,<M_number>**

Name: **:VALue** command, query

constants take nothing

Command Syntax: **:MEASurements:YT:Function:VALue <M_number>, CONSTANTl**

Name: **:ANNotation** command, query

This command is used to add annotation to a particular measurement. <string_data> is an alphanumeric string which can be up to 8 characters long.

Command Syntax: **:MEASurements:YT:ANNotation <M_number>, <string_data>**

Name: **:BOUnds** command, query

This command specifies whether to use, either the portion of the trace between the cursors or, the entire trace or between two previous measurement points as the measurement data source.

The previous measurements are the two results from the closest measurement numbers which give a time reading.

Command Syntax: **:MEASurements:YT:BOUnds <M_number>, BETweenCURsors lSEGment PREVIOUSMEASurements**

MEASurements Subsystem

MEASurements Subsystem

Name: **:CLEARlist** command

This command is used to clear all the measurements from the measurement list and is the same as individually deleting each measurement.

Command Syntax: :MEASurements:YT:CLEARlist

Name: **:CROSSing** command, query

This command specifies for a particular measurement, what level to take as a crossing point. It can be either the voltage datum, the trace ground or a percentage of the Top and base measurement. The percentage of the top/base measurement is set with the CROSSLEVEL command.

Command Syntax: :MEASurements:YT:CROSSing <M_number>, DATum|GROund|PERCENTtopbase|PREVIOUSMEASurements

Name: **:CROSSLEVEL** command, query

This command specifies the percentage level that will be used as the crossing point when PERCENTtopbase has been selected in the CROSSing command. The level for <numeric_data> must be a positive integer from 1 and 99.

Command Syntax: :MEASurements:YT:CROSSLEVEL <M_number>, <numeric_data>

Name: **:DATNOiseFl** command, query

This command is used to control the state of the Datum noise floor for the FFT harmonic measurement.

Command Syntax: :MEASurements:YT:DATNOiseFl <M_number>, <Boolean>

Name: **:DATum** command, query

This command is used to select whether the datum measurement operates on the time datum or the voltage datum. <data> can be either TIME or VOLTage.

Command Syntax: :MEASurements:YT:DATum <M_number>, <data>

Name: **:DATumVALue** command, query

This command only operated when DATum is set to TIME and is used to select whether the datum measurement returns the time at the time datum or the voltage at the point where the trace crosses the time datum. <data> can be either TIME or VOLTage.

Command Syntax: :MEASurements:YT:DATumVAL <M_number>, <data>

Name: **:DELeTe** command

This command is used to delete a specified measurement from the measurement list.

Command Syntax: :MEASurements:YT:DELeTe <M_number>

MEASurements Subsystem

MEASurements Subsystem

Name: **:DISplay** command, query
 This command is used to turn the trace display of the specified measurement number on or off.

Command Syntax: :MEASurements:YT:DISplay <M_number>,<boolean>

Name: **:FACtor** command, query
 This command is used to set the scaling factor to be used for a particular measurement when Scaling is turned on by the SCAling command. <numeric_data> is a floating point number.

Command Syntax: :MEASurements:YT:FACtor <M_number>,<numeric_data>

Name: **:FIXedVALue** command, query
 This command is used to provide a fixed value floating point number for a measurement

Command Syntax: :MEASurements:YT:FIXedVALue <M_number>,<numeric_data>

Name: **:FUNDamental** command, query
 This command is used to select the fundamental for the FFT measurement. When set to DATum, the datum defines the frequency of the fundamental. When set to AUTO, the fundamental is automatically computed. When set to PREVIOUSMEASurement, the result of the latest measurement in the list that returns a frequency will be used as the fundamental. <data> is AUTO, PREVIOUSMEASurement or DATum.

Command Syntax: :MEASurements:YT:FUNDamental <M_number>,<data>

Name: **:HARMonic** command, query
 This command is used to set the harmonic number for the FFT harmonic measurement.

Command Syntax: :MEASurements:YT:HARMonic <M_number>,<numeric_data>

Name: **:HIGHtrans** command, query
 This command specifies the high percentage level to be used as a transition point for risetime and falltime and crossing measurements. The level must be a positive integer from 1 and 99 and must be greater than or equal to the LOWtrans level.

Command Syntax: :MEASurements:YT:HIGHtrans <M_number>,<numeric_data>

Name: **:KNEE** command, query
 This command is used to choose a knee measurement which can be either the TOP or BOTtom knee.

Command Syntax: :MEASurements:YT:KNEE<M_number>,<data>

MEASurements Subsystem

MEASurements Subsystem

Name: **:LEFTtrans** command, query

This command specifies the level of the left-hand side of a pulse that will be used as the measurement point in pulse calculations. The level for <numeric_data> must be a positive integer from 1 and 99.

Command Syntax: :MEASurements:YT:LEFTtrans <M_number>, <numeric_data>

Name: **:LOWtrans** command, query

This command specifies the low percentage level to be used as a transition point for risetime and falltime and crossing measurements. The level must be a positive integer between 1 and 99 and must be lower than or equal to the HIGHtrans level.

Command Syntax: :MEASurements:YT:LOWtrans <M_number>, <numeric_data>

Name: **:MARKers** command, query

This command is used to specify whether or not markers for a particular measurement will be displayed on the instrument's screen.

Command Syntax: :MEASurements:YT:MARKers <M_number>, <boolean>

Name: **:NAME** command, query

This command is used to attach a name of up to 8 characters to the specified measurement number.

Command Syntax: :NAME <M_number>, <string_data>

Name: **:OFFset** command, query

This command is used to set an offset to be applied to a particular measurement. <numeric_data> is a floating point number.

Command Syntax: :MEASurements:YT:OFFset <M_number>, <numeric_data>

Name: **:REFerence** command, query

This command specifies the method of determining the trace reference voltage either using the voltage DATum or the trace GRound.

Command Syntax: :MEASurements:YT:REFerence <M_number>, DATum|GRound|PREVIOUSMEASurements

Name: **:ReSeT** command

This command is used to reset the Average value to the current measurement value to allow a new average to be obtained.

Command Syntax: :MEASurements:YT:ReSeT<M_number>

MEASurements Subsystem

MEASurements Subsystem

Name:	:RESult	query
	This command returns a string which is the same as the screen data. For measurements that produce two lines of data on the screen, both lines are returned but as one long string.	
Command Syntax:	:MEASurements:YT:RESult? <M_number>	
Name:	:RESultNUMeric	query
	This command returns a numeric string containing the numbers from the screen display for the specified measurement number. For measurements that have two results only the most important number for that measurement is returned as listed below.	
	Cursor	Volts
	Crossing	Time
	Max	Volts
	Min	Volts
	Trigger	Time
	Knee	Time
Command Syntax:	:MEASurements:YT:RESultNUMeric? <M_number>	
Name:	:RIGHTtrans	command, query
	This command specifies the level of the right-hand side of a pulse that will be used as the measurement point in pulse calculations. The level for <numeric_data> must be a positive integer from 1 and 99.	
Command Syntax:	:MEASurements:YT <M_number>, <numeric_data>	
Name:	:SCAling	command, query
	This command is used to turn scaling on and off for a particular measurement. The scaling factor is set using the FACtor command.	
Command Syntax:	:MEASurements:YT:SCAling <M_number>, <boolean>	
Name:	:SEGment	command, query
	This command is used to set on which segment the specified measurement operates. This command is ignored if the instrument is not in segmented mode. This query always returns 1 when the instrument is in non-segmented mode.	
Command Syntax:	:MEASurements:YT:SEGment <M_number>, <numeric_data>	
Name:	:SOUrce1YT	command, query
	This command specifies which previous measurement to use as the first source for the next calculation, and can be either the time or the value part of the measurement. For instance, crossing returns both a time and a voltage value of the crossing point.	
Command Syntax:	:MEASurements:YT:SOUrce1YT <M_number>, TIME VALUE	

MEASurements Subsystem

MEASurements Subsystem

Name: **:SOURce2YT** command, query

This command specifies which previous measurement to use as the second source for the next calculation, and can be either the time or the value part of the measurement. For instance, crossing returns both a time and a voltage value of the crossing point.

Command Syntax: :MEASurements:YT:SOURce2YT <M_number>, TIME|VALUE

Name: **:STanDardreference** command, query

This command selects whether the cursor measurement works with respect to the vertical datum or with respect to ground

Command Syntax: :MEASurements:YT:STanDardreference <M_number>, DATum|GROund

Name: **:TimeDATumREference** command, query

This command selects whether the time measurement works with respect to the vertical datum or with respect to ground

Command Syntax: :MEASurements:YT:TimeDATumREference <M_number>, DATum|GROund

Name: **:TOPbase** command, query

This command specifies whether to use the cursor and datum position or statistical analysis (as specified in IEEE) to provide the Top and Base measurement points.

Command Syntax: :MEASurements:YT:TOPbase <M_number>, CURsors|STATistical|MAXMIN

Name: **:TRANSDIRection** command, query

This command specifies the transition direction to be used. The number must be an integer

Command Syntax: :MEASurements:YT:TRANSDIRection <M_number>, POSitive|NEGative

Name: **:TRANSNUMber** command, query

This command specifies the transition number to be used. The number must be an integer

Command Syntax: :MEASurements:YT:TRANSNUMber <M_number>, <numeric_data>

Name: **:UNITS** command, query

This command is used to specify the dimensions of the dimensioned constant measurement

Command Syntax: :MEASurements:YT:UNITS <M_number>, HORIZontal|VERTical

Name: **:WEIGHTing** command, query

This command is used to specify the weighting factor to be applied to an average measurement. The value of <numeric_data> should be an integer in the range 2 - 1024.

Command Syntax: :MEASurements:YT:WEIGHTing <M_number>, <numeric_data>

:XY commands

Name: **:MEASurement** command, query

This command is used to specify the measurement to be performed.

<string_data> is one of: CONSTant, XYCURsor, XYCURsorT, XYRADius, XYANGLE, XYAREA or XINTEGRate

Command Syntax: :MEASurements:XY:Function:MEASurement <M_number>, <string_data>

Name: **:OPERator** command, query

This command is used to perform an operation on previously obtained measurements. This command requires two M_numbers, even if only one is required for instance the LOG function only requires one M_number. The second number is ignored when it is not needed.

<string_data> is the operation to be performed and is one of SUM, DELta, RATio, MULtiply, AVeraGe, LOGarithm, AntiLOG or COSine.

Command Syntax: :MEAS:XY:Function:OPERator <M_number>, <string_data>,<M_number>,<M_number>

Name: **:ANNotation** command, query

This command is used to add annotation to a particular measurement. <string_data> is an alphanumeric string which can be up to 8 characters long.

Command Syntax: :MEASurements:XY:ANNotation <M_number>, <string data>

14

Name: **:CLEARlist** command

This command is used to clear all the measurements from the measurement list and is the same as individually deleting each measurement.

Command Syntax: :MEASurements:XY:CLEARlist

Name: **:DatumLOCKedtotrace** command, query

In Absolute mode, this command fixes the datum to 0V (off) or to the trigger point (on). When the cursor mode is Delta, this command will free the datums in both X and Y directions (off) or the datum will be locked to the selected Y trace (on)

Command Syntax: :MEASurements:XY:DatumLOCKedtotrace <M_number>, <boolean>

Name: **:DELeTe** command

This command is used to delete a specified measurement from the measurement list.

Command Syntax: :MEASurements:XY:DELeTe <M_number>

MEASurements Subsystem

MEASurements Subsystem

Name: **:DispAREAfill** command, query

This command turns the area fill display on and off.

Command Syntax: :MEASurements:XY:DispAREAfill <M_number>, <boolean>

Name: **:DISplay** command, query

This command is used to turn the trace display of the specified measurement number on or off.

Command Syntax: :MEASurements:XY:DISplay <M_number>, <boolean>

Name: **:FACTOR** command, query

This command is used to turn scaling on and off for a particular measurement. The scaling factor is set using the FACTor command.

Command Syntax: :MEASurements:XY:FACTOR <M_number>, <numeric_data>

Name: **:FIXedVALue** command, query

This command is used to provide a fixed value floating point number for a measurement

Command Syntax: :MEASurements:XT:FIXedVALue <M_number>, <numeric_data>

Name: **:NAME** command, query

This command is used to attach a name of up to 8 characters to the specified measurement number.

Command Syntax: :MEASurements:XY:NAME <M_number>, string data

Name: **:OFFset** command, query

This command is used to set an offset to be applied to a particular measurement. <numeric_data> is a floating point number

Command Syntax: :MEASurements:XY:OFFset <M_number>, <numeric_data>

Name: **:REFerence** command, query

This command specifies the method of determining the trace reference voltage either using the voltage DATum or the trace GRound.

Command Syntax: :MEASurements:XY:REFerence <M_number>, DATum|GRound|PREVIOUSMEASurements

Name: **:ReSeT** command

This command is used to reset the Average value to the current measurement value to allow a new average to be obtained.

Command Syntax: :MEASurements:XY:ReSeT <M_number>

MEASurements Subsystem

MEASurements Subsystem

Name:	:RESult	query
	This command returns a string which is the same as the screen data. For measurements that produce two lines of data on the screen, both lines are returned but as one long string.	
Command Syntax:	:MEASurements:XY:RESult? <M_number>	
Name:	:RESultNUMeric	query
	This command returns a numeric string containing the numbers from the screen display for the specified measurement number. For measurements that have two results only the most important number for that measurement is returned.	
Command Syntax:	:MEASurements:XY:RESultNUMeric? <M_number>	
Name:	:SCAling	command, query
	This command is used to turn scaling on and off for a particular measurement. The scaling factor is set using the FACTor command.	
Command Syntax:	:MEASurements:XY:SCAling <M_number>, <boolean>	
Name:	:SOURce1YT	command, query
	This command specifies which previous measurement to use as the first source for the next calculation, and can be either the time or the value part of the measurement. For instance, CURSor returns both a time and a voltage value of the cursor point.	
Command Syntax:	:MEASurements:XY:SOURce1YT <M_number>, TIME VALUE	
Name:	:SOURce2YT	command, query
	This command specifies which previous measurement to use as the second source for the next calculation, and can be either the time or the value part of the measurement. For instance, CURSor returns both a time and a voltage value of the cursor point.	
Command Syntax:	:MEASurements:XY:SOURce2YT <M_number>, TIME VALUE	
Name:	:STanDardreference	command, query
	This command selects whether the cursor measurement works with respect to the vertical datum or with respect to ground	
Command Syntax:	:MEASurements:XY:STanDardreference <M_number>, DATum GROund	
Name:	:UNITS	command, query
	This command is used to specify the dimensions of the dimensioned constant measurement	
Command Syntax:	:MEASurements:XY:UNITS <M_number>, HORIZontal VERTical	
Name:	:WEIGHTing	command, query
	This command is used to specify the weighting factor to be applied to an average measurement. The value of <numeric_data> should be an integer in the range 2 - 1024.	
Command Syntax:	:MEASurements:XY:WEIGHTing <M_number>, <numeric_data>	

15 The MEMory Subsystem is used for controlling the instrument's internal trace memories.

COMMANDS

:MEMory

:CLear

:RECall

:SAVE

Name:	CLear	Command
--------------	--------------	----------------

Function:	This command is used to clear a specified memory.
------------------	---

Command Syntax:	:MEMory:CLear <numeric_data>
------------------------	------------------------------

Example:	:MEMory:CLear 4
-----------------	-----------------

Name:	RECall	Command
--------------	---------------	----------------

Function:	This command is used to recall a specified memory to a particular trace. Note that if the specified trace is not held, a recalled trace could be overwritten almost immediately by the next acquisition.
------------------	--

<Trace_name> is one of TRace1, TRace2, TRace3, TRace4, TRace5, TRace6, TRace7 or TRace8.

Command Syntax:	:MEMory:REcall <numeric_data>, <Trace_name>
------------------------	---

Example:	:MEMory:REcall 2, TRace7
-----------------	--------------------------

Name:	SAVE	Command
--------------	-------------	----------------

Function:	This command is used to save a particular trace to a specified memory.
------------------	--

<Trace_name> is one of TRace1, TRace2, TRace3, TRace4, TRace5, TRace6, TRace7 or TRace8.

Command Syntax:	:MEMory:SAVE <Trace_name>, <numeric_data>
------------------------	---

Example:	:MEMory:SAVE TRace4, 3
-----------------	------------------------

MENU Subsystem

16 The MENU Subsystem is used for controlling the menu transparency.

COMMANDS

:MENU
:TRANSparency

Name:	TRANSparency	Command, query
Function:	This command is used to set whether or not menus are transparent. ON means that menus are transparent.	
Command Syntax:	:MENU:TRANSparency <Boolean>	
Example:	:MENU:TRANSparency ON	
Query Syntax:	:MENU:TRANSparency?	
Example:	:MENU:TRANSparency <Boolean>	

17 The PLOT Subsystem is used for controlling all aspects of the instrument's plot performance.

COMMANDS

:PLOT

:DEst
:DRiVer
:EXTeRnaldest
:GRaticule
:LABELsdatecursor
:LENGth
:LIMitstype
:PERStype

Name:	DEst	Command, Query
Function:	This command sets the destination for plots. The internal option is only valid if an internal thermal plotter is fitted. If EXTeRnal is selected, the plot will be sent via the current plot port which may be either RS423 or GPIB, whichever one is not the current control port. Values for <data> are: INTeRnal or EXTeRnal.	
	This query returns the current plot destination.	
Command Syntax:	:PLOT:DEst <data>	
Example:	:PLOT:DEst INTeRnal	
Query syntax:	:PLOT:DEst?	
Returned Format:	:PLOT:DEst <data>	

Name:	DRiVer	Command, Query
Function:	This command sets the driver for external plots. Values for <data> are: HPGJ, DeskJetMONO or DeskJetCOLOR	
	This query returns the current plot driver.	
Command Syntax:	:PLOT:DRiVer <data>	
Example:	:PLOT:DRiVer DeskJetCOLOR	
Query syntax:	:PLOT:DRiVer?	
Returned Format:	:PLOT:DRiVer <data>	

Name:	EXTeRnaldest	Command, Query
Function:	This command sets the output port to be used for external plots. EXTeRnal should be selected as the plot destination using the DEst command. Values for <data> are: GPIB, RS423 or PARALLEL.	
	This query returns the current external plot port.	
Command Syntax:	:PLOT:EXTeRnaldest <data>	
Example:	:PLOT:EXTeRnaldest PARALLEL	
Query syntax:	:PLOT:EXTeRnaldest ?	
Returned Format:	:PLOT:EXTeRnaldest <data>	

PLOT Subsystem

PLOT Subsystem

Name: **GRaticule** Command, Query

Function: This command determines whether or not the graticule will be plotted. It can be either ON or OFF. Numeric values may be sent, 0 is interpreted as OFF and any other number is interpreted as ON.

This query returns the current state of the plot graticule function

Command Syntax: :PLOT:GRaticule <boolean>

Example: :PLOT:GRaticule ON

Query syntax: :PLOT:GRaticule?

Returned Format: :PLOT:GRaticule <boolean>

Name: **LABELsdatecursor** Command, Query

Function: This command is used to turn trace labels, date stamps and cursors on plots on or off. Numeric values may be sent, 0 is interpreted as OFF and any other number is interpreted as ON.

This query returns the state of the plot labels, date and cursor function.

Command Syntax: :PLOT:LABELsdatecursor <Boolean>

Example: :PLOT:LABELsdatecursor ON

Query syntax: :PLOT:LABELsdatecursor?

Returned Format: :PLOT:LABELsdatecursor <Boolean>

Name: **LENgth** Command, Query

Function: This command is only valid when an internal thermal plotter is fitted to the instrument and is used to define the length of a streaming plot in screens. It can be either 1, 2, 5, 10, 20, 50 or 100.

This query returns the length of a streaming plot.

Command Syntax: :PLOT:LENgth <data>

Example: :PLOT:LENgth 10

Query syntax: :PLOT:LENgth?

Returned Format: :PLOT:LENgth <data>

Name: **LIMitstype** Command, Query

Function: This command is used to define the way in which limit traces will be plotted.

They can be omitted from plots, plotted as trace lines or as a shaded area bounded by the trace lines by sending OFF, LINE or FILLED.

This query returns the LIMitstype function.

Command Syntax: :PLOT:LIMitstype <data>

Example: :PLOT:LIMitstype LINE

Query syntax: :PLOT:LIMitstype?

Returned Format: :PLOT:LIMitstype <data>

PLOT Subsystem

PLOT Subsystem

Name:	PERStyle	Command, Query
Function:	This command sets the style of persisted trace plots. The trace can either be plotted as broken or solid lines or not plotted at all by sending BROKEN, SOLID or OFF. This query returns the plot style of persisted traces.	
Command Syntax:	:PLOT:PERStyle <data>	
Example:	:PLOT:PERStyle BROKEN	
Query syntax:	:PLOT:PERStyle?	
Returned Format:	:PLOT:PERStyle <data>	

17a The PRINT Subsystem is used for controlling all aspects of the instrument’s printing performance.

COMMANDS

:PRINT
:DENSity

Name:	DENSity	Command, Query
Function:	This command sets the desity of solid areas on the internal thermal printer. Thedensity can be either BLack or GRay. Printing large areas of black may cause the print head to overheat and for the thermal trip to operate. Further printing will then have to wait until the print head has cooled sufficiently. This query returns the current print density setting.	
Command Syntax:	:PRINT:DENSity GRay BLack	
Example:	:PRINT:DENSity GRay	
Query syntax:	:PRINT:DENSity?	
Returned Format:	:PRINT:DENSity GRay BLack	

18 The RS423 Subsystem is used for controlling the parameters of the instrument's RS423 port.

COMMANDS

:RS423

:BAUD
:BPC
:ECHOandprompt
:HAnd
:PARity
:STopBits

Name:	BAUD	Command, Query
Function:	This command is used to set the baud rate for the RS423 port. Valid baud rates are 110, 300, 600, 1200, 2400, 4800, 9600, 19200 and 38400. Any other vales sent will be rounded.	
	This query returns the current baud rate for the RS423 port.	
Command Syntax:	:RS423:BAUD <numeric_value>	
Example:	:RS423:BAUD 9600	
Query syntax:	:RS423:BAUD?	
Returned Format:	:RS423:BAUD <numeric_value>	

Name:	BPC	Command, Query
Function:	This command is used to set the number of data bits per byte sent for RS423 plots. Valid numbers are 7 or 8. Other values will be rounded.	
	This query returns the number of data bits per byte sent for RS423 plots.	
Command Syntax:	:RS423:BPC <numeric_value>	
Example:	:RS423:BPC 7	
Query syntax:	:RS423:BPC?	
Returned Format:	:RS423:BPC <numeric_value>	

Name:	ECHOandprompt	Command, Query
Function:	This command is used to turn the echo and prompt feature on and off. Numeric values may be sent, 0 is interpreted as OFF and any other number is interpreted as ON.	
	This query returns the current state of the echo and prompt feature.	
Command Syntax:	:RS423:ECHOandprompt <Boolean>	
Example:	:RS423:ECHOandprompt ON	
Query syntax:	:RS423:ECHOandprompt?	
Returned Format:	:RS423:ECHOandprompt <Boolean>	

RS423 Subsystem

RS423 Subsystem

Name: **HAnd** Command, Query

Function: This command is used to set the handshaking used by the RS423 port. Handshaking can be set to None, XONxoff (software) or RTScts (hardware).
This query returns the handshaking arrangement to be used by the RS423 interface.

Command Syntax: :RS423:HAnd <data>

Example: :RS423:HAnd RTScts

Query syntax: :RS423:HAnd?

Returned Format: :RS423:HAnd <data>

Name: **PARity** Command, Query

Function: This command is used to set the parity used by the RS423 interface. Parity can be set to None, Even, Odd, Mark or Space.
This query returns the parity setting to be used by thr RS423 interface.

Command Syntax: :RS423:PARity <data>

Example: :RS423:PARity None

Query syntax: :RS423:PARity?

Returned Format: :RS423:PARity <data>

Name: **STopBits** Command, Query

Function: This command is used to set the number of stop bits per byte for RS423 communications. Either 1 or 2 stop bits can be sent. Other values will be rounded.
This query returns the number of stop bits to be used for RS423 communications.

Command Syntax: :RS423:STopBits <numeric_value>

Example: :RS423:STopBits 2

Query syntax: :RS423:STopBits?

Returned Format: :RS423:STopBits <numeric_value>

19 The SEquence Subsystem contains all the commands relating to sequences.

COMMANDS

:SEquence
 :CLEAR[12345678] (command only)
 :CONTinue (command only)
 :CYCles[12345678]
 :NAME
 :RUN[12345678] (command only)
 :SOFTkey

Name: **CLEAR**
Type: Command
Function: This command clears the selected sequence.

Command Syntax: :SEquence:CLEAR[12345678]
Example: :SEquence:CLEAR 3

Name: **CONTinue**
Type: Command
Function: This command can be used to enable a sequence to continue running after it has executed the WAIT instruction.

Command Syntax: :SEquence:CONTinue
Example: :SEquence:CONTinue

Name: **CYCles**
Type: Command, Query
Function: This command sets the number of times a sequence will run sequentially. The value can be from 1 to 999 or CONTinuous.

 This query returns the currently number of times a sequence will run.

Command Syntax: :SEquence:CYCles <numeric_value>|CONT
Example: :SEquence:CYCles 27
Query Syntax: :SEquence:CYCles?
Returned format: :SEquence:CYCles <numeric_value>|CONT

SEquence Subsystem

SEquence Subsystem

Name: NAME

Type: Command, Query

Function: This command is used to assign a name to a selected sequence. The name can consist of up to 8 alphanumeric characters in quotes.
This query returns the currently set name of a selected sequence.

Command Syntax: :SEquence:NAME "<data>", <numeric_value>

Example: :SEquence:NAME "John",4

Query Syntax: :SEquence:NAME?

Returned format: :SEquence:NAME "<data>", <numeric_value>

Name: RUN

Type: Command

Function: This command runs the selected sequence. The sequence is selected by its number not its name.

Command Syntax: :SEquence:RUN <numeric_value>

Example: :SEquence:RUN 7

Name: SOFTkey

Type: Command, Query

Function: This command is used to assign a sequence to a numeric button so that it may be run directly from the front panel. Numeric buttons 2, 3 and 4 can be used. If OFF is sent, no button is assigned.
This query returns the currently assigned button number of a particular sequence.

Command Syntax: :SEquence:SOFTkey <numeric_value>, <numeric_value>|OFF

Example: :SEquence:SOFTkey 7,4

Query Syntax: :SEquence:SOFTkey?

Returned format: :SEquence:SOFTkey <numeric_value>, <numeric_value>

20 The SETup Subsystem is used for saving and recalling instrument setups.

COMMANDS

:SETup
:RECALL
:SAVE

Name:	RECALL	Command
Function:	This command is used to recall a setup to the instrument. The recalled setup replaces the current setup.	
Command syntax:	:SETUP:RECALL <nunmeric_data>	
Example:	:SETUP:RECALL 3	

Name:	SAVE	Command
Function:	This command is used to save the current instrument setup to a setup memory.	
Command Syntax:	:SETUP:SAVE <nunmeric_data>	
Example:	:SETUP:SAVE 4	

21 The SYStem Subsystem is used for controlling the GPIB error reporting and format.

COMMANDS

:SYStem

:ERRor

:ERRorCLear

:HEADer

:LONGform

Name:	ERRor	Query
Function:	This query returns the earliest entry in the error list created by the instrument. There are two forms of the command numeric or string. Either the word STRING, the word NUMBER or no parameter can be specified. If STRING is specified, the DSO will return the error numbers together with their descriptions. If NUMBER or no parameter is specified, the error number only is returned .	
	The errors returned depend on the error type as shown in the table below:	
No. Error		
0 No Error		Save And Recall Errors
Special Purpose Errors		120 SETUP EMPTY
1 command failed		121 SETUP CORRUPT
2 invalid mode		122 SEQUENCE EMPTY
5 WARNING:VALUE FORCED INTO RANGE		123 SEQUENCE CORRUPT
Remote Command Processor/Parser Errors		124 ENTRY NOT ALLOWED IN SEQUENCE
10 command not known		125 MEMORY EMPTY
11 argument not known		126 MEMORY CORRUPT
12 argument expected		Main & Zoom Errors
13 string expected		129 TRACE NOT VISIBLE
14 query only		Display Errors
15 command only		130 ROLL:WILL ZOOM WHEN STORED
16 illegal data		Sequence Errors
17 data expected		131 CANNOT CONTINUE SEQUENCE
18 data not expected		132 SEQUENCE ALREADY RUNING
19 excessive data		133 SOFT KEY ALREADY ASSIGNED
20 invalid selection		140 INVALID KEY
21 input too long		141 NO YT MEASUREMENTS IN XY
I/O Errors:		142 NO XY MEASUREMENTS IN YT
55 I/O OPTION FAULT		143 BAD RESULT
Horizontal Errors		144 LIMITS TEST OFF
62 AVERAGING DISABLED IN ROLL		Roll Mode Errors
63 L.TEST AVAILABLE IN REFR ONLY		151 NO AVERAGE IN ROLL
272 NO 16bit TRANSFERS IN 50K STORES		General Errors
Warnings		155 50 OHM: NO AC COUPLING
90 WARNING:NO TRACE SELECTED		159 CANNOT TURN OF TRACE 1 IN XY
91 WARNING:NO Y TRACE VISIBLE		160 CHANNEL HOLD RESTRICTED BY
94 WARNING:INFINITE PERSISTENCE		SAMPLE RATE
95 WARNING:VALUE FORCED INTO RANGE		167 OPTION NOT AVAILABLE
96 WARNING:CALC ALREADY SELECTED		169 NOT ALLOWED IN FFT
99 8 TRACES ONLY IN MAIN & ZOOM		170 AUTOSSETUP ALREADY IN PROGRESS
		171 DATA CHECKSUM FAILURE
		172 INVALID ENTRY FROM NUMERIC
		ENTRY PAD

SYStem Subsystem

SYStem Subsystem

Thermal Plotter Errors

- 174 HEAD RAISED
- 175 PRINTER HOT
- 176 OUT OF PAPER

Cursor Errors

- 180 DATUM LOCKED TO GND IN ABS MODE
- 181 DATUM LOCKED TO TGR IN ABS MODE
- 182 THE TIME CURSOR AND DATUM ARE THE SAME
- 183 THE CURSORS ARE OFF

Analysis and Measurement Errors

- 184 CANNOT SET SCALING IN XY MODE
- 185 TRACE IS NOT A GRAPH
- 186 TRACE IS NOT A HISTOGRAM
- 187 CANNOT GRAPH IN XY
- 188 NO CURSOR TRACE SELECTED
- 189 ILLEGAL MEASUREMENT
- 190 GRAPH RUNNING
- 191 HISTOGRAM RUNNING
- 192 NOT ALLOWED WHEN RUNNING GRAPH
- 193 DESTINATION AS CURSOR TRACE
- 194 MIN IS NOT LESS THAN MAX
- 195 NOT ALLOWED WHEN RUNNING HISTOGRAM

Plotter Errors

- 196 PLOT BUFFER FULL
- 197 NO PLOTTER FITTED

GPIB Errors

- 200 GPIB deadlock
- 201 GPIB interrupted
- 202 GPIB unterminated
- 203 GPIB dcas
- 204 GPIB ifc
- 205 GPIB unexpected EOI
- 206 GPIB received GET

Mass Memory Errors

- 210 GRAPH NOT OVERWRITTEN
- 211 USER ALREADY EXISTS
- 212 FILE OR DIRECTORY DOES NOT EXIST
- 213 DEVICE FULL
- 214 DEVICE WRITE PROTECTED
- 215 SAVE ABORTED
- 216 FILE IN USE
- 217 NO DISPLAYED TRACES TO SAVE
- 218 NO FILES IN RUN NO.
- 219 WRONG TRACE LENGTH
- 220 COULDN'T READ DATA FROM DEVICE
- 221 WRITE ERROR

- 222 READ ERROR
- 223 GENERAL ERROR
- 224 ILLEGAL OPERATION
- 225 COULDN'T INITIALISE DEVICE
- 226 GENERIC ERROR
- 227 BAD RUN NUMBER
- 228 BAD NAME
- 229 BAD SEQUENCE
- 230 BAD TRACE NAME
- 231 BAD SETUP FILE
- 232 BAD SEQUENCE FILE
- 233 BAD FFT FILE
- 234 BAD HISTOGRAM FILE
- 235 BAD GRAPH FILE
- 236 SETUP FILE DOES NOT EXIST
- 237 LIST FULL

Glitch Detect Errors

- 269 GLITCH DETECT NOT AVAILABLE IN ETS

Trigger Errors

- 300 TRIGGER NOT ACTIVE
- 301 TV TRIG: NO AUX SOURCE
- 302 NO AUTO TRIG IN TV TRIGGER
- 310 TV TRIG: NO AUTO TRIGGER
- 311 TV TRIG: NO SLOPE
- 312 TV TRIG: DC COUPLING ONLY
- 320 B TRIG: LINE ONLY
- 330 LINE TRIG: DC hf-rej ONLY
- 331 LINE TRIG: AC hf-rej ONLY
- 340 EXT TRIG: DC COUPLING ONLY
- 341 AUX TRIG: CANNOT CHANGE TRIGGER LEVEL
- 342 TV TRIG: CANNOT CHANGE TRIGGER LEVEL
- 350 TRIGGER LEVEL OFF SCREEN
- 353 TRIG NOT IN USE
- 356 WARNING: NO ETS IN TRIGGER TOOLS
- 357 WARNING: NO TRIGGER TOOLS IN ETS

Multishot Errors

- 449 NO ROLL MODE IN MULTISHOT
- 450 NO TRIGGER TOLLS IN FAST REARM
- 451 NO AVERAGING IN MULTISHOT
- 452 NO MULTISHOT IN AVERAGING

Averaging Errors

- 453 NO AVERAGING IN 200K

TruTrace Errors

- 500 NO PERSISTENCE IN TRUTRACE
- 501 NO XY IN TRUTRACE
- 502 NO ROLL IN TRUTRACE

When an error has been examined using this command it is removed from the list.

Query syntax: :SYStem:ERRor? NUMBERSTRING

Returned Format: :SYStem:ERRor <error_number><string>

SYStem Subsystem

SYStem Subsystem

Name:	ERRorCLear	Command
Function:	This command is used to clear the error list held by the instrument when errors such as invalid commands are received.	
Command Syntax:	:SYStem:ERRorCLear	
Example:	:SYStem:ERRorCLear	

Name:	HEADer	Command, Query
Function:	This command is used to turn the Response Header part of a query reply on or off. Numeric values may be sent, 0 is interpreted as OFF and any other number is interpreted as ON. This query determines whether a Response Header is included as part of a reply to a query.	
Command Syntax:	:SYStem:HEADer <Boolean>	
Example:	:SYStem:HEADer OFF	
Query syntax:	:SYStem:HEADer?	
Returned Format:	:SYStem:HEADer <Boolean>	

Name:	LONGform	Command, Query
Function:	This command determines whether the DSO responds to queries with the minimum format of commands or the longform (complete form) of commands. The DSO will always accept either form of command as an input. Numeric values may be sent, 0 is interpreted as minimum and any other number is interpreted as Longform. This query returns the state of the command format to be sent by the DSO.	
Command Syntax:	:SYStem:LONGform <Boolean>	
Example:	:SYStem:LONGform ON	
Query syntax:	:SYStem:LONGform?	
Returned Format:	:SYStem:LONGform <Boolean>	

22 The TRace Subsystem is used for controlling and querying various aspects of the instrument's traces.

COMMANDS

TRace

```
:DATEandtime[12345678]      (query only)
:SEGmenTtime[12345678]      (query only)
:TRaceOFFset[12345678]
:Y
:TRaceSCaling[12345678]
:X
:Y
:TRaceUNIts[12345678]
:X
:Y
:TRaceUNCal[12345678]
:X
:Y
```

Name:	DATEandtime	Query
Function:	This query returns the date and time at which the trace was acquired.	
Query Syntax:	:TRace:DATEandtime[12345678]?	
Returned format:	:TRace:DATEandtime[12345678] <data>	

Name:	SEGmenTtime	Query
Function:	This query returns the date and time at which the segment was acquired.	
Query Syntax:	:TRace:SEGmenTtime[12345678] <numeric_data>, <numeric_data>?	
Returned format:	:TRace:DATEandtime[12345678] <numeric_data>, <numeric_data>	

Name:	TRaceOFFset[12345678]:Y	Command, query
Function:	This command is used to set the vertical offset voltage for the selected trace. The range is $\pm 1\text{E}\pm 30$. This query returns the vertical offset voltage for the selected trace.	
Command Syntax:	:TRace:TRaceOFFset[12345678]:Y <numeric_value>	
Query Syntax:	:TRace:TRaceOFFset[12345678]:Y?	
Returned format:	TRace:TRaceOFFset[12345678]:Y <numeric_value>	

Name:	TRaceSCaling[12345678]:X	Command, query
Function:	This command is used to set the horizontal scaling factor for the selected trace. This query returns the horizontal scaling factor for the selected trace. e.g. For a trace stored with a timebase of 100 $\mu\text{s}/\text{div}$ the numeric data will be 100E-3. If a trace has not been saved in the selected memory, an error will be reported and an invalid value returned.	
Command Syntax:	:TRace:TRaceSCaling[12345678]:X <numeric_value> EXtErnal	
Query Syntax:	:TRace:TRaceSCaling[12345678]:X?	
Returned format:	TRace:TRaceSCaling[12345678]:X <numeric_value> EXtErnal	

TRace Subsystem

TRace Subsystem

Name:	TRaceSCaling[12345678]:Y	Command, query
Function:	<p>This command is used to set the vertical scaling factor for the selected reference trace.</p> <p>This query returns the vertical scaling factor for the selected reference trace. e.g. For a trace stored with a range of 2 mV/div the numeric data will be 2E-3. If a trace has not been saved in the selected memory, an error will be reported and an invalid value returned.</p>	
Command Syntax:	:TRace:TRaceSCaling[12345678]:Y <numeric_value>	
Query Syntax:	:TRace:TRaceSCaling[12345678]:Y?	
Returned format:	:TRace:TRaceSCaling[12345678]:Y <numeric_value>	
Name:	TRaceUNIts[12345678]:X	Command, query
Function:	<p>This command is used to set the scaling units for the selected trace. <data> can be up to 4 alphanumeric digits.</p> <p>This query returns the scaling units for the selected trace.</p>	
Command Syntax:	:TRace:TRaceUNIts[12345678]:X <data>	
Query Syntax:	:TRace:TRaceUNIts[12345678]:X?	
Returned format:	TRace:TRaceUNIts[12345678]:X <data>	
Name:	TRaceUNIts[12345678]:Y	Command, query
Function:	<p>This command is used to set the scaling units for the selected trace.<data> can be up to 4 alphanumeric digits..</p> <p>This query returns the scaling units for the selected trace.</p>	
Command Syntax:	:TRace:TRaceUNIts[12345678]:Y <data>	
Query Syntax:	:TRace:TRaceUNIts[12345678]:Y?	
Returned format:	TRace:TRaceUNIts[12345678]:Y <data>	
Name:	TRaceUNCa[12345678]:X	Command, query
Function:	<p>This command is used to turn trace horizontal uncal on and off. Numeric valuses may be sent, 0 is interpreted as OFF and any other number is interpreted as ON.</p> <p>This query returns the current state of the horizontal uncal feature.</p>	
Command Syntax:	:TRace:TRaceUNCa[12345678]:X <Boolean>	
Query Syntax:	:TRace:TRaceUNCa[12345678]:X?	
Returned format:	TRace:TRaceUNCa[12345678]:X <Boolean>	
Name:	TRaceUNCa[12345678]:Y	Command, query
Function:	<p>This command is used to turn trace vertical uncal on and off. Numeric valuses may be sent, 0 is interpreted as OFF and any other number is interpreted as ON.</p> <p>This query returns the current state of the vertical uncal feature.</p>	
Command Syntax:	:TRace:TRaceUNCa[12345678]:Y <Boolean>	
Query Syntax:	:TRace:TRaceUNCa[12345678]:Y?	
Returned format:	TRace:TRaceUNCa[12345678]:Y <Boolean>	

23 The TRANSFER Subsystem is used for controlling all aspects of trace and setup data transfer to and from the DSO's display or mass memory storage areas.

COMMANDS

:TRANsfer

:ACQuisition

:ALL

:DATAonly

:WindowSTArt

:WindowSTOp

:FILE

:EXECute

:DEVICE

:RuNnAMe

:RuNnUMber

:SEQuence

:SETup

:TRACeNAME

:TYPE

:USER

:FORMat

:GRAPH

:ALL

:DATAonly

:WindowSTArt

:WindowSTOp

:HISTOgram

:ALL

:DATAonly

:WindowSTArt

:WindowSTOp

:MAIN

:DATAonly

:WindowSTArt

:WindowSTOp

:RADIX

:WORDsize

:ZOOM

:DATAonly

:WindowSTArt

:WindowSTOp

The :ALL, :DATAonly, WindowSTArt and WindowSTOp parts of the command header function in the same way for each of the commands;

:ACQuisition, :GRAPH, :HISTOgram, :MAIN, :MEMory and ZOOM.

The functions of :ALL, :DATAonly, WindowSTArt and WindowSTOp are described below and should be read in conjunction with the relevant command.

TRANSfer Subsystem

TRANSfer Subsystem

Name:	ALL	Command, Query
Function:	<p>This command is used in conjunction with the TRANSfer commands to transfer data to the DSO from a host computer. For the possible trace names see the relevant transfer command.</p> <p>This query is used in conjunction with the TRANSfer commands to transfer data from the DSO to a host computer.</p>	
Command Syntax:	:TRANSfer:{command}:ALL {trace},{Bulk data}	
Example:	:TRANSfer:ACQuisition:ALL 1,{bulk data}	
Query syntax:	:TRANSfer:{command}:ALL? {trace}	
Returned Format:	:TRANSfer:{command}:ALL {trace},{bulk data}	
See also:	DATAonly, ACQuisition, MEMory, Bulk Transfers 1.7	

Name:	DATAonly	Command, Query
Function:	<p>This command is used in conjunction with the TRANSfer commands to transfer data to the DSO from a host computer. For the possible trace names see the relevant transfer command.</p> <p>This query is used in conjunction with the TRANSfer commands to transfer data from the DSO to a host computer.</p>	
Command Syntax:	:TRANSfer:{command}:DATAonly {trace},{bulk data}	
Example:	:TRANSfer:ACQuisition:DATAonly 2,{bulk data}	
Query syntax:	:TRANSfer:{command}:DATAonly?{trace}	
Returned Format:	:TRANSfer:{command}:DATAonly {trace},{bulk data}	
See also:	ALL, ACQuisition, MAIN, MEMory, ZOOM, Bulk Transfers 1.7.	

Name:	ACQuisition	Command, Query
Function:	<p>This command is used to transfer trace data from a host computer to an acquisition store in the instrument. The store number can be 1, 2, 3 or 4. The store length depends upon the memory length in use when the trace was acquired. The bulk data must have been originally generated by and transferred from the DSO. The WindowSTart and WindowSTOp commands should be used to specify which segment of the data is to be transferred.</p> <p>This query is used to transfer trace data from an acquisition store in the DSO to a host computer. The store number can be 1, 2, 3 or 4. The store length depends upon the memory length in use when the trace was acquired. The WindowSTart and WindowSTOp commands should be used to specify which segment of the data is to be transferred.</p>	
Command Syntax:	:TRANSfer:ACQuisition:{ ALL DATAonly } [1234},{bulk data}	
Example:	:TRANSfer:ACQuisition:ALL 4,{bulk data}	
Query syntax:	:TRANSfer:ACQuisition:{ ALL DATAonly }? [1234]	
Returned Format:	:TRANSfer:ACQuisition:{ ALL DATAonly } [1234], {bulk data}	
See also:	ALL, DATAonly, Bulk Transfers 1.7.	

TRANsfer Subsystem

TRANsfer Subsystem

Name:	DEVICE	Command, query
Function:	<p>This command specifies the device to be used for file transfers. The possible values for <string> are FloppyDISK or RamDISK.</p> <p>This query returns the currently specified device used for file transfers.</p>	
Command Syntax:	:TRANsfer:FILE:DEVICE <string>	
Example:	:TRANsfer:FILE:DEVICE RamDISK	
Query Syntax:	:TRANsfer:FILE:DEVICE?	
Returned Format:	:TRANsfer:FILE:DEVICE <string>	

Name:	EXECute	Command, query
Function:	<p>This command is used to transfer data from a host computer to a mass storage file. Note that if the file already exists on the mass storage device, the transfer will not occur and the command will return an error.</p> <p>The query is used to transfer a mass storage file to a host computer.</p> <p>The file to be transferred is specified using the other commands in the :TRANsfer:FILE subsystem.</p> <p>Depending upon the type of file to be transferred, different parameters need to be specified before using this command.</p> <p>For TRACE transfers, the device, user, run name, run number and trace file need to be specified.</p> <p>For SETUP transfers, the device, user, and setup filename need to be specified.</p>	
Command Syntax:	:TRANsfer:FILE:EXECute <bulk_data>	
Example:	:TRANsfer:FILE:EXECute	
Query Syntax:	:TRANsfer:FILE:EXECute? <bulk_data>	
Returned Format:	:TRANsfer:FILE:EXECute <bulk_data>	

Name:	FORMat	Command, Query
Function:	<p>This command is used to define the format of data transmitted from the instrument and can either be raw data or pure text.</p> <p>Raw means Definite Length Arbitrary Response Data. See section 1.7.</p> <p>Text means Numeric response data NR1. See section 1.7.</p> <p>This query returns the current format to be used for data transmission</p>	
Command Syntax:	:TRANsfer:FORMat {TEXT RAW}	
Example:	:TRANsfer:FORMat TEXT	
Query syntax:	:TRANsfer:FORMat?	
Returned Format:	:TRANsfer:FORMat {RAW TEXT}	

Name:	GRAPH	Command, Query
Function:	<p>This command is used to transfer trace data from a host computer to a graph trace on the instrument. The trace can be either 1, 2, 3 or 4. The bulk data must have been originally generated by and transferred from the DSO. The WindowStart and WindowStop commands should be used to specify which segment of the data is to transferred.</p> <p>This query is used to transfer trace data from a graph trace on the DSO to a host computer. The trace can be either 1, 2, 3 or 4. The WindowStart and WindowStop commands should be used to specify which segment of the data is to transferred.</p>	
Command Syntax:	:TRANsfer:GRAPH:{ALL DATAonly} [1234], {bulk data}	
Example:	:TRANsfer:GRAPH:ALL 4, {bulk data}	
Query syntax:	:TRANsfer:GRAPH:{ALL DATAonly}? [1234]	
Returned Format:	:TRANsfer:GRAPH:{ALL DATAonly} [1234], {bulk data}	
See also:	ALL, DATAonly, Bulk Transfers 1.7.	

Name:	HISTOgram	Command, Query
Function:	<p>This command is used to transfer trace data from a host computer to a histogram trace on the instrument. The trace can be either 1, 2, 3 or 4. The bulk data must have been originally generated by and transferred from the DSO. The WindowStart and WindowStop commands should be used to specify which segment of the data is to transferred.</p> <p>This query is used to transfer trace data from a histogram trace on the DSO to a host computer. The trace can be either 1, 2, 3 or 4. The WindowStart and WindowStop commands should be used to specify which segment of the data is to transferred.</p>	
Command Syntax:	:TRANsfer:HISTOgram:{ALL DATAonly} [1234], {bulk data}	
Example:	:TRANsfer:HISTOgram:ALL 4, {bulk data}	
Query syntax:	:TRANsfer:HISTOgram:{ALL DATAonly}? [1234]	
Returned Format:	:TRANsfer:HISTOgram:{ALL DATAonly} [1234], {bulk data}	
See also:	ALL, DATAonly, Bulk Transfers 1.7.	

Name:	MAIN	Query
Function:	<p>This query is used to transfer trace data from a main display store in the DSO to a host computer. The trace is 502 data points. The WindowStart and WindowStop commands should be used to specify which segment of the data is to transferred.</p>	
Query syntax:	:TRANsfer:MAIN:DATAonly? {TRace1 TRace2 TRace3 TRace4 TRace5 TRace6 TRace7 TRace8}	
Returned Format:	:TRANsfer:MAIN:DATAonly {TRace1 TRace2 TRace3 TRace4 TRace5 TRace6 TRace7 TRace8}, {bulk data}	
See also:	ALL, DATAonly, Bulk Transfers 1.7.	

TRANSfer Subsystem

TRANSfer Subsystem

Name: **RADIX** Command, Query

Function: This command is used to set the number base used for data transfer. It can be either Octal, Decimal or Hexadecimal.

This query returns the number base to be used for data transfers.

Command Syntax: :TRANSfer:RADIX { OCTal | DECimal | HEXadecimal }

Example: :TRANSfer:RADIX DECimal

Query syntax: :TRANSfer:RADIX?

Returned Format: :TRANSfer:RADIX { OCTal | DECimal | HEXadecimal }

Name: **RuNnAMe** Command, query

Function: This command specifies the run name used when transferring trace files. The user need not specify the directory name extension, this is done automatically.

This query returns the run name used when transferring trace files.

Command Syntax: :TRANSfer:FILE:RuNnAMe <string>

Example: :TRANSfer:FILE:RuNnAMe

Query Syntax: :TRANSfer:FILE:RuNnAMe?

Returned Format: :TRANSfer:FILE:RuNnAMe <string>

Name: **RuNnUMber** Command, query

Function: This command specifies the run number used when transferring trace files. The user need not specify the full directory name, this is done automatically.

This query returns the run number used when transferring trace files.

Command Syntax: :TRANSfer:FILE:RuNnUMber <string>

Example: :TRANSfer:FILE:RuNnUMber

Query Syntax: :TRANSfer:FILE:RuNnUMber?

Returned Format: :TRANSfer:FILE:RuNnUMber <string>

Name: **SETup** Command, query

Function: This command specifies the setup file name used when transferring setup files. The user need not specify the file name extension, this is done automatically.

This query returns the setup file name used when transferring setup files.

Command Syntax: :TRANSfer:FILE:SETup <string>

Example: :TRANSfer:FILE:SETup

Query Syntax: :TRANSfer:FILE:SETup?

Returned Format: :TRANSfer:FILE:SETup <string>

TRANSfer Subsystem

TRANSfer Subsystem

Name: **TRACeNAME** Command, query

Function: This command specifies the trace name used when transferring trace files. The user must specify the full filename and extension.
This query returns the trace name used when transferring trace files.

Command Syntax: :TRANSfer:FILE:TRACeNAME <string>

Example: :TRANSfer:FILE:TRACeNAME

Query Syntax: :TRANSfer:FILE:TRACeNAME?

Returned Format: :TRANSfer:FILE:TRACeNAME <string>

Name: **TYPE** Command, query

Function: This command specifies the type of file to be transferred. Possible values for <string> are TRACE, SETUP and SEQUENCE.
This query returns the type of file to be transferred.

Command Syntax: :TRANSfer:FILE:TYPE <string>

Example: :TRANSfer:FILE:TYPE SETUP

Query Syntax: :TRANSfer:FILE:TYPE?

Returned Format: :TRANSfer:FILE:TYPE <string>

Name: **USER** Command, query

Function: This command specifies the user name to be used for file transfers.
This query returns the user name to be used for file transfers.

Command Syntax: :TRANSfer:FILE:USER <string>

Example: :TRANSfer:FILE:USER JAMES

Query Syntax: :TRANSfer:FILE:USER?

Returned Format: :TRANSfer:FILE:USER <string>

Name: **WindowSTArt** Command, Query

Function: This command is used in conjunction with a transfer command and the WindowSTOp command to define the portion of the trace transmitted or accepted by DATAonly transfer commands and queries.
For the whole trace, WindowSTArt should be set to 0 and WindowSTOp to 502, 5020 or 50200 depending upon the transfer type and memory length in use.
GRAPH, HISTOgram, MAIN and ZOOM traces are always 502 points.
ACQuisition and MEMOry traces can be either 502, 5020 or 50200 data points
If only the middle two divisions of a 502 point trace are required, WindowSTArt should be 200 and WindowSTOp 299.
This query returns the byte number of the first transmitted data point of DATAonly transfers.

Command Syntax: :TRANSfer:{command} WindowSTArt {numeric}

TRANsfer Subsystem

TRANsfer Subsystem

Example: :TRANsfer:{command} WindowSTArt 470

Query syntax: :TRANsfer:{command} WindowSTArt?

Returned Format: :TRANsfer:{command} WindowSTArt {numeric}

See also: DATAonly, Bulk Transfers 1.7.

Name: **WindowSTOp** Command, Query

Function: This command is used in conjunction with a transfer command and the WindowSTOp command to define the portion of the trace transmitted or accepted by DATAonly transfer commands and queries.

For the whole trace, WindowSTArt should be set to 0 and WindowSTOp to 501, 5019 or 50199 depending upon the transfer type and memory length in use.

GRAPH, HISTOgram, MAIN, ZOOM and LIMIT traces are always 502 points.

ACQuisition, REfERENCETraces and MEMory traces can be either 502, 5020 or 50200 data points

If only the middle two divisions of a 502 point trace are required, WindowSTArt should be 200 and WindowSTOp 299.

This query returns the byte number of the last transmitted data point of DATAonly transfers.

Command Syntax: :TRANsfer:{command} WindowSTOp {numeric}

Example: :TRANsfer:{command} WindowSTOp 502

Query syntax: :TRANsfer:{command} WindowSTOp?

Returned Format: :TRANsfer:{command} WindowSTOp {numeric}

See also: DATAonly, Bulk Transfers 1.7.

Name: **WORDsize** Command, Query

Function: This command sets the remote system word size for use with bulk transfers of data. It is relevant for acquisition, graphs and reference trace data but not for main, zoom or histogram data.

This query returns the current state of the WORDsize parameter.

Command syntax: :TRANsfer:WORDsize 8bit|16bit

Example: :TRANsfer:WORDsize 16bit

Query syntax: :TRANsfer:WORDsize?

Returned Format: :TRANsfer:WORDsize 8bit|16bit

Name: **ZOOM** Query

Function: This query is used to transfer trace data from a zoom display store in the DSO to a host computer. The trace is 502 data points. The WindowSTArt and WindowSTOp commands should be used to specify which segment of the data is to be transferred.

Query syntax: :TRANsfer:ZOOM:DATAonly? {TRace1|TRace2|TRace3|TRace4|TRace5|TRace6|TRace7|TRace8}

Returned Format: :TRANsfer:ZOOM:DATAonly {TRace1|TRace2|TRace3|TRace4|TRace5|TRace6|TRace7|TRace8}, {bulk data}

See also: ALL, DATAonly, Bulk Transfers 1.7.

24 The TRIGger Subsystem is used for controlling the instrument's trigger system.

COMMANDS

:TRIGger
 :SElect
 :SYstem

Name:	SElect	Command, Query
Function:	This command is used to control the trigger gating function of the instrument. The trigger system can be set to A (A triggers only), B (B triggers only) or AGAtEb (A gate B triggers). This query returns the current state of the trigger gating function.	
Command Syntax:	:TRIGger:SElect <data>	
Example:	:TRIGger:SElect B	
Query syntax:	:TRIGger:SElect?	
Returned Format:	:TRIGger:SElect <data>	

Name:	SYstem	Command, Query
Function:	This command sets the type of trigger to be used. This can be either standard triggering, TV triggering or advanced triggering. The permitted values of <data> are STanDard, TV and ADVanced. This query returns the trigger system currently in use.	
Command Syntax:	:TRIGger:SYstem <data>	
Example:	:TRIGger:SYstem TV	
Query syntax	:TRIGger:SYstem?	
Returned Format:	:TRIGger:SYstem <data>	

24a The TRIGger[AB] Subsystem is used for controlling the instrument's A & B trigger systems.

COMMANDS

```
:TRIGger[AB]
  :AUTotrig
  :COUpling
  :DELay
  :LEVel{ 1,2,3,4,EXT}
  :PREPost
  :PREtrig
  :SLOpe{ 1,2,3,4,EXT}
  :Source
```

Name:	AUTotrig	Command, Query
Function:	<p>This command is used to turn auto trigger on or off, i.e. set the trigger mode to either Auto or Normal. Numeric values may be sent, 0 is interpreted as Autotrigger OFF and any other number is interpreted as ON.</p> <p>This query returns the current state of the autotrigger system.</p>	
Command Syntax:	:TRIGger[AB]:AUTotrig <Boolean>	
Example:	:TRIGgerB:AUTotrig ON	
Query syntax:	:TRIGger[AB]:AUTotrig?	
Returned Format:	:TRIGger[AB]:AUTotrig <Boolean>	

Name:	COUpling	Command, Query
Function:	<p>This command is used to set the coupling of the trigger source for the A or B trigger system. If the trigger source is set to line then this command is not available. For high frequency reject coupling, the selection is HFREJ. The complete list of couplings is: AC, DC, ACHFREJ and DCHFREJ.</p> <p>This query returns the current coupling of the trigger source for the A or B trigger system.</p>	
Command Syntax:	:TRIGger[AB]:COUpling <data>	
Example:	:TRIGgerA:COUpling ACHFREJ	
Query syntax:	:TRIGger[AB]:COUpling?	
Returned Format:	:TRIGger[AB]:COUpling <data>	

TRIGger[AB] Subsystem

TRIGger[AB] Subsystem

Name:	DElay	Command, Query
Function:	<p>This command is used to set the post trigger delay time of the A or B trigger. The time is in seconds with a resolution of 5 ns.</p> <p>This query returns the current delay of the A or B trigger.</p>	
Command Syntax:	:TRIGger[AB]:DElay <numeric_value>	
Example:	:TRIGgerA:DElay 0.055	
Query syntax:	:TRIGger[AB]:DElay?	
Returned Format:	:TRIGger[AB]:DElay <numeric_value>	
See Also:	PREPost	

Name:	LEVEL{1,2,3,4,EXT}	Command, Query
Function:	<p>This command is used to set the trigger level for the A or B trigger system. The numeric value is in screen divisions from the source channel ground, and has a range of ± 8 divisions.</p> <p>This query returns the current trigger level for the A or B trigger system.</p>	
Command Syntax:	:TRIGger[AB]:LEVEL{1,2,3,4,EXT} <numeric_value>	
Example:	:TRIGgerB:LEVEL2 5.4	
Query syntax:	:TRIGger[AB]:LEVEL{1,2,3,4,EXT}?	
Returned Format:	:TRIGger[AB]:LEVEL{1,2,3,4,EXT} <numeric_value>	

Name:	PREPost	Command, Query
Function:	<p>This command is used to select either pre or post trigger for the A or B trigger system. PREtrigger or POSTtrigger are the values for <data>.</p> <p>This query returns the state of the Pre or Post trigger selection for the A or B trigger system.</p>	
Command Syntax:	:TRIGger[AB]:PREPost <data>	
Example:	:TRIGgerB:PREPost PREtrigger	
Query syntax:	:TRIGger[AB]:PREPost?	
Returned Format:	:TRIGger[AB]:PREPost <data>	

Name:	PREtrig	Command, Query
Function:	<p>This command is used to set the pre trigger percentage of the A or B trigger. The range is from 0% to 100% in 0.2% steps.</p> <p>This query returns the current pre trigger percentage of the A or B trigger.</p>	
Command Syntax:	:TRIGger[AB]:PREtrig <numeric_value>	
Example:	:TRIGgerA:PREtrig 50	
Query syntax:	:TRIGger[AB]:PREtrig?	
Returned Format:	:TRIGger[AB]:PREtrig <numeric_value>	
See Also:	PREPost	

TRIGger[AB] Subsystem

TRIGger[AB] Subsystem

Name:	SLOPE{1,2,3,4,EXT}	Command, Query
Function:	<p>This command is used to set the slope of the trigger used for the A or B trigger system. PLus means the positive (+) slope and MINus means the negative (–) slope.</p> <p>This query returns the current slope of the trigger used for the A or B trigger system.</p>	
Command Syntax:	:TRIGger[AB]:SLOPE{1,2,3,4,EXT} <data>	
Example:	:TRIGgerA:SLOPE3 PLus	
Query syntax:	:TRIGger[AB]:SLOPE{1,2,3,4,EXT}?	
Returned Format:	:TRIGger[AB]:SLOPE{1,2,3,4,EXT} <data>	

Name:	SOURCE	Command, Query
Function:	<p>This command is used to set the source for either the A or B trigger signals. The source can be any of the instrument's input channels, an EXTernal signal connected to the front panel EXT BNC or the instrument's AC Line supply.</p> <p>The source names are: CHANnel1, CHANnel2, CHANnel3, CHANnel4, EXTernal and LINE.</p> <p>This query returns the source of the A or B trigger signals.</p>	
Command Syntax:	:TRIGger[AB]:SOURCE <data>	
Example:	:TRIGgerA:SOURCE CHANnel1	
Query syntax:	:TRIGger[AB]:SOURCE?	
Returned Format:	:TRIGger[AB]:SOURCE <data>	

25 The TRIGgerTOOLS subsystem contains the commands for controlling all aspects of the Trigger Tools operation of the trigger system that relate to both the A and the B triggers.
To set a parameter use the appropriate command and to set the trigger tool to be used by the instrument, use the MODE command.

COMMANDS

- :TRIGgerTOOLS
 - :ADELayNGatesB
 - :COUNT
 - :ADELayTimeGatesB
 - :PERiod
 - :BAND
 - :WHEN
 - :BGatesaDelayN
 - :COUNT
 - :COMBination
 - :PERiod
 - :DELAybyN
 - :COUNT
 - :DIVidebyN
 - :COUNT
 - :CONTrol
 - :PHASEslip
 - :FREQuency
 - :FREQuency
 - :PERiod
 - :WHENFrequency
 - :WHENPeriod
 - :MISSing
 - :PERiod
 - :MODE
 - :PulseWidth
 - :WIDth
 - :WHEN
 - :RUNT
 - :POLarity
 - :SKEW
 - :PERiod
 - :WHEN
 - :SLEWrate
 - :PERiod
 - :WHEN

Name:	ADELayNGatesB:COUNT	Command, Query
Function:	This command sets the count number for N for the A delayed by N gates B trigger tools. The value of <numeric_data> can be set from 1 to 9999 in steps of 1. This query returns the current count number for A Delayed by N gates B.	
Command Syntax:	:TRIGgerTOOLS:ADELayNGatesB:COUNT <numeric_data>	
Example:	:TRIGgerTOOLS:ADELayNGatesB:COUNT 47	
Query Syntax:	:TRIGgerTOOLS:ADELayNGatesB:COUNT?	
Returned Format:	:TRIGgerTOOLS:ADELayNGatesB:COUNT <numeric_data>	

TRIGgerTOOLS Subsystem

TRIGgerTOOLS Subsystem

Name: **ADElayTimeGatesB:PERiod** Command, Query

Function: This command sets the time for the A delayed by time gates B trigger tools. The value of <numeric_data> can be set from 5 ns to 2.5 ns less than 400 s in 2.5 ns increments.

This query returns the current time for A Delayed by time gates B.

Command Syntax: :TRIGgerTOOLS:ADElayTimeGatesB:PERiod <numeric_data>

Example: :TRIGgerTOOLS:ADElayTimeGatesB:PERiod 76E-6

Query Syntax: :TRIGgerTOOLS:ADElayTimeGatesB:PERiod?

Returned Format: :TRIGgerTOOLS:ADElayTimeGatesB:PERiod <numeric_data>

Name: **BAND:WHEN** Command, Query

Function: This command sets the conditions for band trigger mode. <data> can be either ENTERsband or LEAVEsband.

This query returns the band conditions for Trigger Tools.

Command Syntax: :TRIGgerTOOLS:BAND:WHEN <data>

Example: :TRIGgerTOOLS:BAND:WHEN LEAVEsband

Query Syntax: :TRIGgerTOOLS:BAND:WHEN?

Returned Format: :TRIGgerTOOLS:Band:WHEN<data>

Name: **BGAtesaDelayN:COUNT** Command, Query

Function: This command sets the count number for N for the B gates A delayed by N trigger tools. The value of <numeric_data> can be set from 1 to 9999 in steps of 1.

This query returns the current count number for B gates A delayed by N.

Command Syntax: :TRIGgerTOOLS:BGAtesaDelayN:COUNT <numeric_data>

Example: :TRIGgerTOOLS:BGAtesaDelayN:COUNT 2001

Query Syntax: :TRIGgerTOOLS:BGAtesaDelayN:COUNT?

Returned Format: :TRIGgerTOOLS:BGAtesaDelayN:COUNT <numeric_data>

Name: **COMBination:PERiod** Command, Query

Function: This command sets the time used by the GREATERthan and LESSthan trigger combination commands. The value of <numeric_data> can be set from 5 ns to 2.5 ns less than 400 s in 2.5 ns increments.

This query returns the current time period to which the trigger system will respond.

Command Syntax: :TRIGgerTOOLS:COMBination:PERiod <numeric_data>

Example: :TRIGgerTOOLS:COMBination:PERiod 14E-6

Query Syntax: :TRIGgerTOOLS:COMBination:PERiod?

Returned Format: :TRIGgerTOOLS:COMBination:PERiod <numeric_data>

TRIGgerTOOLS Subsystem

TRIGgerTOOLS Subsystem

Name: **DElaybyN:COUNT** Command, Query

Function: This command sets the count number for N for the delayed by N trigger tools. The value of <numeric_data> can be set from 1 to 9999 in steps of 1.
This query returns the current count number for Delayed by N trigger.

Command Syntax: :TRIGgerTOOLS:DElaybyN:COUNT <numeric_data>

Example: :TRIGgerTOOLS:DElaybyN:COUNT 1407

Query Syntax: :TRIGgerTOOLS:DElaybyN:COUNT?

Returned Format: :TRIGgerTOOLS:DElaybyN:COUNT<numeric_data>

Name: **DIVidebyN:COUNT** Command, Query

Function: This command sets the count number for N for the Divide by N trigger tools. The value of <numeric_data> can be set from 1 to 9999 in steps of 1.
This query returns the current count number for Divide by N trigger.

Command Syntax: :TRIGgerTOOLS:DIVidebyN:COUNT <numeric_data>

Example: :TRIGgerTOOLS:DIVidebyN:COUNT 1959

Query Syntax: :TRIGgerTOOLS:DIVidebyN:COUNT?

Returned Format: :TRIGgerTOOLS:DIVidebyN:COUNT<numeric_data>

Name: **DIVidebyN:CONTRol** Command, Query

Function: This command sets the trigger point control for divide by N trigger to be either COUNT or PHASE. The actual amount of delay is set by the DIVidebyN:COUNT and DIVidebyN:PHASEslip commands.
This query returns the current trigger point control for Divide by N trigger.

Command Syntax: :TRIGgerTOOLS:DIVidebyN:CONTRol <data>

Example: :TRIGgerTOOLS:DIVidebyN:CONTRol COUNT

Query Syntax: :TRIGgerTOOLS:DIVidebyN:CONTRol?

Returned Format: :TRIGgerTOOLS:DIVidebyN:CONTRol<data>

Name: **DIVidebyN:PHASEslip** Command, Query

Function: This command is used to phase slip the Divide by N trigger point. Each time the command is sent, the trigger point will move by one phase. The direction of the move is determined by sending either PLus or MINus.
This query returns the direction of the phase slip.

Command Syntax: :TRIGgerTOOLS:DIVidebyN:PHASEslip <data>

Example: :TRIGgerTOOLS:DIVidebyN:PHASEslip PLus

Query Syntax: :TRIGgerTOOLS:DIVidebyN:PHASEslip?

Returned Format: :TRIGgerTOOLS:DIVidebyN:PHASEslip<data>

TRIGgerTOOLS Subsystem

TRIGgerTOOLS Subsystem

Name: **FREQuency:FREQuency** Command, Query

Function: This command sets the frequency for the frequency trigger tools. The value of <numeric_data> can be set from 2.5 mHz to 133.33 MHz.

This command also changes the setting of the FREQuency:PERiod command.

This query returns the current frequency to which the trigger system will respond.

Command Syntax: :TRIGgerTOOLS:FREQuency:FREQuency <numeric_data>

Example: :TRIGgerTOOLS:FREQuency:FREQuency 60

Query Syntax: :TRIGgerTOOLS:FREQuency:FREQuency?

Returned Format: :TRIGgerTOOLS:FREQuency:FREQuency <numeric_data>

Name: **FREQuency:PERiod** Command, Query

Function: This command sets the time for the frequency trigger tools. The value of <numeric_data> can be set from 7.5 ns to 2.5 ns less than 400 s in 2.5 ns increments.

This command also changes the setting of the FREQuency:FREQuency command.

This query returns the current time period to which the trigger system will respond.

Command Syntax: :TRIGgerTOOLS:FREQuency:PERiod <numeric_data>

Example: :TRIGgerTOOLS:FREQuency:PERiod 20E-3

Query Syntax: :TRIGgerTOOLS:FREQuency:PERiod?

Returned Format: :TRIGgerTOOLS:FREQuency:PERiod<numeric_data>

Name: **FREQuency:WHENFrequency** Command, Query

Function: This command sets the condition for the frequency trigger tools. The condition can be either LESSthan or GREATERthan the frequency set by the FREQuency:FREQuency command.

This query returns the current condition for the frequency trigger.

Command Syntax: :TRIGgerTOOLS:FREQuency:WHENFrequency <data>

Example: :TRIGgerTOOLS:FREQuency:WHENFrequency LESSthan

Query Syntax: :TRIGgerTOOLS:FREQuency:WHENFrequency?

Returned Format: :TRIGgerTOOLS:FREQuency:WHENFrequency<data>

Name: **FREQuency:WHENPeriod** Command, Query

Function: This command sets the condition for the frequency trigger tools. The condition can be either LESSthan or GREATERthan the period set by the FREQuency:PERiod command.

This query returns the current condition for the frequency trigger.

Command Syntax: :TRIGgerTOOLS:FREQuency:WHENPeriod <data>

Example: :TRIGgerTOOLS:FREQuency:WHENPeriod GREATERthan

Query Syntax: :TRIGgerTOOLS:FREQuency:WHENPeriod?

Returned Format: :TRIGgerTOOLS:FREQuency:WHENPeriod<data>

TRIGgerTOOLS Subsystem

TRIGgerTOOLS Subsystem

Name:	MISSing:PERiod	Command, Query
Function:	<p>This command sets the time period for the Missing trigger tools. The value of <numeric_data> can be set from 5 ns to 2.5 ns less than 400 s in 2.5 ns increments.</p> <p>This query returns the current period for Missing Trigger.</p>	
Command Syntax:	:TRIGgerTOOLS:MISSing:PERiod <numeric_data>	
Example:	:TRIGgerTOOLS:MISSing:PERiod 365	
Query Syntax:	:TRIGgerTOOLS:MISSing:PERiod?	
Returned Format:	:TRIGgerTOOLS:MISSing:PERiod<numeric_data>	

Name:	MODE	Command, Query
Function:	<p>This command sets trigger mode to be used by the trigger tools system and can be any one of the trigger type. Allowable values for <data> are: ADElayNGatesB, ADElayTimeGatesB, BAND, BGAtesaDElayN, COMBination, DELaybyN, DIVidebyN, FREQuency, MISSing, PulseWidth, RUNT, SKEW, SLEWrate.</p> <p>This query returns the current trigger tools mode.</p>	
Command Syntax:	:TRIGgerTOOLS:MODE <data>	
Example:	:TRIGgerTOOLS:MODE SLEWrate	
Query Syntax:	:TRIGgerTOOLS:MODE?	
Returned Format:	:TRIGgerTOOLS:MODE<data>	

Name:	PulseWidth:WIDTH	Command, Query
Function:	<p>This command sets the time period for the pulsewidth trigger tools. The value of <numeric_data> can be set from 5 ns to 2.5 ns less than 400 s in 2.5 ns increments.</p> <p>This query returns the current period for Pulsewidth Trigger.</p>	
Command Syntax:	:TRIGgerTOOLS:PulseWidth:WIDTH <numeric_data>	
Example:	:TRIGgerTOOLS:PulseWidth:WIDTH 34E-6	
Query Syntax:	:TRIGgerTOOLS:PulseWidth:WIDTH?	
Returned Format:	:TRIGgerTOOLS:PulseWidth:WIDTH<numeric_data>	

Name:	PulseWidth:WHEN	Command, Query
Function:	<p>This command sets the condition for the pulsewidth trigger tools. The condition can be either LESSthan or GREATERthan the period set by the PulseWidth:WIDTH command.</p> <p>This query returns the current condition for the pulsewidth trigger.</p>	
Command Syntax:	:TRIGgerTOOLS:PulseWidth:WHEN <data>	
Example:	:TRIGgerTOOLS:PulseWidth:WHEN LESSthan	
Query Syntax:	:TRIGgerTOOLS:PulseWidth:WHEN?	
Returned Format:	:TRIGgerTOOLS:PulseWidth:WHEN<data>	

TRIGgerTOOLS Subsystem

TRIGgerTOOLS Subsystem

Name: **RUNT:POLarity[1|2|3|4|EXT]** Command, Query

Function: This command specifies the edge of the signal that will be used by the trigger system. Either the POSitive or NEGative edge can be used.

1, 2, 3, 4 or EXT refer to the channel number or the external source for which the edge is being set.

This query returns the current edge which the trigger system will respond.

Command Syntax: :TRIGgerTOOLS:RUNT:POLarity[1|2|3|4|EXT] <data>

Example: :TRIGgerTOOLS:RUNT:POLarity2 POSitive

Query Syntax: :TRIGgerTOOLS:RUNT:POLarity[1|2|3|4|EXT]?

Returned Format: :TRIGgerTOOLS:RUNT:POLarity[1|2|3|4|EXT] <data>

Name: **SKEW:PERiod** Command, Query

Function: This command sets the time for the Skew trigger tools. The value of <numeric_data> can be set from 5 ns to 2.5 ns less than 400 s in 2.5 ns increments.

This query returns the current time period to which the trigger system will respond.

Command Syntax: :TRIGgerTOOLS:SKEW:PERiod <numeric_data>

Example: :TRIGgerTOOLS:SKEW:PERiod 128E-3

Query Syntax: :TRIGgerTOOLS:SKEW:PERiod?

Returned Format: :TRIGgerTOOLS:SKEW:PERiod <numeric_data>

Name: **SKEW:WHEN** Command, Query

Function: This command sets the condition for the skew trigger tools. The condition can be either LESSthan or GREATERthan the period set by the SKEW:PERiod command.

This query returns the current condition for the Skew trigger.

Command Syntax: :TRIGgerTOOLS:SKEW:WHEN<data>

Example: :TRIGgerTOOLS:SKEW:WHEN GREATERthan

Query Syntax: :TRIGgerTOOLS:SKEW:WHEN?

Returned Format: :TRIGgerTOOLS:SKEW:WHEN <data>

Name: **SLEWrate:PERiod** Command, Query

Function: This command sets the time for the slew rate period. The value of <numeric_data> can be set from 5 ns to 2.5 ns less than 400 s in 2.5 ns increments.

This query returns the current slew rate period.

Command Syntax: :TRIGgerTOOLS:SLEWrate:PERiod <numeric_data>

Example: :TRIGgerTOOLS:SLEWrate:PERiod 12E-9

Query Syntax: :TRIGgerTOOLS:SLEWrate:PERiod?

Returned Format: :TRIGgerTOOLS:SLEWrate:PERiod <numeric_data>

TRIGgerTOOLS Subsystem		TRIGgerTOOLS Subsystem
Name:	SLEWrate:WHEN	Command, Query
Function:	This command sets the condition for the Slew rate trigger tools. The condition can be either LESSthan or GREATERthan the period set by the SLEWrate PERiod command. This query returns the current condition for the slew rate trigger.	
Command Syntax:	:TRIGgerTOOLS:SLEWrate:WHEN <data>	
Example:	:TRIGgerTOOLS:SLEWrate:WHEN LESSthan	
Query Syntax:	:TRIGgerTOOLS:SLEWrate:WHEN?	
Returned Format:	:TRIGgerTOOLS:SLEWrate:WHEN <data>	

TRIGgerTOOLS[AB] Subsystem

TRIGgerTOOLS[AB] Subsystem

25a The TRIGgerTOOLS[AB] subsystem contains the commands for controlling all aspects of the Trigger Tools system that can be applied to either the A or the B triggers.

COMMANDS

:TRIGgerTOOLS[AB]

:COMBination

:AnyChangeQUALification

:HiLowQUALification

:Source

:TRIGger

:FREQuency

:EDGE

:PulseWidth

:POLarity

:SLEWrate

:EDGE

Name:	COMBination:AnyChangeQUALification Command, Query
Function:	This command sets the qualification type for use by Combination trigger when the trigger conditions are Any Change. The qualification can be either NONE or ENabled. This query returns the current qualification type for All Change Combination trigger.
Command Syntax:	:TRIGgerTOOLS[AB]:COMBination:AnyChangeQUALification <data>
Example:	:TRIGgerTOOLS[AB]:COMBination:AnyChangeQUALification ENabled
Query Syntax:	:TRIGgerTOOLS[AB]:COMBination:AnyChangeQUALification?
Returned Format:	:TRIGgerTOOLS[AB]:COMBination:AnyChangeQUALification <data>

Name:	COMBination:HiLowQUALification Command, Query
Function:	This command sets the qualification type for use by Combination trigger when the trigger conditions are any of the Hi or Lo types. The qualification can be either NONE, GREATERthan, LESSthan, CLocked or ENabled. The time period for use with LESSthan or GREATERthan qualifications is set using the COMBination:PERiod command. This query returns the current qualification type for Hi or Lo Combination trigger.
Command Syntax:	:TRIGgerTOOLS[AB]:COMBination:HiLowQUALification <data>
Example:	:TRIGgerTOOLS[AB]:COMBination:HiLowQUALification CLocked
Query Syntax:	:TRIGgerTOOLS[AB]:COMBination:HiLowQUALification?
Returned Format:	:TRIGgerTOOLS[AB]:COMBination:HiLowQUALification<data>

TRIGgerTOOLS[AB] Subsystem

TRIGgerTOOLS[AB] Subsystem

Name:	COMBination: SOurce[1 2 3 4]	Command, Query
Function:	<p>This command sets the state of the signal sources for combination trigger. Depending upon the qualification selected, each of the sources can be OFF, ON, ENabled or CLocked.</p> <p>This query returns the current state of the selected trigger source.</p>	
Command Syntax:	:TRIGgerTOOLS[AB]:COMBination:SOurce[1 2 3 4] <data>	
Example:	:TRIGgerTOOLSa:COMBination:SOurce2 ENabled	
Query Syntax:	:TRIGgerTOOLS[AB]:COMBination:SOurce[1 2 3 4]?	
Returned Format:	:TRIGgerTOOLS[AB]:COMBination:SOurce[1 2 3 4] <data>	

Name:	COMBination: TRIGger	Command, Query
Function:	<p>This command sets the trigger conditions for combination trigger tools. The trigger conditions can be ANYHi, ANYLo, ALLHi, ALLLo or ANYChange.</p> <p>This query returns the current trigger conditions for Combination trigger.</p>	
Command Syntax:	:TRIGgerTOOLS[AB]:COMBination:TRIGger <data>	
Example:	:TRIGgerTOOLSa:COMBination:TRIGger ALLHi	
Query Syntax:	:TRIGgerTOOLS[AB]:COMBination:TRIGger?	
Returned Format:	:TRIGgerTOOLS[AB]:COMBination:TRIGger<data>	

Name:	FREQuency:EDGE[1 2 3 4 EXT]	Command, Query
Function:	<p>This command specifies the edge of the signal that will be used by the trigger system. Either the RISing or FALLing edge can be used.</p> <p>1, 2, 3, 4 or ext refers to the channel number or the external source for which the edge is being set.</p> <p>This query returns the current edge which the trigger system will respond.</p>	
Command Syntax:	:TRIGgerTOOLS[AB]:FREQuency:EDGE[1 2 3 4 EXT] <data>	
Example:	:TRIGgerTOOLSa:FREQuency:EDGE4 FALLing	
Query Syntax:	:TRIGgerTOOLS[AB]:FREQuency:EDGE[1 2 3 4 EXT]?	
Returned Format:	:TRIGgerTOOLS[AB]:FREQuency:EDGE[1 2 3 4 EXT] <data>	

Name:	PWidth: POLarity[1 2 3 4 EXT]	Command, Query
Function:	<p>This command specifies the polarity of the pulse that will be used by the trigger system. Either POSitive or NEGative pulses can be used.</p> <p>1, 2, 3, 4 or ext refers to the channel number or the external source for which the polarity is being set.</p> <p>This query returns the current pulse polarity to which the trigger system will respond.</p>	
Command Syntax:	:TRIGgerTOOLS[AB]:PulseWidth:POLarity[1 2 3 4 EXT] <data>	
Example:	:TRIGgerTOOLSb:PulseWidth:POLarity2 POSitive	
Query Syntax:	:TRIGgerTOOLS[AB]:PulseWidth:POLarity[1 2 3 4 EXT]?	
Returned Format:	:TRIGgerTOOLS[AB]:PulseWidth:POLarity[1 2 3 4 EXT] <data>	

TRIGgerTOOLS[AB] Subsystem

TRIGgerTOOLS[AB] Subsystem

Name:	SLEWrate:EDGE[1 2 3 4 EXT]	Command, Query
Function:	This command specifies the edge of the signal that will be used by the trigger system. Either the RISING or FALLING edge can be used. 1, 2, 3, 4 or ext refers to the channel number or the external source for which the edge is being set. This query returns the current edge which the trigger system will respond.	
Command Syntax:	:TRIGgerTOOLS[AB]:SLEWrate:EDGE[1 2 3 4 EXT] <numeric_data>	
Example:	:TRIGgerTOOLS[AB]:SLEWrate:EDGE3 -6.77	
Query Syntax:	:TRIGgerTOOLS[AB]:SLEWrate:EDGE[1 2 3 4 EXT]?	
Returned Format:	:TRIGgerTOOLS[AB]:SLEWrate:EDGE[1 2 3 4 EXT] <numeric_data>	

26 The TRIGgerTV subsystem controls the actions associated with TV trigger conditions.

COMMANDS

:TRIGgerTV
 :MODE
 :Source
 :syncPOLarity

Name:	MODE	Command, Query
Function:	<p>This command is used to set the TV trigger mode to either LINE or FRAME.</p> <p>This query returns the current TV trigger mode.</p>	
Command Syntax:	:TRIGgerTV:MODE <data>	
Example:	:TRIGgerTV:MODE:FRAME	
Query Syntax:	:TRIGgerTV:MODE?	
Returned Format:	:TRIGgerTV:MODE<data>	

Name:	Source	Command, Query
Function:	<p>This command is used to set the channel that will be used as the trigger source when TV trigger is used. The trigger source <data> can be CHANnel1, CHANnel2, CHANnel3 or CHANnel4.</p> <p>This query returns the channel that will be used as the trigger source in TV trigger mode.</p>	
Command Syntax:	:TRIGgerTV:Source <data>	
Example:	:TRIGgerTV:Source CHANnel4	
Query Syntax:	:TRIGgerTV:Source?	
Returned Format:	:TRIGgerTV:Source <data>	

Name:	syncPOLarity	Command, Query
Function:	<p>This command is used to set the polarity of the sync pulses used by the TV trigger function. The polarity can be either Positive or Negative.</p> <p>This query returns the polarity of the sync pulses used to by the TV trigger function.</p>	
Command Syntax:	:TRIGgerTV:syncPOLarity<data>	
Example:	:TRIGgerTV:syncPOLarity Positive	
Query Syntax:	:TRIGgerTV:syncPOLarity?	
Returned Format:	:TRIGgerTV:syncPOLarity<data>	

Service Facilities

Gould and its distributors and agents maintain comprehensive after sales facilities and, whether or not it is still under guarantee, the instrument should be returned to the local Gould service centre or distributor through whom it was supplied for servicing if this is necessary. The type and serial number of the instrument should always be quoted, together with full details of any fault and service required.

Equipment returned for servicing must be adequately packed, preferably in the box in which the product was supplied and shipped, with transportation charges prepaid. We accept no responsibility for instruments arriving damaged.

Our Sales and Service Departments and those of approved distributors and agents are ready to assist you at all times.

The Gould Service Department and those of approved distributors and agents can provide maintenance and repair information by telephone or letter, if required.

PRINCIPAL GOULD SERVICE OFFICES

UNITED KINGDOM

Gould Instrument Systems,
Roebuck Road, Hainault,
Ilford, Essex, IG6 3UE
Telephone (0181) 500 1000
Fax (0181) 501 0116

GERMANY

Gould Nicolet Messtechnik GmbH,
Waldstrasse 66, 63128 Dietzenbach.
Telephone (06074) 4908-0.
Telex 4 197 644 Gould.
Fax (06074) 4908-48

FRANCE

Gould Instruments S.A.
57 Rue Saint Sauveur, Ballainvilliers,
91165 Longjumeau, Cédex.
Telephone (1) 69 10 22 40.
Telex 600824 GOULD
Fax (1) 6934 2073

USA

Gould Instrument Systems, Inc
8333, Rockside Road,
Valley View, OH. 44125-6104.
Telephone (216) 328 7000
Fax (216) 328 7400

ITALY

Gould Instrument Systems Filiale Italiana,
Via Torino 25,
20063 Cernusco S/N (MI).
Telephone (02) 9210 7461
Fax (02) 9210 7456

Contents

Introduction

1 GPIB Communication

2 Commands

3 ACQquisition Subsystem

4 ANALysis Subsystem

5 CALibrate Subsystem

6 CHANnel Subsystem

7 Common & Root Commands

8 CURSor Subsystem

9 DATE Subsystem

10 DATum Subsystem

11 DISPlay Subsystem

12 LIMits Subsystem

13 MassMEMory Subsystem

14 MEASurement Subsystem

15 MEMory Subsystem

16 MENU Subsystem

17 PLOT & PRINT Subsystem

18 RS423 Subsystem

19 SEQuence Subsystem

20 SETup Subsystem

21 SYStem Subsystem

22 TRace Subsystem

23 TRANsfer Subsystem

24 TRIGger & TRIGger[AB] Subsystems

25 TRIGgerTOOLS and TRIGgerTOOLS[AB] Subsystem

26 TRIGgerTV Subsystem

27 Service Facilities