

# Online Programmer Guide



## **TDS5000B Series Digital Phosphor Oscilloscopes**

**PHP023920**

Adapted from the TDS5000B Series Online Programmer Guide  
Version 2.0, July 15, 2004.



---

# Table of Contents

Copyright and Version Information .....	1
<b>Getting Started .....</b>	<b>2</b>
Introduction .....	2
Setting Up Remote Communications .....	3
Documentation .....	6
<b>Command Syntax .....</b>	<b>7</b>
Syntax Overview .....	7
Command and Query Structure .....	7
Clearing the Instrument .....	9
Command Entry .....	9
Constructed Mnemonics .....	11
Argument Types .....	13
<b>Command Groups .....</b>	<b>15</b>
Acquisition Command Group .....	15
Alias Command Group .....	16
Calibration Command Group .....	17
Cursor Command Group .....	17
Diagnostic Command Group .....	19
Display Control Command Group .....	21
E-mail Command Group .....	23
File System Command Group .....	24
Hard Copy Command Group .....	25
Histogram Command Group .....	26
Horizontal Command Group .....	27
Limit Test Command Group .....	29
Mask Command Group .....	32
Math Command Group .....	37
Measurement Command Group .....	38
Miscellaneous Command Group .....	42
Save and Recall Command Group .....	43
Status and Error Command Group .....	44
Trigger Command Group .....	45
Vertical Command Group .....	51
Waveform Transfer Command Group .....	52
Zoom Command Group .....	57

---

## Commands Listed in Alphabetical Order . . . 58

*CAL? .....	58
*CLS .....	59
*DDT .....	60
*ESE .....	61
*ESR? .....	62
*IDN? .....	63
*LRN? .....	64
*OPC .....	65
*OPT? .....	66
*PSC .....	67
*PUD .....	68
*RCL .....	69
*RST .....	70
*SAV .....	71
*SDS .....	72
*SRE .....	73
*STB? .....	74
*TRG .....	75
*TST? .....	76
*WAI .....	77
ACQuire:MODe .....	78
ACQuire:NUMACq? .....	80
ACQuire:NUMAVg .....	81
ACQuire:NUMEnv .....	82
ACQuire:NUMSAMples .....	83
ACQuire:SAMPlingmode .....	84
ACQuire:STATE .....	86
ACQuire:STOPAfter .....	88
ACQuire? .....	89
ALlas .....	90
ALlas:CATalog? .....	91
ALlas:DEFine .....	92
ALlas:DELEte .....	94
ALlas:DELEte:ALL .....	95
ALlas:DELEte:NAME .....	96
ALlas:STATE .....	97
ALLEV? .....	98
AUTOSet .....	99
AUXout:SOUrce .....	100
AUXout? .....	101
BELI .....	102
BUSY? .....	103
CAL? .....	104
CALibrate:CALProbe:CH<x>? .....	105

---

CALibrate:INTERNAL	106
CALibrate:INTERNAL:START	107
CALibrate:INTERNAL:STATUS?	108
CALibrate:PROBEstate:CH<x>?	109
CALibrate:RESults:SPC?	110
CALibrate:RESults?	111
CH<x>?	112
CH<x>:BANdwidth	113
CH<x>:COUPling	115
CH<x>:DESKew	116
CH<x>:INVert	117
CH<x>:LABel:NAME	118
CH<x>:LABel:XPOS	119
CH<x>:LABel:YPOS	120
CH<x>:OFFSet	121
CH<x>:POSition	123
CH<x>:PROBECal?	124
CH<x>:PRObe:GAIN?	125
CH<x>:PRObe:ID:SERnumber?	126
CH<x>:PRObe:ID:TYPE?	127
CH<x>:PRObe:ID?	128
CH<x>:PRObe:RESistance?	129
CH<x>:PRObe:UNIts?	130
CH<x>:PROBEFunc:EXTAtten	131
CH<x>:PROBEFunc:EXTDBatten	133
CH<x>:PROBEFunc:EXTUnits	134
CH<x>:PRObe?	135
CH<x>:SCALE	136
CH<x>:TERmination	138
CMDBatch	139
CURSor:FUNCTion	140
CURSor:HBArs?	142
CURSor:HBArs:DELTA?	143
CURSor:HBArs:POSITION<x>	144
CURSor:HBArs:UNIts?	145
CURSor:LINEStYle	146
CURSor:MODE	147
CURSor:SCREEN:StYle	148
CURSor:SOUrce<x>	149
CURSor:STATE	151
CURSor:VBArS	152
CURSor:VBArS:DELTA?	153
CURSor:VBArS:POSITION<x>	154
CURSor:VBArS:UNIts	155
CURSor:WAVEform:HDELTA?	156
CURSor:WAVEform:HPOS<x>?	157

---

CURSor:WAVEform:POSition<x>	158
CURSor:WAVEform:SOUrce<x>	159
CURSor:WAVEform:STYle	160
CURSor:WAVEform:UNIts	161
CURSor:WAVEform:VDELTA?	162
CURSor:WAVEform	163
CURSor:XY:PRODELta?	164
CURSor:XY:PRODUCT<x>?	165
CURSor:XY:RADIUS<x>?	166
CURSor:XY:RATDELta?	167
CURSor:XY:RATIO<x>?	168
CURSor:XY:RDELta?	169
CURSor:XY:READOUT	170
CURSor:XY:RECTX<x>	171
CURSor:XY:RECTY<x>	172
CURSor:XY:THDELta?	173
CURSor:XY:THETA<x>?	174
CURSor:XY:XDELta?	175
CURSor:XY:YDELta?	176
CURSor:XY?	177
CURSor?	178
CURVe	179
DATA	181
DATA:DESTination	182
DATA:ENCdg	183
DATA:SOUrce	186
DATA:STARt	187
DATA:STOP	188
DATE	190
DELEte:SETUp	191
DELEte:WAVEform	192
DESE	193
DIAg:CONTROL:HALT	194
DIAg:CONTROL:LOOP	195
DIAg:EXECUTE	196
DIAg:ITEM?	197
DIAg:ITEM:FAILURES?	198
DIAg:ITEM:NAME?	199
DIAg:ITEM:RESULT?	200
DIAg:ITEM:SUBITEMS?	201
DIAg:LEVEL	202
DIAg:LOOPS?	203
DIAg:NAME?	204
DIAg:NAME:AREA?	205
DIAg:NAME:SUBSYS?	206
DIAg:NAME:TEST?	207

---

DIAg:NUMITEMS? .....	208
DIAg:RESULts? .....	209
DIAg:RESULts:VERBoSe? .....	210
DIAg:SELEct:ALL .....	211
DIAg:SELEct:AREA .....	212
DIAg:SELEct:LAST .....	213
DIAg:SELEct:SUBSYS .....	214
DIAg:SELEct:TEST .....	215
DIAg:STATE .....	216
DIAg:STOP .....	217
DISplay:CLOCK .....	218
DISplay:COLOr:MATHCOLOr .....	219
DISplay:COLOr:PALEtte:IMAGEView .....	220
DISplay:COLOr:PALEtte:RECORDView .....	222
DISplay:COLOr:PALEtte:USEr:CARet .....	224
DISplay:COLOr:PALEtte:USEr:CH<x> .....	225
DISplay:COLOr:PALEtte:USEr:GRAticule .....	226
DISplay:COLOr:PALEtte:USEr:HIStoqram .....	227
DISplay:COLOr:PALEtte:USEr:MASK .....	228
DISplay:COLOr:PALEtte:USEr:MASKHighlght .....	229
DISplay:COLOr:PALEtte:USEr:MATH<x> .....	230
DISplay:COLOr:PALEtte:USEr:REF<x> .....	231
DISplay:COLOr:PALEtte:USEr? .....	232
DISplay:COLOr:REFCOLOr .....	233
DISplay:COLOr? .....	234
DISplay:FILTer .....	235
DISplay:FORMat .....	236
DISplay:GRAticule .....	238
DISplay:INTENSITy:AUTOBright .....	240
DISplay:INTENSITy:SCREENSAVER .....	242
DISplay:INTENSITy:SCREENSAVERDELAY .....	243
DISplay:INTENSITy:WAVEform:IMAGEView .....	244
DISplay:INTENSITy:WAVEform:RECORDView .....	245
DISplay:INTENSITy? .....	246
DISplay:PERsistence .....	247
DISplay:PERsistence:RESET .....	248
DISplay:SCREENExt:LABel<x>:NAME .....	249
DISplay:SCREENExt:LABel<x>:XPOS .....	250
DISplay:SCREENExt:LABel<x>:YPOS .....	251
DISplay:SCREENExt:LABel<x>? .....	252
DISplay:SCREENExt:STATE .....	253
DISplay:SCREENExt? .....	254
DISplay:SHOWREmote .....	255
DISplay:STYle .....	256
DISplay:TRIGBar .....	257
DISplay:VARpersist .....	258

---

DISplay? .....	259
EMail .....	261
EMail:ATTempts .....	262
EMail:AUTHLogin .....	263
EMail:AUTHPassword .....	264
EMail:COUnt? .....	265
EMail:FROm .....	266
EMail:HOSTwanted .....	267
EMail:IMage .....	268
EMail:LIMit .....	269
EMail:MASk .....	270
EMail:MAXsize .....	271
EMail:MEASurement .....	272
EMail:NUMemails .....	273
EMail:SMTPPort .....	274
EMail:SMTPServer .....	275
EMail:STATus? .....	276
EMail:TIMEout .....	277
EMail:TO .....	278
EMail:TRIGGER .....	279
EMail:WAVEform .....	280
EVENT? .....	281
EVMsg? .....	282
EVQty? .....	283
EXPort .....	284
EXPort:FILENAME .....	285
EXPort:FORMat .....	286
EXPort:IMAGe .....	287
EXPort:PALEtte .....	288
EXPort:VIEW .....	289
FACTory .....	290
FASTAcq:STATE .....	291
FASTAcq? .....	292
FILESystem:COpy .....	293
FILESystem:CWD .....	294
FILESystem:DELEte .....	295
FILESystem:DIR? .....	296
FILESystem:MKDir .....	297
FILESystem:PRInt .....	298
FILESystem:READFile .....	299
FILESystem:REName .....	300
FILESystem:RMDir .....	301
FILESystem:WRITEFile .....	302
FILESystem? .....	304
HARDCopy .....	305
HARDCopy:FILENAME .....	306



---

HARDCopy:IMAGe	308
HARDCopy:LAYout	309
HARDCopy:PALEtte	310
HARDCopy:PORT	311
HARDCopy:VIEW	312
HDR	313
HEADer	314
HIStogram?	316
HIStogram:Box	317
HIStogram:BOXPcnt	319
HIStogram:COUNt	321
HIStogram:DATA?	322
HIStogram:DISplay	323
HIStogram:FUNcTION	325
HIStogram:MODE	326
HIStogram:SIZE	328
HIStogram:SOURce	329
HIStogram:STATE	331
HORizontal:DIVisions?	332
HORizontal:FASTframe:COUNt	333
HORizontal:FASTframe:LENgth	334
HORizontal:FASTframe:MULtipleframes:FRAMEStart:<wfm>	335
HORizontal:FASTframe:MULtipleframes:MODE	336
HORizontal:FASTframe:MULtipleframes:NUMFRames:<wfm>	337
HORizontal:FASTframe:REF:FRAME	339
HORizontal:FASTframe:REF:SOURce	340
HORizontal:FASTframe:SELECTED:<wfm>	341
HORizontal:FASTframe:STATE	342
HORizontal:FASTframe:SUMFrame	343
HORizontal:FASTframe:TIMEStamp:ALL:<wfm>?	344
HORizontal:FASTframe:TIMEStamp:BETWeen:<wfm>?	345
HORizontal:FASTframe:TIMEStamp:DELTA:<wfm>?	346
HORizontal:FASTframe:TIMEStamp:FRAME:<wfm>?	347
HORizontal:FASTframe:TIMEStamp:REF?	348
HORizontal:FASTframe:TIMEStamp:SELECTED:<wfm>?	349
HORizontal:FASTframe:TRACk	350
HORizontal:FASTframe:XZErO:ALL:CH<x>?	352
HORizontal:FASTframe:XZErO:ALL:REF<x>?	354
HORizontal:FASTframe:XZErO:FRAME:CH<x>?	356
HORizontal:FASTframe:XZErO:FRAME:REF<x>?	357
HORizontal:FASTframe:XZErO:REF?	358
HORizontal:FASTframe:XZErO:SELECTED:CH<x>?	359
HORizontal:FASTframe:XZErO:SELECTED:REF<x>?	360
HORizontal:FASTframe?	361
HORizontal[:MAIN]:DELay:MODE	362
HORizontal[:MAIN]:DELay:POSition	363

---

HORizontal[:MAIn]:DELay:TIME .....	364
HORizontal:MAIn:INTERPRatio? .....	365
HORizontal[:MAIn]:POSition .....	366
HORizontal:MAIn:SAMPLERate .....	367
HORizontal[:MAIn]:SCAle .....	368
HORizontal:MAIn:UNIts .....	369
HORizontal:MAIn:UNIts:STRing .....	370
HORizontal:MAIn? .....	371
HORizontal:POSition .....	372
HORizontal:RECOrdlength .....	373
HORizontal:RESOLution .....	374
HORizontal:ROLL .....	375
HORizontal:SCAle .....	376
HORizontal:TRIGger:POSition .....	377
HORizontal? .....	378
ID? .....	379
LIMit:Beep .....	380
LIMit:COMpare .....	381
LIMit:COMpare:CH<x> .....	382
LIMit:COMpare:MATH<x> .....	383
LIMit:COMpare:REF<x> .....	384
LIMit:EMail .....	385
LIMit:HARDCopy .....	386
LIMit:HIGHLIGHTHits .....	387
LIMit:LOCK .....	388
LIMit:LOG .....	389
LIMit:SAVEWFM .....	390
LIMit:SAVWFM:FILEName .....	391
LIMit:SRQ .....	392
LIMit:STATE .....	393
LIMit:STOPOnviolation .....	394
LIMit:TEMPLate:STORe .....	395
LIMit:TEMPLate:TOLerance:HORizontal .....	396
LIMit:TEMPLate:TOLerance:VERTical .....	397
LIMit? .....	398
LOCK .....	399
MASK? .....	400
MASK:AUTOAdjust .....	401
MASK:AUTOAdjust:HDELTA .....	402
MASK:AUTOAdjust:VDELTA .....	403
MASK:AUTOSet:AUTOAdjust .....	404
MASK:AUTOSet:HPOS .....	405
MASK:AUTOSet:HSCAle .....	406
MASK:AUTOSet:MODE .....	407
MASK:AUTOSet:OFFSETAdj .....	408
MASK:AUTOSet:STANdard (TDS5000B Series Option SM) .....	409

---

MASK:AUTOSet:TRIGger	417
MASK:AUTOSet:VPOS	418
MASK:AUTOSet:VSCAle	419
MASK:COpy:USER	420
MASK:COUNt RESET	421
MASK:COUNt:FAILURES?	422
MASK:COUNt:HITS?	423
MASK:COUNt:SEG<m>:HITS?	424
MASK:COUNt:STATE	425
MASK:COUNt:TESTS?	426
MASK:COUNt:TOTAl?	427
MASK:COUNt:VIOLATIONS?	428
MASK:COUNt:WAVEFORMS?	429
MASK:DISPlay	430
MASK:FILTer	431
MASK:HIGHLIGHTHits	432
MASK:INVert	433
MASK:LOCK	434
MASK:MARgin:PERCent	435
MASK:MARgin:STATE	436
MASK:MASKPRE:AMPlitude	437
MASK:MASKPRE:HSCAle	438
MASK:MASKPRE:HTRIGPOS	439
MASK:MASKPRE:PATTERNBITS	440
MASK:MASKPRE:PRESAMPBITS	441
MASK:MASKPRE:RECOrdlength	442
MASK:MASKPRE:SERIALTRIG	443
MASK:MASKPRE:TRIGTOSAMP	444
MASK:MASKPRE:VOFFSet	445
MASK:MASKPRE:VPOS	446
MASK:MASKPRE:VSCAle	447
MASK:MASKPRE:WIDth	448
MASK:POLarity	449
MASK:SEG<m> DELEte	450
MASK:SEG<m>:NR_Pt?	451
MASK:SEG<m>:POINTS	452
MASK:SOUrce	453
MASK:STANdard (TDS5000B Series Option SM)	454
MASK:STOPOnviolation	463
MASK:TESt:AUX:COMPLetion	464
MASK:TESt:AUX:FAILure	465
MASK:TESt:BEEP:COMPLetion	466
MASK:TESt:BEEP:FAILure	467
MASK:TESt:DELAy	468
MASK:TESt:HARDCopy	469
MASK:TESt:LOG:FAILure	470

---

MASK:TEST:REPeat	471
MASK:TEST:SAMple	472
MASK:TEST:SAMple:THReshold	473
MASK:TEST:SAVEWFM	474
MASK:TEST:SAVEWFM:FILEName	475
MASK:TEST:SRQ:COMPLetion	476
MASK:TEST:SRQ:FAILure	477
MASK:TEST:STATE	478
MASK:TEST:STATUS?	479
MASK:TEST:STOP:FAILure	480
MASK:TEST:THReshold	481
MASK:TEST:WAVEform	482
MASK:USER:AMPlitude	483
MASK:USER:BITRate	484
MASK:USER:HSCAle	485
MASK:USER:HTRIGPOS	486
MASK:USER:LABel	487
MASK:USER:PATTERNBITS	488
MASK:USER:PRESAMPBITS	489
MASK:USER:RECOrdlength	490
MASK:USER:SEG<m> DELEte	491
MASK:USER:SEG<m>:NR_Pt?	492
MASK:USER:SEG<m>:POINTS	493
MASK:USER:TRIGTOSAMP	494
MASK:USER:VOFFSet	495
MASK:USER:VPOS	496
MASK:USER:VSCAle	497
MASK:USER:WIDth	498
MATH<x>:DEFIne	499
MATH<x>:LABel:NAME	501
MATH<x>:LABel:XPOS	502
MATH<x>:LABel:YPOS	503
MATH<x>:NUMAVg	504
MATH<x>:SPECTral?	505
MATH<x>:SPECTral:CENTER	506
MATH<x>:SPECTral:GATEPOS	507
MATH<x>:SPECTral:GATEWIDTH	508
MATH<x>:SPECTral:LOCK	509
MATH<x>:SPECTral:MAG	511
MATH<x>:SPECTral:PHASE	512
MATH<x>:SPECTral:REFLevel	513
MATH<x>:SPECTral:REFLEVELOffset	514
MATH<x>:SPECTral:RESBw	516
MATH<x>:SPECTral:SPAN	517
MATH<x>:SPECTral:SUPPpress	519
MATH<x>:SPECTral:UNWRap	520

---

MATH<x>:SPECTral:WINDow	521
MATH<x>:UNITString	523
MATH<x>:VERTical:POSition	524
MATH<x>:VERTical:SCAlE	526
MATHVAR:VAR<x>	528
MATH<x>?	529
MEASUrement:GATing	530
MEASUrement:IMMed?	531
MEASUrement:IMMed:DELay?	532
MEASUrement:IMMed:DELay:DIREction	533
MEASUrement:IMMed:DELay:EDGE[1]	534
MEASUrement:IMMed:DELay:EDGE2	535
MEASUrement:IMMed:NOISe (TDS5000B Series option SM)	536
MEASUrement:IMMed:REFLevel:ABSolute:HIGH	537
MEASUrement:IMMed:REFLevel:ABSolute:LOW	538
MEASUrement:IMMed:REFLevel:ABSolute:MID[1]	539
MEASUrement:IMMed:REFLevel:ABSolute:MID2	540
MEASUrement:IMMed:REFLevel:METhod	541
MEASUrement:IMMed:REFLevel:PERCent:HIGH	542
MEASUrement:IMMed:REFLevel:PERCent:LOW	543
MEASUrement:IMMed:REFLevel:PERCent:MID[1]	544
MEASUrement:IMMed:REFLevel:PERCent:MID2	545
MEASUrement:IMMed:REFLevel?	546
MEASUrement:SOURCE<x>:SIGType (TDS5000B Series option SM)	547
MEASUrement:IMMed:SOURCE[1]	548
MEASUrement:IMMed:SOURCE2	550
MEASUrement:IMMed:SOURce<x>:SIGType (TDS5000B Series option SM)	552
MEASUrement:IMMed:TYPe	553
MEASUrement:IMMed:UNIts?	560
MEASUrement:IMMed:VALue?	561
MEASUrement:MEAS<x>:COUNT?	562
MEASUrement:MEAS<x>:MAXimum?	563
MEASUrement:MEAS<x>:MEAN?	564
MEASUrement:MEAS<x>:MINimum?	565
MEASUrement:MEAS<x>:NOISe (TDS5000B Series option SM)	566
MEASUrement:MEAS<x>:STDdev?	567
MEASUrement:MEAS<x>:DELay:DIREction	568
MEASUrement:MEAS<x>:DELay:EDGE[1]	569
MEASUrement:MEAS<x>:DELay:EDGE2	570
MEASUrement:MEAS<x>:DELay?	571
MEASUrement:MEAS<x>:REFLevel:ABSolute:HIGH	572
MEASUrement:MEAS<x>:REFLevel:ABSolute:LOW	573
MEASUrement:MEAS<x>:REFLevel:ABSolute:MID[1]	574
MEASUrement:MEAS<x>:REFLevel:ABSolute:MID2	575
MEASUrement:MEAS<x>:REFLevel:METhod	576

---

MEASUrement:MEAS<x>:REFLevel:PERCent:HIGH .....	577
MEASUrement:MEAS<x>:REFLevel:PERCent:LOW .....	578
MEASUrement:MEAS<x>:REFLevel:PERCent:MID[1] .....	579
MEASUrement:MEAS<x>:REFLevel:PERCent:MID2 .....	580
MEASUrement:MEAS<x>:REFLevel? .....	581
MEASUrement:MEAS<x>:SOURCE[1] .....	582
MEASUrement:MEAS<x>:SOURCE2 .....	584
MEASUrement:MEAS<x>:SOURce<x>:SIGType (TDS5000B Series option SM) .....	586
MEASUrement:MEAS<x>:STATE .....	587
MEASUrement:MEAS<x>:TYPe .....	589
MEASUrement:MEAS<x>:UNIts? .....	596
MEASUrement:MEAS<x>:VALue? .....	597
MEASUrement:MEAS<x>? .....	598
MEASUrement:MEthod .....	599
MEASUrement:NOISe (TDS5000B Series option SM) .....	601
MEASUrement:REFLevel:ABSolute:HIGH .....	602
MEASUrement:REFLevel:ABSolute:LOW .....	603
MEASUrement:REFLevel:ABSolute:MID[1] .....	604
MEASUrement:REFLevel:ABSolute:MID2 .....	605
MEASUrement:REFLevel:MEthod .....	606
MEASUrement:REFLevel:PERCent:HIGH .....	607
MEASUrement:REFLevel:PERCent:LOW .....	608
MEASUrement:REFLevel:PERCent:MID[1] .....	609
MEASUrement:REFLevel:PERCent:MID2 .....	610
MEASUrement:REFLevel? .....	611
MEASUrement:STATIstics:COUNT .....	612
MEASUrement:STATIstics:MODE .....	613
MEASUrement:STATIstics:WEIghting .....	614
MEASUrement? .....	615
NEWpass .....	616
PASSWord .....	617
RECAIl:MASK .....	618
RECAIl:SETUp .....	619
RECAIl:WAVEform .....	621
REF<x>:LABel? .....	622
REF<x>:LABel:NAME .....	623
REF<x>:LABel:XPOS .....	624
REF<x>:LABel:YPOS .....	625
REF<x>:HORizontal:POSition .....	626
REF<x>:VERTical:POSition .....	627
REF<x>:VERTical:SCALE .....	629
REM .....	631
ROSc:SOURce .....	632
ROSc:STATE? .....	633
SAVe:MASK .....	634

SAVe:SETUp	635
SAVe:WAVEform	636
SAVe:WAVEform:FILEFormat	637
SElect:<wfm>	639
SElect:CONTRol <wfm>	640
SElect?	641
SETUp:NAME	642
SET?	643
TEKSecure	644
TEST	645
TEST:RESults?	646
TEST:RESults:VERBoSe?	647
TEST:STOP	648
TIME	649
TRIGger	650
TRIGger:A	652
TRIGger:A:COMMunication:AMI...PULSEForm (Option SM Only)	654
TRIGger:A:COMMunication:AMI...THReshold:HIGH (Option SM Only)	655
TRIGger:A:COMMunication:AMI...THReshold:LOW (Option SM Only)	656
TRIGger:A:COMMunication:BITRate (Option SM Only)	657
TRIGger:A:COMMunication:CLOCK:POLarity (Option SM Only)	658
TRIGger:A:COMMunication:CMI:PULSEForm (Option SM Only)	659
TRIGger:A:COMMunication:CODE (Option SM Only)	660
TRIGger:A:COMMunication:SOURce (Option SM Only)	661
TRIGger:A:COMMunication:SOURce:TYPE (Option SM Only)	662
TRIGger:A:COMMunication:STANDard (Option SM Only)	663
TRIGger:A:EDGE:COUPLing	668
TRIGger:A:EDGE:SLOpe	670
TRIGger:A:EDGE:SOURce	671
TRIGger:A:EDGE?	672
TRIGger:A:HOLDoff:ACTUal?	673
TRIGger:A:HOLDoff:BY	674
TRIGger:A:HOLDoff:TIME	675
TRIGger:A:HOLDoff?	676
TRIGger:A:LEVel	677
TRIGger:A:LOGIc:CLAss	678
TRIGger:A:LOGIc:FUNcTION	680
TRIGger:A:LOGIc:INPut:CH<x>	681
TRIGger:A:LOGIc:INPut?	683
TRIGger:A:LOGIc:PATtern:INPut:CH2	684
TRIGger:A:LOGIc:PATtern:INPut:CH4	685
TRIGger:A:LOGIc:PATtern:WHEn	686
TRIGger:A:LOGIc:PATtern:WHEn:LESSLimit	688
TRIGger:A:LOGIc:PATtern:WHEn:MORELimit	689

TRIGger:A:LOGic:PATtern? .....	690
TRIGger:A:LOGic:SETHold:CLOCK:EDGE .....	691
TRIGger:A:LOGic:SETHold:CLOCK:SOUrce .....	692
TRIGger:A:LOGic:SETHold:CLOCK:THReshold .....	693
TRIGger:A:LOGic:SETHold:CLOCK? .....	694
TRIGger:A:LOGic:SETHold:DATA:SOUrce .....	695
TRIGger:A:LOGic:SETHold:DATA:THReshold .....	696
TRIGger:A:LOGic:SETHold:DATA? .....	697
TRIGger:A:LOGic:SETHold:HOLDTime .....	698
TRIGger:A:LOGic:SETHold:SETTime .....	699
TRIGger:A:LOGic:SETHold? .....	700
TRIGger:A:LOGic:STATE:INPUt:CH2 .....	701
TRIGger:A:LOGic:STATE:INPUt:CH4 .....	702
TRIGger:A:LOGic:STATE:WHEn .....	703
TRIGger:A:LOGic:STATE? .....	704
TRIGger:A:LOGic:THReshold:CH<x> .....	705
TRIGger:A:LOGic:THReshold? .....	706
TRIGger:A:LOGic? .....	707
TRIGger:A:MODE .....	708
TRIGger:A:PULse:CLAss .....	709
TRIGger:A:PULse:GLItch:POLarity .....	711
TRIGger:A:PULse:GLItch:TRIGIF .....	712
TRIGger:A:PULse:GLItch:WIDth .....	713
TRIGger:A:PULse:GLItch? .....	714
TRIGger:A:PULse:RUNT:LOGic:INPUt:CH<x> .....	715
TRIGger:A:PULse:RUNT:LOGic:INPUt? .....	716
TRIGger:A:PULse:RUNT:LOGic:THReshold:CH<x> .....	717
TRIGger:A:PULse:RUNT:LOGic? .....	718
TRIGger:A:PULse:RUNT:POLarity .....	719
TRIGger:A:PULse:RUNT:THReshold:BOTH .....	720
TRIGger:A:PULse:RUNT:THReshold:HIGH .....	721
TRIGger:A:PULse:RUNT:THReshold:LOW .....	722
TRIGger:A:PULse:RUNT:THReshold? .....	723
TRIGger:A:PULse:RUNT:WHEn .....	724
TRIGger:A:PULse:RUNT:WIDth .....	726
TRIGger:A:PULse:RUNT? .....	727
TRIGger:A:PULse:SOUrce .....	728
TRIGger:A:PULse:TIMEOut:POLarity .....	729
TRIGger:A:PULse:TIMEOut:TIME .....	730
TRIGger:A:PULse:TIMEOut? .....	731
TRIGger:A:PULse:TRANSition:DELTATime .....	732
TRIGger:A:PULse:TRANSition:POLarity .....	733
TRIGger:A:PULse:TRANSition:THReshold:BOTH .....	734
TRIGger:A:PULse:TRANSition:THReshold:HIGH .....	735
TRIGger:A:PULse:TRANSition:THReshold:LOW .....	736
TRIGger:A:PULse:TRANSition:THReshold? .....	737



---

TRIGger:A:PULse:TRANSition:WHEn	738
TRIGger:A:PULse:TRANSition?	739
TRIGger:A:PULse:WIDth:HIGHLimit	740
TRIGger:A:PULse:WIDth:LOWLimit	741
TRIGger:A:PULse:WIDth:POLarity	742
TRIGger:A:PULse:WIDth:WHEn	743
TRIGger:A:PULse:WIDth?	745
TRIGger:A:PULse:WINDow:LOGIc:INPUT:CH<x>	746
TRIGger:A:PULse:WINDow:LOGIc:INPUT?	747
TRIGger:A:PULse:WINDow:LOGIc:THReshold:CH<x>	748
TRIGger:A:PULse:WINDow:LOGIc?	749
TRIGger:A:PULse:WINDow:THReshold:BOTH	750
TRIGger:A:PULse:WINDow:THReshold:HIGH	751
TRIGger:A:PULse:WINDow:THReshold:LOW	752
TRIGger:A:PULse:WINDow:THReshold?	753
TRIGger:A:PULse:WINDow:TYPE	754
TRIGger:A:PULse:WINDow:WHEn	755
TRIGger:A:PULse:WINDow:WIDTh	756
TRIGger:A:PULse:WINDow?	757
TRIGger:A:PULse?	758
TRIGger:A:TYPE	759
TRIGger:A:VIDeo:CUSTom:FORMAt	761
TRIGger:A:VIDeo:CUSTom:SCAN	762
TRIGger:A:VIDeo:CUSTom?	764
TRIGger:A:VIDeo:FIELD	765
TRIGger:A:VIDeo:HOLdoff:FIELD	767
TRIGger:A:VIDeo:LINE	768
TRIGger:A:VIDeo:POLarity	770
TRIGger:A:VIDeo:SCAN	771
TRIGger:A:VIDeo:SOUrce	773
TRIGger:A:VIDeo:STANdard	774
TRIGger:A:VIDeo?	776
TRIGger:B	777
TRIGger:B:BY	778
TRIGger:B:EDGE:COUPling	780
TRIGger:B:EDGE:SLOpe	781
TRIGger:B:EDGE:SOUrce	782
TRIGger:B:EDGE?	783
TRIGger:B:EVENTS:COUNt	784
TRIGger:B:EVENTS?	785
TRIGger:B:LEVel	786
TRIGger:B:STATE	787
TRIGger:B:TIME	788
TRIGger:B:TYPE	789
TRIGger:STATE?	790
UNLock	792

---

VERBose	793
WFMInpre?	794
WFMInpre:BIT_Nr	795
WFMInpre:BN_Fmt	796
WFMInpre:BYT_Nr	797
WFMInpre:BYT_Or	798
WFMInpre:ENCdg	799
WFMInpre:NR_FR?	800
WFMInpre:NR_Pt	801
WFMInpre:PT_Fmt	802
WFMInpre:PT_Off	804
WFMInpre:WFId	805
WFMInpre:XINcr	806
WFMInpre:XUNit	807
WFMInpre:XZEro	808
WFMInpre:YMUlt	809
WFMInpre:YOff	810
WFMInpre:YUNit	811
WFMInpre:YZEro	812
WFMOutpre?	813
WFMOutpre:BIT_Nr	814
WFMOutpre:BN_Fmt	815
WFMOutpre:BYT_Nr	816
WFMOutpre:BYT_Or	817
WFMOutpre:ENCdg	818
WFMOutpre:NR_FR?	819
WFMOutpre:NR_Pt?	820
WFMOutpre:PT_Fmt?	821
WFMOutpre:PT_Off?	822
WFMOutpre:PT_OR?	823
WFMOutpre:WFId?	824
WFMOutpre:XINcr?	826
WFMOutpre:XUNit?	827
WFMOutpre:XZEro?	828
WFMOutpre:YMUlt?	829
WFMOutpre:YOff?	830
WFMOutpre:YUNit?	831
WFMOutpre:YZEro?	832
WFMPre:NR_FR?	833
WAVFrm?	834
ZOOM	836
ZOOM:GRAticule:SIZE	838
ZOOM:MODe	839
ZOOM:SCROLL:DIREction	840
ZOOM:SCROLL:LOCK	841
ZOOM:SCROLL:SPEED	842

---

ZOOm:GRAticule:SPLit .....	843
ZOOm:ZOOM<x> .....	844
ZOOm:ZOOM<x>:<wfm>:HORizontal:POSition .....	845
ZOOm:ZOOM<x>:<wfm>:HORizontal:SCAle .....	846
ZOOm:ZOOM<x>:<wfm>:VERTical:POSition .....	847
ZOOm:ZOOM<x>:<wfm>:VERTical:SCAle .....	848
ZOOm:ZOOM<x>:SCROLLLock .....	849
ZOOm:ZOOM<x>:STATE .....	850

## **Status and Events ..... 851**

Registers .....	851
Queues .....	855
Event Handling Sequence .....	855
Synchronization Methods .....	856
Messages .....	863

## **Programming Examples ..... 868**

Overview .....	868
Compiling GPIB Programs .....	869
Visual C++ .....	869

## **Miscellaneous ..... 871**

Character Charts .....	871
Reserved Words .....	872
Factory Default Setup Values .....	876
GPIB Interface Specifications .....	895



---

# Copyright and Version Information

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

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

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

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

Tektronix, Inc. P.O. Box 500, Beaverton, OR 97077

TEKTRONIX, TEK, and TEKPROBE are registered trademarks of Tektronix, Inc.

FrameScan is a trademark of Tektronix, Inc.

TDS5000B Series Programmer Online Guide

OLH0239, Version 2.00, July 15, 2004

---

# Getting Started

## Introduction

This online programmer guide provides you with the information required to use GPIB commands for remotely controlling your instrument. With this information, you can write computer programs that will perform functions such as setting the front-panel controls, taking measurements, performing statistical calculations, and exporting data for use in other programs, such as spreadsheets.

Besides the traditional GPIB electronic interface (referred to as the physical GPIB interface), your instrument is provided with a *TekVISA* GPIB-compatible interface (referred to as the virtual GPIB interface). This is a software Application Programming Interface (API) which enables you to communicate with the instrument in a variety of ways, including via the internet. With the following two exceptions, these interfaces are completely independent:

- **HEADER.** Command headers enabled or disabled on one interface are correspondingly enabled or disabled on the other interface. Refer to the command descriptions for more detailed information.
- **VERBOSE.** Verbosity enabled or disabled on one interface is correspondingly enabled or disabled on the other interface. Refer to the command description for more detailed information.

Refer to Documentation for information on related manuals and documents.

The programmer guide is divided into the following major topics (books):

- **Getting Started.** This topic introduces you to the online help and provides basic information about setting up your instrument for remote control.
- **Command Syntax.** This topic provides an overview of the command syntax that you will use to communicate with the instrument and other general information about commands, such as how commands and queries are constructed, how to enter commands, constructed mnemonics, and argument types.
- **Command Groups.** This topic contains all the commands listed in functional groups. Each group consists of an overview of the commands in that group and a table that lists all the commands and queries for that group. You can click a command in the listing to display a detailed description of the command.
- **Status and Events.** This topic discusses the status and event reporting system for the GPIB interfaces. This system informs you of certain significant events that occur within the instrument. Topics that are discussed include registers, queues, event handling sequences, synchronization methods, and messages that the instrument may return, including error messages.

- **Miscellaneous.** This topic contains miscellaneous information, such as a list of reserved words, a table of the factory initialization (default) settings, and GPIB interface specifications that may be helpful when using GPIB commands to remotely control the instrument.

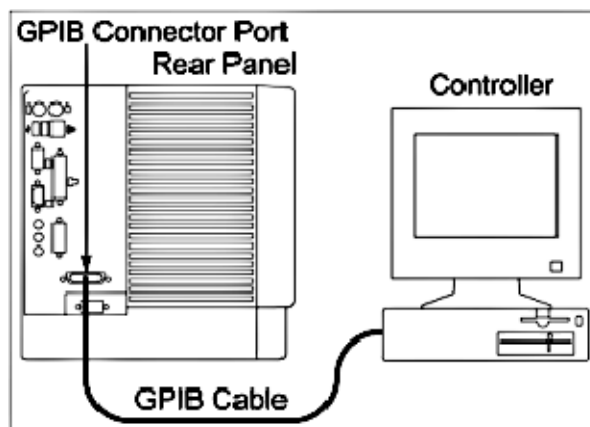
## Setting Up Remote Communications

Before setting up the instrument for remote communications using the electronic (physical) GPIB interface, you should familiarize yourself with the following GPIB requirements:

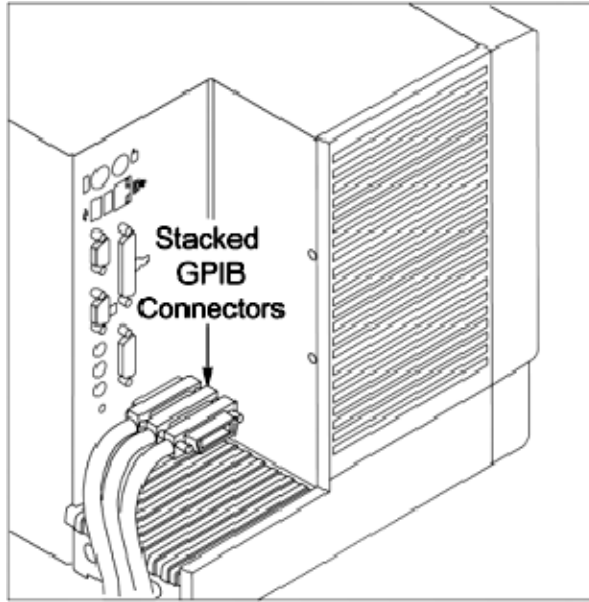
- A unique device address must be assigned to each device on the bus. No two devices can share the same device address.
- No more than 15 devices can be connected to any one line.
- One device should be connected for every 6 feet (2 meters) of cable used.
- No more than 65 feet (20 meters) of cable should be used to connect devices to a bus.
- At least two-thirds of the devices on the network should be powered on while using the network.
- Connect the devices on the network in a star or linear configuration. Do not use loop or parallel configurations.

### Connecting to the Instrument

Your instrument has a 24-pin GPIB connector on its rear (side) panel. This connector has a D-type shell and conforms to IEEE Std 488.1—1987. Attach an IEEE Std 488.1—1987 GPIB cable to this connector and to your controller as shown in the following figure.



If necessary, the GPIB connectors can be stacked as shown in the figure below.



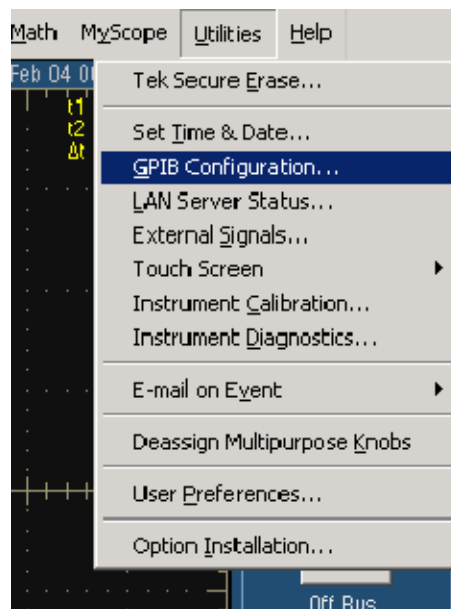
### Setting the GPIB Address

To function correctly, your instrument must have a unique device address. The default settings for the GPIB configuration are:

- GPIB Address 1
- GPIB Mode GPIB Talk/Listen

To change either of the GPIB settings, do the following:

1. Select GPIB Configuration... from the Utilities menu.





2. Click the Configuration Talk/Listen button.



3. Change the GPIB Address to a unique address.
4. Click the Close button.

The instrument is now set up for bidirectional communication with your controller.

---

# Documentation



***TDS5000B Series Digital Phosphor Oscilloscopes Quick Start User Manual.*** The user manual has information about installing and operating the instrument. It also provides concepts and theories about using the instrument that are not covered in the online help.

***TDS5000B Series Online Help.*** This is an online help system that is integrated with the User Interface application that ships with this product. The online help provides in-depth operation and user interface help.

***Getting Started with OpenChoice™ Solutions Manual.*** A book that explores some options for getting data from your instrument into any one of several available analysis tools.

***TDS5000B Series Digital Phosphor Oscilloscopes Specifications and Performance Verification.*** Instrument specifications and a performance verification procedure is available as a printable PDF file on the *TDS5000B Series Product Software CD-ROM*.

***TekVISA Programmer Manual.*** This manual is available as a printable PDF file on the *TDS5000B Series Product Software CD-ROM*. The manual describes TekVISA, the Tektronix implementation of the VISA Application Programming Interface (API). TekVISA is industry-compliant software for writing interoperable instrument drivers in a variety of Application Development Environments (ADEs).

***Optional Applications Software for Tektronix Windows-Based Instruments (CD-ROM).*** This CD-ROM contains trial versions of application-specific programs that you can install and run five times per application. To purchase an application, contact your local Tektronix representative.

***Other Included Documentation.*** Installation booklets are included in the *TDS5000B Series Product Software* and *TDS5000B Series Operating System Restore Software CD-ROM* packages.

***TDS5000B Series Digital Phosphor Oscilloscopes Service Manual.*** A printed service manual is available as an optional accessory. The service manual includes procedures to service the instrument to module levels. This manual is also available as a printable PDF file on the *TDS5000B Series Product Software CD-ROM*.

---

# Command Syntax

## Syntax Overview

You can control the operations and functions of the instrument through the GPIB interface using commands and queries. The related topics listed below describe the syntax of these commands and queries. The topics also describe the conventions that the instrument uses to process them. See the *Command Groups* topic in the table of contents for a listing of the commands by command group, or use the index to locate a specific command.

### Backus-Naur Form Notation

This documentation describes the commands and queries using Backus-Naur Form (BNF) notation. Refer to the following table for the symbols that are used.

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

## Command and Query Structure

### Overview

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

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

---

## Messages

A command message is a command or query name followed by any information the instrument needs to execute the command or query. Command messages may contain five element types, defined in the following table.

Command Message Elements	
Symbol	Meaning
<Header>	This is the basic command name. If the header ends with a question mark, the command is a query. The header may begin with a colon (:) character. If the command is concatenated with other commands, the beginning colon is required. Never use the beginning colon with command headers beginning with a star (*).
<Mnemonic>	This is a header subfunction. Some command headers have only one mnemonic. If a command header has multiple mnemonics, a colon (:) character always separates them from each other.
<Argument>	This is a quantity, quality, restriction, or limit associated with the header. Some commands have no arguments while others have multiple arguments. A <space> separates arguments from the header. A <comma> separates arguments from each other.
<Comma>	A single comma is used between arguments of multiple-argument commands. Optionally, there may be white space characters before and after the comma.
<Space>	A white space character is used between a command header and the related argument. Optionally, a white space may consist of multiple white space characters.

## Commands

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

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

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

---

## Queries

Queries cause the instrument to return status or setting information. Queries have the structure:

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

You can specify a query command at any level within the command tree unless otherwise noted. These branch queries return information about all the mnemonics below the specified branch or level. For example, HISTogram:STATistics:STDdev? returns the standard deviation of the histogram, while HISTogram:STATistics? returns all the histogram statistics, and HISTogram? returns all the histogram parameters.

## Headers

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

Comparison of Header Off and Header On Responses		
Query	Header Off	Header On
TIME?	"14:30:00"	:TIME "14:30:00"
ACQuire:NUMAVg?	100	:ACQUIRE:NUMAVG 100

## Clearing the Instrument

You can clear the Output Queue and reset the instrument to accept a new command or query by using the selected Device Clear (DCL) GPIB function. Refer to your GPIB library documentation for further details about the selected Device Clear operation.

## Command Entry

### Rules

The following rules apply when entering commands:

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

- 
- The instrument ignores commands consisting of any combination of white space characters and line feeds.

## Abbreviating

You can abbreviate many instrument commands. Each command in this documentation shows the abbreviations in capitals. For example, you can enter the command `ACQuire:NUMAVg` simply as `ACQ:NUMA` or `acq:numa`.

Abbreviation rules may change over time as new instrument models are introduced. Thus, for the most robust code, use the full spelling.

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

## Concatenating

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

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

1. Separate completely different headers by a semicolon and by the beginning colon on all commands except the first one. For example, the commands `TRIGger:MODE NORMAl` and `ACQuire:NUMAVg 10`, can be concatenated into the following single command:

```
TRIGger:MODE NORMAl;:ACQuire:NUMAVg 10
```

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

```
ACQuire:MODE ENVeloPe; NUMAVg 10
```

The longer version works equally well:

```
ACQuire:MODE ENVeloPe;:ACQuire:NUMAVg 10
```

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

```
ACQuire:MODE ENVeloPe;*OPC
```

Any commands that follow will be processed as if the star command was not there so the commands, `ACQuire:MODE ENVeloPe;*OPC;NUMAVg 10` will set the acquisition mode to envelope and set the number of acquisitions for averaging to 10.

4. When you concatenate queries, the responses to all the queries are concatenated into a single response message. For example, if the display background color is white and the display foreground color is black, the concatenated query `DISplay:COLor:BACKGround?;FOREGround?` will return the following.

---

If the header is on:

```
:DISPLAY:COLOR:BACKGROUND 7;  
:DISPLAY:COLOR:FOREGROUND 0
```

If the header is off:

```
7;0
```

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

```
ACQUIRE:MODE SAMPLE;NUMAVG?;STATE?
```

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

Here are some invalid concatenations:

- `DISPlay:STyle:NORMAl;ACQuire:NUMAVg 10`  
(no colon before ACQuire)
- `DISPlay:COLor:CURSor1 1;:CURSor2 5`  
(extra colon before CURSor2; use `DISPlay:COLor:CURSor1 1:CURSor2 5` instead)
- `DISPlay:STyle:NORMAl;:*OPC`  
(colon before a star (\*) command)
- `DISPlay:COLor:CURSor1 1;COLor:CURSor2 5`  
(levels of the mnemonics are different; either remove the second use of COLor or place `:DISPlay:` in front of `COLor:CURSor2 5`)

## Terminating

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

End of Message Terminator	
Symbol	Meaning
<EOM>	Message terminator

The end-of-message terminator must be the END message (EOI asserted concurrently with the last data byte). The last data byte may be an ASCII linefeed (LF) character.

This instrument does not support ASCII LF only message termination. The instrument always terminates outgoing messages with LF and EOI.

## Constructed Mnemonics

Some header mnemonics specify one of a range of mnemonics. For example, a channel mnemonic can be CH1, CH2, CH3, or CH4 for

four-channel instruments or CH1 or CH2 for two-channel instruments. You use these mnemonics in the command just as you do any other mnemonic. For example, there is a CH1:POSition command, and there is also a CH2:POSition command. In the command descriptions, this list of choices is abbreviated as CH<x>.

### Cursor Position Mnemonics

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

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

### Math Specifier Mnemonics

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

Math Specifier Mnemonics	
Symbol	Meaning
MATH<x>	A math waveform specifier; <x> is 1 through 4 for four-channel instruments or 1 through 2 for two-channel instruments.

### Measurement Specifier Mnemonics

Commands can specify which measurement to set or query as a mnemonic in the header. Up to eight automated measurements may be displayed on a four-channel instrument or four automated measurements on a two-channel instrument.

Measurement Specifier Mnemonics	
Symbol	Meaning
MEAS<x>	A measurement specifier; <x> is 1 through 8 for four-channel instruments or 1 through 4 for two-channel instruments.

### Channel Mnemonics

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

Channel Mnemonics	
Symbol	Meaning
CH<x>	A channel specifier; <x> is 1 through 4 for four-channel instruments and 1 through 2 for two-channel instruments.



---

## Reference Waveform Mnemonics

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

Reference Waveform Mnemonics	
Symbol	Meaning
REF<x>	A reference waveform specifier; <x> is 1 through 4 for four-channel instruments or 1 through 2 for two-channel instruments.

## Argument Types

### Numeric

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

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

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

### Quoted String

Some commands accept or return data in the form of a quoted string, which is simply a group of ASCII characters enclosed by a single quote (') or double quote ("). The following is an example of a quoted string: "This is a quoted string". This documentation represents these arguments as follows:

Quoted String Argument	
Symbol	Meaning
<QString>	Quoted string of ASCII text

A quoted string can include any character defined in the 7-bit ASCII character set. Follow these rules when you use quoted strings:

1. Use the same type of quote character to open and close the string. For example: "this is a valid string".

2. You can mix quotation marks within a string as long as you follow the previous rule. For example, "this is an 'acceptable' string".
3. You can include a quote character within a string by repeating the quote. For example: "here is a "" mark".
4. Strings can have upper or lower case characters.
5. If you use a GPIB network, you cannot terminate a quoted string with the END message before the closing delimiter.
6. A carriage return or line feed embedded in a quoted string does not terminate the string, but is treated as just another character in the string.
7. The maximum length of a quoted string returned from a query is 1000 characters.

Here are some invalid strings:

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

## Block

Several instrument commands use a block argument form (see the following table).

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

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

---

# Command Groups

## Acquisition Command Group

Use the commands in the Acquisition Command Group to set up the modes and functions that control how the instrument acquires the signals you input to the channels and processes them into waveforms.

Using these commands for acquiring waveforms, you can do the following:

- Start and stop acquisitions.
- Control whether each waveform is simply acquired, averaged, or enveloped over successive acquisitions of that waveform.
- Set the controls or conditions that start and stop acquisitions.
- Determine the action the system takes upon completing an acquisition, such as saving all waveforms and taking a measurement when the acquisition is stopped.
- Control acquisition of acquired channel waveforms.
- Set acquisition parameters.

Command	Description
ACQuire:MODE	Sets or returns acquisition mode
ACQuire:NUMAcq?	Returns the number of acquisitions that have occurred
ACQuire:NUMAVg	Sets or returns number of acquisitions for an averaged waveform
ACQuire:NUMEnv	Sets or returns number of acquisitions for envelope waveform
ACQuire:NUMSAMPles	Sets or returns the number of samples that make up a WfmDB for single sequence mode and Mask Pass/Fail Completion Test
ACQuire:SAMPlingmode	This command sets or queries the sampling mode
ACQuire:STATE	Starts, stops, or returns acquisition state
ACQuire:STOPAfter	Sets or returns whether the acquisition is continuous or single sequence
ACQuire?	Returns acquisition parameters
FASTAcq:STATE	Enables, disables, or returns state of Fast Acquisition mode
FASTAcq?	Returns the Fast Acquisition state

---

# Alias Command Group

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

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

- The alias name must consist of a valid IEEE 488.2 message unit, which may not appear in a message preceded by a colon, comma, or a command or query program header.
- The alias name may not appear in a message followed by program data, a colon, comma, or question mark.
- An alias name must be distinct from any keyword or keyword short form.
- An alias name cannot be redefined without first being deleted using one of the alias deletion functions.
- Alias names do not appear in response messages.
- The Alias commands are defined in Tektronix Standard Codes and Formats. Deviations between that standard and what is specified here will be considered TDS5000 series errors unless specifically noted in the command description in this document.

<b>Command</b>	<b>Description</b>
ALias	Sets or returns the alias state
ALias:CATalog?	Returns a list of the currently defined alias labels
ALias:DEFine	Assigns a sequence of program messages to an alias label
ALias:DELEte	Removes a specified alias
ALias:DELEte:ALL	Deletes all existing aliases
ALias:DELEte:NAME	Removes a specified alias
ALias:STATE	Sets or returns the alias state

---

## Calibration Command Group

The calibration commands provide information about the current state of instrument calibration and allow you to initiate internal signal path calibration (SPC). Commands that are specific to factory calibration are not described in this manual; however, they are described in the service manual (located on your TDS5000B Product Software CD-ROM in PDF format). You can also order a printed copy (see Recommended Accessories in the online help for the instrument).

Command	Description
*CAL?	Instructs the instrument to perform self-calibration
CAL?	Returns the internal and factory calibration status and the calibration due date
CALibrate:CALProbe:CH<x>?	Performs a probe calibration for the selected channel and returns the calibration status
CALibrate:INTERNaL	Starts the internal signal path calibration
CALibrate:INTERNaL:START	Starts the internal signal path calibration
CALibrate:INTERNaL:STATus?	Returns the current status of the internal signal path calibration.
CALibrate:PROBEstate:CH<x>?	Returns the probe calibration status for the probe of the selected channel
CALibrate:RESults?	Returns the status of all calibration subsystems without performing an SPC operation
CALibrate:RESults SPC?	Returns the results of the last SPC operation

## Cursor Command Group

Use the commands in the Cursor Command Group to control the cursor display and readout. You can use these commands to control the setups for cursor 1 and cursor 2, such as waveform source, cursor position, and cursor color.

You can also use the commands to select one of the following cursor functions:

- **Off.** Shuts off the display of all cursors.
- **Vertical Bars.** Displays vertical bar cursors, which provide traditional horizontal unit readouts for Cursor 1 (bar1), Cursor 2 (bar2), the delta between them, and 1/delta (results in frequency when the horizontal unit is time).
- **Horizontal Bars.** Displays horizontal bar cursors, which provide traditional vertical unit readouts for Cursor 1 (bar1), Cursor 2 (bar2), and the delta between them.

- **Waveform Cursors.** Consists of two cursors you can independently assign to a waveform. These cursors provide the same readouts that the vertical and horizontal bar cursors provide. Waveform cursors enable you to conveniently measure waveform amplitude and time. In XY or XYZ format, waveform cursors indicate the amplitude position of an XY pair (Ch1 vs Ch2 voltage, where Ch1 is the X axis and Ch2 is the Y axis) relative to the trigger.
- **Screen Cursors.** Consists of two pairs of independent horizontal and vertical cursors. You can use these cursors to indicate an arbitrary position within the waveform display area. Screen cursors, depending on the style selected, consist of the intersection of a vertical and horizontal line, an X, or a vertical line with an X. These cursors have no association with any waveform, other than they inherit the color of the waveform they are assigned too.

Command	Description
CURSor?	Returns all cursor settings
CURSor:FUNcTION	Sets or returns the cursor type
CURSor:HBArs?	Returns hbar cursor settings
CURSor:HBArs:DELTA?	Returns hbars cursors vertical difference
CURSor:HBArs:POSITION<x>	Sets or returns the hbar cursor<x> vertical position
CURSor:HBArs:UNIts?	Returns hbar cursor units
CURSor:LINEStYle	Sets or returns the cursor line style
CURSor:MODe	Sets or returns whether cursors move in unison or separately
CURSor:SCREEN:StYle	Sets or returns the cursor type for screen mode
CURSor:SOUrce<x>	Sets or returns the source for cursor <x>
CURSor:StATE	Turns cursors on or off or returns their state
CURSor:VBArS	Sets or returns the position of vertical bar cursors
CURSor:VBArS:DELTA?	Returns the difference between vbar cursors
CURSor:VBArS:POSITION<x>	Sets or returns the vbar cursor<x> horizontal position
CURSor:VBArS:UNIts	Sets or returns the units for vbar cursors
CURSor:WAVEform:HDELTA?	Returns the horizontal difference between waveform cursors
CURSor:WAVEform:HPOS<x>?	Returns the position of waveform cursor <x>
CURSor:WAVEform:POStion<x>	Sets or returns the position of waveform cursor <x>
CURSor:WAVEform:SOUrce<x>	Sets or returns the source for waveform cursor <x>
CURSor:WAVEform:StYle	Sets or returns the cursor type for waveform mode
CURSor:WAVEform:UNIts	Sets or returns the units for waveform cursors

CURSor:WAVEform:VDELTA?	Returns the vertical difference between waveform cursors
CURSor:WAVEform	Sets or returns the current settings for waveform cursors
CURSor:XY:PRODELta?	Returns the product of the difference between the cursors X positions and Y positions
CURSor:XY:PRODUCT <x>?	Returns the product of the X and Y positions for the specified cursor
CURSor:XY:RADIUS <x>	Returns the radius of the specified cursor
CURSor:XY:RATDELta?	Returns ratio of the difference between the cursors X position and Y position
CURSor:XY:RATIO <x>	Returns ratio of the X (horizontal) and Y (vertical) position for the specified cursor
CURSor:XY:RDELta?	Returns the $\Delta r$ value
CURSor:XY:READOUT	Sets or returns the XY cursor readout mode.
CURSor:XY:RECTX <x>	Sets or returns the X cursor position in rectangular coordinates
CURSor:XY:RECTY <x>	Sets or returns the Y cursor position in rectangular coordinates
CURSor:XY:THDELta?	Returns the XY cursor angle delta in polar coordinates
CURSor:XY:THETA <x>?	Returns the XY cursor angle in polar coordinates
CURSor:XY:XDELta?	Returns the XY cursor $\Delta X$ value in rectangular coordinates
CURSor:XY:YDELta?	Returns the XY cursor $\Delta Y$ value in rectangular coordinates
CURSor:XY?	Returns the current settings for XY cursors

## Diagnostic Command Group

The diagnostic commands control the selection and execution of diagnostic tests.

<b>Command</b>	<b>Description</b>
DIAG:CONTROL:HALT	Enables or disables halting on first diagnostic failure
DIAG:CONTROL:LOOP	Enables or disables looping of diagnostics
DIAG:EXECUTE	Executes currently selected set of diagnostics
DIAG:ITEM?	Returns all data associated with a selected menu item
DIAG:ITEM:FAILURES?	Returns the total number of failures that occurred
DIAG:ITEM:NAME?	Returns the name of the selected menu item
DIAG:ITEM:RESULT?	Returns the results of the last test executed on this item

---

DIAG:ITEM:SUBITEMS?	Returns the number of subitems associated with this item
DIAG:LEVEL	Sets the current level of diagnostic test hierarchy
DIAG:LOOPS?	Returns the number of times the diagnostics were completed during the last execution
DIAG:NAME?	Returns the subsystem name, area, and test name of the current diagnostic test
DIAG:NAME:AREA?	Returns the selected area of the current diagnostic test
DIAG:NAME:SUBSYS?	Returns the subsystem of the current diagnostic test
DIAG:NAME:TEST?	Returns the name of the current diagnostic test
DIAG:NUMITEMS?	Returns the number of items on the currently selected level of test hierarchy
DIAG:RESuLts?	Returns a brief pass or fail status of the last test execution
DIAG:RESuLts:VERBose?	Returns a more explanatory message about the results of the last diagnostic execution
DIAG:SElect:ALL	Selects all available diagnostics
DIAG:SElect:AREA	Selects one of the available diagnostic areas
DIAG:SElect:LAST	Sets the last item of a group of items from the same level of test hierarchy
DIAG:SElect:SUBSYS	Selects one of the available diagnostic subsystems
DIAG:SElect:TEST	Selects one of the available diagnostic tests
DIAG:STATE	Sets the instrument operating state
DIAG:STOP	Terminates the execution of diagnostics
TEST	Selects and executes any item at any level of the test hierarchy
TEST:RESuLts?	Returns a brief pass or fail status of the last test execution
TEST:RESuLts:VERBose?	Returns a more explanatory message about the results of the last test execution
TEST:STOP	Terminates the execution of the test



---

# Display Control Command Group

Use the commands in the Display Control Command Group to change the graticule style, the displayed intensities, and to set the characteristics of the waveform display.

You can set the display of date and time; cursor, histogram, mask, and measurement readouts; measurement annotations, and the mode in which waveforms are displayed.

There are six color palettes from which you can select:

- **Normal** displays hues and lightness levels for best overall viewing.
- **Temp** displays areas of the waveform with the highest sample density in warmer colors (red shades) while the areas of lowest sample density appear in cooler colors (blue shades).
- **Spectral** displays areas of the waveform with the highest sample density in blue shades while the areas of lowest sample density appear in red shades.
- **Green** displays waveforms in shades of green. Areas of the waveform with the highest sample density appear in lighter green shades while the areas of lowest sample density appear in darker green shades.
- **Gray** displays waveforms in shades of gray. Areas of the waveform with the highest sample density appear in lighter gray shades while the areas of lowest sample density appear in darker gray shades.
- **User** allows you to create a customized color palette.

Use the commands to set the style that best displays your waveforms and graticule display properties. Note that the mode you choose globally affects all displayed waveforms.

Command	Description
DISplay:CLOCK	Sets or returns the display of the date/time stamp
DISplay:COLOr:MATHCOLOr	Sets or returns the color to be used for math traces
DISplay:COLOr:PALEtte:IMAGView	Sets or returns the color palette for imageview waveforms
DISplay:COLOr:PALEtte:RECORDView	Sets or returns the color palette for recordview waveforms
DISplay:COLOr:PALEtte:USER:CARet	Sets or returns the user caret color
DISplay:COLOr:PALEtte:USER:CH<x>	Sets or returns the user palette channel colors
DISplay:COLOr:PALEtte:USER:GRAticule	Sets or returns the user palette graticule
DISplay:COLOr:PALEtte:USER:HISTogram	Sets or returns the user palette histogram color
DISplay:COLOr:PALEtte:USER:MASK	Sets or returns the user palette mask color

---

DISplay:COLOr:PALEtte:USEr: MASK:HighLight	Sets or returns the user palette mask hits color
DISplay:COLOr:PALEtte:USEr: MATH<x>	Sets or returns the user palette math colors
DISplay:COLOr:PALEtte:USEr: REF<x>	Sets or returns the user palette reference colors
DISplay:COLOr:PALEtte:USEr?	Returns the user palette group settings
DISplay:COLOr:REFCOLOr	Sets or returns the color to be used for reference traces
DISplay:COLOr?	Returns color group settings
DISplay:FILTer	Sets or returns the type of inter- polation to use for the display
DISplay:FORMat	Sets or returns the display format
DISplay:GRATICule	Sets or returns the type of grati- cule that is displayed
DISplay:INTENSITy:AUTOBright	Sets or returns whether automatic adjustment of waveform intensity is enabled
DISplay:INTENSITy:SCRENSAVER	Sets or returns automatic display protection features
DISplay:INTENSITy:SCRENSAVER DELAY	Sets or returns the display protec- tion timeout state
DISplay:INTENSITy:WAVEform: IMAGEView	Sets or returns the waveform sat- uration level for imageview wave- forms
DISplay:INTENSITy:WAVEform: RECORDView	Sets or returns the waveform sat- uration level for recordview wave- forms
DISplay:INTENSITy?	Returns the waveform saturation level, autobright state and screen- saver settings
DISplay:PERsistence	Sets or returns display persis- tence setting
DISplay:PERsistence:RESET	Sets or returns the clearing of per- sistence data
DISplay:SCREntExt:LABel<x>: NAME	Sets or returns the text to be dis- played for a given label
DISplay:SCREntExt:LABel<x>: XPOS	Sets or returns the horizontal posi- tion of a given label
DISplay:SCREntExt:LABel<x>: YPOS	Sets or returns the vertical posi- tion of a given label
DISplay:SCREntExt:LA- Bel<x>?	Return screen text setting for a given label
DISplay:SCREntExt:STATE	Sets or returns the state of the display of screen text
DISplay:SCREntExt?	Return all screen text settings
DISplay:SHOWREmote	Sets or returns the state of the re- mote display feature
DISplay:STYle	Sets or returns data display style
DISplay:TRIGBar	Sets or returns the display setting of the trigger level indicator bar(s)
DISplay:VARpersist	Sets or returns the persistence decay time
DISplay?	Returns current display settings

---

## E-mail Command Group

Commands in the E-mail group allow you to send e-mail to one or more designated recipients whenever a selected event, such as a trigger, mask test failure (Option SM only), or a limit test failure occurs in the instrument.

Using this feature, you do not have to continually monitor the instrument for the event.

When an event occurs, the instrument will send an SMTP mail message to one or more designated recipients through the specified mail server. The message can include any of the following:

- Screen image
- Waveform data
- Measurement data at the time of the event

**NOTE:** Your instrument must be connected to the Local Area Network (LAN) using Transfer C Protocol/Internet Protocol (TCP/IP). The recipient must also have an SMTP mail server. To send e-mail for a mask test failure, you must have Mask Testing (Option SM) installed.

<b>Command</b>	<b>Description</b>
EMail	Sends a test e-mail message or sets the current e-mail sent count to zero.
EMail:ATTEMpts	Sets or queries the number of times that an attempt will be made to send e-mail to the SMTP e-mail server
EMail:AUTHLogin	Sets or queries the login name that will be used if the SMTP e-mail server requires one for authentication
EMail:AUTHPassword	Sets the password that will be used if the SMTP e-mail server requires one for authentication
EMail:COUnt?	Returns the number of e-mails that have been sent since Email on Event was armed
EMail:FRom	Sets or queries the From line in the e-mail
EMail:HOSTwanted	Sets or queries the hostname that will be used when e-mail is sent to the SMTP e-mail server
EMail:IMage	Sets or queries whether image data is included in the e-mail
EMail:LIMit	Sets or queries whether e-mail is sent when a limit test failure occurs
EMail:MASk	Sets or queries whether e-mail is sent when a mask test failure occurs
EMail:MAXsize	Sets or queries the maximum size (in megabytes) of e-mail that can be sent to the SMTP server
EMail:MEASurement	Sets or queries whether measurement data is included in the e-mail
EMail:NUMemails	Sets or queries the number e-mails that can be sent when Email onEvent is armed

---

E <code>Mail:SMTPPort</code>	Sets or queries the SMTP port number that the e-mail server uses
E <code>Mail:SMTPServer</code>	Sets or queries the address of the SMTP mail server.
E <code>Mail:STATus?</code>	Queries the status of e-mail.
E <code>Mail:TIMEout</code>	Sets or queries the global timeout in seconds.
E <code>Mail:TO</code>	Sets or queries the address of the recipient(s) of an e-mail. Multiple addresses are separated with a semicolon (;)
E <code>Mail:TRigger</code>	Sets or queries whether e-mail is sent when a trigger occurs
E <code>Mail:WAVEform</code>	sets or queries whether waveform data is included in the e-mail.

## File System Command Group

Use the commands in the File System Command Group to help you use the built-in hard disk drive and floppy disk drive. You can use the commands to do the following:

- List the contents of the default directory
- Create and delete directories
- Create, copy, read, rename, or delete a file

When using these commands, keep the following points in mind:

- File arguments are always enclosed within double quotes:  
`"C:\MYDIR\TEK00001.SET"`
- File names follow the MSDOS format: `[DRIVE:][\PATH]filename`
- Path separators may be either forward slashes (/) or back slashes (\)

**NOTE:** Using back slash as a path separator may produce some unexpected results, depending on how your GPIB controller application treats escaped characters. Many applications recognize the sequence of back slash followed by an alphabetic character as an escaped character, and, as such, interpret that alphabetic character as a control character. For example, the sequence `"\n"` may be interpreted as a newline character; `"\t"` may be interpreted as a tab character. To ensure that this interpretation does not occur, you can use double back slashes. For example, `"C:\\testfile.txt"`.

- Some FILESystem commands may fail because a file has read-only attributes. You will not be able to delete or replace such files until this attribute is removed. Refer to the operating system help on file properties for further information.

---

<b>Command</b>	<b>Description</b>
FILESystem?	Returns the file system state
FILESystem:COpy	Copies one or more files to a new file
FILESystem:CWD	Sets or returns the current working directory for FILESystem GPIB commands.
FILESystem:DELEte	Deletes a named file or directory
FILESystem:DIR?	Returns a list of directory contents
FILESystem:MKDir	Makes a new directory
FILESystem:PRInt	Prints a named file to the named port
FILESystem:READFile	Copies the named file to the GPIB port
FILESystem:REName	Assigns a new name to an existing file
FILESystem:RMDir	Deletes the named directory
FILESystem:WRITEFile	Copies the GPIB port block data to a named file

## Hard Copy Command Group

Hard copy commands enable you to make hard copies of data files or send the data to a specified file. In addition, these commands provide you with information about (and the ability to set) file path settings.

<b>Command</b>	<b>Description</b>
EXPort	Copies a waveform to a specified file or returns formatting and file information
EXPort:FILEName	Sets or returns the export file path
EXPort:FORMat	Sets or returns the export image format
EXPort:IMAGe	Sets or returns the export inksaver mode
EXPort:PALEtte	Sets or returns the export color palette
EXPort:VIEW	Sets or returns the export screen capture area
HARDCopy	Sends a screen copy to the selected port or returns the selected port and file path
HARDCopy:FILEName	Sets or returns the hardcopy file path
HARDCopy:IMAGe	Sets or returns the inksaver mode for hardcopy printing
HARDCopy:LAYout	Sets or returns the page orientation for hardcopy
HARDCopy:PALEtte	Sets or returns the hardcopy color palette
HARDCopy:PORT	Sets or returns whether the data will be sent to a file or printed on the next hardcopy
HARDCopy:VIEW	Sets or returns the area of the screen to be hard-copied

---

# Histogram Command Group

Histogram commands let you select the type of histogram, what part of the waveform should go into the histogram, and histogram statistics. Use commands from this group to do the following:

- Create a histogram of vertical or horizontal values for a channel, math, or reference waveform
- Adjust the limits of the waveform area from which the histogram data is obtained
- Clear histogram count and restart
- Retrieve the histogram data

<b>Command</b>	<b>Description</b>
HIStogram?	Return all histogram parameters
HIStogram:BOX	Sets or returns the left, top, right, and bottom positions of the histogram box, in source waveform coordinates
HIStogram:BOXPcnt	Sets or returns same as HIStogram:BOX, but in percentage coordinates, with 0,0 upper left and 100,100 lower right
HIStogram:COUNT	Clears histogram count source data and restarts counting
HIStogram:DATA?	Returns a comma-separated list of histogram data numbers
HIStogram:DISplay	Sets or returns whether histogram data is displayed and the current display type setting
HIStogram:FUNCTion	Sets the type of histogram to create Or returns the current histogram type
HIStogram:MODE	Sets type of histogram to create or turns histograms off Or returns the current histogram type or that histogram display is disabled
HIStogram:SIZE	Sets or returns the width (or height) of the histogram on the screen in divisions
HIStogram:SOURce	Sets or returns which source waveform will be compared against the histogram box when the histogram testing is enabled
HIStogram:STATE	Sets or returns whether histogram calculations are enabled

---

# Horizontal Command Group

Horizontal commands control the time bases of the instrument. You can set the time per division (or time per point) of the main time base. You

- can use the Horizontal commands to do the following:
  - Set the scale, horizontal position and reference, and units of the time base
  - Get the screen resolution, time of first point and time of last point, or get all the horizontal settings
  - Enable or disable the display of the time base

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

Command	Description
HORizontal:DIVisions?	Returns the number of graticule divisions over which the waveform is displayed
HORizontal:FASTframe:COUNT	Sets or returns FastFrame frame count
HORizontal:FASTframe:LENGth	Sets or returns the horizontal record length to the number of sample points per frame
HORizontal:FASTframe:MUltipleframes:FRAMEStart: <wfm>	Sets or returns the start frame number on the specified waveform for the FastFrames' multiple frames feature
HORizontal:FASTframe:MUltipleframes:MODE	Sets or returns the multiple frames mode for the FastFrames' multiple frames feature
HORizontal:FASTframe:MUltipleframes:NUMFRames: <wfm>	Sets or returns the number of frames on the specified waveform for the FastFrames' multiple frames feature
HORizontal:FASTframe:REF:FRAME	Sets or returns the FastFrame reference frame number
HORizontal:FASTframe:REF:SOURce	Sets or returns the FastFrame reference waveform source
HORizontal:FASTframe:SELECTED: <wfm>	Sets or returns the FastFrame selected frame number on the specified waveform
HORizontal:FASTframe:STATE	Sets or returns the state of the FastFrame acquisition
HORizontal:FASTframe:SUMFrame	This command sets or queries the summary frame mode.
HORizontal:FASTframe:TIMEStamp: ALL:<wfm>?	Returns the frame number and time stamp for each frame between requested frames within the specified waveform
HORizontal:FASTframe:TIMEStamp: BETWeen:<wfm>?	Returns the relative trigger for the delta time between the specified frames

HORizontal:FASTframe:TIMESamp:DELTA:<wfm>?	Returns the relative time between the triggers of the FastFrame Selected and the FastFrame Reference, within the specified waveform
HORizontal:FASTframe:TIMESamp:FRAME:<wfm>?	Returns the absolute trigger date and time for the specified frame and waveform
HORizontal:FASTframe:TIMESamp:REF?	Returns the absolute trigger date and time for FastFrame reference
HORizontal:FASTframe:TIMESamp:SELECTED:<wfm>?	This query-only command returns the absolute trigger date and time for the FastFrame Selected, within the specified waveform
HORizontal:FASTframe:TRACK	Sets or returns the state of the FastFrame tracking feature
HORizontal:FASTframe:XZEro:ALL:CH<x>?	Returns the frame number and xzero for each frame between requested frames within the specified channel waveform
HORizontal:FASTframe:XZEro:ALL:REF<x>?	Returns the frame number and xzero for each frame between requested frames within the specified reference waveform
HORizontal:FASTframe:XZEro:FRAME:CH<x>?	Returns xzero for the specified frame of the specified channel waveform
HORizontal:FASTframe:XZEro:FRAME:REF<x>?	Returns xzero for the specified frame of the specified reference waveform
HORizontal:FASTframe:XZEro:REF<x>?	Returns xzero of the first frame of the immediate FastFrame reference waveform
HORizontal:FASTframe:XZEro:SELECTED:CH<x>?	Returns xzero for FastFrame selected within the specified channel waveform
HORizontal:FASTframe:XZEro:SELECTED:REF<x>?	Returns xzero for FastFrame selected within the specified reference waveform
HORizontal:FASTframe?	Returns the horizontal FastFrame settings
HORizontal[:MAIN]:DELay:MODE	Sets or returns the main time base trigger delay mode
HORizontal[:MAIN]:DELay:POSITION	Sets or returns the main time base position when Horizontal Delay Mode is turned off
HORizontal[:MAIN]:DELay:TIME	Sets or returns the main time base trigger delay time
HORizontal:MAIN:INTERPRatio	Returns the main horizontal time base interpolation ratio
HORizontal[:MAIN]:POSITION	Sets or returns the waveform horizontal position on the display
HORizontal:MAIN:SAMPLERate	Sets the horizontal sample rate to the desired number of samples per second  Or returns the current horizontal sample rate



---

HORizontal[:MAIn]:SCAlE	Sets time per division for the main time base Or returns the main time base horizontal scale
HORizontal:MAIn:UNIts	Sets or returns the units for the horizontal main time base
HORizontal:MAIn:UNIts:STRing	Sets or returns the units string for the horizontal main time base trigger delay
HORizontal:MAIn?	Returns the time per division of the main time base
HORizontal:POSition	Sets or returns the waveform's horizontal position on the display
HORizontal:RECOrdlength	Sets the horizontal record length to the number of data points in each frame Or returns the current horizontal record length
HORizontal:RESOlution	Sets the horizontal record length to the number of data points in each frame and simultaneously adjusts the sample rate to maintain a constant time/division Or returns the current horizontal record length
HORizontal:ROLL	Sets or returns the horizontal roll mode status
HORizontal:SCAlE	Sets or returns the time per division for the main time base
HORizontal:TRIGger:POSition	Sets or returns the position of the horizontal trigger
HORizontal?	Returns all learnable settings for the horizontal commands

## Limit Test Command Group

Use the commands in the Limit Test Command Group to perform limit testing by selecting channel, math, or reference waveforms and comparing them against a template waveform.

Using the limit commands, you can:

- Select a source waveform to compare against the selected template during a limit test. The source waveform can be a Ch1–Ch4 waveform, a Math1–Math4 waveform, or a Ref1–Ref4 waveform. (For two-channel instruments, the choices are limited to Ch1–Ch2, Math1–Math2, or Ref1–Ref2 waveforms.)
- Select a template from the following choices: None and Ref1–Ref4 (only Ref1–Ref2 are available on two-channel instruments).
- Set the limits around a template waveform.
- Reset the instrument to the original limit test default settings. The default setting for Source is Channel 1 and for Template is None.

- 
- Set up test failure notification, lock the template to a waveform, and highlight hits.
  - Sound a beep when a limit test fails.
  - Send a screen print to the default printer when a limit test fails.
  - Save waveform data to either a default directory or to a directory of your choice when a limit test fails.
  - Record the date and time for each failed limit test in a date log file.
  - Send an e-mail to specified recipients when a limit test fails.
  - Highlight hits so that any points on the source waveform that fall outside of the template are highlighted in a different color from the rest of the waveform.
  - Reset the highlighted hits.

<b>Command</b>	<b>Description</b>
LIMit:BEEP	Sounds beep when limit exceeded
LIMit:COMpare	Resets the Limit test comparison template
LIMit:COMpare:CH<x>	Sets or returns the waveform template to which the specified waveform is compared
LIMit:COMpare:MATH<x>	Sets or returns the waveform template to which the specified math waveform is compared
LIMit:COMpare:REF<x>	Sets or returns the waveform template to which the specified reference waveform is compared
LIMit:EMail	Sets or queries whether an e-mail is generated when the source waveform data exceeds test limits
LIMit:HARDCopy	Sets or returns whether a hardcopy operation is executed on the waveform when any waveform data exceeds the limit set in the limit test
LIMit:HIGHLIGHTHits	Sets or queries whether violation highlighting occurs when limit testing is active; also, clears any highlighted hits in the mask display
LIMit:LOCK	Sets or queries whether vertical scaling and positioning affect both the source and template for template comparison pairs
LIMit:LOG	Sets or queries whether a log file is saved when the source waveform data exceeds the test limits
LIMit:SAVEWFM	Sets or queries whether the source waveform is saved when the source waveform data exceeds the test limits
LIMit:SAVEWFM:FILEName	Sets or queries the path where waveforms or log files will be saved when the waveform data exceeds the limits set by the limit test
LIMit:SRQ	Sets or queries whether an SRQ is sent when the waveform data exceeds the test limits
LIMit:STATE	Sets or returns whether limit testing is on or off
LIMit:STOPOnviolation	Sets or queries whether acquisitions are stopped when the waveform data exceeds the test limits
LIMit:TEMPlate:STORe	Saves the specified source waveform to a specified file or reference
LIMit:TEMPlate:TOLerance:HORizontal	Sets or queries the amount, in units of horizontal divisions, by which the source waveform is varied horizontally when creating the destination waveform

---

LIMit:TEMPLate:TOLer-  
ance:VERTical

Sets or queries the amount, in units of vertical divisions, by which the source waveform is varied vertically when creating the destination waveform

Limit?

Returns all settings for the Limit commands

## Mask Command Group

Use the commands in the Mask Command Group to compare incoming waveforms against standard or user-defined telecommunications masks. Standard masks are only available when option SM is installed. Without option SM, you need to define your masks. When the instrument detects waveforms that fall inside or outside of the mask limits, it can perform tasks such as generating hard copy, saving the waveform, emitting a beep or an SRQ, stopping and waiting for your input, sending an email message with specified data, logging the time/date of the acquired waveform, or some combination of these actions.

A mask is a set of polygonal regions on the screen. Every vertical line of the screen intersects the polygon in zero, one, or two places, but never in more than two places. (A vertical line that intersects a vertical mask border is counted.) You have to break up more complicated polygons into two separate segments. Unlike limit testing, the inside of a mask is the region where waveform data would not normally fall. A telecommunications standard requires up to eight of these masks. Pulse standards always have two masks. Standards with eye patterns usually have three masks, but some have four.

You use the commands in the Mask Command Group to do the following:

- Specify the waveform source to test and the mask to use
- Specify whether to use, and the size of, mask margins, which allow you to shrink or expand an existing set of polygons by a specified percentage.
- Specify whether to display a readout of hits and the mask on screen. Options also exist for auto fitting the incoming waveforms to match the mask you choose.
- Select industry-standard masks that support a variety of electrical and optical communication standards (available only with option SM).
- Define and edit your own custom mask; create an entirely new mask, or use a standard mask (option SM only) as a starting reference, and edit it to meet your needs.
- Enable, disable, or reset the mask counts. Once you turn on mask counting, it remains on until you explicitly turn it off.

Option SM also provides communication trigger commands that you can use in conjunction with the mask commands, such as TRIGGER:A:TYPE Communication. Refer to the Trigger Command Group and

<b>Command</b>	<b>Description</b>
MASK?	Returns the states of all settable mask parameters
MASK:AUTOAdjust	Optimizes signal position within the mask to minimize hits
MASK:AUTOAdjust:HDEL-TA	Sets or returns how far autofit searches horizontally
MASK:AUTOAdjust:VDEL-TA	Sets or returns how far autofit searches vertically
MASK:AUTOSet:AUTOAdjust	Controls what happens at the end of a mask autose. If ON, the horz/vert auto adjustment is run ONCE to potentially improve the result of mask autose
MASK:AUTOSet:HPOS	Controls whether the autose algorithm will attempt to change the horizontal position of the signal while attempting to autose
MASK:AUTOSet:HSCAle	Controls whether the autose algorithm will attempt to change the horizontal scale while attempting to autose
MASK:AUTOSet:MODE	Controls whether or not an autose will be automatically done after a standard mask is selected
MASK:AUTOSet:OFFSE-TAdj	Sets autose to not enforce the rule that, for pulse standards, requires 0V to be in a certain place in the mask. Instead, autose will attempt to measure the DC offset in the signal and use scope offset or position controls to compensate for the DC offset in the signal
MASK:AUTOSet:STANdard	Selects the standard mask in autose
MASK:AUTOSet:TRIGger	Controls if a MASK:AUTOSet affects the trigger level
MASK:AUTOSet:VPOS	Controls whether the autose algorithm will affect the vertical position (or offset) of the signal while attempting to autose
MASK:AUTOSet:VSCAle	Controls whether the autose algorithm will attempt to change the vertical scale while attempting to autose
MASK:COPy:USER	Copies the currently displayed mask to the User mask
MASK:COUNt:FAILURES?	Returns the number of pass/fail tests that have failed
MASK:COUNt:HITS?	Returns the total number of mask hits on all mask segments
MASK:COUNt RESET	Resets to zero the number of hits for all mask segments on the specified source
MASK:COUNt:SEG<m>:HITS?	Returns the number of hits in mask segment <m>
MASK:COUNt:STATE	Sets or returns the mask hits count state
MASK:COUNt:TESTS?	Returns the number of pass/fail tests that have occurred

---

MASK:COUNt:TOTal?	Returns the sum of all hits in all mask segments
MASK:COUNt:VIOLA-TIONS?	Returns the number of test violations that have occurred in the current pass/fail test
MASK:COUNt:WAVE-FORMS?	Returns the number of waveforms that have been acquired and processed during pass/fail testing
MASK:DISPlay	Sets or returns whether or not defined masks are displayed on the screen
MASK:FILTer	Sets or returns whether the selected optical reference receiver mask filter is enabled
MASK:HIGHLIGHtHits	Sets or returns whether or not hits in a mask are highlighted in different colors than other waveform data
MASK:INVert	Controls whether the mask is drawn inverted
MASK:LOCK	Sets or returns the mask lock to waveform mode
MASK:MARgin:PERCent	Sets or queries the mask margin percentage
MASK:MARgin:STATE	Sets or returns the state of the mask margins
MASK:MASKPRE:AMPLitude	Sets or returns the current mask's nominal pulse amplitude in volts.
MASK:MASKPRE:HSCALE	Sets or returns the nominal timing resolution used to draw the current mask, in time/division
MASK:MASKPRE:HTRIG-POS	Sets or returns the nominal tripper position.(pulse leading edge) used to draw the current mask as a fraction of the display width
MASK:MASKPRE:PAT-TERNBITS	Sets or returns the number of bits used for serial trigger for the current mask standard
MASK:MASKPRE:PRE-AMPBITS	Sets or returns the number of bits before the (isolated one) pulse leading edge in the serial trigger pass/fail testing
MASK:MASKPRE:RE-CORdlength	Sets or returns the nominal record length for pulse mask testing
MASK:MASKPRE:SERIAL-TRIG	Sets or returns the type of triggering used in pass/fail testing of the current mask
MASK:MASKPRE:TRIG-TOSAMP	Sets or returns the time, in seconds, from the (leading edge) trigger position to the pulse bit sampling position
MASK:MASKPRE:VOFF-Set	Sets or returns the nominal vertical offset, in volts, used to vertically offset the input channels
MASK:MASKPRE:VPOS	Sets or returns the nominal vertical position, in divisions, used to vertically position the input channels
MASK:MASKPRE:VSCALE	Sets or returns the nominal vertical scale, in volts per division, used to vertically scale the input channels

---

MASK:MASKPRE:WIDth	Sets or returns the nominal bit width in seconds
MASK:POLarity	Sets or returns the input waveform polarity for the pass/fail test
MASK:SEG<m> DELEte	Deletes the specified mask segment from the Current mask
MASK:SEG<m>:NR_Pt?	Returns the number of points that make up the specified mask segment of the Current mask
MASK:SEG<m>:POINTS	Sets or returns the X-Y user coordinates of all points in the Current mask segment
MASK:SOUrce	Sets or reports which source will be compared against the mask(s) when counting is turned on
MASK:STANdard	Deletes the existing mask and sets the selected standard mask
MASK:STOPOnviolation	Sets or returns the stop on violation setting
MASK:TEST:AUX:COMPLetion	Sets or returns the test:aux on pass/fail test completion mode
MASK:TEST:AUX:FAILure	Sets or returns the test:aux status on pass/fail test failure mode
MASK:TEST:BEEP:COMPLetion	Sets or returns the beep on pass/fail test completion mode
MASK:TEST:BEEP:FAILure	Sets or returns the beep status on pass/fail test failure mode
MASK:TEST:DELay	Sets or returns the amount of time, in seconds, to wait after starting pass/fail testing to evaluate waveforms
MASK:TEST:HARDCopy	Sets or returns the hard copy in pass/fail failure mode
MASK:TEST:LOG:FAILure	Sets or returns the log status on pass/fail test failure mode
MASK:TEST:REPeat	Sets or returns the repeat pass/fail test on completion mode
MASK:TEST:SAMple	Sets or returns the minimum number of waveform database (WfmDB) points the oscilloscope can acquire before it stops a single sequence acquisition or stops running a mask test
MASK:TEST:SAM-ple:THReshold	Sets or returns the minimum number of hits in mask regions needed to cause the pass/fail status to change from PASSING to FAILING
MASK:TEST:SAVEWFM	Sets or returns the save waveform to file on failure setting
MASK:TEST:SA-VEWFM:FILEName	Sets or returns the name of the directory and file to use with the MASK:TEST:SAVEWFM command
MASK:TEST:SRQ:COMPLetion	Sets or returns an SRQ on pass/fail test completion mode
MASK:TEST:SRQ:FAILure	Sets or returns the SRQ status on pass/fail test failure mode
MASK:TEST:STATE	Sets or returns the state of mask pass/fail testing

---

MASK:TEST:STATUS?	Returns the pass/fail test status
MASK:TEST:STOP:FAILure	Sets or returns the stop status on pass/fail test failure mode
MASK:TEST:THReshold	Sets or returns the number of failed tested waveforms needed in each pass/fail test to cause the pass/fail status to change from PASSING to FAILING
MASK:TEST:WAVEform	Sets or returns the number of waveforms to test during pass/fail testing
MASK:USER:AMPlitude	Sets or returns the current mask's nominal pulse amplitude in volts
MASK:USER:BITRate	Sets or returns the user mask's bit rate
MASK:USER:HSCAle	Sets or returns the nominal timing resolution used to draw the mask, in time/division
MASK:USER:HTRIGPOS	Sets or returns the nominal tripper position.(pulse leading edge) used to draw the mask as a fraction of the display width
MASK:USER:LABel	Sets or returns the label assigned to a user mask
MASK:USER:PATTERN-BITS	Sets or returns the number of bits used for serial trigger for the User mask standard
MASK:USER:PRESAMP-BITS	Sets or returns the number of bits before the (isolated one) pulse leading edge in the serial trigger pass/fail testing
MASK:USER:RECOrd-length	Sets or returns the nominal record length for pulse mask testing
MASK:USER:SEG<m> DELETE	Deletes the specified mask segment from the User mask
MASK:USER:SEG<m>:NR_Pt?	Returns the number of points that make up the specified User mask segment
MASK:USER:SEG<m>:POINTS	Sets or returns the X-Y user coordinates of all points in the specified User mask segment
MASK:USER:TRIGTO-SAMP	Sets or returns the time, in seconds, from the (leading edge) trigger position to the pulse bit sampling position
MASK:USER:VOFFSet	Sets or returns the nominal vertical offset, in volts, used to vertically offset the input channels
MASK:USER:VPOS	Sets or returns the nominal vertical position, in divisions, used to vertically position the input channels
MASK:USER:VSCAle	Sets or returns the nominal vertical scale, in volts per division, used to vertically scale the input channels
MASK:USER:WIDth	Sets or returns the nominal bit width in seconds
RECAI:MASK	Recalls a stored mask as the User mask.
SAVe:MASK	Saves the User mask to a mass storage file



---

# Math Command Group

Use the commands in the Math Command Group to create and define math waveforms. You can define and display up to four math waveforms simultaneously on four channel models and up to two on two channel models. Use the available math functions to define your math waveform.

The math waveform you create depends on sources listed in the math expression. If you change these sources, the math waveforms you previously defined will be affected.

Math expressions can be simple, containing no mathematical computation, such as CH1, which specifies that a waveform shows the signal source of channel 1. Math expressions can also be complex, consisting of 100 plus characters and comprising many sources, functions, and operands.

The acquisition of a live waveform can stop for several reasons: You can turn off the channel, stop the waveform (via Run/Stop from the Horiz/Acq menu), or stop the trigger (via Run/Stop from the Trig menu). When you turn off the channel, math continues and data is acquired but is not displayed. When you stop either the waveform or the trigger, the math calculation stops, and the last math calculation performed is displayed.

When a live waveform update or reference waveform is altered, math waveforms containing those waveforms as sources are also updated to reflect the changes. Also, sources must exist but do not need to be displayed to be used in and to update math waveforms.

Command	Description
MATH<x>?	Returns the specified math waveform settings
MATH<x>:DEFine	Sets or returns the math<x> waveform definition for the specified waveform
MATH<x>:LABel:NAME	Sets or returns the label string used for annotating the displayed math waveform
MATH<x>:LABel:XPOS	Sets or returns the X screen offset where the math waveform label is displayed
MATH<x>:LABel:YPOS	Sets or returns the Y screen offset where the math waveform label is displayed
MATH<x>:NUMAVg	Sets or returns the acquisition number at which the averaging algorithm will begin exponential averaging
MATH<x>:SPECTral?	Returns the specified math waveform spectral setups
MATH<x>:SPECTral:CENTER	Sets or returns the center frequency of the spectral analyzer output data span
MATH<x>:SPECTral:GATEPOS	Sets or returns the position of the center of the gate
MATH<x>:SPECTral:GATEWIDTH	Sets or returns the gate width input to the spectral analyzer
MATH<x>:SPECTral:LOCK	Sets or returns the state of the spectral locking for the specified math waveform

---

MATH<x>:SPECTral:MAG	Sets or returns the units of the SpectralMag function in the specified math string
MATH<x>:SPECTral:PHASE	Sets or returns the units of the SpectralPhase function in the specified math string
MATH<x>:SPECTral:REFLevel	Sets or returns the vertical position of the output data from the spectral analyzer
MATH<x>:SPECTral:REFLEVELOffset	Sets or returns the spectral reference level offset used for calculating the dB value
MATH<x>:SPECTral:RESBw	Sets or returns the resolution bandwidth of the spectral analyzer
MATH<x>:SPECTral:SPAN	Sets the top of the span to a value that is closest to the specified value Or returns the specified math waveform's current span value
MATH<x>:SPECTral:SUPPress	Sets or returns the phase suppression threshold
MATH<x>:SPECTral:UNWRap	Enables or disables or returns the state of the phase unwrap of the spectral analyzer output data
MATH<x>:SPECTral:WINDow	Sets or returns the window function used to multiply the input data to the spectral analyzer
MATH<x>:UNITString	Sets or returns the units used for the math waveform; command overrides the default unit string
MATH<x>:VERTical:POSition	Sets or returns the specified math waveform vertical position
MATH<x>:VERTical:SCALE	Sets or returns the specified math waveform vertical scale (per div)
MATHVAR:VAR<x>	Sets or returns numerical values you can use within math expressions

## Measurement Command Group

Use the commands in the Measurement Command Group to control the automated measurement system.

Up to eight automated measurements can be displayed on the screen. In the commands, these measurement readouts are named MEAS<x>, where <x> can be 1 through 8.

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

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

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

<b>Command</b>	<b>Description</b>
MEASUrement:GATing	Sets or returns the measurement gating
MEASUrement:IMMed?	Returns all immediate measurement setup parameters
MEASUrement:IMMed:DElay?	Returns information about the immediate delay measurement
MEASUrement:IMMed:DElay:DIREction	Sets or returns the search direction to use for immediate delay measurements
MEASUrement:IMMed:DElay:EDGE[1]	Sets or returns the slope of the edge used for immediate delay "from" waveform measurements
MEASUrement:IMMed:DElay:EDGE2	Sets or returns the slope of the edge used for immediate delay "to" waveform measurements
MEASUrement:IMMed:NOISe	Sets or returns the location of the noise measurement on the eye waveform
MEASUrement:IMMed:REFLevel:ABSolute:HIGH	Sets or returns the high reference level
MEASUrement:IMMed:REFLevel:ABSolute:LOW	Sets or returns the low reference level
MEASUrement:IMMed:REFLevel:ABSolute:MID[1]	Sets or returns the middle reference level
MEASUrement:IMMed:REFLevel:ABSolute:MID2	Sets or returns the middle reference level for delay measurements
MEASUrement:IMMed:REFLevel:METhod	Sets or returns the method used to calculate measurements
MEASUrement:IMMed:REFLevel:PERCent:HIGH	Sets or returns the top level for rise time (90% level)
MEASUrement:IMMed:REFLevel:PERCent:LOW	Sets or returns the low level for rise time (10% level)
MEASUrement:IMMed:REFLevel:PERCent:MID[1]	Sets or returns the mid level for measurements
MEASUrement:IMMed:REFLevel:PERCent:MID2	Sets or returns the mid level for delay measurements
MEASUrement:IMMed:REFLevel?	Returns the reference levels
MEASUrement:IMMed:SOURCE[1]	Sets or returns the "from" source for all single channel immediate measurements
MEASUrement:IMMed:SOURCE2	Sets or returns the source to measure "to" for phase or delay immediate measurements
MEASUrement:IMMed:SOURce<x>:SIGType	Sets or returns the type of input signal used for the specified measurement source
MEASUrement:IMMed:TYPe	Sets or returns the type of the immediate measurement
MEASUrement:IMMed:UNIts?	Returns the units of the immediate measurement
MEASUrement:IMMed:VALue?	Returns the value of the immediate measurement

MEASUrement:MEAS<x>:COUNT?	Returns the number of values accumulated since the last statistical reset
MEASUrement:MEAS<x>:DELaY:DIRection	Sets or returns the search direction to use for delay measurements
MEASUrement:MEAS<x>:DELaY:EDGE[1]	Sets or returns the slope of the edge to use for delay "from" waveform measurements
MEASUrement:MEAS<x>:DELaY:EDGE2	Sets or returns the slope of the edge to use for delay "to" waveform measurements
MEASUrement:MEAS<x>:DELaY?	Returns the delay measurement parameters for specified measurement
MEASUrement:MEAS<x>:MAXimum?	Returns the maximum value found since the last statistical reset
MEASUrement:MEAS<x>:MEAN?	Returns the mean value accumulated since the last statistical reset
MEASUrement:MEAS<x>:MINImum?	Returns the minimum value found since the last statistical reset
MEASUrement:MEAS<x>:NOISe	Sets or returns the location of the noise measurement on the eye waveform for MEAS<x>
MEASUrement:MEAS<x>:REFLevel:ABSolute:HIGH	Sets or returns the top level for rise time (90% level)
MEASUrement:MEAS<x>:REFLevel:ABSolute:LOW	Sets or returns the low level for rise time (10%)
MEASUrement:MEAS<x>:REFLevel:ABSolute:MID[1]	Sets or returns the mid level for measurements
MEASUrement:MEAS<x>:REFLevel:ABSolute:MID2	Sets or returns the mid level for delay measurements
MEASUrement:MEAS<x>:REFLevel:METhod	Sets or returns the HIGH and LOW levels to either % or units
MEASUrement:MEAS<x>:REFLevel:PERCent:HIGH	Sets or returns the top level for rise time (90% level)
MEASUrement:MEAS<x>:REFLevel:PERCent:LOW	Sets or returns the low level for rise time (10% level)
MEASUrement:MEAS<x>:REFLevel:PERCent:MID[1]	Sets or returns the mid level for measurements
MEASUrement:MEAS<x>:REFLevel:PERCent:MID2	Sets or returns the mid level for delay measurements
MEASUrement:MEAS<x>:REFLevel?	Returns the reference levels
MEASUrement:MEAS<x>:SOURCE[1]	Sets or returns the channel from which measurements are taken
MEASUrement:MEAS<x>:SOURCE[2]	Sets or returns the channel to which measurements are sent
MEASUrement:MEAS<x>:SOUrce<x>:SIGType	Sets or returns the type of input signal used for the specified measurement source for the specified measurement slot
MEASUrement:MEAS<x>:STATE	Sets or returns whether the specified measurement slot is computed and displayed

---

MEASUrement:MEAS<x>:STDdev?	Returns the standard deviation of values accumulated since the last statistical reset
MEASUrement:MEAS<x>:TYPe	Sets or returns the measurement<x> type
MEASUrement:MEAS<x>:UNIts?	Returns measurement<x> units
MEASUrement:MEAS<x>:VALue?	Returns the value of measurement<x>
MEASUrement:MEAS<x>?	Returns all measurement parameters for the displayed measurement
MEASUrement:METhod	Sets or returns the method used for calculating reference levels
MEASUrement:NOISe	Sets or returns the location of the noise measurement on the eye waveform
MEASUrement:REFLevel:ABSolute:HIGH	Sets or returns the top reference level for rise time
MEASUrement:REFLevel:ABSolute:LOW	Sets or returns the low reference level for rise time
MEASUrement:REFLevel:ABSolute:MID[1]	Sets or returns the mid reference level for measurements
MEASUrement:REFLevel:ABSolute:MID2	Sets or returns the mid reference level for delay "to" measurements
MEASUrement:REFLevel:METhod	Sets or returns the method for assigning high and low reference levels
MEASUrement:REFLevel:PERCent:HIGH	Sets or returns the top reference percent level for rise time
MEASUrement:REFLevel:PERCent:LOW	Sets or returns the low reference percent level for rise time
MEASUrement:REFLevel:PERCent:MID[1]	Sets or returns the mid reference percent level for waveform measurements
MEASUrement:REFLevel:PERCent:MID2	Sets or returns the mid reference percent level for second waveform measurements
MEASUrement:REFLevel?	Returns reference level parameters
MEASUrement:SOURCE<x>:SIGType	Sets or returns the type of input signal used for the specified measurement source
MEASUrement:STATIstics:COUNt	Clears existing measurement statistics from memory
MEASUrement:STATIstics:MODE	Turns management statistics on or off and sets or returns which pair of statistical accumulations is displayed
MEASUrement:STATIstics:WEIghting	Sets or returns the 'time constant' for mean and standard deviation statistical accumulations
MEASUrement?	Returns all measurement parameters

---

# Miscellaneous Command Group

Miscellaneous commands do not fit into other categories.

Several commands and queries are common to all 488.2–1987 devices on the GPIB bus. The 488.2–1987 standard defines these commands. The common commands begin with an asterisk (\*) character.

<b>Command</b>	<b>Description</b>
*DDT	Sets or returns the commands that will be executed by the group execute trigger
*IDN?	Returns the instrument identification code
*LRN?	Returns a listing of instrument settings
*TRG	Performs the group execute trigger (GET)
*TST?	Tests the GPIB interface and returns status
AUTOSet	Sets the vertical, horizontal and trigger controls to provide a stable display of the selected waveform
AUXout?	Returns the auxiliary out setup
AUXout:SOUrce	Sets or returns the trigger source at the BNC connection
BELI	In previous TDS models, this command sounded the audio indicator. Provided for backward compatibility.
CMDBatch	Turns command batching on or off Or returns the state of command batching
DATE	Sets or returns the date that the instrument can display
HDR	Sets or returns the Response Header Enable State
HEADer	Sets or returns the Response Header Enable State
ID?	Returns identifying information about the instrument and its firmware
LOCK	Sets or returns the front panel lock state
NEWpass	Changes the password for user protected data
PASSWord	Provides access for changing user protected data
REM	Specifies a comment, which is ignored by the instrument
ROSc:SOUrce	Sets or returns the source for the 10 MHz time base
ROSc:STATE?	Returns the state of the 10 MHz time base
SET?	Returns a listing of instrument settings
TEKSecure	Initializes both waveform and setup memories
TIME	Sets or returns the time displayed by the instrument
UNLock	Unlocks front panel
VERBose	Sets or returns the verbose state

---

## Save and Recall Command Group

Use the commands in the Save and Recall Command Group to store and retrieve internal waveforms and settings. When you save a setup, you save all the settings of the instrument. When you recall a setup, the instrument restores itself to the state that it was in when you originally saved that setting.

<b>Command</b>	<b>Description</b>
*RCL	Restores the state of the instrument from a copy of the setting stored in memory
*SAV	Stores the current instrument state to a specified memory location
*SDS	Changes the specified setup to reference the factory setup instead of the user setup
DELEte:SETUp	Removes stored setups from memory and initializes the location with the factory default setups
DELEte:WAVEform	Deletes (one or all of) the stored reference waveforms from memory
FACTory	Resets the instrument to factory default settings
RECALL:SETUp	Recalls saved instrument settings
RECALL:WAVEform	Recalls a stored waveform to a reference location
SAVE:SETUp	Saves the current front-panel setup to a specified memory location or file
SAVE:WAVEform	Saves a waveform to one of four reference memory locations or a mass storage file
SAVE:WAVEform:FILEFormat	Sets or returns the format for saved waveforms
SETUp:NAME	Sets or returns the user-defined setup label

---

# Status and Error Command Group

Use the commands in the Status and Error command Group to determine the status of the instrument and control events.

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

<b>Com- mand</b>	<b>Description</b>
*CLS	Clears status
*ESE	Sets or returns the bits in the Event Status Enable Register
*ESR?	Returns the contents of the Standard Event Status Register
*OPC	Generates the operation complete message in the standard event status register when all pending operations are finished  Or returns "1" when all current operations are finished
*OPT?	Returns a list of options installed in the instrument
*PSC	Sets or returns the power on status flag
*PUD	Sets or returns a string of protected user data
*RST	Resets the instrument to factory default settings
*SRE	Sets or returns the bits in the Service Request Enable Register
*STB?	Returns the contents of the Status Byte Register
*WAI	Prevents the instrument from executing further commands until all pending operations finish
ALLEv?	Returns all events and their messages
BUSY?	Returns instrument status
DESE	Sets or returns the bits in the Device Event Status Enable Register
EVENT?	Returns event code from the event queue
EVMsg?	Returns event code, message from the event queue
EVQty?	Return number of events in the event queue



---

# Trigger Command Group

Use the commands in the Trigger Command Group to control all aspects of triggering for the instrument.

There are two triggers: A and B. Where appropriate, the command set has parallel constructions for each trigger.

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

You can also set A triggers to pulse, logic or video modes. With pulse triggering, the instrument triggers whenever it detects a pulse of a certain width or height. Logic triggering lets you logically combine the signals on one or more channels. The instrument then triggers when it detects a certain combination of signal levels. Video triggering enables you to trigger on the most common Standard Definition, High Definition, and custom video standards. You can trigger on all fields, all lines, or individual video lines.

If you have Option SM (Mask Testing) installed, you also have access to a communication mode. When this mode is selected, several communication trigger commands are available that work in conjunction with the mask commands. Refer to Mask Command Group and

Command	Description
TRIGger	Forces a trigger event to occur or returns current trigger parameters for the instrument
TRIGger:A	Sets A trigger level to 50% or returns current A trigger parameters
TRIGger:A:COMMunication:AMI: . . PULSEForm	Sets the AMI pulse form
TRIGger:A:COMMunication:AMI: . . THReshold:HIGH	Sets the AMI threshold high level
TRIGger:A:COMMunication:AMI: . . THReshold:LOW	Sets the AMI threshold low level
TRIGger:A:COMMunication:BITRate	Sets and returns the bit rate
TRIGger:A:COMMunication:CLOCK: POLarity	Sets or returns the polarity of the clock signal
TRIGger:A:COMMunication:CMI: PULSEForm	Selects the CMI pulse form
TRIGger:A:COMMunication:CODE	Sets the signal code
TRIGger:A:COMMunication:SOUrce	Sets the source channel
TRIGger:A:COMMunication:SOUrce: TYPE	Sets the type of source channel
TRIGger:A:COMMunication:STANdard	Selects the standard that identifies the code and bit rate
TRIGger:A:EDGE:COUPling	Sets or returns the type of coupling for the A edge trigger

TRIGger:A:EDGE:SLOpe	Sets or returns the slope for the A edge trigger
TRIGger:A:EDGE:SOUrce	Sets or returns the source for the A edge trigger
TRIGger:A:EDGE?	Returns the source, coupling and source for the A edge trigger
TRIGger:A:HOLDoff:ACTUal?	Returns the holdoff time actually used by the A trigger
TRIGger:A:HOLDoff:BY	Sets or returns the type of holdoff for the A trigger
TRIGger:A:HOLDoff:TIMe	Sets or returns the A trigger holdoff time
TRIGger:A:HOLDoff?	Returns the A trigger holdoff parameters
TRIGger:A:LEVel	Sets or returns the level for the A trigger
TRIGger:A:LOGic:CLAss	Sets or returns the type of A trigger logic
TRIGger:A:LOGic:FUNcTion	Sets or returns the logical combination of the input channels for the A logic trigger
TRIGger:A:LOGic:INPut:CH<x>	Specifies or returns the input settings for the specified logic trigger channel
TRIGger:A:LOGic:INPut?	Returns the A logic trigger input for channels 1-3.
TRIGger:A:LOGic:PATtern:INPut:CH2	Sets or returns the A logic trigger input for channel 2
TRIGger:A:LOGic:PATtern:INPut:CH4	Sets or returns the A logic trigger input for channel 4
TRIGger:A:LOGic:PATtern:WHEn	Sets or returns the condition for generating the A logic pattern trigger
TRIGger:A:LOGic:PATtern:WHEn:LESSLimit	Sets or returns the maximum time that the selected pattern may be true and still generate an A logic pattern trigger
TRIGger:A:LOGic:PATtern:WHEn:MORELimit	Sets or returns the minimum time that the selected pattern may be true and still generate an A logic pattern trigger
TRIGger:A:LOGic:PATtern?	Returns the conditions for generating an A logic pattern trigger
TRIGger:A:LOGic:SETHold:CLOCK:EDGE	Sets or returns the clock edge polarity for setup and hold triggering
TRIGger:A:LOGic:SETHold:CLOCK:SOUrce	Sets or returns the clock source for the A logic trigger setup and hold input
TRIGger:A:LOGic:SETHold:CLOCK:THREShold	Sets or returns the clock voltage threshold for setup and hold trigger
TRIGger:A:LOGic:SETHold:CLOCK?	Returns clock edge polarity, voltage threshold and source input for setup/hold triggering

---

TRIGger:A:LOGic:SETHold:DATA:SOUrce	Sets or returns the data source for the setup and hold trigger
TRIGger:A:LOGic:SETHold:DATA:THReshold	Sets or returns the data voltage threshold for setup and hold trigger
TRIGger:A:LOGic:SETHold:DATA?	Returns the voltage threshold and data source for the setup/hold trigger
TRIGger:A:LOGic:SETHold:HOLDTime	Sets or returns the hold time for the setup and hold violation triggering
TRIGger:A:LOGic:SETHold:SETTime	Sets or returns the setup time for setup and hold violation triggering
TRIGger:A:LOGic:SETHold?	Returns clock edge polarity, voltage threshold and source; data voltage threshold and source; and setup/hold times for violation triggering
TRIGger:A:LOGic:STATE:INPut:CH2	Sets or returns the slope for channel 2 of the A logic state trigger
TRIGger:A:LOGic:STATE:INPut:CH4	Sets or returns the slope for channel 4 of the A logic state trigger
TRIGger:A:LOGic:STATE:WHEn	Sets or returns the condition for generating an A logic state trigger
TRIGger:A:LOGic:STATE?	Returns the data input and trigger criteria for the A logic trigger
TRIGger:A:LOGic:THReshold:CH<x>	Sets or returns the A logic trigger threshold voltage for the specified channel
TRIGger:A:LOGic:THReshold?	Returns the threshold voltage for all channels in A logic trigger
TRIGger:A:LOGic?	Returns all of the A trigger logic parameters
TRIGger:A:MODE	Sets or returns the A trigger mode
TRIGger:A:PULse:CLAss	Sets or returns the type of pulse on which to trigger
TRIGger:A:PULse:GLItch:POLarity	Sets or returns the polarity for the A pulse glitch trigger
TRIGger:A:PULse:GLItch:TRIGIF	Sets or returns the acceptance or rejection of pulse glitch trigger, based on width.
TRIGger:A:PULse:GLItch:WIDth	Sets or returns the width of the A pulse glitch trigger
TRIGger:A:PULse:GLItch?	Returns the current A glitch pulse trigger parameters
TRIGger:A:PULse:RUNT:LOGic:INPut:CH<x>	Sets or returns the input setting for the logic condition with runt for channel 1 through channel 4

TRIGger:A:PULse:RUNT:LOGIc:INPUT?	Returns the current A runt pulse trigger logic input parameters
TRIGger:A:PULse:RUNT:LOGIc:THReshold:CH<x>	Sets or returns logic threshold channels with runt for channel 1 through channel 4
TRIGger:A:PULse:RUNT:LOGIc?	Returns the current A runt pulse trigger logic parameters
TRIGger:A:PULse:RUNT:POLarity	Sets or returns the polarity for the A pulse runt trigger
TRIGger:A:PULse:RUNT:THReshold:BOTH	Sets the upper and lower switching thresholds for the A pulse runt trigger
TRIGger:A:PULse:RUNT:THReshold:HIGh	Sets or returns the upper limit for the A pulse runt trigger
TRIGger:A:PULse:RUNT:THReshold:LOW	Sets or returns the lower limit for the A pulse runt trigger
TRIGger:A:PULse:RUNT:THReshold?	Returns the upper and lower thresholds for the A pulse runt trigger
TRIGger:A:PULse:RUNT:WHEN	Sets or returns the type of pulse width the trigger checks for when it uncovers a runt
TRIGger:A:PULse:RUNT:WIDth	Sets or returns the minimum width for A pulse runt trigger
TRIGger:A:PULse:RUNT?	Returns the current A runt pulse trigger parameters
TRIGger:A:PULse:SOURce	Sets or returns the source for the A pulse trigger
TRIGger:A:PULse:TIMEOut:POLarity	Sets or returns the polarity for the A pulse timeout trigger
TRIGger:A:PULse:TIMEOut:TIME	Sets or returns the pulse timeout trigger time
TRIGger:A:PULse:TIMEOut?	Returns the parameters for the A pulse timeout trigger
TRIGger:A:PULse:TRANSition:DELTATime	Sets or returns the delta time used in calculating the transition value
TRIGger:A:PULse:TRANSition:POLarity	Sets or returns the polarity for the A pulse transition trigger
TRIGger:A:PULse:TRANSition:THReshold:BOTH	Sets the upper and lower thresholds for the transition trigger
TRIGger:A:PULse:TRANSition:THReshold:HIGh	Sets or returns the upper transition trigger threshold
TRIGger:A:PULse:TRANSition:THReshold:LOW	Sets or returns the lower transition trigger threshold
TRIGger:A:PULse:TRANSition:THReshold?	Returns the upper and lower threshold limits for the transition time trigger
TRIGger:A:PULse:TRANSition:WHEN	Sets or returns the relationship of delta time to transitioning signal

TRIGger:A:PULse:TRANSition?	Returns the delta time, polarity, and both upper and lower threshold limits for the transition time trigger
TRIGger:A:PULse:WIDth:HIGHLimit	Sets or returns the upper limit for the A pulse width trigger
TRIGger:A:PULse:WIDth:LOWLimit	Sets or returns the lower limit for the A pulse width trigger
TRIGger:A:PULse:WIDth:POLarity	Sets or returns the polarity for the A pulse width trigger
TRIGger:A:PULse:WIDth:WHEn	Sets or returns the criteria for width specification of pulse width trigger events
TRIGger:A:PULse:WIDth?	Returns the trigger A pulse width parameters
TRIGger:A:PULse:WINDow:LOGic:INPUT:CH<x>	Sets or returns input setting for logic condition with window for channel 1 through channel 4
TRIGger:A:PULse:WINDow:LOGic:INPUT?	Returns the current A window pulse trigger logic input parameters
TRIGger:A:PULse:WINDow:LOGic:THReshold:CH<x>	Sets or returns logic threshold channels with window for channel 1 through channel 4
TRIGger:A:PULse:WINDow:LOGic?	Returns the current A window pulse trigger logic parameters
TRIGger:A:PULse:WINDow:THReshold:BOTH	Sets both window thresholds
TRIGger:A:PULse:WINDow:THReshold:HIGh	Sets or returns high window threshold
TRIGger:A:PULse:WINDow:THReshold:LOW	Sets or returns low window threshold
TRIGger:A:PULse:WINDow:Threshold?	Returns the upper and lower thresholds for the A pulse window trigger.
TRIGger:A:PULse:WINDow:TYPE	Sets or returns type of window trigger
TRIGger:A:PULse:WINDow:WHEn	Sets or returns condition for use of window violation
TRIGger:A:PULse:WINDow:WIDTh	Sets or returns minimum width of violation
TRIGger:A:PULse:WINDow?	Returns the current A window pulse trigger parameters
TRIGger:A:PULse?	Returns the A pulse trigger parameters
TRIGger:A:TYPE	Sets or returns the type of A trigger
TRIGger:A:VIDeo:CUSTom:FORMat	Sets or returns the video scan format when the video standard is set to Custom
TRIGger:A:VIDeo:CUSTom:SCAN	Sets or returns the video scan rate
TRIGger:A:VIDeo:CUSTom?	Returns the custom video parameters for the A trigger
TRIGger:A:VIDeo:FIELD	Sets or returns the video field trigger

---

TRIGger:A:VIDeo:HOLdoff:FIELD	Sets or returns the trigger holdoff in video field units
TRIGger:A:VIDeo:LINE	Sets or returns the trigger delay as a number of video lines
TRIGger:A:VIDeo:POLarity	Sets or returns the polarity of the video trigger
TRIGger:A:VIDeo:SCAN	Sets or returns the video scan rate
TRIGger:A:VIDeo:SOURce	Sets or returns the video trigger source
TRIGger:A:VIDeo:STANdard	Sets or returns the video standard
TRIGger:A:VIDeo?	Returns the video parameters for the A trigger
TRIGger:B	Sets the B trigger level to 50% or returns the B trigger parameters
TRIGger:B:BY	Sets or returns B trigger time or event qualifiers
TRIGger:B:EDGE:COUPling	Sets or returns the type of B trigger coupling
TRIGger:B:EDGE:SLOpe	Sets or returns the B edge trigger slope
TRIGger:B:EDGE:SOURce	Sets or returns the B edge trigger source
TRIGger:B:EDGE?	Returns B trigger edge type parameters
TRIGger:B:EVENTS:COUNT	Sets or returns the number of events that must occur before the B trigger occurs
TRIGger:B:EVENTS?	Returns the current B trigger events parameter
TRIGger:B:LEVel	Sets or returns the level for the B trigger
TRIGger:B:STATE	Sets or returns the state of the B trigger
TRIGger:B:TIME	Sets or returns the B trigger delay time
TRIGger:B:TYPE	Sets or returns the type of B trigger
TRIGger:STATE?	Returns the current state of the triggering system

---

# Vertical Command Group

Use the commands in the Vertical Command Group to control the vertical setup of all live (channel) waveforms for acquisition and to control the display of channel, reference, and math waveforms.

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

<b>Command</b>	<b>Description</b>
CH<x>?	Returns vertical parameters for the specified channel
CH<x>:BANdwidth	Sets or returns the bandwidth of the specified channel
CH<x>:COUPling	Sets or returns the coupling setting for the specified channel
CH<x>:DESKew	Sets or returns the deskew time for the specified channel
CH<x>:INVert	Sets or returns the invert function for the specified channel
CH<x>:LABel:NAME	Defines or returns the label for the channel waveform
CH<x>:LABel:XPOS	Sets or returns the X display coordinate for the channel waveform label
CH<x>:LABel:YPOS	Sets or returns the Y display coordinate for the channel waveform label
CH<x>:OFFSet	Sets or returns the channel offset
CH<x>:POSition	Sets or returns the channel vertical position
CH<x>:PRObe?	Returns the gain, resistance, units, and ID of the probe that is attached to the specified channel
CH<x>:PROBECal?	Returns the probe calibration status
CH<x>:PRObe:GAIN?	Returns the gain of the probe that is attached to the specified channel
CH<x>:PRObe:ID?	Returns the type and serial number of the probe that is attached to the specified channel
CH<x>:PRObe:ID:TYPe	Returns the type of probe that is attached to the specified channel
CH<x>:PRObe:ID:SERnumber?	Returns the serial number of the probe that is attached to the specified channel
CH<x>:PRObe:RESistance?	Returns the resistance of the probe that is attached to the specified channel
CH<x>:PRObe:UNits?	Returns the units of the probe that is attached to the specified channel
CH<x>:PROBEFunc:EXTAtten	Sets the attenuation value for the specified channel to the specified scale factor Or returns the user-specified attenuation
CH<x>:PROBEFunc:EXTDBatten	Sets the attenuation value for the specified channel to the specified value, in decibels Or returns the user-specified attenuation, in decibels

---

CH<x>:PROBEFunc:EXTUnits	Sets the unit of measurement for the external attenuator of the specified channel Or returns the user-specified unit of measurement for the external attenuator
CH<x>:SCALE	Sets or returns the vertical scale of the specified channel
CH<x>:TERmination	Sets channel input termination
MATH<x>:POSITION	Sets or returns the vertical position for math display
MATH<x>:SCALE	Sets or returns the vertical scale (per div) for math display
REF<x>:HORizontal:POSITION	Sets or returns the horizontal position of the specified reference waveform in percent of the waveform that is displayed to the right of the center vertical graticule
REF<x>:LABel?	Returns a branch query containing the waveform label name and the coordinates at which the label is displayed
REF<x>:LABel:NAME	Sets or returns the label of the designated waveform
REF<x>:LABel:XPOS	Sets or returns the position of the reference waveform label on the X axis
REF<x>:LABel:YPOS	Sets or returns the position of the reference waveform label on the Y axis
REF<x>:VERTical:POSITION	Sets or returns the vertical position of the specified reference waveform
REF<x>:VERTical:SCALE	Sets or returns the reference waveform vertical scale in vertical units/div
SElect?	Returns information on which waveforms are on or off and which waveform is selected.
SElect:<wfm>	Turns on the specified waveform Or returns whether the specified channel is on or off
SElect:CONTRol <wfm>	Sets or returns the waveform controlled by the front panel

## Waveform Transfer Command Group

Use the commands in the Waveform Transfer Command Group to transfer waveform data points to and from the instrument. Waveform data points are a collection of values that define a waveform. One data value usually represents one data point in the waveform record. When working with envelope waveforms, each data value is either the minimum or maximum of a min/max pair.

Before you transfer waveform data, you must specify the data format, record length, and waveform source.



---

## Data Formats

Acquired waveform data uses eight or more bits to represent each data point. The number of bits used depends on the acquisition mode specified when you acquired the data. Data acquired in **SAM**ple or **EN**velope mode uses eight bits per waveform data point. Data acquired in **AVER**age mode uses up to 14 bits per point.

The instrument can transfer waveform data in either ASCII or binary format. You specify the format with the **DATA:ENCd**g command. The instrument uses signed, 4 byte integers and floating point values; it does not support unsigned floating point values.

**ASCII data** is represented by signed integer or floating point values. An example ASCII waveform data string may look like this:

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

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

**Binary data** can be represented by signed integer or floating point values. The range of the values depends on the byte width specified. When the byte width is one, signed integer data ranges from -128 to 127, and positive integer values range from 0 to 255. When the byte width is two, the values range from -32768 to 32767. When a **MATH** (or **REF** that came with a **MATH**) is used, 32 bit floating point values are used that are four bytes in width.

The defined binary formats specify the order in which the bytes are transferred. The following are the four binary formats:

- **RIBinary** specifies signed integer data-point representation with the most significant byte transferred first.
- **SRIBinary** is the same as **RIBinary** except that the byte order is swapped, meaning that the least significant byte is transferred first. This format is useful when transferring data to PCs.
- **RFBinary** specifies floating point data-point representation with the most significant byte transferred first.
- **SRFBinary** is the same as **RFBinary** except that the byte order is swapped, meaning that the least significant byte is transferred first. This format is useful when transferring data to PCs.

### Waveform Data and Record Lengths

You can transfer multiple points for each waveform record. You can transfer a portion of the waveform or you can transfer the entire record. You can use the **DATA:STAR**t and **DATA:STOP** commands to specify the first and last data points of the waveform record.

When transferring data into the instrument, you must first specify the record length of the destination waveform record. You do this with the **WFMI**nopre:NR\_Pt command. Next, specify the first data point within the waveform record. For example, when you set **DATA:STAR**t to 1, data points will be stored starting with the first point in the record. The instrument will ignore the value set by **DATA:STOP** when reading in data. It will stop reading data when there is no more data to read or when it has reached the specified record length.

---

When transferring data from the instrument, you must specify the first and last data points in the waveform record. Setting DATA:START to 1 and DATA:STOP to the record length will always return the entire waveform.

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

The DATA:SOURce command specifies the waveform source when transferring a waveform from the instrument. You can only transfer one waveform at a time. Waveforms sent to the instrument are always stored in one of the four reference memory locations. Use the DATA:DESTination command to specify a reference memory location.

### **Waveform Preamble**

Each waveform that you transfer has an associated waveform preamble that contains information such as the horizontal scale, the vertical scale, and other settings in effect when the waveform was created. Refer to the individual WFMInpre and WFMOupre commands for more information.

### **Scaling Waveform Data**

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

### **Transferring Waveform Data from the Instrument**

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

1. Select the waveform source(s) using DATA:SOURce.
2. Specify the waveform data format using DATA:ENCdg.
3. Specify the number of bytes per data point using WFMOupre:BYT\_Nr.  
**Note:** MATH waveforms (and REF waveforms that came from a MATH) are always set to four bytes.
4. Specify the portion of the waveform that you want to transfer using DATA:START and DATA:STOP.
5. Transfer waveform preamble information using WFMOupre?.
6. Transfer waveform data from the instrument using CURVe?.

### **Transferring Waveform Data to the Instrument**

1. Specify waveform reference memory using DATA:DESTination.
2. Set WFMInpre:NR\_Pt to equal the number of data points to be sent.
3. Specify the waveform data format using WFMInpre:ENCdg.
4. Specify the number of bytes per data point using WFMInpre:BYT\_Nr.
5. Specify first data point in the waveform record using DATA:START.
6. Transfer waveform preamble information using WFMInpre.

**Note:** FastAcq waveforms cannot be returned to the instrument via the CURVe command.

<b>Command</b>	<b>Description</b>
CURVe	The command format transfers waveform data to the instrument (reference memory location specified by DATA:DESTination)  The query format transfers waveform data from instrument specified by the DATA:SOURce command
DATA	Sets the format and location of the waveform data that is transferred with the CURVe? Command  Or returns the format and location of the waveform data that is transferred with the CURVe? command
DATA:DESTination	Sets or returns the reference memory location for storing waveform data sent to the instrument
DATA:ENCdg	Sets or returns the format of outgoing waveform data
DATA:SOURce	Sets or returns the location of waveform data transferred from the instrument
DATA:START	Sets or returns the starting point in waveform transfer
DATA:STOP	Sets or returns the ending data point in waveform transfer
WAVFrm?	Returns a branch query containing waveform data in either binary or ASCII format, waveform formatting data, and the location of the waveform data source
WFMImpre?	Returns the waveform formatting specification to be applied to the next incoming CURVE command data
WFMImpre:BIT_Nr	Sets or returns the number of bits per binary waveform point
WFMImpre:BN_Fmt	Sets or returns the format of binary data for the incoming waveform
WFMImpre:BYT_Nr	Sets or returns the binary field data width for the first ordered waveform
WFMImpre:BYT_Or	Sets or returns the byte order of waveform points
WFMImpre:ENCdg	Sets or returns the type of encoding for incoming waveform data
WFMImpre:NR_FR?	Returns one frame, indicating the number of frames in the transmitted waveform record
WFMImpre:NR_Pt	Sets or returns the number of points in the transmitted waveform record
WFMImpre:PT_Fmt	Sets or returns the point format of incoming waveform data
WFMImpre:PT_Off	Sets or returns the trigger point within the incoming waveform data
WFMImpre:WFId	Provided to allow a waveform extracted from the instrument to be more easily sent back
WFMImpre:XINcr	Sets or returns the horizontal sampling interval between incoming waveform points
WFMImpre:XUNit	Sets or returns the horizontal units of the incoming waveform

---

WFMinpre:XZEro	Sets or returns the (sub-sample) time between the trigger sample and the occurrence of the actual incoming waveform trigger
WFMinpre:YMUlt	Sets or returns the vertical scale factor, per digitizing level, of the incoming waveform points
WFMinpre:YOFF	Sets or returns the vertical position of the incoming waveform in digitizing levels
WFMinpre:YUNit	Sets or returns the vertical units of the incoming waveform
WFMinpre:YZEro	Sets or returns the offset of the incoming waveform
WFMOutpre?	Returns the waveform formatting data for the waveform specified by the DATA:SOURCE command
WFMOutpre:BIT_Nr	Sets or returns the number of bits per waveform point that outgoing waveforms contain
WFMOutpre:BN_Fmt	Sets or returns the format of binary data for the waveform
WFMOutpre:BYT_Nr	Sets or returns the data width for the waveform
WFMOutpre:BYT_Or	Sets or returns the byte order of waveform points
WFMOutpre:ENCdg	Sets or returns the type of encoding for outgoing waveforms
WFMOutpre:NR_FR?	Returns the number of frames for the DATA:SOURce waveform transmitted in response to a CURVe? query
WFMOutpre:NR_Pt?	Returns the number of points for the waveform transmitted in response to a CURVe? query
WFMOutpre:PT_Fmt?	Returns the point format for the waveform
WFMOutpre:PT_Off?	Returns the trigger point relative to DATA:START for the waveform
WFMOutpre:PT_OR?	Returns whether the source waveform is DPO
WFMOutpre:WFId?	Returns a string describing the acquisition parameters for the waveform
WFMOutpre:XINcr?	Returns the horizontal sampling interval
WFMOutpre:XUNit?	Returns the horizontal units for the waveform
WFMOutpre:XZEro?	Returns the (sub-sample) time between the trigger sample and the occurrence of the actual waveform trigger
WFMOutpre:YMUlt?	Returns the vertical scale factor per digitizing level for the waveform
WFMOutpre:YOFF?	Returns the vertical offset in digitizing levels for the waveform
WFMOutpre:YUNit?	Returns the vertical units for the waveform
WFMOutpre:YZEro?	Returns the vertical offset for the waveform
WFMPre:NR_FR?	Returns the number of frames for the waveform transmitted in response to a CURVe? query

---

# Zoom Command Group

Zoom commands let you expand and position the waveform display horizontally and vertically, without changing the time base or vertical settings.

You can specify up to four zoom areas. You can also specify a zoomed waveform as a zoom source, which enables you to zoom in on a zoomed waveform.

<b>Command</b>	<b>Description</b>
ZOOM	Sets the zoom transforms to default values for either live traces or all traces  Or returns the current vertical and horizontal positioning and scaling of the display
ZOOM:GRATICULE:SIZE	Sets or returns the size of the zoom graticule
ZOOM:GRATICULE:SPLIT	Sets or returns the sizes of the acquisition and zoom windows when Zoom is selected.
ZOOM:MODE	Sets or returns the zoom mode
ZOOM:SCROLL:DIRECTION	Sets or returns the direction of AutoScroll for zoomed waveforms
ZOOM:SCROLL:LOCK	Sets or returns the state of Scroll Lock, on or off
ZOOM:SCROLL:SPEED	Sets or returns the speed of AutoScroll for zoomed waveforms
ZOOM:ZOOM<x>	Sets or returns zoom factors for the specified zoom
ZOOM:ZOOM<x>:<wfm>:HORIZONTAL:POSITION	Sets or returns the horizontal zoom position for the specified waveform in the specified zoom
ZOOM:ZOOM<x>:<wfm>:HORIZONTAL:SCALE	Sets or returns the horizontal zoom scale of the specified waveform in the specified zoom
ZOOM:ZOOM<x>:<wfm>:VERTICAL:POSITION	Sets or returns the vertical zoom position for the specified waveform in the specified zoom
ZOOM:ZOOM<x>:<wfm>:VERTICAL:SCALE	Sets or returns the vertical zoom scale for the specified waveform in the specified zoom
ZOOM:ZOOM<x>:SCROLLLOCK	Sets or returns Scroll Lock for the specified zoom
ZOOM:ZOOM<x>:STATE	Specifies or returns a trace as zoomed, on or off

---

# Commands Listed in Alphabetical Order

## \*CAL?

### Description

This query-only command instructs the digitizing oscilloscope to perform signal path compensation and return its calibration status. This command is equivalent to selecting Instrument Calibration from the Utilities menu and then clicking the Calibrate button.

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

### Group

Calibration

### Related Commands

CALibrate:RESults:SPC?

### Syntax

\*CAL?

### Returns

- 0

<NR1>

Means the calibration did not complete successfully.

- 1 (or any non zero value)

<NR1>

Means the calibration completed without errors.

### Example

\*CAL?

This query might return

1

This indicates that the calibration was successful.

---

# \*CLS

## Description

This command (no query form) clears the following:

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

If the \*CLS command immediately follows an <EOI>, the Output Queue and MAV bit (Status Byte Register bit 4) are also cleared. MAV indicates that information is in the output queue. The device clear (DCL) GPIB control message will clear the output queue and thus MAV. \*CLS does not clear the output queue or MAV.

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

## Group

Status and Error

## Related Commands

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

## Syntax

\*CLS

## Example

\*CLS

This command clears the instrument status data structures.

---

# \*DDT

## Description

This command allows you to specify a command or a list of commands that are executed when the instrument receives a \*TRG command or the GET GPIB interface message. Define Device Trigger (\*DDT) is a special alias that the \*TRG command uses.

## Group

Miscellaneous

## Related Commands

ALIAS, \*TRG

## Syntax 1

```
*DDT {<Block>|<Qstring>}
```

## Syntax 2

```
*DDT?
```

## Arguments

### ■ <Block>

This is a complete sequence of program messages. The messages can contain only valid commands that must be separated by semicolons and must follow all rules for concatenating commands. The sequence must be less than or equal to 80 characters. The format of this argument is always returned as a query.

### ■ <Qstring>

This is a complete sequence of program messages. The messages can contain only valid commands that must be separated by semicolons and must follow all rules for concatenating commands. The sequence must be less than or equal to 80 characters.

## Example

```
*DDT #OACQUIRE:STATE RUN
```

This command specifies that the acquisition system will be started each time a \*TRG command is sent.



---

# \*ESE

## Description

This command sets and queries the bits in the Event Status Enable Register (ESER). The ESER prevents events from being reported to the Status Byte Register (STB). For a more detailed discussion of the use of these registers, see Registers.

## Group

Status and Error

## Related Commands

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

## Syntax 1

\*ESE <NR1>

## Syntax 2

\*ESE?

## Argument

■ <NR1>

This specifies the binary bits of the ESER according to this value, which ranges from 0 through 255.

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

**Note:** Setting the DESER and the ESER to the same values allows only those codes to be entered into the Event Queue and summarized on the ESB bit (bit 5) of the Status Byte Register. Use the DESE command to set the DESER.

## Example 1

\*ESE 209

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

## Example 2

\*ESE?

This query form of this command returns 186, showing that the ESER contains the binary value 10111010.

---

# **\*ESR?**

## **Description**

This query-only command returns the contents of the Standard Event Status Register (SESR). \*ESR? also clears the SESR (since reading the SESR clears it). For a more detailed discussion of the use of these registers, see Registers.

## **Group**

Status and Error

## **Related Commands**

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

## **Syntax**

\*ESR?

## **Example**

\*ESR?

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

---

# **\*IDN?**

## **Description**

This query-only command returns the instrument identification code.

## **Group**

Miscellaneous

## **Related Commands**

ID?

## **Syntax**

IDN?

## **Example**

\*IDN?

This query might return

```
:TEKTRONIX,TDS5054B,10000001,CF:91.1CT  
FV:01.00.912,
```

indicating the instrument model number, serial number, configured number, and firmware version number.

---

# \*LRN?

## Description

This query-only command returns the commands that list the instrument settings (except for configuration information for the calibration values), allowing you to record or "learn" the current instrument settings. You can use these commands to return the instrument to the state it was in when you made the \*LRN? query. This command is identical to the SET? Command.

## Group

Miscellaneous

## Related Commands

SET?

## Syntax

\*LRN?

## Example

\*LRN?

This query might return the following response:

```
:ACQUIRE:STOPAFTER RUNSTOP;STATE 1;MODE SAMPLE;NUMENV 10;NUMAVG 16;NUMSAMPLES 16000;SAMPLINGMODE IT;:FASTACQ:STATE 0;:APPLICATION:GPKNOB1:ACTIVE 0;:APPLICATION:GPKNOB2:ACTIVE 0;:APPLICATION:WINDOW:HEIGHT 236;WIDTH 640;:APPLICATION:SCOPEAPP:STATE RUNNING;WINDOW FULL-SCREEN;:APPLICATION:EXTAPP:STATE NOTRUNNING;:VARIABLE:EVENT:REPORT BOTH;:AUX-OUT:SOURCE ATRIGGER;EDGE FALLING;:CMDBATCH 1;:HEADER 1;:LOCK NONE;:ROSC:SOURCE INTERNAL;:VERBOSE 1;:ALIAS:STATE 0;:DISPLAY:CLOCK 1;COLOR:PALETTE:IMAGEVIEW TEMPERATURE;RECORD-VIEW NORMAL;USER:GRATICULE 165,50,15;CH1 180,50,100;CH2 300,50,100;CH3 60,50,100;CH4 240,50,100;REF1 0,90,0;REF2 0,90,100;REF3 60,90,100;REF4 240,90,100;MATH1 160,50,100;MATH2 40,60,100;MATH3 120,60,100;MATH4 195,50,100;HISTOGRAM 320,50,100;CARET 150,50,100;MASK 0,25,75;MASK-HIGHLIGHT 140,50,100;:DISPLAY:COLOR:MATHCOLOR DEFAULT..
```

# \*OPC

## Description

This command generates the operation complete message in the Standard Event Status Register (SESR) when all pending commands that generate an OPC message are complete. The \*OPC? query places the ASCII character "1" into the output queue when all such OPC commands are complete. The \*OPC? response is not available to read until all pending operations finish. For a complete discussion of the use of these registers and the output queue, see Registers and Queues.

The \*OPC command allows you to synchronize the operation of the instrument with your application program. For more information, see Synchronization Methods.

Commands that generate an OPC Message	
Operation	Command
Single sequence acquisition	ACQuire:STATE ON or ACQuire:STATE RUN or ACQuire:STATE 1 (when ACQuire:STOPAfter is set to SEQUENCE).
Calibration step	Refer to the optional <i>TDS5000 Series Digital Phosphor Oscilloscopes Service Manual</i>

## Group

Status and Error

## Related Commands

BUSY?, \*WAI

## Syntax 1

\*OPC

## Syntax 2

\*OPC?

## Example 1

\*OPC

Upon completion of all pending OPC operations, this command generates the operation complete message in the SESR.

## Example 2

\*OPC?

This query might return 1 to indicate that all pending OPC operations are finished.

---

# \*OPT?

## Description

This query-only command returns a list of the options installed in your instrument.

## Group

Status and Error

## Syntax

\*OPT?

## Example

\*OPT?

This query might return

:3M:16 Meg Max 4M/CH

---

# \*PSC

## Description

This command sets and queries the power-on status flag that controls the automatic power-on handling of the DESER, SRER, and ESER registers. When \*PSC is true, the DESER register is set to 255 and the SRER and ESER registers are set to 0 at power-on. When \*PSC is false, the current values in the DESER, SRER, and ESER registers are preserved in nonvolatile memory when power is shut off and are restored at power-on.

## Group

Status and Error

## Related Commands

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

## Syntax 1

\*PSC <NR1>

## Syntax 2

\*PSC?

## Argument

- <NR1>

A 0 sets the power-on status clear flag to false, disables the power-on clear and allows the instrument to possibly assert SRQ after power-on; any other value sets the power-on status clear flag to true, enabling the power-on status clear and prevents any SRQ assertion after power on.

## Example 1

\*PSC 0

This command sets the power-on status clear flag to false.

## Example 2

\*PSC?

The query form of this command might return 1 to indicate that the power-on status clear flag is set to true.

---

# \*PUD

## Description

This command sets or queries a string of Protected User Data. This data is protected by the PASSWord command. You can modify it only by first entering the correct password. This password is not necessary to query the data.

## Group

Status and Error

## Related Commands

PASSWord

## Syntax 1

```
*PUD {<Block>|<String>}
```

## Syntax 2

```
*PUD?
```

## Argument

- <Block>

This is a block containing up to 100 characters.

- <String>

This is a string containing up to 100 characters.

## Example 1

```
*PUD #229This instrument belongs to me
```

This command stores the string "This oscilloscope belongs to me" in the user protected data area.

## Example 2

```
*PUD?
```

This query might return #221Property of Company X



---

# \*RCL

## Description

This command (no query form) restores the state of the instrument from a copy of the settings stored in memory (The settings are stored using the \*SAV command). If 'factory' is referenced (by specifying '0'), the factory default values will be restored.

This command is equivalent to RECALL:SETUp and performs the same function as selecting Recall from the File menu, and then pressing the Setup button.

## Group

Save and Recall

## Related Commands

DELEte:SETUp, FACtory, \*LRN?, RECALL:SETUp, \*RST, \*SAV, SAVE:SETUp

## Syntax

\*RCL <NR1>

## Argument

- <NR1>

This specifies a setup storage location value ranging from 0 through 10. Using an out-of-range value causes an execution error.

## Example

\*RCL 3

This command restores the instrument from a copy of the settings stored in memory location 3.

---

# \*RST

## Description

This command (no query form) resets the instrument to the factory default settings. The \*RST command does not alter the following:

- The state of the IEEE Std 488.1-1987 interface
- The selected IEEE Std 488.1-1987 address of the instrument
- Calibration data that affect device specifications
- The Output Queue
- The Service Request Enable Register setting
- The Power-on status clear flag setting
- Alias definitions
- Stored settings
- The \*PUD? Response
- Any of the values associated with the DATA command.

This command is equivalent to pressing the **DEFAULT SETUP** button on the front panel.

## Group

Status and Error

## Related Commands

FACtory, RECAll:SETUp, SAVe:SETUp

## Syntax

\*RST

## Argument

None

## Example

\*RST

This command resets the instrument settings to factory defaults.

---

# **\*SAV**

## **Description**

This command (no query form) stores the state of the instrument to a specified memory location. You can later use the \*RCL command to restore the instrument to this saved state. \

This command is equivalent to selecting Save As from the File menu, and then pressing the Setup button.

## **Group**

Save and Recall

## **Related Commands**

\*RCL, RECALL:SETUp, SAVe:SETUp

## **Syntax**

\*SAV <NR1>

## **Argument**

- <NR1>

This specifies a location in which to save the state of the instrument. Location values range from 1 through 10. Using an out-of-range location value causes an execution error. Any settings that have been stored previously at this location will be overwritten.

## **Example**

\*SAV 2

This command saves the current instrument state in memory location 2.

---

# \*SDS

## Description

This command (no query form) changes the specified setup to reference the factory setup instead of the specific user setup slot. The content of the setup slot is unchanged, but the data will no longer be accessible to you.

This command is equivalent to selecting Delete from the File menu, and then choosing the All Setups.

## Group

Save and Recall

## Related Commands

DELEte:SETUp

## Syntax

\*SDS <NR1>

## Argument

- <NR1>

This specifies a user setup location to delete. Setup storage location values range from 1 through 10; using an out-of-range value causes an error.

## Example

\*SDS 2

This command changes setup slot 2 to reference the factory setup.

---

# \*SRE

## Description

The \*SRE (Service Request Enable) command sets and queries the bits in the Service Request Enable Register. For more information, refer to Registers.

## Group

Status and Error

## Related Commands

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

## Syntax 1

\*SRE <NR1>

## Syntax 2

\*SRE?

## Argument

■ <NR1>

This is a value in the range from 0 through 255. The binary bits of the SRER are set according to this value. Using an out-of-range value causes an execution error. The power-on default for SRER is 0 if \*PSC is 1. If \*PSC is 0, the SRER maintains the previous power cycle value through the current power cycle.

## Example 1

\*SRE 48

This command sets the bits in the SRER to binary 00110000.

## Example 2

\*SRE?

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

---

# **\*STB?**

## **Description**

The \*STB? (Read Status Byte) query returns the contents of the Status Byte Register (SBR) using the Master Summary Status (MSS) bit. For more information, refer to Registers.

## **Group**

Status and Error

## **Related Commands**

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

## **Syntax**

\*STB?

## **Returns**

<NR1>

## **Example**

\*STB?

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

---

# **\*TRG**

## **Description**

This command (no query form) performs the group execute trigger on commands defined by \*DDT.

## **Group**

Miscellaneous

## **Related Commands**

\*DDT

## **Syntax**

\*TRG

## **Example**

\*TRG

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

---

# \*TST?

## Description

This query-only command tests (self-test) the GPIB interface and returns a 0.

## Group

Miscellaneous

## Syntax

\*TST?

## Example

\*TST?

This query always returns 0.



---

# **\*WAI**

## **Description**

The \*WAI (Wait) command (no query form) prevents the instrument from executing further commands or queries until all pending commands that generate an OPC message are complete. This command allows you to synchronize the operation of the instrument with your application program. For more information, refer to Synchronization Methods.

## **Group**

Status and Error

## **Related Commands**

BUSY?, \*OPC

## **Syntax**

\*WAI

## **Example**

\*WAI

This command prevents the instrument from executing any further commands or queries until all pending commands that generate an OPC message are complete.

---

# ACQuire:MODE

## Description

This command sets or queries the acquisition mode of the instrument. This affects all live waveforms. This command is equivalent to selecting Horizontal/Acquisition from the Horiz/Acq menu, and then choosing the desired mode from the Acquisition Mode group box.

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

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

## Group

Acquisition

## Related Commands

ACQuire:NUMAVg, ACQuire:NUMenv, CURVe?

## Syntax 1

```
ACQuire:MODE{SAMPLE|PEAKdetect|HIRes|AVER-  
age|ENVELOPE}
```

## Syntax 2

```
ACQuire:MODE?
```

## Arguments

### ■ SAMPLE

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

### ■ PEAKdetect

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

### ■ HIRes

Specifies Hi Res mode where the displayed data point value is the average of all the samples taken during the acquisition interval. This is a form of averaging, where the average comes from a single wave-

---

form acquisition. The number of samples taken during the acquisition interval determines the number of data values that compose the average.

#### ■ AVErAge

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

#### ■ ENVeloPe

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

#### ■ WFMDB

WfmDB (Waveform Database) mode acquires and displays a waveform pixmap. A pixmap is the accumulation of one or more acquisitions.

The oscilloscope acquires data after each trigger event using Sample mode; it then determines the pix map location of each sample point and accumulates it with stored data from previous acquisitions.

A Pix map is a two dimensional array. The value at each point in the array is a counter that reflects the hit intensity. Infinite and noninfinite persist display modes affect how pix maps are accumulated. Zoom, Math, FastAcq, FastFrame, XY, XYZ, Roll, and Interpolated Time (IT) Sampling Mode are conflicting features to WFMDB acqMode. Turning on one of them generally turns the other one off. Selection of some standard masks (for example, eye masks, which require option SM) changes the acquisition mode to WFMDB.

### Example 1

```
ACQuire:MODE ENVeloPe
```

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

### Example 2

```
ACQuire:MODE?
```

This command might return

```
:ACQuire:MODE AVERAGE
```

indicating that the displayed waveform is the average of the specified number of waveform acquisitions.

---

# ACQUIRE:NUMACQ?

## Description

This query-only command returns the number of waveform acquisitions that have occurred since starting acquisition with the ACQUIRE:STATE RUN command. This value is reset to zero when any acquisition, horizontal, or vertical arguments that affect the waveform are changed. The maximum number of acquisitions that can be counted is  $2^{30}-1$ . The instrument stops counting when this number is reached. This is the same value that displays in the upper center of the screen when the acquisition system is stopped.

## Group

Acquisition

## Related Commands

ACQUIRE:STATE

## Syntax

ACQUIRE:NUMACQ?

## Example

ACQUIRE:NUMACQ?

This command might return

```
:ACQUIRE:NUMACQ 350
```

indicating that 350 acquisitions have occurred since executing an ACQUIRE:STATE RUN command.

---

# ACQuire:NUMAVg

## Description

This command sets or queries the number of waveform acquisitions that make up an averaged waveform. Use the ACQuire:MODE command to enable the Average mode. Sending this command is equivalent to selecting Horizontal/Acquisition Setup from the Horiz/Acq menu, selecting the Acquisition tab, and choosing Average from the Acquisition Mode group box. Then enter the desired number of waveforms that will be used to make up an averaged waveform in the # of Wfms box.

## Group

Acquisition

## Related Commands

ACQuire:MODE

## Syntax 1

```
ACQuire:NUMAVg <NR1>
```

## Syntax 2

```
ACQuire:NUMAVg?
```

## Arguments

- NR1

This is the number of waveform acquisitions to average.

## Example 1

```
ACQuire:NUMAVg 10
```

This command specifies that 10 waveform averages will be performed before exponential averaging starts.

## Example 2

```
ACQuire:NUMAVg?
```

This command might return

```
:ACQUIRE:NUMAVG 75
```

indicating that there are 75 acquisitions specified for averaging.

---

# ACQUIRE:NUMENV

## Description

This command sets or queries the number of waveform acquisitions that make up an envelope waveform. Sending this command is equivalent to setting the # of Wfms in the Acquisition Setup menu when Envelope Acquisition mode is selected.

## Group

Acquisition

## Syntax 1

```
ACQUIRE:NUMENV {<NR1>|INFINITE}
```

## Syntax 2

```
ACQUIRE:NUMENV?
```

## Arguments

- NR1

This is the number of waveform acquisitions to be enveloped.

## Example 1

```
ACQUIRE:NUMENV 10
```

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

## Example 2

```
ACQUIRE:NUMENV?
```

This command might return

```
:ACQUIRE:NUMENV 0
```

indicating that acquisitions are acquired infinitely for enveloped waveforms.

---

# ACQUIRE:NUMSAMPLES

## Description

This command sets or queries the minimum number of acquired samples that make up a waveform database (WfmDB) waveform for single sequence mode and Mask Pass/Fail Completion Test. This is equivalent to setting the Waverform Database Samples in the Acquisition Mode side menu.

## Group

Acquisition

## Syntax 1

```
ACQUIRE:NUMSAMPLES {<NR1>}
```

## Syntax 2

```
ACQUIRE:NUMSAMPLES?
```

## Arguments

- NR1

This is the minimum number of acquired samples that make up a waveform database (WfmDB) waveform for single sequence mode and Mask Pass/Fail Completion Test. The default value is 16,000 samples. The range is 5,000 to 2,147,400,000 samples.

In real time mode, if the entire trace is displayed, one triggered acquisition should acquire 5000 samples for a record length of 5000. If you want 5001 samples, the instrument will need to make two acquisitions (10,000 samples). If you request fewer samples than the record length, the number of samples will be the same as the record length.

## Example 1

```
ACQUIRE:NUMSAMPLES 5000
```

This command specifies that 5000 samples will be acquired in WfmDB mode for single sequence mode and Mask Pass/Fail Completion Test.

## Example 2

```
ACQUIRE:NUMSAMPLES?
```

This command might return

```
:ACQUIRE:NUMSAMPLES 10000
```

indicating that there are 10,000 samples to be acquired in WfmDB mode for single sequence and Mask Pass/Fail Completion Test.

---

# ACQuire:SAMPlingmode

## Description

This command sets or queries the sampling mode.

## Group

Acquisition

## Syntax 1

```
ACQuire:SAMPlingmode{RT|IT|ET}
```

## Syntax 2

```
ACQuire:SAMPlingmode?
```

## Arguments

- RT

This argument sets the sampling mode to real time only.

- IT

This argument sets the sampling mode to interpolation allowed.

- ET

This argument sets the sampling mode to equivalent time allowed.

The number of channels selected and the sampling speed affects the mode that the instrument uses to acquire data. At slow sampling speeds the instrument uses real time sampling. At fast sampling speeds, the instrument uses equivalent time sampling or interpolation. For speeds between these extremes, the instrument creates waveform records differently, depending on the number of input channels selected. Refer to the following table, which assumes 500-point records when indicating the mode.

Time Base	One Channel	Two Channel	Three or Four Channel
≥40 ns/div	RT	RT	RT
20 ns/div	RT	RT	ET or IT
10 ns/div	RT	ET or IT	ET or IT
≤4 ns/div	ET or IT	ET or IT	ET or IT

≥ means slower than; ≤ means faster than



---

### **Example 1**

`ACQUIRE:NUMSAMPLES 5000`

This command specifies that 5000 samples will be acquired in WfmDB mode for single sequence mode and Mask Pass/Fail Completion Test.

### **Example 2**

`ACQUIRE:NUMSAMPLES?`

This command might return

`:ACQUIRE:NUMSAMPLES 10000`

indicating that there are 10,000 samples to be acquired in WfmDB mode for single sequence and Mask Pass/Fail Completion Test.

---

# ACQuire:STATE

## Description

This command starts or stops acquisitions. When state is set to ON or RUN, a new acquisition will be started. If the last acquisition was a single acquisition sequence, a new single sequence acquisition will be started. If the last acquisition was continuous, a new continuous acquisition will be started.

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

Sending this command is equivalent to pressing the front-panel RUN/STOP button.

## Group

Acquisition

## Related Commands

ACQuire:STOPAfter

## Syntax 1

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

## Syntax 2

```
ACQuire:STATE?
```

## Arguments

- OFF

This argument stops acquisitions.

- STOP

This argument stops acquisitions.

- ON

This argument starts acquisitions.

- RUN

This argument starts acquisitions.

- NR1

0 stops acquisitions; any other value starts acquisitions.

---

### **Example 1**

`ACQuire:STATE RUN`

This command starts the acquisition of waveform data and resets the count of the number of acquisitions.

### **Example 2**

`ACQuire:STATE?`

This query might return

`:ACQUIRE:STATE 0`

indicating that the acquisition is stopped.

---

# ACQUIRE:STOPAFTER

## Description

This command sets or queries whether the instrument continually acquires acquisitions or acquires a single sequence. This command is equivalent to pressing SINGLE from the front panel.

## Group

Acquisition

## Related Commands

ACQUIRE:STATE

## Syntax 1

```
ACQUIRE:STOPAFTER {RUNSTOP|SEQUENCE}
```

## Syntax 2

```
ACQUIRE:STOPAFTER?
```

## Arguments

- RUNSTOP

Specifies that the instrument will continually acquire data, if ACQUIRE:STATE is turned on.

- SEQUENCE

Specifies that the next acquisition will be a single-sequence acquisition.

## Example 1

```
ACQUIRE:STOPAFTER RUNSTOP
```

This command sets the instrument to continually acquire data.

## Example 2

```
ACQUIRE:STOPAFTER?
```

This query might return

```
:ACQUIRE:STOPAFTER SEQUENCE
```

indicating that the next acquisition the instrument makes will be of the single-sequence type.

---

# ACQuire?

## Description

This query-only command returns the following current acquisition parameters:

- Stop after
- Acquisition state
- Mode
- Number of envelopes
- Number of averages
- Repetitive signals

## Group

Acquisition

## Related Commands

ACQuire:MODE, ACQuire:NUMACq, ACQuire:NUMAVg, ACQuire:NUMENV, ACQuire:STOPAfter

## Syntax

ACQuire?

## Example

ACQuire?

This query might return the following string for the current acquisition parameters:

```
:ACQUIRE:STOPAFTER RUNSTOP;STATE 1;MODE SAMPLE;NUMENV 10;NUMAVG 16;NUMSAMPLES 16000;SAMPLINGMODE IT
```

---

# ALias

## Description

This command sets or queries the state of alias functionality.

## Group

Alias

## Related Commands

ALias:STATE

## Syntax 1

```
ALias {OFF | ON | <NR1>}
```

## Syntax 2

```
ALias?
```

## Arguments

- OFF

This turns Alias expansion off. If a defined alias is sent when ALias:STATE is off, a command error (102) will be generated.

- ON

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

- NR1

A 0 disables Alias mode; any other value enables Alias mode.

## Example 1

```
ALias ON
```

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

## Example 2

```
ALias?
```

This query might return

```
:ALIAS:STATE 1
```

indicating that the alias feature is on.

---

# ALias:CATalog?

## Description

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

## Group

Alias

## Syntax

```
ALias:CATalog?
```

## Example

```
ALias:CATalog?
```

This query might return the string

```
:ALIAS:CATALOG "SETUP1","TESTMENU1","DEFAULT"
```

showing that there are three aliases named SETUP1, TESTMENU1, and DEFAULT.

---

# ALias:DEFine

## Description

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

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

## Group

Alias

## Related Commands

ALias:STATE

## Syntax 1

```
ALias:DEFine <Qstring><,>{<Qstring>|<Block>}
```

## Syntax 2

```
ALias:DEFine?
```

## Arguments

- The first <Qstring> is the alias label.

This label cannot be a command name. Labels must start with a letter and can contain only letters, numbers, and underscores; other characters are not allowed. The label must be less than or equal to 12 characters.

- The second <Qstring> or <Block> is a complete sequence of program messages.

The messages can contain only valid commands that must be separated by semicolons and must follow all rules for concatenating commands. The sequence must be less than or equal to 256 characters.

## Example 1

```
ALias:DEFine "ST1",":RECALL:SETUP 5;:AUTOSET  
EXECUTE;:SELECT:CH1 ON"
```

This command defines an alias named "ST1" that sets up the instrument.



---

## Example 2

ALIAS:DEFINE? "ST1"

This command returns

```
:ALIAS:DEFINE "ST1",#246  
:RECALL:SETUP 5;:AUTOSET EXECUTE;:SELECT:CH1  
ON
```

---

# ALias:DELEte

## Description

This command removes a specified alias and is identical to ALias:DELEte:NAME. An error message is generated if the named alias does not exist.

## Group

Alias

## Related Commands

\*ESR?, ALias:DELEte:ALL

## Syntax

```
ALias:DELEte <Qstring>
```

## Argument

- <Qstring>

This is the name of the alias to be removed. Using ALias:DELEte without specifying an alias causes an execution error. <Qstring> must be a previously defined value.

## Example

```
ALias:DELEte "SETUP1"
```

This command deletes the alias named SETUP1.

---

# ALias:DELEte:ALL

## Description

This command deletes all existing aliases.

## Group

Alias

## Related Commands

ALias:DELEte, ALias:DELEte:NAME

## Syntax

```
ALias:DELEte:ALL
```

## Example

```
ALias:DELEte:ALL
```

This command deletes all existing aliases.

---

# ALias:DELEte:NAME

## Description

This command removes a specified alias. An error message is generated if the named alias does not exist. This command is identical to ALias:DELEte.

## Group

Alias

## Syntax

```
ALias:DELEte:NAME <Qstring>
```

## Argument

- <Qstring>

This is the name of the alias to remove. Using ALias:DELEte:NAME without specifying an alias causes an exception error. <Qstring> must be a previously defined alias.

## Example

```
ALias:DELEte:NAME "STARTUP"
```

This command deletes the alias named STARTUP.

---

# ALias:STATE

## Description

This command turns aliases on or off and is identical to the ALIas command. The query form of this command returns the state of the aliases.

## Group

Alias

## Syntax 1

```
ALIAS:STATE {OFF|ON|<NR1>}
```

## Syntax 2

```
ALIAS:STATE?
```

## Arguments

- OFF

This turns alias expansion off. If a defined alias is sent when ALIAS:STATE is OFF, the instrument generates a command error (102).

- ON

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

- <NR1>

A 0 turns off aliases; any other value turns on aliases.

## Example 1

```
ALIAS:STATE OFF
```

This command turns off the alias feature.

## Example 2

```
ALIAS:STATE?
```

This query might return

```
:ALIAS:STATE ON
```

indicating that alias expansion is currently turned on.

---

# ALLEV?

## Description

This query-only command prompts the instrument to return all events and their messages (delimited by commas), and removes the returned events from the Event Queue. Use the \*ESR? query to enable the events to be returned. This command is similar to repeatedly sending \*EVMsg? queries to the instrument.

## Group

Status and Error

## Related Commands

\*ESR?, \*EVMsg?

## Syntax

ALLev?

## Example

ALLev?

This query might return

```
:ALLEV 2225,"Measurement error, No waveform to  
measure; "420,"Query UNTERMINATED;"
```

---

# AUTOSet

## Description

This command (no query format) sets the vertical, horizontal, and trigger controls of the instrument to automatically acquire and display the selected waveform. (To autoselect a video waveform, the video trigger must be set to video standard, not custom.) This is equivalent to pressing the front-panel AUTOSET button. For a detailed description of autoselect functionality, see *Autoset* in the index of the online help for your instrument.

## Group

Miscellaneous

## Related Command

DISplay:GRATICule

## Syntax

```
AUTOSet {EXECute|UNDo|VIDeo|VLines|VFields}
```

## Arguments

### ■ EXECute

This argument autoselects the displayed waveform; this is equivalent to pressing the front-panel AUTOSET button. If the display is set to a PAL, MV, or IRE graticule, this argument forces the graticule display to full mode (frame, grid, and cross hair).

### ■ UNDo

This argument returns the instrument to the setting prior to executing an autoselect.

### ■ VIDeo

This autoselects the displayed waveform.

### ■ VLines

This autoselects the displayed waveform.

### ■ VFields

This autoselects the displayed waveform.

## Example

```
AUTOSet VFields
```

This command sets the instrument to trigger on all video fields.

---

# AUXout:SOUrce

## Description

This command sets or queries the trigger source at the BNC connection. This command is equivalent to selecting AUX OUT Configuration from the Utilities menu and then selecting the desired Configuration setting.

## Group

Miscellaneous

## Related Commands

AUXout?

## Syntax 1

```
AUXout:SOUrce {ATRIGger|BTRIGger}
```

## Syntax 2

```
AUXout:SOUrce?
```

## Arguments

- ATRIGger

This sets the source at the BNC connector to the main trigger.

- BTRIGger

This sets the source at the BNC connector to the delayed trigger.

## Example

```
AUXout:SOUrce?
```

This query might return :AUXOUT:SOURCE ATRIGGER, indicating that the source at the BNC connector is set to the A trigger.



---

# AUXout?

## Description

This query-only command returns the auxiliary output setup and is equivalent to selecting External Signals... from the Utilities menu, and then viewing the current settings for the AUX OUT Configuration.

## Group

Miscellaneous

## Related Commands

AUXout:SOURce

## Syntax

AUXout?

## Example

AUXout?

This query might return

```
:AUXOUT:SOURCE ATRIGGER;EDGE RISING
```

indicating that the source at the BNC connector is set to the A trigger and the polarity is set to the rising edge of the trigger output signal.

---

# BEL1

## Description

This command was previously used to beep an audio indicator and is provided for backward compatibility.

## Group

Miscellaneous

## Syntax

BEL1

## Example

BEL1

This command is accepted but does nothing.

---

# BUSY?

## Description

This query-only command returns the status of the instrument. This command allows you to synchronize the operation of the instrument with your application program.

## Group

Status and Error

## Related Commands

\*OPC, \*WAI

## Syntax

BUSY?

## Returns

<NR1>

If 0 is returned, it means that the instrument is not busy processing a command whose execution time is extensive. If 1 is returned, it means that the instrument is busy processing one of the commands listed in the table below.

Commands that Affect BUSY? Response	
Operation	Command
Single sequence acquisition	ACQure:STATE ON or ACQure:STATE RUN or ACQure:STATE 1 (when ACQure:STOPAfter is set to SEQuence).
Hardcopy operation	HARDCopy STArt
Calibration step	Refer to the optional <i>TDS5000 Series Digital Phosphor Oscilloscopes Service Manual</i>

## Example

BUSY?

This query might return

:BUSY 1

This indicates that the instrument is currently busy.

---

# CAL?

## Description

This query-only command returns the status of instrument calibration, including internal signal path calibration, factory calibration, and probe calibration. This query also returns the time until notification when the next factory calibration is due. Refer to the service manual for factory calibration command descriptions.

## Group

Calibration

## Related Commands

CALibrate:RESults:SPC?

## Syntax

CAL?

## Example

CAL?

This query might return

```
:CALIBRATE:INTERNAL:STATUS PASS;:CALI-  
BRATE:TEMPERATURE 32;FACTORY:FANCONTROL 2;STA-  
TUS PASS;STEPSTATUS FAIL;STEPSTIMULUS  
"0.0,0,0,0,END";NOTIFY:HOURS 10500;YEARS  
5.5000;DUE DUE;:CALIBRATE:RESULTS:SPC  
PASS;FACTORY PASS;FACTORY:VOLTAGE PASS;FRE-  
QUENCY PASS;TRIGGER PASS;:CALIBRATE:PROBES-  
TATE:CH1 1;CH2 1;CH3 1;CH4 1
```

---

# CALibrate:CALProbe:CH<x>?

## Description

This query-only command instructs the instrument to perform a probe calibration for the selected channel and returns the calibration status. The Channel <x> range is 1 through 4. This command is equivalent to selecting Probe Cal from the Vertical menu. You must warm up the instrument for at least 20 minutes before running this command.

**Note:** A probe query takes approximately 30 seconds to complete. No other commands will be executed until calibration is complete.

## Group

Calibration

## Related Commands

CALibrate:PRObestate:CH<x>?, CH<x>:PROBECal?

## Syntax

CALibrate:CALProbe:CH<x>?

## Returns

- -1

Indicating the probe calibration failed for the selected channel.

- 0

Indicating the probe calibration passed for the selected channel.

- 1

Indicating the probe calibration is initialized for the selected channel.

## Example

CALibrate:CALProbe:CH2?

This query might return

:CALIBRATE:CALPROBE:CH2 0

indicating the probe calibration passed for Channel 2.

---

# CALibrate:INTERNAL

## Description

This command (no query form) starts the internal signal path calibration (SPC) of the instrument. You can use the CALibrate:INTERNAL:STATUS? query to return the current status of the internal signal path calibration of the instrument.

## Group

Calibration

## Related Commands

CALibrate:RESults:SPC?

## Syntax

```
CALibrate:INTERNAL
```

## Example

```
CALibrate:INTERNAL
```

This command starts the internal signal path calibration of the instrument.

---

# CALibrate:INTERNaL:START

## Description

This command (no query form) starts the internal signal path calibration (SPC) of the instrument. This command is the same as the CALibrate:INTERNaL command. You can use the CALibrate:INTERNaL:STATus? query to return the current status of the internal signal path calibration of the instrument.

## Group

Calibration

## Related Commands

CALibrate:RESults:SPC?

## Syntax

```
CALibrate:INTERNaL:START
```

## Example

```
CALibrate:INTERNaL:START
```

This command starts the internal signal path calibration of the instrument.

---

# CALibrate:INTERNAL:STATUS?

## Description

This query-only command returns the current status of the instrument internal signal path calibration for the last SPC operation.

## Group

Calibration

## Related Commands

\*CAL?

## Syntax

CALibrate:INTERNAL:STATUS?

## Returns

This query will return one of the following:

- INIT

Indicates the instrument has not had internal signal path calibration run.

- WARMUP

Indicates the instrument has not warmed up for the required time to perform signal path calibration.

- PASS

Indicates that the signal path calibration completed successfully.

- FAIL

Indicates that the signal path calibration did not complete successfully.

- TEMPDRIFT

Indicates that the instrument has not reached a state of stable temperature adequate for performing signal path calibration.

- RUNNING

Indicates that the signal path calibration is currently running.

## Example

CALibrate:INTERNAL:STATUS?

This query might return

```
:CALIBRATE:INTERNAL:STATUS INIT
```

indicating that the current status of the internal signal path calibration is that it has not been run.



---

# CALibrate:PROBEstate:CH<x>?

## Description

This query-only command returns the probe calibration status for the probe of the selected channel, 1 through 4.

## Group

Calibration

## Related Commands

CH<x>:PROBECal?

## Syntax

CALibrate:PROBEstate:CH<x>?

## Returns

- -1

Indicating the probe calibration failed for the selected channel.

- 0

Indicating the probe calibration passed for the selected channel.

- 1

Indicating the probe calibration is initialized for the selected channel.

- 2

Indicating that the probe calibration for the selected channel is running.

## Example

CALibrate:PROBEstate:CH1?

This query might return

:CALIBRATE:PROBESTATE:CH1 0

Indicating that the probe calibration for Channel 1 passed.

---

# CALibrate:RESults:SPC?

## Description

This query-only command returns the results of the last SPC operation. However, this query does not cause an SPC to be run. This command is equivalent to selecting Instrument Calibration from the Utilities menu.

## Group

Calibration

## Related Commands

\*CAL?

## Syntax

```
CALibrate:RESults:SPC?
```

## Example

```
CALibrate:RESults:SPC?
```

This query returns the results of the last SPC operation: either **PASS** or **FAIL**.

---

# CALibrate:RESults?

## Description

This query-only command returns the status of internal and factory calibrations, without performing any calibration operations. The results returned do not include the calibration status of attached probes. The CALibration:RESults? query is intended to support GO/NoGO testing of the oscilloscope calibration readiness: all returned results should indicate PASS status if the instrument is "fit for duty". It is quite common, however, to use uncalibrated probes (particularly when the instrument inputs are connected into a test system with coaxial cables).

This command is equivalent to selecting Instrument Calibration from the Utilities menu and then viewing the contents of the Status field.

## Group

Calibration

## Related Commands

\*CAL?

## Syntax

CALibrate:RESults?

## Example

CALibrate:RESults?

This query returns the status of all the calibration subsystems. The query might return :

```
:CALIBRATE:RESULTS:SPC PASS;FACTORY PASS;FACTORY:VOLTAGE PASS;FREQUENCY PASS;TRIGGER PASS
```

---

# CH<x>?

## Description

This query-only command returns the vertical parameters for the specified channel. The channel is specified by x. The value of x can range from 1 through 4 for four-channel instruments or 1 through 2 for two-channel instruments. This command is similar to selecting Vertical Setup from the Vertical menu.

## Group

Vertical

## Syntax

CH<x>?

## Example

CH1?

This query might return the following vertical parameters for channel 1:

```
:CH1:BANDWIDTH 1.0000E+09;COUPLING DC;DES-  
KEW0.0000E+00;OFFSET 0.0000E+00;POSITION  
0.0000E+00;SCALE 5.0000E-01;TERMINATION  
1.0000E+06;PROBCAL INIT;PROBE:GAIN  
1.0000E+00;RESISTANCE 1.0000E+06;UNITS  
"V";ID:TYPE "1X";SERNUMBER "N/A";:CH1:PROBE-  
FUNC:EXTATTEN 1.0000E+00;EXTUNITS  
"None";:CH1:LABEL:NAME "";XPOS 5;YPOS 5
```

---

# CH<x>:BANdwidth

## Description

This command sets or queries the selectable low-pass bandwidth limit filter of the specified channel. The channel is specified by x. The value of x can range from 1 through 4 for four-channel instruments or 1 through 2 for two-channel instruments. This is equivalent to selecting Bandwidth from the Vertical menu.

The query form of this command always returns the maximum bandwidth of the channel.

## Group

Vertical

## Syntax 1

```
CH<x>:BANdwidth {TWenty|ONEfifty|FIVE|FUL1|<NR3>}
```

## Syntax 2

```
CH<x>:BANdwidth?
```

## Arguments

- TWenty

This sets the upper bandwidth limit to 20 MHz.

- ONEfifty

This sets the upper bandwidth limit to 150 MHz.

- FIVE

This argument sets the upper bandwidth limit to 500 MHz.

- FUL1

This disables any optional bandwidth limiting. The specified channel operates at its maximum attainable bandwidth.

- <NR3>

This argument is a double-precision ASCII string. The instrument rounds this value to an available bandwidth using geometric rounding and then uses this value set the upper bandwidth.

---

### **Example 1**

```
CH1:BAWdwidth TWEnty
```

This command sets the bandwidth of channel 1 to 20 MHz.

### **Example 2**

```
CH2:BAWdwidth?
```

For TDS5052B and TDS5054B instruments, this query might return

```
:CH2:BAWdWIDTH 500.0000E+06
```

indicating that there is no bandwidth limiting on channel 2.

For TDS5104 instruments, this query might return

```
:CH2:BAWdWIDTH 1.0000E+09
```

indicating that there is no bandwidth limiting on channel 2.

---

# CH<x>:COUPling

## Description

This command sets or queries the input attenuator coupling setting for the specified channel. The channel is specified by x. The value of x can range from 1 through 4 for four-channel instruments or 1 through 2 for two-channel instruments. This command is equivalent to selecting Coupling from the Vertical menu.

## Group

Vertical

## Syntax 1

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

## Syntax 2

```
CH<x>:COUPling?
```

## Arguments

- AC

This sets the specified channel to AC coupling.

- DC

This sets the specified channel to DC coupling.

- GND

This sets the specified channel to ground. Only a flat, ground-level waveform will be displayed.

## Example 1

```
CH2:COUPling GND
```

This command sets channel 2 to ground.

## Example 2

```
CH3:COUPling?
```

This query might return

```
:CH3:COUPling DC
```

indicating that channel 3 is set to DC coupling.

---

# CH<x>:DESKew

## Description

This command sets or queries the deskew time for the specified channel. The channel is specified by x. The value of x can range from 1 through 4 for four-channel instruments or 1 through 2 for two-channel instruments. This command is equivalent to selecting Deskew from the Vertical Setup menu.

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

## Group

Vertical

## Syntax 1

```
CH<x>:DESKew <NR3>
```

## Syntax 2

```
CH<x>:DESKew?
```

## Argument

- <NR3>

This is the deskew time for this channel, ranging from -25 ns to +25 ns with a resolution of 1 ps. Out-of-range values are clipped.

## Example 1

```
CH4:DESKew 5.0E-9
```

This command sets the deskew time for channel 4 to 5 ns.

## Example 2

```
CH2:DESKew?
```

This query might return

```
:CH2:DESKEW 2.0000E-09
```

indicating that the deskew time for channel 2 is set to 2 ns.



---

# CH<x>:INVert

## Description

This command sets or queries the invert function for the specified channel. The channel is specified by x. The value of x can range from 1 through 4 for four-channel instruments or 1 through 2 for two-channel instruments. When on, the invert function inverts the waveform for the specified channel. This command is equivalent to selecting On or Off for the Invert function in the Vertical Setup control window.

**Note:** This command inverts the waveform for display purposes only. The instrument does not use an inverted waveform for triggers or trigger logic inputs.

## Group

Vertical

## Syntax 1

```
CH<x>:INVert {ON|OFF|NR1}
```

## Syntax 2

```
CH<x>:INVert?
```

## Argument

- OFF

This argument turns off the invert function for the specified channel.

- ON

This argument turns on the invert function for the specified channel.

- <NR1>

A 0 turns off the invert function; any other value turns on the invert function.

## Example 1

```
CH4:INVert ON
```

This command inverts the waveform on channel 4.

## Example 2

```
CH2:INVert?
```

This query might return

```
:CH2:INVERT 0
```

indicating that the invert function for channel 2 is off.

---

# CH<x>:LABel:NAME

## Description

This command sets or queries the label attached to the displayed waveform for the specified channel. The channel is specified by x. The value of x can range from 1 through 4 for four-channel instruments or 1 through 2 for two-channel instruments. This command is equivalent to selecting Label from the Vertical menu.

## Group

Vertical

## Syntax 1

```
CH<x>:LABel:NAME <str>
```

## Syntax 2

```
CH<x>:LABel:NAME?
```

## Argument

- <str>

This is an alphanumeric character string, ranging from 1 through 32 characters in length.

## Example 1

```
CH2:LABel:NAME "Pressure"
```

This command changes the waveform label for the CH2 waveform to "Pressure".

## Example 2

```
CH3:LABel:NAME?
```

This query might return

```
:CH3:LABEL:NAME "CH3"
```

indicating that the waveform label for the CH 3 waveform is "CH3".

---

# CH<x>:LABel:XPOS

## Description

This command sets or queries the X screen offset at which the label (attached to the displayed waveform of the specified channel) is displayed, relative to the left edge of the screen. The channel is specified by x. The value of x can range from 1 through 4 for four-channel instruments or 1 through 2 for two-channel instruments. This command is equivalent to selecting Label from the Vertical menu and either viewing or setting X Pos.

## Group

Vertical

## Syntax 1

```
CH<x>:LABel:XPOS <NR1>
```

## Syntax 2

```
CH<x>:LABel:XPOS?
```

## Argument

- <NR1>

This is the location (in pixels) where the waveform label for the selected channel is displayed, relative to the left edge of the screen. Arguments should be integers ranging from 0 through 500.

## Example 1

```
CH3:LABel:XPOS 50
```

This command moves the waveform label for the CH3 waveform so that it begins 50 pixels to the right of the left edge of the screen.

## Example 2

```
CH2:LABel:XPOS?
```

This query might return

```
:CH2:LABEL:XPOS 50
```

indicating that the waveform label for the CH2 waveform is currently 50 pixels to the right of the left edge of the screen.

---

# CH<x>:LABel:YPOS

## Description

This command sets or queries the Y screen offset at which the label (attached to the displayed waveform of the specified channel) is displayed, relative to the top edge of the screen. The channel is specified by x. The value of x can range from 1 through 4 for four-channel instruments or 1 through 2 for two-channel instruments. This command is equivalent to selecting Label from the Vertical menu and either viewing or setting Y Pos.

## Group

Vertical

## Syntax 1

```
CH<x>:LABel:YPOS <NR1>
```

## Syntax 2

```
CH<x>:LABel:YPOS?
```

## Argument

- <NR1>

This is the location (in pixels) where the waveform label for the selected channel is displayed, relative to the top edge of the screen. Arguments should be integers ranging from 0 to 385.

## Example 1

```
CH3:LABel:YPOS -25
```

This command moves the waveform label for the CH3 waveform to just beneath (25 pixels below) the top of the screen.

## Example 2

```
CH2:LABel:YPOS?
```

This query might return

```
:CH2:LABEL:YPOS 0
```

indicating that the waveform label for the CH2 waveform is currently located just beneath the top of the screen.

---

# CH<x>:OFFSet

## Description

This command sets or queries the vertical offset for the specified channel. The channel is specified by x. The value of x can range from 1 through 4 for four-channel instruments or 1 through 2 for two-channel instruments. This command is equivalent to selecting Offset from the Vertical menu.

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

Offset adjusts only the vertical center of the acquisition window for channel waveforms to help determine what data is acquired. The instrument always displays the input signal minus the offset value. The channel reference marker will move to the vertical graticule position given by the negative of the offset value divided by the scale factor, unless that position is off-screen. If the computed coordinate for the reference mark is off-screen, the mark moves to the nearest screen limit and changes from a right-pointing arrow (→) to an arrow pointing in the appropriate off-screen direction.

The settable range of a channel offset is either  $\pm 100$  V,  $\pm 10$  V or  $\pm 1.0$  V, depending on the vertical scale factor.

Vertical Scale Adjust Factors			
When vertical scale is:	From 0.001 to 0.1	From 0.101 to 1.0	From 1.01 to 100.0
Product of Offset range, probe gain and transducer gain is:	$\pm 1.0$ V	$\pm 10.0$ V	$\pm 100.0$ V

**Note:** The above table describes instrument behavior only when no probe is attached, and when the external attenuation factor is 1.0.

## Group

Vertical

## Related Commands

CH<x>POSition

## Syntax 1

CH<x>:OFFSet <NR3>

## Syntax 2

CH<x>:OFFSet?

---

## Argument

- <NR3>

This is the offset value for the specified channel.

### Example 1

```
CH3:OFFSet 2.0E-3
```

This command sets the offset for channel 3 to 2 mV.

### Example 2

```
CH4:OFFSet?
```

This query might return

```
:CH4:OFFSET 1.0000E-03
```

indicating that the offset for channel 4 is set to 1 mV.

---

# CH<x>:POSition

## Description

This command sets or queries the vertical position of the specified channel. The channel is specified by x. The value of x can range from 1 through 4 for four-channel instruments or 1 through 2 for two-channel instruments. The position value is applied to the signal before it is digitized. This command is equivalent to selecting Position/Scale from the Vertical menu and either viewing or setting Position.

Increasing the position value of a waveform causes the waveform to move up, and decreasing the position value causes the waveform to move down. Position adjusts only the display position of a waveform, whether it is a channel, math, or reference waveform. The position value determines the vertical graticule coordinate at which input signal values, minus the present offset setting for that channel, are displayed. For example, if the position for Channel 3 is set to 2.0 and the offset is set to 3.0, then input signals equal to 3.0 units are displayed 2.0 divisions above the center of the screen.

## Group

Vertical

## Related Commands

CH<x>:OFFSet, REF<x>:VERTical:POSition, MATH<x>:VERTical:POSition

## Syntax 1

CH<x>:POSition <NR3>

## Syntax 2

CH<x>:POSition?

## Argument

- <NR3>

This is the position value, in divisions from the center graticule, ranging from 8 to -8 divisions.

## Example 1

```
CH2:POSition 1.3E+00
```

This command positions the Channel 2 input signal 1.3 divisions above the center graticule.

## Example 2

```
CH1:POSition?
```

This query might return

```
:CH1:POSITION -1.3000E+00
```

indicating that the current position of Channel 1 is 1.3 divisions below the center graticule.

---

# CH<x>:PROBECal?

## Description

This query-only command returns the probe calibration state for the selected channel. The channel is specified by x. The value of x can range from 1 through 4 for four-channel instruments or 1 through 2 for two-channel instruments. This command is equivalent to selecting Probe Cal from the Vertical menu.

## Group

Vertical

## Related Commands

## Syntax

CH<x>:PROBECal?

## Outputs

- FAIL

This signifies that the probe calibration has failed for the selected channel.

- INIT

This signifies that the probe calibration has not yet been run for the selected channel.

- PASS

This signifies the probe calibration has passed for the selected channel.

## Example

CH2:PROBECal?

This query might return

:CH2:PROBECAL PASS

indicating that the probe calibration has passed for channel 2.



---

# CH<x>:PRObe:GAIN?

## Description

This query-only command returns the gain factor of the probe that is attached to the specified channel. The channel is specified by x. The value of x can range from 1 through 4 for four-channel instruments or 1 through 2 for two-channel instruments. The "gain" of a probe is the output divided by the input transfer ratio. For example, a common 10x probe has a gain of 0.1.

## Group

Vertical

## Related Commands

CH<x>:SCAlE

## Syntax

CH<x>:PRObe:GAIN?

## Example

CH2:PRObe:GAIN?

This query might return

:CH2:PROBE:GAIN 0.1000E+00

indicating that the attached 10x probe delivers 0.1 V to the channel 2 BNC for every 10 V applied to the probe input.

---

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

## Description

This query-only command returns the serial number of the probe that is attached to the specified channel. The channel is specified by x. The value of x can range from 1 through 4 for four-channel instruments or 1 through 2 for two-channel instruments.

**Note:** For Level 0 and 1 probes, the serial number will be "N/A".

## Group

Vertical

## Syntax

CH<x>:PRObe:ID:SERnumber?

## Example

CH1:PRObe:ID:SERnumber?

This query might return

:CH1:PROBE:ID:SERNUMBER "B010289"

indicating that the serial number of the probe attached to channel 1 is B010289.

---

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

## Description

This query-only command returns the type of probe that is attached to the specified channel. The channel is specified by x. The value of x can range from 1 through 4 for four-channel instruments or 1 through 2 for two-channel instruments. Level 2 (or higher) probes supply their exact product nomenclature; for Level 0 or 1 probes, a generic 'type string' of "nnX" is returned.

## Group

Vertical

## Syntax

```
CH<x>:PRObe:ID:TYPe?
```

## Example

```
CH1:PRObe:ID:TYPe?
```

This query might return

```
:CH1:PROBE:ID:TYPE "P6203"
```

indicating that P6203-type probe is attached to channel 1.

---

## CH<x>:PRObe:ID?

### Description

This query-only command returns the type and serial of the probe that is attached to the specified channel. The channel is specified by x. The value of x can range from 1 through 4 for four-channel instruments or 1 through 2 for two-channel instruments.

### Group

Vertical

### Syntax

CH<x>:PRObe:ID?

### Example

CH2:PRObe:ID?

This query might return

```
:CH2:PROBE:ID:TYPE "10X";SERNUMBER "N/A"
```

indicating that a passive 10x probe of unknown serial number is attached to channel 2.

---

# CH<x>:PRObe:RESistance?

## Description

This query-only command returns the resistance factor of the probe that is attached to the specified channel. The channel is specified by x. The value of x can range from 1 through 4 for four-channel instruments or 1 through 2 for two-channel instruments.

## Group

Vertical

## Syntax

```
CH<x>:PRObe:RESistance?
```

## Example

```
CH2:PRObe:RESistance?
```

This query might return

```
:CH2:PROBE:RESISTANCE 10.0000E+06
```

indicating that the input resistance of the probe attached to Channel 2 is 10.0 MΩ.

---

## CH<x>:PRObe:UNIts?

### Description

This query-only command returns a string describing the units of measure for the probe attached to the specified channel. The channel is specified by x. The value of x can range from 1 through 4 for four-channel instruments or 1 through 2 for two-channel instruments.

### Group

Vertical

### Related Commands

CH<x>:PROBEFunc:EXTUnits

### Syntax

CH<x>:PRObe:UNIts?

### Example

CH4:PRObe:UNIts?

This query might return

:CH4:PROBE:UNITS "V"

indicating that the units of measure for the probe attached to channel 4 are volts.

---

# CH<x>:PROBEFunc:EXTAtten

## Description

This command is used to specify the input-output ratio, of external attenuation or gain, between the signal and the instrument input channels.

The channel is specified by x. The value of x can range from 1 through 4 for four-channel instruments, or 1 through 2 for two-channel instruments.

The query form of this command returns the user-specified attenuation. Note that this command deals with an attenuation factor, not a gain factor, unlike CH<x>:PRObe? (This command returns a value independent of the external attenuation). For example, if you specify a 20x attenuation factor for channel 1, the commands return the following values (assuming that a 1x probe is presently attached, since the external attenuation is used in combination with the probe attenuation):

```
CH1:PRObe:EXTA? 20.00E+0
```

```
CH1:PRObe? 1.0E+0
```

This command is equivalent to selecting Attenuation from the Vertical menu, and then either viewing or setting Ext Atten.

## Group

Vertical

## Related Commands

CH<x>:PROBEFunc:EXTDBatten

## Syntax 1

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

## Syntax 2

```
CH<x>:PROBEFunc:EXTAtten?
```

## Argument

- <NR3>

This is the attenuation value, which is specified as a multiplier in the range from 1.00E-10 to 1.00E+10.

## Example 1

```
CH1:PROBEFunc:EXTAtten 167.00E-3
```

This command specifies an external attenuation, which is connected between the user's input signal and the input of the probe attached to channel 1.

---

## Example 2

CH2:PROBEFunc:EXTAtten?

This query might return

:CH2:PROBEFUNC:EXTATTEN 1.0000E+00

indicating that the probe attached to channel 2 is connected directly to the user's signal.



---

# CH<x>:PROBEFunc:EXTDBatten

## Description

Use this command to specify the input-output ratio (expressed in decibel units) of external attenuation or gain between the signal and the instrument input channels.

The channel is specified by x. The value of x can range from 1 through 4 for four-channel instruments, or 1 through 2 for two-channel instruments.

The query form of this command returns the user-specified attenuation in decibels: 1X = 0 dB, 10X = 20 dB, 100X = 40 dB, etc.

This command is equivalent to selecting Attenuation from the Vertical menu, and then either viewing or setting Ext Att(dB).

## Group

Vertical

## Related Commands

CH<x>:PROBEFunc:EXTAtten

## Syntax 1

CH<x>:PROBEFunc:EXTDBatten <NR3>

## Syntax 2

CH<x>:PROBEFunc:EXTDBatten?

## Argument

- <NR3>

This is the attenuation value, which is specified in the range from -200.00 to 200.00 dB.

## Example 1

```
CH3:PROBEFunc:EXTDBatten 2.5
```

This command specifies an external 2.5 dB attenuator on channel 3.

## Example 2

```
CH1:PROBEFunc:EXTDBatten?
```

This query might return

```
:CH1:PROBEFUNC:EXTDBATTEEN 2.5000E+00
```

indicating that the attenuation for channel 1 is 2.5 dB.

---

# CH<x>:PROBEFunc:EXTUnits

## Description

This command sets the unit of measurement for the external attenuator of the specified channel. The channel is specified by x. The value of x can range from 1 through 4 for four-channel instruments or 1 through 2 for two-channel instruments. There is also a corresponding query that returns the user-specified unit of measurement for the external attenuator. Unless these units are set to the factory default string value of "None", they become the attenuated units of measurement for that channel. It is assumed that the probe connected to the specified channel is of the correct type to receive the output of the user's external transducer or network.

## Group

Vertical

## Related Commands

CH<x>:PRObe:UNITS

## Syntax 1

```
CH<x>:PROBEFunc:EXTUnits <str>
```

## Syntax 2

```
CH<x>:PROBEFunc:EXTUnits?
```

## Argument

- <str>

This can contain a string of up to eight characters to indicate the attenuation unit of measurement for the specified channel. However, most instrument attenuators only display the first two characters.

## Example 1

```
CH4:PROBEFunc:EXTUnits "Pa"
```

This command sets the unit of measurement for the Channel 4 external attenuator.

## Example 2

```
CH2:PROBEFunc:EXTUnits?
```

This query might return

```
:CH2:PROBEFUNC:EXTUNITS "Pa"
```

indicating that the Channel 2 external attenuator units of measurement are Pascals.

---

# CH<x>:PRObe?

## Description

This query-only command returns all information concerning the probe that is attached to the specified channel. The channel is specified by x. The value of x can range from 1 through 4 for four-channel instruments or 1 through 2 for two-channel instruments.

## Group

Vertical

## Related Commands

CH<x>:PROBECal?

## Syntax

CH<x>:PROBE?

## Example

CH2:PROBE?

This query might return

```
:CH2:PROBE:GAIN 1.0000E-01; RESISTANCE  
1.0000E+07;UNITS "V";ID:TYPE "10X" 'SERNUMBER  
"N/A"
```

for a 10x probe, indicating that (among other parameters) the attenuation factor for the probe attached to channel 2 is 100.0 mV (assuming that probe units are set to volts).

---

# CH<x>:SCALE

## Description

This command sets or queries the vertical scale of the specified channel. The channel is specified by x. The value of x can range from 1 through 4 for four-channel instruments or 1 through 2 for two-channel instruments. Sending this command is equivalent to selecting Vertical Setup from the Vertical menu and then viewing or setting the Scale.

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

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

- For channel waveforms, this setting controls the vertical size of the acquisition window as well as the display scale. The range and resolution of scale values depends on the probe attached and any other external factors you have specified.
- For reference and math waveforms, this setting controls the display only, graphically scaling these waveforms and having no affect on the acquisition hardware.

## Group

Vertical

## Related Commands

CH<x>OFFSet, CH<x>:POSition, REF<x>:VERTical:SCALE, MATH<x>:VERTical:SCALE

## Syntax 1

CH<x>:SCALE <NR3>

## Syntax 2

CH<x>:SCALE?

## Argument

- <NR3>

This is the vertical channel scale in units per division.

## Example 1

CH4:SCALE 100E-03

This command sets the channel 4 scale to 100 mV per division.

---

## Example 2

CH2:SCALE?

This query might return

:CH2:SCALE 1.0000E+00

indicating that the current scale setting of channel 2 is 1 V per division.

---

# CH<x>:TERmination

## Description

This command sets the connected/disconnected status of a 50 Ω resistor, which may be connected between the specified channel's coupled input and instrument ground. The channel is specified by x. The value of x can range from 1 through 4 for four-channel instruments or 1 through 2 for two-channel instruments. There is also a corresponding query that requests the termination parameter and translates this enumeration into one of the two float values. This command is equivalent to selecting Termination from the Vertical menu or toggling between termination values from the VERTICAL area of the front panel.

## Group

Vertical

## Syntax 1

```
CH<x>:TERmination <NR3>
```

## Syntax 2

```
CH<x>:TERmination?
```

## Argument

- <NR3>

This specifies the channel input resistance, which can be specified as 50 or 1,000,000 Ω.

## Example 1

```
CH4:TERmination 50.0E+0
```

This command establishes 50 Ω impedance on channel 1.

## Example 2

```
CH2:TERmination?
```

This query might return

```
:CH2:TERMINATION 50.0E+0
```

indicating that channel 2 is set to 50 Ω impedance.

---

# CMDBatch

## Description

This command sets or queries the state of command batching. By batching commands, database transactions can be optimized, increasing command throughput. Also, batching allows for ALL commands in an individual batch to be order independent and accomplish the same result as if the commands were coupled.

The Batch state is persistent and will be saved across power cycles, but will not be saved and recalled as part of a setup. In a setup scenario, the factory initial value is enabled.

## Group

Miscellaneous

## Syntax 1

```
CMDBatch {OFF|ON|<NR1>}
```

## Syntax 2

```
CMDBatch?
```

## Arguments

- OFF

This turns command batching off.

- ON

This turns command batching on.

- <NR1>

A 0 turns command batching off; any other value turns command batching on.

## Example 1

```
CMDBatch OFF
```

This command disables command batching.

## Example 2

```
CMDBatch?
```

This query might return

```
:CMDBATCH 1
```

indicating that command batching is turned on.

---

# CURSor:FUNcTION

## Description

This command sets or queries the cursor type. Sending this command is equivalent to selecting Cursor Type from the Cursors menu, and then choosing from the drop-down list.

## Group

Cursor

## Related Commands

CURSor:STAtE

## Syntax 1

```
CURSor:FUNcTION {OFF | HBArS | VBArS | SCREEN | WAVE-  
form}
```

## Syntax 2

```
CURSor:FUNcTION?
```

## Arguments

### ■ OFF

Removes the cursors from the display but does not change the cursor type.

### ■ HBArS

Specifies horizontal bar cursors, which measure in vertical units.

### ■ VBArS

Specifies vertical bar cursors, which measure in horizontal units.

### ■ SCREEN

Specifies both horizontal and vertical bar cursors, which measure in horizontal and vertical units specified by the Cursor 1 and Cursor 2 Sources. Use these cursors to measure anywhere in the waveform display area.

### ■ WAVEform

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

## Example 1

```
CURSOR:FUNcTION VBArS
```

This command selects the vertical bar cursors type



---

## Example 2

`CURSOR:FUNCTION?`

This query might return

`:CURSOR:FUNCTION HBArS`

indicating that the horizontal bar cursors are currently selected.

---

# CURSor:HBArS?

## Description

This query-only command returns the current settings for the horizontal bar cursors.

## Group

Cursor

## Syntax

```
CURSor:HBArS?
```

## Example

```
CURSor:HBArS?
```

This command might return the horizontal bar setting as

```
:CURSOR:HBARS:POSITION1 320.0000E-03;POSITION2-320.0000E-03;UNITS BASE
```

---

# CURSor:HBArS:DELTA?

## Description

This query-only command returns the vertical difference between the two horizontal bar cursors.

## Group

Cursor

## Related Command

CURSor:HBArS:UNIts?

## Syntax

CURSor:HBArS:DELTA?

## Returns

A floating point value with an exponent.

## Example

CURSOR:HBARS:DELTA?

This command might return

:CURSOR:HBARS:DELTA 5.0800E+00

indicating that the voltage difference between the two cursors is 5.08 V.

---

# CURSor:HBArS:POSITION<x>

## Description

This command sets or queries the horizontal bar cursor position relative to ground, which is expressed in vertical units (usually volts). The cursor is specified by x, which can be 1 or 2. This command is the equivalent to selecting Cursor Position from the Cursors menu, selecting the H Bars Cursor Type, and then viewing or editing the desired cursor position.

## Group

Cursor

## Related Commands

CURSor:FUNcTion

## Syntax 1

```
CURSor:HBArS:POSITION<x> <NR3>
```

## Syntax 2

```
CURSor:HBArS:POSITION<x>?
```

## Argument

- <NR3>

Specifies the cursor position relative to ground.

## Example 1

```
CURSOR:HBARS:POSITION1 25.0E-3
```

This command positions Cursor 1 of the horizontal cursors at 25 mV.

## Example 2

```
CURSOR:HBARS:POSITION2?
```

This query might return

```
:CURSOR:HBARS:POSITION2 -64.0000E-03
```

indicating that Cursor 2 of the horizontal bar cursors is at -64 mV.

---

# CURSor:HBArS:UNIts?

## Description

This query-only command returns the units for the horizontal bar cursors.

This query always returns `BASE`, indicating that the units for the horizontal bar cursors are those of the base waveform to which they were assigned.

## Group

Cursor

## Syntax

```
CURSor:HBArS:UNIts?
```

## Example

```
CURSor:HBArS:UNIts?
```

This query always returns

```
:CURSOR:HBARS:UNITS BASE
```

indicating that the units for the horizontal bar cursors are base.

---

# CURSOR:LINEStyle

## Description

This command sets or queries the cursors line style used when cursors are displayed.

## Group

Cursor

## Syntax 1

```
CURSOR:LINEStyle {SOLID|DASHed}
```

## Syntax 2

```
CURSOR:LINEStyle?
```

## Arguments

- SOLID

Displays the cursors as solid lines.

- DASHed

Displays the cursors as dashed lines.

## Example 1

```
CURSOR:LINEStyle SOLID
```

This command displays the cursors as solid lines.

## Example 2

```
CURSOR:LINEStyle?
```

This query might return

```
:CURSOR:LINESTYLE DASHED
```

indicating that the cursors are displayed as dashed lines.

---

# CURSOR:MODE

## Description

This command sets or queries whether the two cursors move together in unison or separately. This command is the equivalent of selecting Cursor Mode from the Cursors menu, and then choosing from the drop-down list.

## Group

Cursor

## Syntax 1

```
CURSOR:MODE {TRACK|INDEPENDENT}
```

## Syntax 2

```
CURSOR:MODE?
```

## Arguments

- TRACK

Ties the navigational functionality of the two cursors together.

For cursor 1 adjustments, this ties the movement of the two cursors together; however, cursor 2 continues to move independently of cursor 1.

- INDEPENDENT

Allows independent adjustment of the two cursors.

## Example 1

```
CURSOR:MODE TRACK
```

This command specifies that the cursor positions move in unison.

## Example 2

```
CURSOR:MODE?
```

This query might return

```
:CURSOR:MODE TRACK
```

indicating that the two cursors move in unison.

---

# CURSOR:SCREEN:STYLE

## Description

This command sets or queries the cursor type for Screen mode.

## Group

Cursor

## Syntax 1

```
CURSOR:SCREEN:STYLE {LINE_X|LINES|X}
```

## Syntax 2

```
CURSOR:SCREEN:STYLE?
```

## Arguments

- LINE\_X

Specifies the cursor style to be a line with superimposed X.

- LINES

Specifies the cursor style to be a line.

- X

Specifies the cursor style to be an X.

## Example 1

```
CURSOR:SCREEN:STYLE LINES
```

This command sets the cursor style to lines.

## Example 2

```
CURSOR:SCREEN:STYLE?
```

This query might return

```
:CURSOR:SCREEN:STYLE X
```

Indicating that the style for screen cursors is X.



---

# CURSor:SOUrce<x>

## Description

This command sets or queries the source(s) for the currently selected cursor type (H Bars, V Bars, Waveform, or Screen). The cursor is specified by x, which can be 1 or 2. If the cursor is not specified, it defaults to cursor 1.

This command is equivalent to selecting Cursor Setup from the Cursors menu, and then choosing the desired cursor source.

## Group

Cursor

## Syntax 1

```
CURSor:SOUrce {CH<x>|MATH<x>|REF<x>}
```

## Syntax 2

```
CURSor:SOUrce?
```

## Arguments

### ■ CH<x>

Sets an input channel waveform as the cursor source. The valid channel waveform range is from 1 through 4 for four-channel instruments or 1 through 2 for two channel-instruments.

### ■ MATH<x>

Sets a math waveform as the cursor source. The valid math waveform range is from 1 through 4 for four-channel instruments or 1 through 2 for two channel-instruments.

### ■ REF<x>

Sets a reference waveform as the cursor source. The valid reference waveform range is from 1 through 4 for four-channel instruments or 1 through 2 for two channel-instruments.

## Returns

Any valid waveform.

## Example 1

```
CURSor:SOUrce1 CH2
```

This command sets the Cursor1 source to channel 2.

---

## **Example 2**

`CURSOR:SOURCE1?`

This query might return

`:CURSOR:SOURCE1 CH2`

indicating that the Cursor1 source is channel 2.

## **Example 3**

`CURSOR:SOURCE?`

This query might return

`:CURSOR:SOURCE1 CH2`

indicating that the Cursor1 source is channel 2. If you do not specify the cursor, cursor 1 is returned as the source.

---

# CURSor:STATE

## Description

This command sets or queries the state of cursors. Note that setting the cursor state does not modify the cursor type.

This command is equivalent to pressing the CURSOR button on the front panel.

## Group

Cursor

## Related Commands

CURSor:FUNcTION OFF

## Syntax 1

```
CURSor:STATE {ON|OFF|<NR1>}
```

## Syntax 2

```
CURSor:STATE?
```

## Arguments

- ON

Displays the cursors.

- OFF

Removes cursors from the display.

- <NR1>

A 0 disables the display cursors; any other value enables the display cursors.

## Example 1

```
CURSor:STATE ON
```

This command displays the cursors.

## Example 2

```
CURSor:STATE?
```

This query might return

```
:CURSOR:STATE 1
```

indicating that the cursors are active on the display.

---

# CURSor:VBArS

## Description

This command returns the current settings for the vertical bar cursors or forces them to snap to positions specified by the DATA:START and DATA:STOP commands.

## Group

Cursor

## Related Commands

DATA:START, DATA:STOP

## Syntax 1

CURSor:VBArS?

Returns the units and positions of the vertical bar cursors.

## Syntax 2

CURSor:VBArS SNAp

## Argument

- SNAp

Forces the position of vertical bar cursor 1 and 2 to snap to the waveform record points specified by the DATA:START and DATA:STOP commands, respectively.

## Example 1

CURSor:VBArS?

This query might return the following vertical bar settings

```
:CURSOR:VBARS:UNITS SECONDS;POSITION1  
1.0000E-06;POSITION2 9.0000E-06
```

## Example 1

CURSor:VBArS SNAp

This command forces vertical bar cursor1 to the DATA:START position and vertical bar cursor2 to the DATA:STOP position.

---

# CURSor:VBArS:DELTA?

## Description

This query-only command returns the horizontal difference between the two vertical bar cursors. The units are specified by the CURSor:VBArS:UNIts command. The position can appear in units of Base or 1/Base.

## Group

Cursor

## Related Commands

CURSor:VBArS:UNIts

## Syntax

CURSor:VBArS:DELTA?

## Returns

A floating point value with an exponent.

## Example

CURSor:VBArS:DELTA?

This command might return

```
:CURSOR:VBARS:DELTA 1.0640E+00
```

indicating that the time between the vertical bar cursors is 1.064 s.

---

# CURSor:VBArS:POSITION<x>

## Description

This command sets or queries the horizontal position for vertical bar cursors. The cursor is specified by x, which can be 1 or 2. Values are with respect to trigger position or the zero reference point for the designated waveform (if horizontal units are not set to time). Use the CURSor:VBArS:UNIts command to specify units. The position can appear in units of base or 1/base.

This command is the equivalent of selecting Cursor Setup from the Cursors menu, selecting V Bars Cursor Type, and then viewing or editing the desired cursor position.

## Group

Cursor

## Related Commands

CURSor:VBArS:UNIts

## Syntax 1

```
CURSor:VBArS:POSITION<x> {<NR3>}
```

## Syntax 2

```
CURSor:VBArS:POSITION<x>?
```

## Argument

- <NR3>

Specifies the cursor position.

## Returns

A floating point value with an exponent.

## Example 1

```
CURSor:VBArS:POSITION2 9.00E-6
```

Positions the cursor2 vertical bar cursor at 9 ms.

## Example 2

```
CURSor:VBArS:POSITION1?
```

This command might return

```
:CURSOR:VBARS:POSITION1 1.0000E-06
```

indicating that the cursor1 vertical bar is positioned at 1 ms.

---

# CURSor:VBArS:UNIts

## Description

This command sets or queries the units for the vertical bar cursors.

## Group

Cursor

## Syntax 1

```
CURSor:VBArS:UNIts {SECOnds|HERtz}
```

## Syntax 2

```
CURSor:VBArS:UNIts?
```

## Arguments

- SECOnds

Sets the units of the vertical bar cursors for the time domain (seconds).

- HERtz

Sets the units of the vertical bar cursors for the frequency domain (hertz).

## Returns

A string for SECOnds or HERtz, depending on the current vertical bar cursor units.

## Example 1

```
CURSor:VBArS:UNIts HERtz
```

Sets the units of the VBArS cursors to 1/seconds.

## Example 2

```
CURSor:VBArS:UNIts?
```

This command might return

```
:CURSOR:VBARS:UNITS SECONDS
```

indicating that the units for the vertical bar cursor are currently set to seconds.

---

# CURSor:WAVEform:HDELTA?

## Description

This query-only command returns the horizontal difference between the waveform cursors. This is the absolute value of the horizontal position of the first cursor minus the horizontal position of the second cursor. This is equivalent to the waveform delta-time readout value.

## Group

Cursor

## Syntax

```
CURSor:WAVEform:HDELTA?
```

## Example

```
CURSor:WAVEform:HDELTA?
```

This query might return

```
:CURSOR:WAVEFORM:HDELTA 1.03E-9
```

indicating that the time between the waveform cursors is 1.03 ns.



---

# CURSor:WAVEform:HPOS<x>?

## Description

This query-only command returns the position of the specified waveform cursor. The cursor is specified by x, which can be 1 or 2. This corresponds to the v1 or v2 (cursor 1 or cursor 2) cursor readout.

## Group

Cursor

## Syntax

```
CURSor:WAVEform:HPOS<x>?
```

## Example

```
CURSor:WAVEform:HPOS1?
```

This query might return

```
:CURSOR:WAVEFORM:HPOS2 0.247344970703
```

indicating that cursor 2 is at 247.3 mV with respect to ground on the source waveform.

---

# CURSor:WAVEform:POSition<x>

## Description

This command sets or queries the position of a waveform cursor, which is constrained to be visible in the selected time base. (Use the CURSor:FUNcTION command to set the cursor function to Waveform.) The cursor is specified by x, which can be 1 or 2. This is equivalent to setting a value in the Position control in the Cursor1 or Cursor2 sections of the Cursor Setup dialog box (when Waveform is selected in the Function section).

## Group

Cursor

## Related Commands

CURSor:FUNcTION

## Syntax 1

```
CURSor:WAVEform:POSition<x> {<NR3>}
```

## Syntax 2

```
CURSor:WAVEform:POSition<x>?
```

## Arguments

- <NR3>

Specifies the cursor position measured relative to the time of the trigger point of the source waveform.

## Example 1

```
CURSor:WAVEform:POSition2 9.00E-6
```

## Example 2

```
CURSor:WAVEform:POSition2?
```

This query might return

```
:CURSOR:WAVEFORM:POSITION2 51.0701402806E-009
```

indicating waveform cursor 1 is at 51.1 ns.

---

# CURSor:WAVEform:SOUrce<x>

## Description

This command sets or queries the source for a waveform cursor. The cursor is specified by x, which can be 1 or 2.

## Group

Cursor

## Syntax 1

```
CURSor:WAVEform:SOUrce  
{CH<x>|MATH<x>|REF<x>}
```

## Syntax 2

```
CURSor:SOUrce?
```

## Arguments

### ■ CH<x>

Sets an input channel waveform as the cursor source. The valid channel waveform range is from 1 through 4 for four-channel instruments or 1 through 2 for two channel-instruments.

### ■ MATH<x>

Sets a math waveform as the cursor source. The valid math waveform range is from 1 through 4 for four-channel instruments or 1 through 2 for two channel-instruments.

### ■ REF<x>

Sets a reference waveform as the cursor source. The valid reference waveform range is from 1 through 4 for four-channel instruments or 1 through 2 for two channel-instruments.

## Example 1

```
CURSor:WAVEform:SOUrce CH2
```

This command sets the Cursor1 source to channel 2.

## Example 2

```
CURSor:WAVEform:SOUrce?
```

This query might return

```
:CURSOR:WAVEFORM:SOURCE1 CH2
```

indicating that the Cursor1 source is channel 2.

---

# CURSor:WAVEform:STYle

## Description

This command sets or queries the cursor type for Waveform mode.

## Group

Cursor

## Syntax 1

```
CURSor:WAVEform:STYle {LINE_X|LINES|X}
```

## Syntax 2

```
CURSor:WAVEform:STYle?
```

## Arguments

- LINE\_X

Specifies the cursor style to be a line with superimposed X.

- LINES

Specifies the cursor style to be a line.

- X

Specifies the cursor style to be an X.

## Example 1

```
CURSor:WAVEform:STYle LINES
```

This command sets the cursor style to lines.

## Example 2

```
CURSor:WAVEform:STYle?
```

This query might return

```
:CURSOR:WAVEFORM:STYLE X
```

indicating that the style for waveform cursors is X.

---

# CURSor:WAVEform:UNIts

## Description

This command sets or queries the units for waveform cursors.

## Group

Cursor

## Syntax 1

```
CURSor:WAVEform:UNIts BASE
```

## Syntax 2

```
CURSor:WAVEform:UNIts?
```

## Arguments

- BASE

Sets cursor units to base.

## Example 1

```
CURSor:WAVEform:UNIts BASE
```

This command sets the waveform cursor units to base.

## Example 2

```
CURSor:WAVEform:UNIts?
```

This query might return

```
:CURSOR:WAVEFORM:UNITS BASE
```

indicating that the units for waveform cursors are base.

---

# CURSor:WAVEform:VDELTA?

## Description

This query-only command returns the vertical difference between the waveform cursors. This is the absolute value of the vertical position of the first cursor minus the vertical position of the second cursor. This is equivalent to the waveform delta-voltage readout value.

## Group

Cursor

## Syntax

```
CURSor:WAVEform:VDELTA?
```

## Example

```
CURSor:WAVEform:VDELTA?
```

This command might return

```
CURSOR:WAVEFORM:VDELTA 1.06E-3
```

indicating that the voltage between the waveform cursors is 1.06 mV.

---

# CURSor:WAVEform

## Description

This command returns the current settings for the waveform cursors or forces them to snap to positions specified by the DATA:START and DATA:STOP commands.

## Group

Cursor

## Related Commands

DATA:START, DATA:STOP

## Syntax 1

```
CURSor:WAVEform?
```

Returns the units and positions of the waveform cursors.

## Syntax 2

```
CURSor:WAVEform SNAp
```

## Argument

- SNAp

Forces the position of waveform cursor 1 and 2 to snap to the waveform record points specified by the DATA:START and DATA:STOP commands, respectively.

## Example 1

```
CURSor:WAVEform?
```

This query might return the following waveform cursor settings

```
:CURSOR:WAVEFORM:UNITS SECONDS;POSITION1  
1.0000E-06;POSITION2 9.0000E-06
```

## Example 2

```
CURSor:WAVEform SNAp
```

This command forces waveform cursor1 to the DATA:START position and waveform cursor2 to the DATA:STOP position.

---

# CURSor:XY:PRODELta?

## Description

This query-only command returns the product of the difference between the cursors X positions and Y positions when the CURSor:XY:READOUT is set to PRODUct. The  $\Delta X \times \Delta Y$  value is calculated as (Cursor 2 X – Cursor 1 X) X (Cursor 2 Y – Cursor 1 Y).

**Note:** This query is only valid when the CURSor:XY:READOUT is set to PRODUct. If the readout is set other than product, the IEEE representation of NAN (Not a Number) is returned: 99.1000E+36.

## Group

Cursor

## Syntax

```
CURSor:XY:PRODELta?
```

## Example

```
CURSor:XY:PRODELta?
```

This query might return

```
:CURSOR:XY:PRODELTA -24.9856
```

indicating that the  $\Delta X \times \Delta Y$  value is  $-24.9856$  VV.



---

# CURSOR:XY:PRODUCT<x>?

## Description

This query-only command returns the product of the X and Y positions for the specified cursor when the CURSOR:XY:READOUT is set to PRODUCT. The cursor is specified by x, which can be 1 or 2.

**Note:** This query is only valid when the CURSOR:XY:READOUT is set to PRODUCT. If the readout is set other than product, the IEEE representation of NAN (Not a Number) is returned: 99.1000E+36.

## Group

Cursor

## Syntax

```
CURSOR:XY:PRODUCT<x>?
```

## Example

```
CURSOR:XY:PRODUCT1?
```

This query might return

```
:CURSOR:XY:PRODUCT1 806.4004E-3
```

indicating that the product of the X position times the Y position for Cursor 1 is 806.4004 mV.

---

## CURSOR:XY:RADIUS<x>?

### Description

This query-only command returns the radius of the specified cursor when the CURSOR:XY:READOUT is set to POLARCoord. The cursor is specified by x, which can be 1 or 2.

**Note:** This query is only valid when the CURSOR:XY:READOUT is set to POLARCoord. If the readout is set other than polar coordinates, the IEEE representation of NAN (Not a Number) is returned: 99.1000E+36.

### Group

Cursor

### Syntax

```
CURSOR:XY:RADIUS<x>?
```

### Example

```
CURSOR:XY:RADIUS2?
```

This query might return

```
:CURSOR:XY:RADIUS2 4.9600
```

indicating that the Cursor 2 radius is 4.9600 V.

---

# CURSor:XY:RATDELta?

## Description

This query-only command returns ratio of the difference between the cursors X position and Y position ( $\Delta Y \div \Delta X$ ) when the CURSor:XY:READOUT is set to RATIo. The ratio is calculated as  $(\text{Cursor 2 Y} - \text{Cursor 1 Y}) \div (\text{Cursor 2 X} - \text{Cursor 1 X})$ .

**Note:** This query is only valid when the CURSor:XY:READOUT is set to RATIo. If the readout is set other than ratio, the IEEE representation of NAN (Not a Number) is returned: 99.1000E+36.

## Group

Cursor

## Syntax

```
CURSor:XY:RATDELta?
```

## Example

```
CURSor:XY:RATDELta?
```

This query might return

```
:CURSOR:XY:RATDELTA -1.0167
```

indicating that the  $\Delta Y \div \Delta X$  ratio is -1.0167.

---

## CURSor:XY:RATIO<x>?

### Description

This query-only command returns ratio the of the X (horizontal) and Y (vertical) position for the specified cursor when the CURSor:XY:READOUT is set to RATIo. The cursor is specified by x, which can be 1 or 2. The ratio is calculated as  $Y \div X$ .

**Note:** This query is only valid when the CURSor:XY:READOUT is set to RATIo. If the readout is set other than ratio, the IEEE representation of NAN (Not a Number) is returned: 99.1000E+36.

### Group

Cursor

### Syntax

```
CURSor:XY:RATIO<x>?
```

### Example

```
CURSor:XY:RATIO1?
```

This query might return

```
:CURSOR:XY:RATIO1 15.8730E-3
```

indicating that the ratio of the X and Y positions for Cursor 1 is 15.8730E-3.

---

# CURSor:XY:RDELta?

## Description

This query-only command returns the delta radius ( $\Delta r$ ) value when the CURSor:XY:READOUT is set to POLARCoord.

**Note:** This query is only valid when the CURSor:XY:READOUT is set to POLARCoord. If the readout is set other than polar coordinates, the IEEE representation of NAN (Not a Number) is returned: 99.1000E+36.

## Group

Cursor

## Syntax

```
CURSor:XY:RDELta?
```

## Example

```
CURSor:XY:RDELta?
```

This query might return

```
:CURSOR:XY:RDELTA 6.9600
```

indicating that the difference between the cursors radius is 6.9600 V.

---

# CURSor:XY:READOUT

## Description

This command sets or queries the XY cursor readout (available when DISPlay:FORMat is set to XY).

## Group

Cursor

## Related Commands

DISPlay:FORMat

## Syntax 1

```
CURSor:XY:READOUT {RECTangular|POLAR-Coord|PRODUct|RATio}
```

## Syntax 2

```
CURSor:XY:READOUT?
```

## Arguments

### ■ RECTangular

Displays the following values: X1, X2, ΔX, Y1, Y2, ΔY, t1, t2, Δt.

### ■ POLARCoord

Displays the following values: r1, r2, Δr, θ1, θ2, Δθ, t1, t2, Δt.

### ■ PRODUct

Displays the following values: X1 x Y1, X2 x Y2, ΔX x ΔY, t1, t2, Δt.

### ■ RATio

Displays the following values: Y1 ÷ X1, Y2 ÷ X2, ΔY ÷ ΔX, t1, t2, Δt.

## Example 1

```
CURSor:XY:READOUT RECTangular
```

This command sets the XY cursor readout to rectangular.

## Example 2

```
CURSor:XY:READOUT?
```

This query might return

```
:CURSOR:XY:READOUT RECTANGULAR
```

indicating that the XY cursor readout is set to rectangular mode.

---

# CURSor:XY:RECTX<x>

## Description

This command sets or queries the X cursor position in rectangular coordinates. The cursor is specified by x, which can be 1 or 2.

**Note:** This query is only valid when the CURSor:XY:READOUT is set to RECTangular. If the readout is set other than rectangular coordinates, the IEEE representation of NAN (Not a Number) is returned: 99.1000E+36.

## Group

Cursor

## Syntax 1

```
CURSor:XY:RECTX<x> {<NR3>}
```

## Syntax 2

```
CURSor:XY:RECTX<x>?
```

## Arguments

- <NR3>

This is the X position of the specified cursor in rectangular coordinates.

## Example 1

```
CURSor:XY:RECTX2 5.04
```

This command sets the X position of Cursor 2 to 5.04 V in rectangular coordinates.

## Example 2

```
CURSor:XY:RECTX1?
```

This query might return

```
:CURSOR:XY:RECTX1 5.1200
```

indicating that the X position of Cursor 1 is set to 5.1200 in rectangular coordinates.

---

# CURSor:XY:RECTY<x>

## Description

This command sets or queries the Y cursor position in rectangular coordinates. The cursor is specified by x, which can be 1 or 2.

**Note:** This query is only valid when the CURSor:XY:READOUT is set to RECTangular. If the readout is set to other than rectangular coordinates, the IEEE representation of NAN (Not a Number) is returned: 99.1000E+36.

## Group

Cursor

## Syntax 1

```
CURSor:XY:RECTY<x> {<NR3>}
```

## Syntax 2

```
CURSor:XY:RECTY<x>?
```

## Arguments

- <NR3>

This is the Y position of the specified cursor in rectangular coordinates.

## Example 1

```
CURSor:XY:RECTY1 2.5
```

This command sets the Y position of Cursor 1 to 2.5 V.

## Example 2

```
CURSor:XY:RECTY2?
```

This query might return

```
:CURSOR:XY:RECTY2 4.8800
```

indicating that the position of Cursor 2 in rectangular coordinates is 4.88 V.



---

# CURSor:XY:THDELta?

## Description

This query-only command returns the XY cursor angle delta in polar coordinates.

**Note:** This query is only valid when the CURSor:XY:READOUT is set to POLARCoord. If the readout is set other than polar coordinates, the IEEE representation of NAN (Not a Number) is returned: 99.1000E+36.

## Group

Cursor

## Syntax

```
CURSor:XY:THDELta?
```

## Example

```
CURSor:XY:THDELta?
```

This query might return

```
:CURSOR:XY:THDELTA 136.3748
```

indicating that the  $\Delta\theta$  XY cursor angle is 136.3748 degrees.

---

# CURSor:XY:THETA<x>?

## Description

This query-only command returns the XY cursor angle in polar coordinates. The cursor is specified by x, which can be 1 or 2.

**Note:** This query is only valid when the CURSor:XY:READOUT is set to POLARCoord. If the readout is set other than polar coordinates, the IEEE representation of NAN (Not a Number) is returned: 99.1000E+36.

## Group

Cursor

## Syntax

```
CURSor:XY:THETA<x>?
```

## Example

```
CURSor:XY:THETA2?
```

This query might return

```
:CURSOR:XY:THETA2 -89.0906
```

indicating that the Channel 2 XY cursor angle in polar coordinates is -89.0906 degrees.

---

# CURSor:XY:XDELta?

## Description

This query-only command returns the XY cursor X delta value in rectangular coordinates.

**Note:** This query is only valid when the CURSor:XY:READOUT is set to RECTangular. If the readout is set to other than rectangular coordinates, the IEEE representation of NAN (Not a Number) is returned: 99.1000E+36.

## Group

Cursor

## Syntax

```
CURSor:XY:XDELta?
```

## Example

```
CURSor:XY:XDELta?
```

This query might return

```
:CURSOR:XY:XDELTA 40.0000E-3
```

indicating that the  $\Delta X$  value is 40 mV.

---

# CURSor:XY:YDELta?

## Description

This query-only command returns the XY cursor Y delta value in rectangular coordinates.

**Note:** This query is only valid when the CURSor:XY:READOUT is set to RECTangular. If the readout is set to other than rectangular coordinates, the IEEE representation of NAN (Not a Number) is returned: 99.1000E+36.

## Group

Cursor

## Syntax

```
CURSor:XY:YDELta?
```

## Example

```
CURSor:XY:YDELta?
```

This query might return

```
:CURSOR:XY:YDELTA 40.0000E-6
```

indicating that the  $\Delta Y$  value is 40 microvolts.

---

# CURSor:XY?

## Description

This query-only command returns all of the XY cursor parameters.

**Note:** The values returned for all XY cursor readouts, that do not pertain to the current CURSor:XY:READOUT setting, are returned as the IEEE representation of NAN (Not a Number): 99.1000E+36.

## Group

Cursor

## Syntax

CURSor:XY?

## Example

CURSor:XY?

This query might return

```
:CURSOR:XY:PRODELTA 99.1000E+36;PRODUCT1
99.1000E+36;PRODUCT2 99.1000E+36;READOUT REC-
TANGULAR;RADIUS1 99.1000E+36;RADIUS2
99.1000E+36;RATDELTA 99.1000E+36;
RATIO 199.1000E+36;RATIO2 99.1000E+36;RDELTA
99.1000E+36;RECTX1 1.0000;RECTX2 1.0200;RECTY1
360.0001E-6;RECTY2360.0001E-6;
THDELTA 99.1000E+36;THETA1 99.1000E+36;THETA2
99.1000E+36;XDELTA 20.0000E-3;YDELTA 0.0000
```

---

# CURSor?

## Description

This query-only command returns all of the current cursor settings.

## Group

Cursor

## Syntax

CURSor?

## Example:

CURSOR?

This query might return the following as the current cursor settings:

```
:CURSOR:STATE 1;FUNCTION VBARS;LINESTYLE SOL-  
ID;MODE INDEPENDENT;SOURCE1 CH1;  
SOURCE2 CH1;SCREEN:STYLE LINES;  
:CURSOR:VBARS:UNITS SECONDS;POSITION1  
-1.6000E-6;POSITION2 1.6000E-6;  
:CURSOR:HBARS:POSITION1 300.0000E-3;  
POSITION2 -300.0000E-3;UNITS BASE;  
:CURSOR:WAVEFORM:POSITION1 -1.6000E-6;POSI-  
TION2 1.6000E-6;  
SOURCE2 CH1;UNITS BASE;STYLE LINE_X;:CURSOR:XY  
:PRODELTA 99.1000E+36;PRODUCT1  
99.1000E+36;PRODUCT2 99.1000E+36;  
READOUT RECTANGULAR;RADIUS1 99.1000E+36;RA-  
DIUS2 99.1000E+36;  
RATDELTA 99.1000E+36;RATIO1 99.1000E+36;RATIO2  
99.1000E+36;  
RDELTA 99.1000E+36;RECTX1 0.0000;RECTX2  
0.0000;RECTY1 0.0000;  
RECTY2 0.0000;THDELTA 99.1000E+36;THETA1  
99.1000E+36;  
THETA2 99.1000E+36;XDELTA 0.0000;YDELTA 0.0000
```

---

# CURVe

## Description

This command transfers waveform data to and from the instrument. Each waveform that is transferred has an associated waveform preamble that contains information such as data format and scale.

The CURVe? query transfers data from the instrument. The data source is specified by the DATA:SOURce command. The first and last data points are specified by the DATA:START and DATA:STOP commands.

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

## Group

Waveform Transfer

## Related Commands

DATA:DESTination, DATA:SOURce, DATA:START, DATA:STOP, SAVE:WAVEform, SAVE:WAVEform:FILEFormat, WFMInpre, WFMInpre:BYT\_Nr, WFMOutpre, WFMInpre:NR\_FR?, WFMOutpre:NR\_FR?

## Syntax 1

```
CURVe {<Block>|<asc curve>}
```

## Syntax 2

```
CURVe?
```

## Arguments

### ■ <Block>

This is the waveform data in binary format. The waveform is formatted as: #<x><yyy><data><newline>, where

<x> is the number of y bytes. For example, if <yyy>=500, then <x>=3)

<yyy> is the number of bytes to transfer. If width is 1, then all bytes on the bus are single data points. If width is 2, then all bytes on the bus are 2-byte pairs. If width is 4, then all bytes on the bus are 4-byte pairs. Use the WFMInpre:BYT\_Nr command to set the width.

<data> is the curve data.

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

### ■ <asc curve>

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

---

## Example 1

CURVe?

This query with ASCII encoding, start and stop of 1 and 10 respectively, and a width set to 1 might return

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

**Note:** Curve data is transferred from the instrument asynchronously and, dependent upon the length of the curve record, such transfers can require several seconds to complete. During this time, the instrument will not respond to user controls. You can interrupt these asynchronous data transfers by sending a device clear message to the instrument or by interrupting the query with another command or query. In order to verify that curve data has been completely transferred, it is recommended that you follow such queries with an \*ESR? Query and verify that the error bit returned and, if set, check the event queue to ascertain the reason for the error. If the error was caused by an interrupted query, then the asynchronous data transfer had not completed when the \*ESR? query was sent. In this case, you may need increase your program's timeout value in order to ensure that all data is transferred and read.

## Example 2

CURVe <Block>

This command sets the format of the waveform data, transferred to and from the instrument, to binary format.



---

# DATA

## Description

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

## Group

Waveform Transfer

## Related Commands

CURVe, DATA:START, DATA:STOP, DATA:ENCdg SAVE:WAVEform, SAVE:WAVEform:FILEFormat, WFMinpre:NR\_FR?, WFMinpre:NR\_Pt, WFMOupre:NR\_FR?, WFMOupre:NR\_Pt?

## Syntax 1

```
DATA {INIT|SNAP}
```

## Syntax 2

```
DATA?
```

## Argument

- INIT

This initializes the waveform data parameters to their factory defaults.

- SNAP

This sets DATA:START and DATA:STOP to match the current V Bar/ Paired/Split cursor positions.

## Example 1

```
DATA?
```

This query might return

```
:DATA:DESTINATION REF1:ENCDG RIBINARY;SOURCE  
CH1;START 1;STOP 500
```

## Example 2

```
DATA INIT
```

This command initializes the waveform data parameters to their factory defaults.

---

# DATA:DESTINATION

## Description

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

## Group

Waveform Transfer

## Related Commands

CURVe

## Syntax 1

```
DATA:DESTINATION REF<x>
```

## Syntax 2

```
DATA:DESTINATION?
```

## Argument

- REF<x>

This is the reference where the waveform will be stored. The reference number is specified by x, which ranges from 1 through 4.

## Example 1

```
DATA:DESTINATION?
```

This query might return

```
:DATA:DESTINATION REF3
```

indicating that reference 3 is the currently selected reference memory location for waveform data.

## Example 2

```
DATA:DESTINATION REF1
```

This command indicates that incoming waveform data be stored in reference 1.

---

# DATA:ENCdg

## Description

This command sets or queries the format of outgoing waveform data. This command is equivalent to setting WFMOutpre:ENCdg, WFMOutpre:BN\_Fmt, and WFMOutpre:BYT\_Or. Setting the DATA:ENGdg value causes the corresponding WFMOutpre values to be updated and visa versa.

**Note:** Values are constrained (for outbound data) to the format of the data specified by DATA:SOURce.

## Group

Waveform Transfer

## Related Commands

WFMOutpre:ENCdg, WFMOutpre:BN\_Fmt, WFMOutpre:BYT\_Or

## Syntax 1

```
DATA:ENCdg {ASCIi|FAStest|RIBinary|RPBina-  
ry|FPBinary|SRIBinary  
|SRPbinary|SFPbinary}
```

## Syntax 2

```
DATA:ENCdg?
```

## Argument

### ■•ASCIi

This specifies the ASCII representation of signed INT, FLOAT. If ASCII is the value, then :BN\_Fmt and :BYT\_Or are ignored.

### ■•FAStest

This specifies that the data be sent in the fastest possible manner consistent with maintaining accuracy and is interpreted with respect to the first waveform specified in the DATA:SOURce list.

:ENCdg will always be BIN, :BYT\_Or will always be LSB, but :BN\_Fmt and :BYT\_Nr will depend on the first DATA:SOURce waveform. :BN\_Fmt will be RI unless the waveform is internally stored as a floating point number, in which case the FP format will be used.

### ■•RIBinary

This specifies signed integer data point representation with the most significant byte transferred first.

When :BYT\_Nr is 1, the range is from -128 through 127. When :BYT\_Nr is 2, the range is from -32,768 through 32,767. When :BYT\_Nr is 8, then the waveform being queried has been set to Fast Acquisition mode. Center screen is 0 (zero). The upper limit is one division above the top of the screen and the lower limit is one division below the bottom of the screen. This is the default argument.

### ■•RPBinary

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

When :BYT\_Nr is 1, the range is from 0 through 255. When :BYT\_Nr is 2, the range is from 0 to 65,535. When :BYT\_Nr is 8, then the waveform being queried has been set to Fast Acquisition mode. The center of the screen is 127. The upper limit is one division above the top of the screen and the lower limit is one division below the bottom of the screen.

### ■•FPBinary

This specifies the floating point (width = 4) data.

The range is from  $-3.4 \times 10^{38}$  to  $3.4 \times 10^{38}$ . The center of the screen is 0. The upper limit is one division above the top of the screen and the lower limit is one division below the bottom of the screen.

The FPBinary argument is only applicable to math waveforms or ref waveforms saved from math waveforms.

### ■•SRBinary

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

### ■•SRPbinary

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

### ■•SFPbinary

This specifies floating point data in IBM PC format.

The SFPbinary argument only works on math waveforms or ref waveforms saved from math waveforms

DATA and WFMOUTPRE		Parameter Settings		
DATA:ENCdg Setting	WFMOUTPRE Settings			
	:ENCdg	:BN_Fmt	:BYT_Or	:BYT_NR
ASCii	ASC	N/A	N/A	1, 2, 4, 8
FASText	BIN	RI/FP	MSB	1, 2, 4
RBinary	BIN	RI	MSB	1, 2, 8
RPBinary	BIN	RP	MSB	1, 2, 8
FPBinary	BIN	FP	MSB	4
SRBinary	BIN	RI	LSB	1, 2, 8
SRPbinary	BIN	RP	LSB	1, 2, 8
SFPbinary	BIN	FP	LSB	4

---

### **Example 1**

`DATA:ENCdg?`

This query might return

`:DATA:ENCDG SRPBINARY`

for the format of the outgoing waveform data.

### **Example 2**

`DATA:ENGdg RPBinary`

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

---

# DATA:SOURce

## Description

This command sets or queries the location of waveform data that is transferred from the instrument by the CURVe? Query.

## Group

Waveform Transfer

## Related Commands

CURVe?

## Syntax 1

```
DATA:SOURce <wfm>[<,><wfm>]
```

## Syntax 2

```
DATA:SOURce?
```

## Argument

■•<wfm>

This is the location of the waveform data that will be transferred from the instrument to the controller.

## Example 1

```
DATA:SOURce?
```

This query might return

```
:DATA:SOURCE REF3
```

indicating that the source for the waveform data which is transferred using a CURVe? Query is reference 3.

## Example 2

```
DATA:SOURce CH1
```

This command specifies that the CH1 waveforms will be transferred in the next CURVe? query.

---

# DATA:START

## Description

This command sets or queries the starting data point for waveform transfer. This command allows for the transfer of partial waveforms to and from the instrument.

## Group

Waveform Transfer

## Related Commands

CURVe, DATA, DATA:STOP, SAve:WAVEform, SAve:WAVEform:FILEFormat, WFMInpre:NR\_Pt, WFMOutpre:NR\_Pt, WFMInpre:NR\_FR?, WFMOutpre:NR\_FR?

## Syntax 1

DATA:START <NR1>

## Syntax 2

DATA:START?

## Argument

- <NR1>

This is the first data point that will be transferred, which ranges from 1 to the record length. Data will be transferred from <NR1> to DATA:STOP or the record length, whichever is less. If <NR1> is greater than the record length, the last data point in the record is transferred.

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

## Example 1

DATA:START?

This query might return

:DATA:START 214

indicating that data point 214 is the first waveform data point that will be transferred.

## Example 2

DATA:START 10

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

---

# DATA:STOP

## Description

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

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

**Note:** When using the CURVe command, DATA:STOP is ignored and WFMinpre:NR\_Pt is used.

## Group

Waveform Transfer

## Related Commands

CURVe, DATA, DATA:START, SAVE:WAVEform, SAVE:WAVEform:FILEFormat, WFMinpre:NR\_Pt, WFMinpre:NR\_FR?, WFMinpre:NR\_FR?, WFMinpre:NR\_FR?, WFMinpre:NR\_FR?, WFMinpre:NR\_FR?, WFMinpre:NR\_FR?

## Syntax 1

DATA:STOP <NR1>

## Syntax 2

DATA:STOP?

## Argument

- <NR1>

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

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

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



---

### **Example 1**

`DATA:STOP?`

This query might return

`:DATA:STOP 14900`

indicating that 14900 is the last waveform data point that will be transferred.

### **Example 2**

`DATA:STOP 15000`

This command specifies that the waveform transfer will stop at data point 15000.

---

# DATE

## Description

This command sets or queries the date that the instrument can display. This command is equivalent to selecting Set Date & Time from the Utilities menu and then setting the fields in the Date group box.

## Group

Miscellaneous

## Related Commands

TIME

## Syntax 1

DATE <Qstring>

## Syntax 2

DATE?

## Argument

- <QString>

This is a date in the form "yyyy-mm-dd" where yyyy refers to a four-digit year number, mm refers to a two-digit month number from 01 to 12, and dd refers to a two-digit day number in the month.

## Example 1

```
DATE "2000-01-24"
```

This command specifies that the date is set to January 24, 2000.

## Example 2

```
DATE?
```

This query might return

```
:DATE 2000-01-24
```

indicating the current date is set to January 24, 2000.

---

# DELEte:SETUp

## Description

This command (no query form) changes the setup to reference the factory setup instead of the specific user setup slot. The content of the setup slot is unchanged, but the data will no longer be accessible to you.

This command is equivalent to selecting Delete from the File menu, and then clicking the specific setup you want to delete (user setups are shown as Setup-User and are ordered in the list from 1 through 10, if defined) or All Setups.

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

## Group

Save and Recall

## Related Commands

\*RCL?, RECall:SETUp, \*RST, \*SAV, SAVe:SETUp, \*SDS

## Syntax

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

## Arguments

- ALL

This argument deletes all the stored setups.

- <NR1>

This argument specifies a setup storage location to delete. Setup storage location values range from 1 through 10; using an out-of-range value causes an error.

## Example

```
DELEte:SETUp ALL
```

This command deletes all stored setups. All ten storage locations are initialized to the factory default setup.

---

# DELEte:WAVEform

## Description

This command (no query form) deletes one or all stored reference waveforms from memory.

This command is equivalent to selecting Delete from the File menu, and then choosing the reference waveform you want to delete; choosing All Refs deletes all of the reference waveforms.

## Group

Save and Recall

## Related Commands

RECALL:WAVEform, Save:WAVEform

## Syntax

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

## Arguments

- ALL

This specifies to delete all the stored reference waveforms.

- REF<x>

This specifies to delete one of the reference memory locations. Reference memory location values range from 1 through 4.

## Example 1

```
DELEte:WAVEform ALL
```

This command removes all waveforms stored in reference memory.

## Example 2

```
DELEte:WAVEform REF2
```

This command removes the waveform stored at REF2.

---

# DESE

## Description

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

## Group

Status and Error

## Related Commands

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

## Syntax 1

```
DESE <NR1>
```

## Syntax 2

```
DESE?
```

## Argument

- <NR1>

The binary bits of the DESER are set according to this value, which ranges from 1 through 255. For example, DESE 209 sets the DESER to the binary value 11010001 (that is, the most significant bit in the register is set to 1, the next most significant bit to 1, the next bit to 0, etc.).

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

**Note:** Setting the DESER and ESER to the same value allows only those codes to be entered into the Event Queue and summarized on the ESB bit (bit 5) of the Status Byte Register. Use the \*ESE command to set the ESER.

## Example 1

```
DESE 209
```

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

## Example 2

```
DESE?
```

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

---

# DIAG:CONTROL:HALT

## Description

This command (no query form) determines whether the next execution of diagnostics will stop on the first diagnostic failure that occurs or will execute the selected set of diagnostic functions. This command is equivalent to selecting Instrument Diagnostics from the Utilities menu and then enabling Halt on Fail.

## Group

Diagnostics

## Related Commands

DIAG:CONTROL:LOOP

## Syntax

```
DIAG:CONTROL:HALT (ON|OFF|<NR1>}
```

## Arguments

- ON

This enables the halt function, causing the execution of diagnostics to halt at the first diagnostic failure that occurs.

- OFF

This disables the halt function, allowing the oscilloscope to execute the entire set of diagnostics before halting, even if diagnostic failure occurs.

- <NR1>

A 0 enables the halt function; any other value disables the halt function.

## Example

```
DIAG:CONTROL:HALT ON
```

This command enables the halt function, causing the execution of diagnostics to halt at the first diagnostic failure that occurs.

---

# DIAG:CONTROL:LOOP

## Description

This command (no query form) determines whether the next execution of diagnostics executes once or continuously loops on the selected set of diagnostics (assuming the halt control is set to off using the DIAG:CONTROL:HALT command or that the halt control is set to ON but no failures occur). This command is equivalent to selecting Instrument Diagnostics from the Utilities menu and then enabling Loop Control.

## Group

Diagnostics

## Related Commands

DIAG:CONTROL:HALT, DIAG:STOP

## Syntax

```
DIAG:CONTROL:LOOP {ON|OFF|<NR1>}
```

## Arguments

- ON

This enables the loop function, causing the execution of diagnostics to continuously loop.

- OFF

This disables the loop function, causing the oscilloscope to execute the entire set of diagnostics once and then halt.

- <NR1>

A 0 enables the loop function; any other value disables the loop function.

## Example

```
DIAG:CONTROL:LOOP ON
```

This command enables the loop function.

---

# DIAG:EXECUTE

## Description

This command (no query form) starts the execution of the currently selected set of diagnostics. This command is equivalent to selecting Instrument Diagnostics from the Utilities menu and then pressing Run.

## Group

Diagnostics

## Related Commands

DIAG:STATE

## Syntax

DIAG:EXECUTE

## Example

DIAG:EXECUTE

This command starts the execution of the entire set of diagnostics.



---

# DIAG:ITEM?

## Description

This query-only command returns the diagnostics settings. This command is equivalent to selecting Instrument Diagnostics from the Utilities menu, and then reviewing the diagnostics settings.

This query-only command can be issued while diagnostics are still in progress.

## Group

Diagnostics

## Related Commands

DIAG:ITEM:FAILURES?, DIAG:ITEM:NAME?, DIAG:ITEM:RESULT?, DIAG:ITEM:SUBITEMS?

## Syntax

```
DIAG:ITEM? <NR1>
```

## Arguments

- <NR1>

This sets the index item about which data will be returned, which ranges from 0 through 15.

## Example

```
DIAG:ITEM? 2
```

This query might return:

```
:DIAG:ITEM 2,"Acquisition","Pass",0,8
```

This indicates that the acquisition test passed and that there were 0 failures of the 8 subitems for that test.

---

# DIAG:ITEM:FAILURES?

## Description

This query-only command returns the total number of failures. This command is equivalent to selecting Instrument Diagnostics from the Utilities menu and then reviewing the Diagnostic Status.

This query-only command can be issued while diagnostics are still in progress.

## Group

Diagnostics

## Related Commands

DIAG:ITEM?, DIAG:ITEM:NAME?, DIAG:ITEM:RESULT?,  
DIAG:ITEM:SUBITEMS?

## Syntax

```
DIAG:ITEM:FAILURES? <NR1>
```

## Arguments

- <NR1>

This sets the index item about which data will be returned, which ranges from 0 through 15.

## Example

```
DIAG:ITEM:FAILURES? 2
```

This query might return:

```
:DIAG:ITEM:FAILURES 2,0
```

This indicates that there were 0 failures for item 2.

---

# DIAG:ITEM:NAME?

## Description

This query-only command returns the name of the selected menu item. This command is equivalent to selecting Instrument Diagnostics from the Utilities menu and then reviewing the Subsystem, Area and Test settings.

This query-only command can be issued while diagnostics are still in progress.

## Group

Diagnostics

## Related Commands

DIAG:ITEM?, DIAG:ITEM:FAILURES?, DIAG:ITEM:RESULT?,  
DIAG:ITEM:SUBITEMS?

## Syntax

```
DIAG:ITEM:NAME? <NR1>
```

## Arguments

- <NR1>

This sets the index item about which data will be returned, which ranges from 0 through 15.

## Example

```
DIAG:ITEM:NAME? 2
```

This query might return

```
:DIAG:ITEM:NAME 2,"Acquisition"
```

This indicates that the selected menu item (2) is Acquisition.

---

# DIAG:ITEM:RESULT?

## Description

This query-only command returns the result from the last execution of the item. This command is equivalent to selecting Instrument Diagnostics from the Utilities menu and then reviewing the Diagnostic Status.

This query-only command can be issued while diagnostics are still in progress.

## Group

Diagnostics

## Related Commands

DIAG:ITEM?, DIAG:ITEM:NAME?, DIAG:ITEM:FAILURES?,  
DIAG:ITEM:SUBITEMS?

## Syntax

```
DIAG:ITEM:RESULT? <NR1>
```

## Arguments

- <NR1>

This sets the index item about which data will be returned, which ranges from 0 through 15.

## Example 1

```
DIAG:ITEM:RESULT? 2
```

This query might return

```
:DIAG:ITEM:RESULT 2,"Pass"
```

This indicates that the item passed during the last execution.

## Example 2

```
DIAG:ITEM:RESULT? 2
```

This query might return

```
:DIAG:ITEM:RESULT 2,"Fail"
```

This indicates that the item failed during the last execution.

## Example 3

```
DIAG:ITEM:RESULT? 2
```

This query might return

```
:DIAG:ITEM:RESULT 2," "
```

This indicates that the item was not run.

---

# DIAG:ITEM:SUBITEMS?

## Description

This query-only command returns the number of sub-items associated with the item. This command is equivalent to selecting Instrument Diagnostics from the Utilities menu, choosing the Subsystem, Area or Test setting and then reviewing the resulting sub-items.

This query-only command can be issued while diagnostics are still in progress.

## Group

Diagnostics

## Related Commands

DIAG:ITEM?, DIAG:ITEM:NAME?, DIAG:ITEM:FAILURES?,  
DIAG:ITEM:RESULT?

## Syntax

```
DIAG:ITEM:SUBITEMS? <NR1>
```

## Arguments

- <NR1>

This sets the index item about which data will be returned, which ranges from 0 through 15.

## Example

```
DIAG:ITEM:SUBITEMS? 2
```

This query might return:

```
:DIAG:ITEM:SUBITEMS 2,8
```

This indicates that there are 8 subitems associated with the selected item.

---

# DIAG:LEVEL

## Description

This command sets or returns the currently selected level of diagnostic test hierarchy. This command is equivalent to selecting Instrument Diagnostics from the Utilities menu and then reviewing the Diagnostic Status.

## Group

Diagnostics

## Syntax 1

```
DIAG:LEVEL (SUBSYS|AREA|TEST}
```

## Syntax 2

```
DIAG:LEVEL?
```

## Arguments

- SUBSYS

This sets diagnostic testing to the subsystem level.

- AREA

This sets diagnostic testing to the area level.

- TEST

This sets diagnostic testing to the test level.

## Example 1

```
DIAG:LEVEL AREA
```

This command sets the level of diagnostic test hierarchy to Area.

## Example 2

```
DIAG:LEVEL?
```

This query might return:

```
:DIAG:LEVEL SUBSYSI
```

This indicates that the current level of diagnostic test hierarchy is Subsys.

---

# DIAG:LOOPS?

## Description

This query-only command returns the number of times that the selected diagnostics set was completed during the last diagnostic execution. This command is equivalent to selecting Instrument Diagnostics from the Utilities menu and then reviewing the Elapsed Loops.

This query-only command can be issued while diagnostics are still in progress.

## Group

Diagnostics

## Syntax

DIAG:LOOPS?

## Example

DIAG:LOOPS?

This query might return

:DIAG:LOOPS 5

This indicates that the selected set of diagnostics was completed five times during the last diagnostic execution.

---

# DIAG:NAME?

## Description

This query-only command returns the names of the subsystem, area, and test of the current diagnostic test. This command can be issued while diagnostics are still in progress.

## Group

Diagnostics

## Related Commands

DIAG:NAME:AREA?, DIAG:NAME:SUBSYS?, DIAG:NAME:TEST?

## Syntax

DIAG:NAME?

## Example

DIAG:NAME?

This query might return

```
:DIAG:NAME:SUBSYS "Processor";AREA  
"Memory";TEST"DRAMWalk1"
```

This indicates the subsystem name, area name, and test name of the currently selected diagnostic test.



---

# DIAG:NAME:AREA?

## Description

This query-only command returns the selected area of the current diagnostic test. There are three levels of diagnostic test hierarchy: subsystem, area and test. This command is equivalent to selecting Instrument Diagnostics from the Utilities menu and then reviewing the Diagnostic Status.

This query-only command can be issued while diagnostics are still in progress.

## Group

Diagnostics

## Related Commands

DIAG:NAME?, DIAG:NAME:SUBSYS?, DIAG:NAME:TEST?

## Syntax

DIAG:NAME:AREA?

## Example

DIAG:NAME:AREA?

This query might return

```
:DIAG:NAME:AREA "Memory"
```

This indicates the selected area name of the current diagnostic test.

---

# DIAG:NAME:SUBSYS?

## Description

This query-only command returns the subsystem of the current diagnostic test. This command is equivalent to selecting Instrument Diagnostics from the Utilities menu and then reviewing the Diagnostic Status.

This query-only command can be issued while diagnostics are still in progress.

## Group

Diagnostics

## Related Commands

DIAG:NAME?, DIAG:NAME:AREA?, DIAG:NAME:TEST?

## Syntax

DIAG:NAME:SUBSYS?

## Example

DIAG:NAME:SUBSYS?

This query might return

```
:DIAG:NAME:SUBSYS "Acquisition"
```

This indicates the subsystem name of the current diagnostic test.

---

# DIAG:NAME:TEST?

## Description

This query-only command returns the name of the current diagnostic test. This command is equivalent to selecting Instrument Diagnostics from the Utilities menu and then reviewing the Diagnostic Status.

This query-only command can be issued while diagnostics are still in progress.

## Group

Diagnostics

## Related Commands

DIAG:NAME?, DIAG:NAME:AREA?, DIAG:NAME:SUBSYS?

## Syntax

DIAG:NAME:TEST?

## Example

DIAG:NAME:TEST?

This query might return:

```
:DIAG:NAME:TEST "diagDataFormatConf"
```

This indicates the test name of the current diagnostic test.

---

# DIAG:NUMITEMS?

## Description

This query-only command returns the number of items on the currently selected level of test hierarchy, which ranges from 1 through 15. This command is equivalent to selecting Instrument Diagnostics from the Utilities menu and then reviewing the Diagnostic Status.

This query-only command can be issued while diagnostics are still in progress.

## Group

Diagnostics

## Syntax

```
DIAG:NUMITEMS?
```

## Example

```
DIAG:NUMITEMS?
```

This query might return

```
:DIAG:NUMITEMS 7
```

This indicates the number of items on the currently selected level of test hierarchy.

---

# DIAG:RESULts?

## Description

This query-only command returns an abbreviated status about the results of the last diagnostic (or test) execution. For a more explanatory status message, use the DIAG:RESULts:VERBose? query. This command is equivalent to selecting Instrument Diagnostics from the Utilities menu and then reviewing the Diagnostic Status.

This query-only command can be issued while diagnostics are still in progress.

## Group

Diagnostics

## Related Commands

DIAG:RESULts:VERBose?

## Syntax

DIAG:RESULts?

## Example 1

DIAG:RESULts?

This query might return

```
:DIAG:RESULTS "135"
```

indicating the specific failure code of the first test failure that occurred.

## Example 2

DIAG:RESULts?

This query might return

```
:DIAG:RESULTS "Pass"
```

indicating that the last diagnostic test passed.

## Example 3

DIAG:RESULts?

This query might return

```
:DIAG:RESULTS "*****"
```

This indicates that the diagnostic test was not run.

---

# DIAG:RESULts:VERBoSe?

## Description

This query-only command returns a more explanatory message about the results of the last diagnostic (or test) execution than the DIAG:RESULts? query. This command is equivalent to selecting Instrument Diagnostics from the Utilities menu and then reviewing the Diagnostic Status.

This query-only command can be issued while diagnostics are still in progress.

## Group

Diagnostics

## Related Commands

DIAG:RESULts?

## Syntax

DIAG:RESULts:VERBoSe?

## Example

DIAG:RESULts:VERBoSe?

This query might return:

```
:DIAG:RESULTS:VERBOSE "DIAG ABORTED, ERROR 15  
in A2D-Dmux 1 interconnects"
```

---

# DIAG:SElect:ALL

## Description

This command (no query form) selects all available diagnostics. This command is equivalent to selecting Instrument Diagnostics from the Utilities menu and then choosing ALL from the Subsystem, Area and Test pull-down lists.

## Group

Diagnostics

## Related Commands

DIAG:NAME:AREA?, DIAG:NAME:SUBSYS?, DIAG:NAME:TEST?,  
DIAG:EXECUTE

## Syntax

```
DIAG:SElect:ALL ALL
```

## Example

```
DIAG:SElect:ALL ALL
```

This command selects all available diagnostics.

---

# DIAG:SElect:AREA

## Description

This command (no query form) selects one of the available areas. This command is equivalent to selecting Instrument Diagnostics from the Utilities menu and then selecting an Area from the pull-down list.

**Note:** This command should be executed only if DIAG:LEVEL is currently set to AREA.

## Group

Diagnostics

## Related Commands

DIAG:SElect:SUBSYS, DIAG:SElect:TEST, DIAG:SElect:LAST

## Syntax

```
DIAG:SElect:AREA <NR1>
```

## Argument

- <NR1>

This selects an area by number, which can range from 0 (zero selects all) through 15 (as specified by DIAG:NUMITEMS).

## Example

```
DIAG:SElect:AREA 1
```

This command selects Area 1 as the area to be diagnosed.



---

# DIAG:SElect:LAST

## Description

This command (no query form) selects one or more diagnostic menu items to be executed via the DIAG:EXECUTE command. If you specify DIAG:LEVEL SUBSYS, then menu items come from this diagnostic level and are limited to the value returned by the DIAG:NUMITEMS? query. For example, specifying DIAG:SElect:SUBSYS 2 and DIAG:NUMITEMS 5 indicates that diagnostics will start from subsystem 2 and that you can specify a range from 2 through 5 for DIAG:SElect:LAST.

If you enter:

DIAG:SElect:LAST 2, only subsystem 2 will be executed.

DIAG:SElect:LAST 4, subsystems 2 *through* 4 will be executed.

## Group

Diagnostics

## Related Commands

DIAG:EXECUTE, DIAG:LEVEL, DIAG:NAME:AREA?,  
DIAG:NAME:SUBSYS?, DIAG:NAME:TEST?, DIAG:NUMITEMS?,  
DIAG:SElect:SUBSYS

## Syntax

```
DIAG:SElect:LAST <NR1>
```

## Arguments

- <NR1>

This selects an integer that identifies the number of the last item that will be executed when the DIAG:EXECUTE command is run.

## Example

```
DIAG:SElect:LAST 2
```

This command specifies that (based on the previous example) only subsystem 2 will be executed.

---

# DIAG:SElect:SUBSYS

## Description

This command (no query form) selects one of the available subsystems. This command is equivalent to selecting Instrument Diagnostics from the Utilities menu and then choosing a Subsystem from the drop-down list.

**Note:** This command should be executed only if DIAG:LEVEL is currently set to SUBSYS.

## Group

Diagnostics

## Related Commands

DIAG:SElect:AREA, DIAG:SElect:LAST, DIAG:SElect:TEST

## Syntax

```
DIAG:SElect:SUBSYS <NR1>
```

## Argument

- <NR1>

This selects a subsystem by number, which can range from 0 (zero selects ALL) through 15 (as specified by DIAG:NUMITEMS).

## Example

```
DIAG:SElect:SUBSYS
```

This command selects Subsystem 1 as the subsystem to be diagnosed.

---

# DIAG:SElect:TEST

## Description

This command (no query form) selects one of the available tests. This command is equivalent to selecting Instrument Diagnostics from the Utilities menu and then choosing a Test from the drop-down list.

**Note:** This command should be executed only if DIAG:LEVEL is currently set to TEST.

## Group

Diagnostics

## Related Commands

DIAG:SElect:AREA, DIAG:SElect:LAST, DIAG:SElect:SUBSYS

## Syntax

```
DIAG:SElect:TEST <NR1>
```

## Arguments

- <NR1>

This selects a test by number, which can range from 0 (zero selects ALL) through 15 (as specified by DIAG:NUMITEMS).

## Example

```
DIAG:SElect:TEST 1
```

This command selects Test 1 as the test to be executed.

---

# DIAG:STATE

## Description

This command (no query form) changes the oscilloscope operating state. Depending on the argument, diagnostics capabilities are either turned on or off. This command is equivalent to opening the DIAG:STATE dialog (ON) or closing it (OFF).

## Group

Diagnostics

## Related Commands

TEST:STOP

## Syntax

```
DIAG:STATE {ON|OFF}
```

## Arguments

### ■ ON

This puts the instrument into the state in which diagnostics can be run. This argument is thrown automatically if either the DIAG:EXECUTE or DIAG:TEST commands are executed.

### ■ OFF

This disables diagnostics capabilities and returns the oscilloscope to a normal operating state.

## Example

```
DIAG:STATE OFF
```

This command turns off diagnostics capabilities and returns the instrument to a normal operating state.

---

# DIAG:STOP

## Description

This command (no query form) causes diagnostics (or test) execution to terminate at the end of the next low-level test. This command is equivalent to selecting Instrument Diagnostics from the Utilities menu and then clicking Abort.

## Group

Diagnostics

## Related Commands

TEST:STOP

## Syntax

DIAG:STOP

## Example

DIAG:STOP

This command terminates diagnostics (or test) execution at the end of the next low-level test.

---

# DISplay:CLOCK

## Description

This command sets or queries the display of the date and time. This is equivalent to selecting Display Date & Time from the Display menu. The query form of this command returns an ON (1) or an OFF (0).

## Group

Display Control

## Related Commands

DATE, TIME

## Syntax 1

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

## Syntax 2

```
DISplay:CLOCK?
```

## Arguments

- ON

This enables display of date and time.

- OFF

This disables display of date and time.

- <NR1>

A 0 disables display of date and time; any other value enables display of date and time.

## Example 1

```
DISplay:CLOCK ON
```

This command enables display of date and time.

## Example 2

```
DISplay:CLOCK?
```

This query might return

```
:DISPLAY:CLOCK 1
```

This indicates that the display of date and time is currently enabled.

---

# DISplay:COLOr:MATHCOLOr

## Description

This command sets or queries the color to be used for math traces, either in the standard palette's nominal Math color, or according to the color of the source waveform. This command is equivalent to selecting Display Setup from the Display menu and then choosing the Colors tab.

## Group

Display Control

## Related Commands

DISplay:COLOr:PALETTE:IMAGEView, DISplay:COLOr:PALETTE:RECORDView

## Syntax 1

```
DISplay:COLOr:MATHCOLOr {DEFAULT|INHERIT}
```

## Syntax 2

```
DISplay:COLOr:MATHCOLOr?
```

## Arguments

### ■ DEFAULT

This sets color math traces in nominal palette math color, which is red.

### ■ INHERIT

This sets color math traces in the source waveform color. Math waveforms are drawn in the same color as their primary source waveform.

## Example 1

```
DISplay:COLOr:MATHCOLOr DEFAULT
```

This command sets the color math traces in the nominal palette math color.

## Example 2

```
DISplay:COLOr:MATHCOLOr?
```

This query might return:

```
:DISPLAY:COLOR:MATHCOLOR INHERIT
```

This indicates that the primary source waveform color is used for each math trace.

---

# DISplay:COLOr:PALEtte:IMAGEView

## Description

This command sets or queries the color palette for all image view (DPO and WfmDB) traces.

## Group

Display Control

## Syntax 1

```
DISplay:COLOr:PALEtte:IMAGEView {NORMal|MONO-  
GREEN|MONOGRAY|TEMPERature|SPECTral|USER}
```

## Syntax 2

```
DISplay:COLOr:PALEtte:IMAGEView?
```

## Arguments

### ■ NORMal

This colors traces according to their channel. This is the default color palette.

### ■ MONOGREEN

This colors traces green, emulating a traditional oscilloscope color palette.

### ■ MONOGRAY

This colors traces gray, emulating a monochrome TDS instrument.

### ■ TEMPERature

All traces share a multicolored palette, where "intensity" is represented by hue; blue for least frequently hit, red for most frequently hit.

### ■ SPECTral

All traces share a multicolored palette, where "intensity" is represented by hue; red for least frequently hit, blue for most frequently hit.

### ■ USER

All traces share a user-defined palette.

## Example 1

```
DISplay:COLOr:PALEtte:IMAGEView MONOGRAY
```

This command sets the palette for all display types.



---

## Example 2

DISplay:COLOr:PALEtte:IMAGEView?

This query might return

:DISPLAY:COLOR:PALETTE:IMAGEVIEW NORMAL

indicating that the Normal color palette is currently selected, which colors traces according to their channel.

---

# DISplay:COLOr:PALEtte:RECORDView

## Description

This command sets or queries the color palette for all record view (non image view) waveforms.

## Group

Display Control

## Syntax 1

```
DISplay:COLOr:PALEtte:RECORDView {NORMAL|MONO-  
GREEN|MONOGRAY|TEMPERature|SPECTral|USER}
```

## Syntax 2

```
DISplay:COLOr:PALEtte:RECORDView?
```

## Arguments

### ■ NORMAL

This colors traces according to their channel. This is the default color palette.

### ■ MONOGREEN

This colors traces green, emulating a traditional oscilloscope color palette.

### ■ MONOGRAY

This colors traces gray, emulating a monochrome TDS instrument.

### ■ TEMPERATURE

All traces share a multi-colored palette, where "intensity" is represented by hue; blue for least frequently hit, red for most frequently hit.

### ■ SPECTRAL

All traces share a multi-colored palette, where "intensity" is represented by hue; red for least frequently hit, blue for most frequently hit.

### ■ USER

All traces share a user-defined palette.

## Example 1

```
DISplay:COLOr:PALEtte:RECORDView MONOGRAY
```

This command sets the palette for all display types.

---

## Example 2

DISplay:COLOr:PALEtte:RECORDView?

This query might return

:DISPLAY:COLOR:PALETTE:RECORDVIEW NORMAL

indicating that the Normal color palette is currently selected, which colors traces according to their channel.

---

# DISplay:COLOR:PALEtte:USER:CARet

## Description

This command sets or queries the caret color for the user palette. It assigns the HUE, light, Saturation (HLS) triplet used for the caret color. The caret is the solid, inverted delta positioned on the top graticule line, which indicates the trigger position within the waveform record.

## Group

Display Control

## Syntax 1

```
DISplay:COLOr:PALEtte:USER:CARet  
<NR1>,<NR1>,<NR1>
```

## Syntax 2

```
DISplay:COLOr:PALEtte:USER:CARet?
```

## Arguments

- <NR1>

Hue. Range of 0 to 360.

- <NR1>

Lightness. Range of 0 to 100.

- <NR1>

Saturation. Range of 0 to 100.

## Example 1

```
DISplay:COLOr:PALEtte:USER:CARet 330,75,100
```

This command sets the caret color hue to 330, the lightness to 75, and the saturation to 100.

## Example 2

```
DISplay:COLOr:PALEtte:USER:CARet?
```

This query might return

```
:DISPLAY:COLOR:PALETTE:USER:CARET 250, 85, 90
```

indicating that the color hue is set at 250, the lightness is set to 85, and the saturation is set to 90.

---

# DISplay:COLOr:PALEtte:USER:CH<x>

## Description

This command sets or queries the hue-lightness-saturation (HLS) triplet to be used for the specified channel color for the user palette. The channel is specified by x. The value of x can range from 1 through 4 for four-channel instruments or 1 through 2 for two-channel instruments.

## Group

Display Control

## Syntax 1

```
DISplay:COLOr:PALEtte:USER:CH{1|2|3|4}  
<NR1>,<NR1>,<NR1>
```

## Syntax 2

```
DISplay:COLOr:PALEtte:USER:CH{1|2|3|4}?
```

## Arguments

- <NR1>

Hue. Range of 0 to 360.

- <NR1>

Lightness. Range of 0 to 100.

- <NR1>

Saturation. Range of 0 to 100.

## Example 1

```
DISplay:COLOr:PALEtte:USER:CH1 330,75,100
```

This command sets the channel 1 color for the user palette to 330, 75, or 100.

## Example 2

```
DISplay:COLOr:PALEtte:USER:CH2?
```

This query might return

```
:DISPLAY:COLOR:PALETTE:USER:CH2 325,65,100
```

indicating that channel 2 color for the user palette is set at 325 for hue, the set to 65 for lightness, and 100 for saturation.

---

# DISplay:COLOr:PALEtte:USER:GRAticule

## Description

This command sets or queries the user palette graticule color. It assigns the hue-lightness-saturation triplet to be used for the graticule color for the user palette.

## Group

Display Control

## Syntax 1

```
DISplay:COLOr:PALEtte:USER:GRAticule  
<NR1>,<NR1>,<NR1>
```

## Syntax 2

```
DISplay:COLOr:PALEtte:USER:GRAticule?
```

## Arguments

- <NR1>

Hue. Range of 0 to 360.

- <NR1>

Lightness. Range of 0 to 100.

- <NR1>

Saturation. Range of 0 to 100.

## Example 1

```
DISplay:COLOr:PALEtte:USER:GRAticule  
330,75,100
```

This command sets the graticule color for the user palette.

## Example 2

```
DISplay:COLOr:PALEtte:USER:GRAticule?
```

This query might return

```
:DISPLAY:COLOR:PALETTE:USER:GRATICULE  
300,60,85
```

indicating that the user palette graticule color is assigned a hue of 300, a lightness of 60, and a saturation of 85.

---

# DISplay:COLOr:PALEtte:USEr:HIStogram

## Description

This command sets or queries the user palette histogram color. It assigns the hue-saturation-lightness (HLS) triplet to be used for the histogram color for the user palette.

## Group

Display Control

## Syntax 1

```
DISplay:COLOr:PALEtte:USEr:HIStogram  
<NR1>,<NR1>,<NR1>
```

## Syntax 2

```
DISplay:COLOr:PALEtte:USEr:HIStogram?
```

## Arguments

- <NR1>

Hue. Range of 0 to 360.

- <NR1>

Lightness. Range of 0 to 100.

- <NR1>

Saturation. Range of 0 to 100.

## Example 1

```
DISplay:COLOr:PALEtte:USEr:HIStogram  
330,75,100
```

This command sets the histogram color for the user palette.

## Example 2

```
DISplay:COLOr:PALEtte:USEr:HIStogram?
```

This query might return

```
:DISPLAY:COLOR:PALETTE:USER:HISTOGRAM  
330,50,90
```

indicating that the user palette histogram color is assigned a hue of 330, a lightness of 50, and a saturation of 90.

---

# DISplay:COLOr:PALEtte:USEr:MASK

## Description

This command sets or queries the user palette mask color. It assigns the hue-lightness-saturation (HLS) color for the mask color for the user palette.

## Group

Display Control

## Syntax 1

```
DISplay:COLOr:PALEtte:USEr:MASK  
<NR1>,<NR1>,<NR1>
```

## Syntax 2

```
DISplay:COLOr:PALEtte:USEr:MASK?
```

## Arguments

- <NR1>

Hue. Range of 0 to 360.

- <NR1>

Lightness. Range of 0 to 100.

- <NR1>

Saturation. Range of 0 to 100.

## Example 1

```
DISplay:COLOr:PALEtte:USEr:MASK 330,75,100
```

This command sets the mask color for the user palette.

## Example 2

```
DISplay:COLOr:PALEtte:USEr:MASK?
```

This query might return

```
:DISPLAY:COLOR:PALETTE:USEr:MASK 330,50,90
```

indicating that the user palette mask color is assigned a hue of 330, a lightness of 50, and a saturation of 90.



---

# DISplay:COLOr:PALEtte:USEr:MASKHighlight

## Description

This command sets or queries the user palette mask hits color. It assigns the hue-lightness-saturation (HLS) triplet to be used for the mask highlight (mask hits) color for the user palette.

## Group

Display Control

## Syntax 1

```
DISplay:COLOr:PALEtte:USEr:MASKHighlight  
<NR1>,<NR1>,<NR1>
```

## Syntax 2

```
DISplay:COLOr:PALEtte:USEr:MASK:Highlight?
```

## Arguments

- <NR1>

Hue. Range of 0 to 360.

- <NR1>

Lightness. Range of 0 to 100.

- <NR1>

Saturation. Range of 0 to 100.

## Example 1

```
DISplay:COLOr:PALEtte:USEr:MASK:Highlight  
330,75,100
```

This command sets the mask highlight (mask hits) color for the user palette.

## Example 2

```
DISplay:COLOr:PALEtte:USEr:MASK:Highlight?
```

This query might return

```
:DISPLAY:COLOR:PALETTE:USER:MASK:HIGHLIGHT  
325,65,100
```

indicating that the user palette mask highlight color is assigned a hue of 325, a lightness of 65, and a saturation of 100.

---

# DISplay:COLOr:PALEtte:USER:MATH<x>

## Description

This command sets or queries the user palette math colors. It assigns the hue-lightness-saturation (HLS) triplet to be used for the specified math waveform for the user palette. The math waveform is specified by x. The value of x can range from 1 through 4 for four-channel instruments or 1 through 2 for two-channel instruments.

## Group

Display Control

## Syntax 1

```
DISplay:COLOr:PALEtte:USER:MATH<x>  
<NR1>,<NR1>,<NR1>
```

## Syntax 2

```
DISplay:COLOr:PALEtte:USER:MATH<x>?
```

## Arguments

- <NR1>

Hue. Range of 0 to 360.

- <NR1>

Lightness. Range of 0 to 100.

- <NR1>

Saturation. Range of 0 to 100.

## Example 1

```
DISplay:COLOr:PALEtte:USER:MATH2 330,75,100
```

This command sets the color of the specified math 2 waveform for the user palette.

## Example 2

```
DISplay:COLOr:PALEtte:USER:MATH2?
```

This query might return

```
:DISPLAY:COLOR:PALETTE:USER:MATH2 345,75,98
```

indicating that the user palette color of the math 2 waveform is set to a hue of 345, a lightness of 75, and a saturation of 98.

---

# DISplay:COLOr:PALEtte:USER:REF<x>

## Description

This command sets or queries the user palette reference colors assigned to reference waveforms. It assigns the hue-lightness-saturation (HLS) triplet to be used for the specified reference waveform color for the user palette. The reference waveform is specified by x. The value of x can range from 1 through 4 for four-channel instruments or 1 through 2 for two-channel instruments.

## Group

Display Control

## Syntax 1

```
DISplay:COLOr:PALEtte:USER:REF<x>  
<NR1>,<NR1>,<NR1>
```

## Syntax 2

```
DISplay:COLOr:PALEtte:USER:REF<x>?
```

## Arguments

- <NR1>

Hue. Range of 0 to 360.

- <NR1>

Lightness. Range of 0 to 100.

- <NR1>

Saturation. Range of 0 to 100.

## Example 1

```
DISplay:COLOr:PALEtte:USER:REF1 330,75,100
```

This command sets the reference waveform 1 color for the user palette.

## Example 2

```
DISplay:COLOr:PALEtte:USER:REF2?
```

This query might return

```
:DISPLAY:COLOR:PALETTE:USER:REF2 300,90,100
```

indicating that the user palette color for reference waveform 2 is assigned a hue of 300, and lightness of 90, and a saturation of 100.

---

# DISplay:COLOR:PALEtte:USER?

## Description

This command queries the color palette for group settings. It outputs settings from the DISPlay CARET, CH<x>, GRATICULE, HISTOGRAM, MASK, MASKHIGHLIGHT, MATH<x>, and REF<x> commands.

## Group

Display Control

## Syntax

```
DISplay:COLOr:PALEtte:USER?
```

## Example 1

```
DISplay:COLOR:PALETTE:USER?
```

This query might return

```
:DISPLAY:COLOR:PALETTE:USER:GRATICULE  
165,50,15;CH1 180,50,100;CH2 300 50,100;CH3  
60,50,100;CH4 240,50,100;REF1 0,90,0;REF2  
0,90,100;REF3 60,90,100;REF4 240,90,100;MATH1  
160,50,100;MATH2 40,60,100;MATH3  
120,60,100;MATH4 195,50,100;HISTOGRAM  
320,50,100;CARET 150,50,100;MASK 0,25,75;MASK-  
HIGHLIGHT 140,50,100
```

---

# DISplay:COLOr:REFCOLOr

## Description

This command sets or queries the color to be used for reference traces, either in the standard palette's nominal REF color or according to the color of the source waveform. This command is equivalent to selecting Display Setup from the Display menu and then choosing the Colors tab.

## Group

Display Control

## Related Commands

DISplay:INTENSITY:WAVEform:IMAGEView, DISplay:INTENSITY:WAVEform:RECORDView

## Syntax 1

```
DISplay:COLOr:REFCOLOr {DEFAULT|INHERIT}
```

## Syntax 2

```
DISplay:COLOr:REFCOLOr?
```

## Arguments

### ■ DEFAULT

This assigns color reference traces in nominal palette reference color, which is off-white.

### ■ INHERIT

This assigns color reference traces in the source waveform color.

## Example 1

```
DISplay:COLOr:REFCOLOr DEFAULT
```

This command assigns color reference traces in nominal palette reference color, which is off-white.

## Example 2

```
DISplay:COLOr:REFCOLOr?
```

This query might return:

```
:DISPLAY:COLOR:REFCOLOR INHERIT
```

This indicates that source waveform color is used for reference traces.

---

# DISplay:COLOr?

## Description

This query-only command returns the settings from the PALETTE, MATHCOLOR and REFCOLOR commands. This is equivalent to selecting Colors from the Display menu.

## Group

Display Control

## Syntax

DISplay:COLOr?

## Arguments

None

## Example

DISplay:COLOr?

This query might return:

```
:DISPLAY:COLOR:PALETTE:IMAGEVIEW TEMPERA-  
TURE;RECORDVIEW NORMAL;USER:GRATICULE  
165,50,15;CH1 180,50,100;CH2 300,50,100;CH3  
60,50,100;CH4 240,50,100;REF1 0,90,0;REF2  
0,90,100;REF3 60,90,100;REF4 240,90,100;MATH1  
160,50,100;MATH2 40,60,100;MATH3  
120,60,100;MATH4 195,50,100;HISTOGRAM  
320,50,100;CARET 150,50,100;MASK 0,25,75;MASK-  
HIGHLIGHT 140,50,100;:DISPLAY:COLOR:MATHCOLOR  
DEFAULT;REFCOLOR DEFAULT
```

indicating that the primary source waveform color is used for each math trace.

---

# DISplay:FILTer

## Description

This command sets or queries the type of interpolation to use for the display. Filtering only applies to normal-mode acquisition. The DISplay:FILTer command also provides selection for acquisition interpolation type. This command is equivalent to selecting Waveform Interpolation from the Display menu.

## Group

Display Control

## Syntax 1

```
DISplay:FILTer {LINEAr|SINX}
```

## Syntax 2

```
DISplay:FILTer?
```

## Arguments

- LINEAr

This specifies linear interpolation, where acquired points are connected with straight lines.

- SINX

This specifies  $\sin(x)/x$  interpolation, where acquired points are fit to a curve.

## Example 1

```
DISplay:FILTer SINX
```

This command specifies sine-curve interpolation, when magnifying waveforms.

## Example 2

```
DISplay:FILTer?
```

This query might return:

```
:DISPLAY:FILTER LINEAR
```

This indicates that straight-line interpolation is specified for magnifying waveforms.

---

# DISplay:FORMat

## Description

This command sets or queries the display format. This command is equivalent to selecting Format from the Display menu.

## Group

Display Control

## Syntax 1 (Two-channel instruments)

```
DISplay:FORMat {YT|XY}
```

## Syntax 2 (Four-channel instruments)

```
DISplay:FORMat {YT|XY|XYZ}
```

## Syntax 3

```
DISplay:FORMat?
```

## Arguments

- YT

This sets the display to a voltage versus time format and is the default mode.

- XY

This argument displays one waveform against another. The source pairs that make up an XY trace are predefined and are listed in the following table. Selecting one source causes its corresponding source to be implicitly selected, producing a single trace from the two input waveforms.

XY Format Pairs	
X-Axis Source	Y-Axis Source
Ch 1	Ch 2
Ch 3 (Four-channel models only)	Ch 4 (Four-channel models only)
Ref 1	Ref 2
Ref 3 (Four-channel models only)	Ref 4 (Four-channel models only)

- XYZ

This argument is available only for four-channel instruments. The argument combines channel 1 and channel 2 for X and Y coordinates and uses channel 3 to provide the intensity value for the sample. XYZ groups channels 1, 2 and 3 to form a single trace. Other channel, math, and reference waveforms are turned off.

**Note:** Selecting XY or XYZ forces a switch to FASTAcq mode.



---

### **Example 1**

`DISplay:FORMat YT`

This command selects a voltage versus time format for the display.

### **Example 2**

`DISplay:FORMat?`

This query might return:

`:DISPLAY:FORMAT XYZ`

This is for the display format, indicating that the display is in the XYZ mode.

---

# DISplay:GRATicule

## Description

This command selects or queries the type of graticule that is displayed. This command is equivalent to selecting Graticule Style from the Display menu.

## Group

Display Control

## Syntax 1

```
DISplay:GRATicule {CROSS-  
Hair | FRame | FULL | GRID | IRE | NTSC | MV | PAL}
```

## Syntax 2

```
DISplay:GRATicule?
```

## Arguments

### ■ CROSSHair

This specifies a frame and cross hairs.

### ■ FRame

This specifies a frame only.

### ■ FULL

This specifies a frame, a grid and cross hairs.

### ■ GRID

This specifies a frame and grid only.

### ■ IRE

This specifies an IRE video graticule, and sets the vertical scale to 143 mV per division.

### ■ NTSC

This specifies an INTSC video graticule (same as the IRE graticule), and sets the vertical scale to 143 mV per division.

### ■ MV

This specifies an mV video graticule and sets the vertical scale to 143 mV per division. This graticule is used to measure PAL standard video signals.

### ■ PAL

This specifies a PAL video graticule (same as the mV graticule) and sets the vertical scale to 143 mV per division. This graticule is used to measure PAL standard video signals.

---

### **Example 1**

`DISplay:GRAticule FRaMe`

This command sets the graticule type to display the frame only.

### **Example 2**

`DISplay:GRAticule?`

This query might return:

`:DISPLAY:GRATICULE FULL`

This indicates that all graticule elements are selected.

---

# DISplay:INTENSITY:AUTOBright

## Description

This command enables automatic, ongoing adjustment of the intensity to display images. The query form returns a 1 (ON) or a 0 (OFF). This command is equivalent to selecting Display Setup from the Display menu and choosing the Appearance tab.

## Group

Display Control

## Related Commands

DISplay:INTENSITY:WAVEform:IMAGEView, DISplay:INTENSITY:WAVEform:RECORDView

## Syntax 1

```
DISplay:INTENSITY:AUTOBright {OFF|ON|<NR1>}
```

## Syntax 2

```
DISplay:INTENSITY:AUTOBright?
```

## Arguments

### ■ OFF

This argument allows the system to use the manually set waveform intensity value against an absolute scale. The display simulates the appearance of signals on an analog oscilloscope. Waveforms that trigger more frequently appear brighter than waveforms that trigger less frequently.

### ■ ON

This argument allows the system to adjust the settings automatically to provide a visible waveform.

### ■ <NR1>

A 0 allows the system to use the manually-set waveform intensity value against an absolute scale; any other value allows the system to adjust settings.

## Example 1

```
DISplay:INTENSITY:AUTOBright ON
```

This command allows the system to adjust settings using the waveform intensity value and the pixel intensity found in the acquired waveform.

---

## Example 2

DISplay:INTENSITY:AUTOBright?

This query might return:

:DISPLAY:INTENSITY:AUTOBRIGHT 1

This indicates that system adjustment of waveform intensity is enabled.

---

# DISplay:INTENSITY:SCRENSAVER

## Description

This command sets and queries the screen saver features of the MS Windows operating system. When enabled, a delay timer (set in seconds by the DISplay:INTENSITY:SCRENSAVERDELAY command) begins counting down. When this screen saver delay times out, the screen low-power mode engages. This causes the LCD backlight to switch off and clears both waveform and text displays. Any control (front panel, mouse or keyboard) or touch screen activity resets the delay timer and restores normal instrument display. This command is equivalent to selecting LCD Save Enabled from the Display menu.

## Group

Display Control

## Related Commands

DISplay:INTENSITY:SCRENSAVERDELAY

## Syntax 1

```
DISplay:INTENSITY:SCRENSAVER {OFF|ON|<NR1>}
```

## Syntax 2

```
DISplay:INTENSITY:SCRENSAVER?
```

## Arguments

- OFF

This disables the screen saver feature.

- ON

This enables the screen saver feature after the specified screen saver delay seconds of control activity have passed.

- <NR1>

A 0 disables the screen saver feature; a 1 enables the screen saver protection features.

## Example 1

```
DISplay:INTENSITY:SCRENSAVER OFF
```

This command disables the screen saver feature.

## Example 2

```
DISplay:INTENSITY:SCRENSAVER?
```

This query might return

```
:DISPLAY:INTENSITY:SCRENSAVER 0
```

This indicates that the screen saver feature is disabled.

---

# DISplay:INTENSITY:SCRENSAVERDELAY

## Description

This command sets or queries the timeout of the screen saver features of the display system. When enabled (after the specified screen saver delay seconds of control activity and when the screen saver feature is enabled) the instrument activates the screen saver feature. Normal instrument displays are restored and the delay timer is reset upon any control activity. The instrument continues to acquire and process data normally while in screen saver mode; only the display is disabled. This command is equivalent to selecting Objects from the Display menu and entering a time in the Delay field of the LCD Backlight Timeout control.

## Group

Display Control

## Related Commands

DISplay:INTENSITY:SCRENSAVER

## Syntax 1

```
DISplay:INTENSITY:SCRENSAVERDELAY {<NR1>}
```

## Syntax 2

```
DISplay:INTENSITY:SCRENSAVERDELAY?
```

## Arguments

- <NR1>

This sets the screen saver timeout, which ranges from 30 through 28800 seconds.

## Example 1

```
DISplay:INTENSITY:SCRENSAVERDELAY 120
```

This command sets the screen saver delay feature to activate after 120 seconds of control inactivity.

## Example 2

```
DISplay:INTENSITY:SCRENSAVERDELAY?
```

This query might return:

```
:DISPLAY:INTENSITY:SCRENSAVERDELAY 300
```

This indicates that the screen saver delay feature is set to 300 seconds of control inactivity.

---

# DISplay:INTENSITY:WAVEform:IMAGEView

## Description

This command sets or queries the saturation level for image view waveforms.

## Group

Display Control

## Related Commands

DISplay:INTENSITY:WAVEform:RECORDView, DISplay:INTENSITY:AUTOBright

## Syntax 1

```
DISplay:INTENSITY:WAVEform:IMAGEView <NR1>
```

## Syntax 2

```
DISplay:INTENSITY:WAVEform:IMAGEView?
```

## Arguments

- <NR1>

This is the waveform saturation and ranges from 10 to 100 percent.

## Example 1

```
DISplay:INTENSITY:WAVEform:IMAGEView 100
```

This command sets the image view waveform saturation to 100 percent.

## Example 2

```
DISplay:INTENSITY:WAVEform:IMAGEView?
```

This query might return

```
:DISPLAY:INTENSITY:WAVEFORM:IMAGEVIEW 60.0000
```

indicating that the saturation level of the image view waveforms is currently set to 60 percent.



---

# DISplay:INTENSITY:WAVEform:RECORDView

## Description

This command sets or queries the saturation level for record view waveforms.

## Group

Display Control

## Related Commands

DISplay:INTENSITY:WAVEform:IMAGEView, DISplay:INTENSITY:AUTOBright

## Syntax 1

```
DISplay:INTENSITY:WAVEform:RECORDView <NR1>
```

## Syntax 2

```
DISplay:INTENSITY:WAVEform:RECORDView?
```

## Arguments

- <NR1>

This is the waveform saturation and ranges from 10 to 100 percent.

## Example 1

```
DISplay:INTENSITY:WAVEform:RECORDView 100
```

This command sets the record view waveform saturation to 100 percent.

## Example 2

```
DISplay:INTENSITY:WAVEform:RECORDView?
```

This query might return

```
:DISPLAY:INTENSITY:WAVEFORM:RECORDVIEW 60.0000
```

indicating that the saturation of the record view waveforms is currently set to 60 percent.

---

# DISplay:INTENSITY?

## Description

This query-only command returns the waveform saturation level, auto-bright state, and screen saver settings. This command is equivalent to selecting Display Setup from the Display menu and choosing the Appearance tab.

## Group

Display Control

## Related Commands

DISplay:INTENSITY:AUTPBright, DISplay:INTENSITY:WAVEform:IMAGEView, DISplay:INTENSITY:WAVEform:RECORDView, DISplay:INTENSITY:SCRENSAVER, DISplay:INTENSITY:SCRENSAVERDELAY

## Syntax

```
DISplay:INTENSITY?
```

## Arguments

None

## Example

```
DISplay:INTENSITY?
```

This query might return:

```
:DISPLAY:INTENSITY:WAVEFORM:IMAGEVIEW  
75.0000;RECORDVIEW 10.0000;:DISPLAY:INTENSITY:  
AUTOBRIGHT 1;SCRENSAVER 1;SCRENSAVERDELAY  
28800
```

---

# DISplay:PERsistence

## Description

This command sets or queries the persistence aspect of the display. This affects the display only and is equivalent to selecting Display Persistence from the Display menu.

## Group

Display Control

## Related Commands

DISplay:VARPersist

## Syntax 1

```
DISplay:PERsistence {OFF|INFPersist|VARPersist}
```

## Syntax 2

```
DISplay:PERsistence?
```

## Arguments

### ■ OFF

This disables the persistence aspect of the display.

### ■ INFPersist

This sets a display mode where any pixels, once touched by samples, remain set until cleared by a mode change.

VARPersist

This sets a display mode where set pixels are gradually dimmed.

## Example 1

```
DISplay:PERsistence VARPersist
```

This command sets the persistence aspect of the display to fade set pixels according to the time set in the DISplay:VARPersist command.

## Example 2

```
DISplay:PERsistence?
```

This query might return:

```
:DISPLAY:PERSISTENCE OFF
```

This indicates that the persistence aspect of the display is disabled.

---

# DISplay:PERSiStence:RESEt

## Description

This command controls the clearing of persistence data.

## Group

Display Control

## Related Commands

DISplay:PERSiStence, DISplay:VARPersist

## Syntax

```
DISplay:PERSiStence:RESEt
```

## Example

```
DISplay:PERSiStence:RESEt
```

This command clears the display of persistence data.

---

# DISplay:SCREENTExt:LABel<x>:NAME

## Description

This command sets the text to be displayed for a given label. The label is specified by <x>. The value of <x> can range from 1 through 8.

## Group

Display Control

## Related Commands

DISplay:SCREENTExt?, DISplay:SCREENTExt:STATE, DISplay:SCREENTExt:LABel<x>?, DISplay:SCREENTExt:LABel<x>:XPOS, DISplay:SCREENTExt:LABel<x>:YPOS

## Syntax

```
DISplay:SCREENTExt:LABel<x>:NAME <String>
```

## Arguments

- <String>

This argument is the text to be displayed for a given label.

## Example 1

```
:DISplay:SCREENTExt:LABel2:NAME "THIS IS  
SCREEN TEXT"
```

This command sets the screen text for label 2 to THIS IS SCREEN TEXT.

## Example 2

```
:DISplay:SCREENTExt:LABel2:NAME?
```

This query might return

```
:DISPLAY:SCREENTEXT:LABEL2:NAME "MyWaveform"
```

indicating that the label 2 screen text is set to MyWaveform.

---

# DISplay:SCREENTExt:LABel<x>:XPOS

## Description

This command sets the horizontal position of a given screen text label. The label is specified by x. The value of x can range from 1 through 8.

## Group

Display Control

## Related Commands

DISplay:SCREENTExt?, DISplay:SCREENTExt:STATE, DISplay:SCREENTExt:LABel<x>?, DISplay:SCREENTExt:LABel<x>:NAME, DISplay:SCREENTExt:LABel<x>:YPOS

## Syntax

<NR1>

## Arguments

- <NR1>

This argument sets the horizontal position of the screen text label. The value can range from 0 to 500.

## Example 1

```
DISplay:SCREENTExt:LABel3:XPOS 100
```

This command sets the horizontal position of label 3 to 100.

## Example 2

```
DISplay:SCREENTExt:LABel3:XPOS?
```

This query might return.

```
:DISPLAY:SCREENTEXT:LABEL3:XPOS 250
```

indicating that screen label 3 is horizontally positioned to 250 (mid-screen horizontally).

---

# DISplay:SCREENTExt:LABel<x>:YPOS

## Description

This command sets the vertical position of a given label. The label is specified by x. The value of x can range from 1 through 8.

## Group

Display Control

## Related Commands

DISplay:SCREENTExt?, DISplay:SCREENTExt:STATE, DISplay:SCREENTExt:LABel<x>?, DISplay:SCREENTExt:LABel<x>:NAME, DISplay:SCREENTExt:LABel<x>:XPOS

## Syntax

<NR1>

## Argument

- <NR1>

This argument sets the vertical position of the screen text label. The value can range from 0 to 385.

## Example 1

```
DISplay:SCREENTExt:LABel3:YPOS 100
```

This command sets the vertical position of label 3 to 100.

## Example 2

```
DISPLAY:SCREENTEXT:LABEL3:YPOS?
```

This query might return

```
:DISPLAY:SCREENTEXT:LABEL3:YPOS 192
```

indicating that screen label 3 is vertically positioned to 192 (midscreen vertically).

---

# DISplay:SCREENTExt:LABel<x>?

## Description

This command returns the screen text setting for a given label. Specifically, it provides the name, horizontal position (XPOS), and vertical position (YPOS).

## Group

Display Control

## Related Commands

DISplay:SCREENTExt?, DISplay:SCREENTExt:STATE, DISplay:SCREENTExt:LABel<x>:NAME, DISplay:SCREENTExt:LABel<x>:XPOS, DISplay:SCREENTExt:LABel<x>:YPOS

## Syntax

```
DISplay:SCREENTExt:LABel<x>?
```

## Example

```
DISplay:SCREENTExt:LABel3??
```

This query might return

```
:DISPLAY:SCREENTEXT:LABEL3:NAME "THIS IS  
SCREEN TEXT"; :XPOS 100; :YPOS 35
```

indicating that the name of label 3 is THIS IS SCREEN TEXT, and the label is position at 100 horizontally and at 35 vertically.



---

# DISplay:SCREExt:STATE

## Description

This command controls the display of screen text.

## Group

Display Control

## Related Commands

DISplay:SCREExt:?, DISplay:SCREExt:LABel<x>?, DISplay:SCREExt:LABel<x>:NAME, DISplay:SCREExt:LABel<x>:XPOS, DISplay:SCREExt:LABel<x>:YPOS

## Syntax

```
DISplay:SCREExt:STATE {ON|OFF|<NR1>}
```

## Arguments

- ON

This turns on the display of screen text.

- OFF

This turns off the display of screen text.

- <NR1>

A 0 disables screen text; any other value enables screen text.

## Example 1

```
DISplay:SCREExt:STATE ON
```

This command turns the screen text display on.

## Example 2

```
DISplay:SCREExt:STATE?
```

This query might return

```
:DISPLAY:SCREExt:STATE 1
```

indicating that the screen text is displayed.

---

# DISplay:SCREENTExt?

## Description

This command returns all screen text settings.

## Group

Display Control

## Related Commands

DISplay:SCREENTExt:STATE, DISplay:SCREENTExt:LABel<x>?,  
DISplay:SCREENTExt:LABel<x>:NAME, DISplay:SCREEN-  
Text:LABel<x>:XPOS, DISplay:SCREENTExt:LABel<x>:YPOS

## Syntax

```
DISplay:SCREENTExt?
```

## Example 1

```
DISPLAY:SCREENTEXT?
```

This command might reply:

```
:DISPLAY:SCREENTEXT:STATE 0;LABEL1:NAME  
"";XPOS 100;YPOS 5;:DISPLAY:SCREENTEXT:LA-  
BEL2:NAME "";XPOS 100;YPOS 20;:DISPLAY:SCREEN-  
TEXT:LABEL3:NAME "";XPOS 100;YPOS 35;:DIS-  
PLAY:SCREENTEXT:LABEL4:NAME "";XPOS 100;YPOS  
50;:DISPLAY:SCREENTEXT:LABEL5:NAME "";XPOS  
100;YPOS 65;:DISPLAY:SCREENTEXT:LABEL6:NAME  
"";XPOS 100;YPOS 80;:DISPLAY:SCREENTEXT:LA-  
BEL7:NAME "";XPOS 100;YPOS 95;:DISPLAY:SCREEN-  
TEXT:LABEL8:NAME "";XPOS 100;YPOS 110
```

---

# DISplay:SHOWREmote

## Description

This command sets or queries the state of the remote display feature and is equivalent to selecting Display Remote from the Display menu. The query form of this command returns ON (1) or OFF (0).

This feature allows you to view waveforms and other graticule data on a remote display using remote control software like VNC (Virtual Network Computing) or Symantec pcAnywhere.

## Group

Display Control

## Syntax 1

```
DISplay:SHOWREmote {ON|OFF|NR1}
```

## Syntax 2

```
DISplay:SHOWREmote?
```

## Arguments

- ON

This enables the remote display of waveform and other graticule data.

- OFF

This disables the remote display of waveform and other graticule data.

- NR1

A 0 disables remote display of waveform and other graticule data; any other value enables remote display of waveform and other graticule data.

## Example 1

```
DISplay:SHOWREmote ON
```

This command enables remote display of waveform and other graticule data.

## Example 2

```
DISplay:SHOWREmote?
```

This query might return:

```
:DISPLAY:SHOWREMOTE 1
```

This indicates that the remote display of waveform and other graticule data is currently enabled.

---

# DISplay:STYLE

## Description

This command sets or queries how the data is displayed for normal and FastAcq modes. This command is equivalent to selecting Display Style from the Display menu and choosing a style.

## Group

Display Control

## Syntax 1

```
DISplay:STYLE {VECTors|DOTs|INTENSIFied}
```

## Syntax 2

```
DISplay:STYLE?
```

## Arguments

### ■ VECTors

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

### ■ DOTs

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

### ■ INTENSIFied

This causes the display to show interpolated samples with dark spaces (Only the "real" samples are displayed).

When FastAcq mode is enabled, intensified samples display as dots. However, turning off FastAcq mode causes the display style to snap back to its previous setting.

## Example 1

```
DISplay:STYLE VECTors
```

This command sets the display to connect adjacent data points.

## Example 2

```
DISplay:STYLE?
```

This query might return:

```
:DISPLAY:STYLE INTENSIFIED
```

This indicates that interpolated samples are not displayed.

---

# DISplay:TRIGBar

## Description

This command controls or queries the display of the trigger-level indicator bar/s. Indicator bars show where the trigger voltage level is set.

The instrument will only display the bar if the associated trigger source is also displayed. If both a main and a delayed trigger are displayed, then two bars will appear. One will accompany each source. If a logic trigger is selected, then multiple bars may appear. One will show the upper threshold and one will show the lower threshold. This command is equivalent to selecting Display Setup from the Display menu and then choosing the Objects tab.

## Group

Display Control

## Syntax 1

```
DISplay:TRIGBar {OFF|SHORT|LONG}
```

## Syntax 2

```
DISplay:TRIGBar?
```

## Arguments

- OFF

This removes the trigger indicator bar from the display.

- SHORT

This displays, as the indicator, a short arrow at the right side of the graticule for each displayed trigger signal.

- LONG

This displays, as the indicator, a horizontal line across the width of the graticule for each displayed trigger signal.

## Example 1

```
DISplay:TRIGBar LONG
```

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

## Example 2

```
DISplay:TRIGBar?
```

This query might return:

```
:DISPLAY:TRIGBAR OFF,
```

This indicates that the indicator bar is removed from the display.

---

# DISplay:VARpersist

## Description

This command sets or queries persistence decay time, which is the approximate decay time for a freshly struck persistence sample. This command is equivalent to selecting Display Setup from the Display menu, selecting the Appearance tab, and then entering the desired Persist Time.

## Group

Display Control

## Related Commands

DISplay:PERSistence

## Syntax 1

```
DISplay:VARpersist <NR3>
```

## Syntax 2

```
DISplay:VARpersist?
```

## Arguments

- <NR3>

This indicates the persistence decay time setting in numeric seconds. The persistence time ranges from 32 ms to 10 s.

## Example 1

```
DISplay:VARpersist 5E-1
```

This command sets the persistence decay time to 500 ms.

## Example 2

```
DISplay:VARpersist?
```

This query might return:

```
:DISPLAY:VARPERSIST 5.0000E-01
```

This indicates that persistence decay time is currently set to 500 ms.

---

# DISplay?

## Description

This query-only command returns the current Display settings. This command is equivalent to selecting Display Setup from the Display menu and then viewing the contents of each tab.

## Group

Display Control

## Syntax

DISplay?

## Example

DISplay?

This query might return

```
:DISPLAY:CLOCK 1;COLOR:PALETTE:IMAGEVIEW MONO-  
GREEN;  
RECORDVIEW TEMPERATURE;USER:GRATICULE  
165,50,15;CH1 180,50,100;  
CH2 300,50,100;CH3 60,50,100;CH4  
240,50,100;REF1 0,90,0;  
REF2 0,90,100;REF3 60,90,100;REF4  
240,90,100;MATH1 160,50,100;  
MATH2 40,60,100;MATH3 120,60,100;MATH4  
195,50,100;  
HISTOGRAM 320,50,100;CARET 150,50,100;MASK  
0,25,75;MASKHIGHLIGHT 140,50,100;:DISPLAY:COL-  
OR:MATHCOLOR DEFAULT;REFCOLOR DEFAULT;  
:DISPLAY:FILTER SINX;FORMAT YT;GRATICULE  
IRE;INTENSITY  
:WAVEFORM:IMAGEVIEW 81.0000;RECORDVIEW  
81.0000;:DISPLAY  
:INTENSITY:AUTOBRIGHT 0;SCRENSAVER 1;SCREEN-  
SAVERDELAY 28800;  
:DISPLAY:PERSISTENCE OFF;STYLE DOTS;TRIGBAR  
OFF;TRIGT 1;  
CURSORTICK LONG;VARPERSIST 2.6000;SCREEN-  
TEXT:STATE 1;  
LABEL1:NAME "";XPOS 100;YPOS 5;:DIS-  
PLAY:SCREENTEXT  
:LABEL2:NAME "THIS IS SCREEN TEXT";XPOS  
100;YPOS 20;  
:DISPLAY:SCREENTEXT:LABEL3:NAME "";XPOS  
100;YPOS 35;  
:DISPLAY:SCREENTEXT:LABEL4:NAME "";XPOS  
100;YPOS 50;  
:DISPLAY:SCREENTEXT:LABEL5:NAME "";XPOS  
100;YPOS 343;  
:DISPLAY:SCREENTEXT:LABEL6:NAME "";XPOS
```

---

```
100;YPOS 80;  
:DISPLAY:SCREENTEXT:LABEL7:NAME "";XPOS  
100;YPOS 95;  
:DISPLAY:SCREENTEXT:LABEL8:NAME "";XPOS  
100;YPOS 110;  
:DISPLAY:WAVEFORM 1
```



---

# EMail

## Description

This command (no query form) sends a test e-mail message or sets the current e-mail sent count to zero.

## Group

Email

## Syntax 1

```
EMail {TESt|RESet}
```

## Arguments

- TESt

This argument sends a test e-mail message.

- RESet

This argument sets the e-mail sent count to zero.

## Example 1

```
EMail TESt
```

This command sends a test e-mail.

## Example 2

```
EMail RESet
```

This command resets the sent mail count to zero.

---

# EMail:ATTempts

## Description

This command sets or queries the number of times that an attempt will be made to send e-mail to the SMTP e-mail server. The default is 1.

## Group

Email

## Syntax 1

```
EMail: ATTempts <NR1>
```

## Syntax 2

```
EMail:ATTempts?
```

## Argument

- NR1

The value for NR1 can be 1 through 20. This number is the specified number of attempts that will be made to send e-mail to the SMTP server.

## Example 1

```
EMail:ATTempts 15
```

This command specifies that 15 attempts will be made to send the e-mail to the SMTP e-mail server.

## Example 2

```
EMail:ATTempts?
```

This query might return

```
:email:ATTEMPTS 10
```

This indicates that 10 attempts will be made to send the e-mail to the SMTP e-mail server.

---

# E-Mail:AUTHLogin

## Description

This command sets or queries the login name that will be used if the SMTP e-mail server requires one for authentication.

## Group

Email

## Syntax 1

```
E-Mail:AUTHLogin {<String>}
```

## Syntax 2

```
E-Mail:AUTHLogin?
```

## Argument

- <String>

This argument is a string that specifies the login name to be used with the SMTP e-mail server.

## Example 1

```
E-Mail:AUTHLogin kentb
```

This command specifies that kentb will be used as the login name with the SMTP e-mail server.

## Example 2

```
E-Mail:AUTHLogin?
```

This query might return

```
:EMAIL:AUTHLOGIN "bobk"
```

This indicates that bobk will be used as the login name used with the SMTP e-mail server.

---

# Email:AUTHPassword

## Description

This command (no query form) sets the password that will be used if the SMTP e-mail server requires one for authentication.

## Group

Email

## Syntax

```
Email:AUTHPassword {<String>}
```

## Argument

- <String>

This argument is a string that specifies the password to be used with the SMTP e-mail server.

## Example

```
Email:AUTHPassword "123xyz"
```

This command specifies that 123xyz will be used as the password with the SMTP e-mail server.

---

# EMail:COUnt?

## Description

This query only command returns the number of e-mails that have been sent since E-mail on Event was armed. The range of values returned can be from 0 to 50.

## Group

Email

## Syntax

```
EMail:COUnt?
```

## Example

```
EMail:COUnt?
```

This query might return

```
:email:COUNT 15
```

This indicates that 15 e-mails have been sent since e-mail on event was armed.

---

# EMail:FRom

## Description

This command sets or queries the From line in the e-mail. The default for the From line is the instrument model and serial number.

## Group

Email

## Syntax 1

```
EMail:FRom {<String>}
```

## Syntax 2

```
EMail:FRom?
```

## Argument

- <String>

This argument is a string that is placed in the From line of the e-mail. An example is "johnz".

## Example 1

```
EMail:FRom "kentb"
```

This command specifies that kentb will be placed in the From line of the e-mail.

## Example 2

```
EMail:FRom?
```

This query might return

```
:email:FROM "BOBK"
```

This indicates that bobk will appear in the From line of the e-mail.

---

# EMail:HOSTwanted

## Description

This command sets or queries the hostname that will be used when e-mail is sent to the SMTP e-mail server if the TDS hostname will not work.

## Group

Email

## Syntax 1

```
EMail:HOSTwanted {<String>}
```

## Syntax 2

```
EMail:HOSTwanted?
```

## Argument

- <String>

This argument is a string that specifies the host name to be used with the SMTP e-mail server if the default TDS host name will not work.

## Example 1

```
EMail:HOSTwanted "LazyEight"
```

This command specifies that LazyEight will be used as the hostname when sending e-mail to the SMTP e-mail server.

## Example 2

```
EMail:HOSTwanted?
```

This query might return

```
:EMAIL:HOSTWANTED "LazyEight"
```

This indicates that LazyEight is the hostname used with the SMTP e-mail server.

---

# EMail:IMage

## Description

This command sets or queries whether a screen image is included in e-mail. If this command is set to ON, the format and content of the screen image included is specified using the EXPort commands within the Hard Copy Command group.

## Group

Email

## Related Commands

EXPort

## Syntax 1

```
EMail:Image {ON|OFF|NR1}
```

## Syntax 2

```
EMail:Image?
```

## Arguments

- ON

This argument enables the inclusion of a screen image in e-mail.

- OFF

This argument disables the inclusion of a screen image in e-mail.

- <NR1>

A 0 disables the inclusion of a screen image in e-mail; any other value enables the inclusion of a screen image in e-mail.

## Example 1

```
EMail:Image 1
```

This command enables the inclusion of a screen image in e-mail.

## Example 2

```
EMail:Image?
```

This query might return

```
:EMAIL:IMAGE 0
```

This indicates that a screen image is not included in the e-mail.



---

# EMail:LIMit

## Description

This command sets or queries whether e-mail is sent when a limit test failure occurs. This command is the same as the LIMit:EMail command.

## Group

Email

## Related Commands

LIMit:EMail

## Syntax 1

```
EMail:LIMit {ON|OFF|NR1}
```

## Syntax 2

```
EMail:LIMit?
```

## Arguments

- ON

This argument enables sending e-mail when a limit test failure occurs.

- OFF

This disables sending e-mail when a limit test failure occurs.

- <NR1>

A 0 disables sending e-mail when a limit test failure occurs; any other value enables sending e-mail when a limit test failure occurs.

## Example 1

```
EMail:LIMit 1
```

This command enables sending e-mail when a limit test failure occurs.

## Example 2

```
EMail:LIMit?
```

This query might return

```
:EMAIL:LIMIT 0
```

This indicates that e-mail will not be sent when a limit test failure occurs.

---

# EMail:MASk

## Description

This command sets or queries whether e-mail is sent when a mask test failure occurs.

## Group

Email

## Syntax 1

```
EMail:MASk {ON|OFF|NR1}
```

## Syntax 2

```
EMail:MASk?
```

## Arguments

- ON

This argument enables sending e-mail when a mask test failure occurs.

- OFF

This disables sending e-mail when a mask test failure occurs.

- <NR1>

A 0 disables sending e-mail when a mask test failure occurs; any other value enables sending e-mail when a mask test failure occurs.

## Example 1

```
EMail:MASk 1
```

This command enables sending e-mail when a mask test failure occurs.

## Example 2

```
EMail:MASk?
```

This query might return

```
:EMAIL:MASK 0
```

This indicates that e-mail will not be sent when a mask test failure occurs.

---

# EMail:MAXsize

## Description

This command sets or queries the maximum size (in megabytes) of e-mail that can be sent to the SMTP server.

## Group

Email

## Syntax 1

```
EMail:MAXsize {<NR1>}
```

## Syntax 2

```
EMail:MAXsize?
```

## Argument

- <NR1>

The value for NR1 can be 0 through 2000 megabytes. This is the maximum size of each e-mail that can be sent to the SMTP server.

## Example 1

```
EMail:MAXsize 5
```

This command sets the maximum size of each e-mail enclosure to 5 megabytes of data.

## Example 2

```
EMail:MAXsize?
```

This query might return

```
:EMAIL:MAXSIZE 10
```

This indicates that each e-mail sent to the SMTP server can contain up to 10 megabytes of data.

---

# E<sub>M</sub>ail:MEASurement

## Description

This command sets or queries whether measurement data is included as an attachment to e-mail. You must specify the saved measurement selection (for all displayed measurements, measurement snapshots, or histogram data) and the measurement format using the Email on Event setup menu. There are no remote commands for this purpose.

## Group

Email

## Syntax 1

```
EMail:MEASurement {ON|OFF|NR1}
```

## Syntax 2

```
EMail:MEASurement?
```

## Arguments

- ON

This enables the inclusion of measurement data in e-mail.

- OFF

This disables the inclusion of measurement data in e-mail.

- <NR1>

A 0 disables the inclusion of measurement data in the e-mail; any other value enables including measurement data in the e-mail.

## Example 1

```
EMail:MEASurement 1
```

This command enables the inclusion of measurement data in e-mail.

## Example 2

```
EMail:MEASurement?
```

This query might return

```
:EMAIL:MEASUREMENT 0
```

This indicates that measurement data is not included in the e-mail.

---

# EMail:NUMemails

## Description

This command sets or queries the number of e-mails you can send when Email onEvent is armed, from 1 to 50. This limit is to restrict e-mail floods.

## Group

Email

## Related Commands

EMail RESet

## Syntax 1

```
EMail:NUMemails {<NR1>}
```

## Syntax 2

```
EMail:NUMemails?
```

## Argument

- <NR1>

The value of NR1 is 1 through 50. This is the number of e-mails you can send before the number of sent e-mails must be reset with the EMail RESet command.

## Example 1

```
EMail:NUMemails 10
```

This command sets the number of e-mails that can be sent to 10.

## Example 2

```
EMail:NUMemails?
```

This query might return

```
:EMAIL:NUMEMAILS 5
```

This indicates that five e-mails can be sent.

---

# EMail:SMTPPort

## Description

This command sets or queries the SMTP port number that the e-mail server uses if other than the default of 25.

## Group

Email

## Syntax 1

```
EMail: SMTPPort {<NR1>}
```

## Syntax 2

```
EMail:SMTPPort?
```

## Argument

- <NR1>

The value for NR1 can be 1 through 65535. This number specifies the TCPIP port number.

## Example 1

```
EMail:SMTPPort 55
```

This command specifies that the SMTP e-mail server will communicate on TCPIP port 55.

## Example 2

```
EMail:SMTPPort?
```

This query might return

```
:EMAIL:SMTPPORT 10
```

This indicates that TCPIP port 10 will be used to communicate with the SMTP e-mail server.

---

# Email:SMTPServer

## Description

This command sets or queries the address of the SMTP mail server.

## Group

Email

## Syntax 1

```
Email:SMTPServer <string>
```

## Syntax 2

```
Email:SMTPServer?
```

## Argument

- <string>

This argument is the address of the SMTP mail server that will handle the mail service request. For example, "smtp.tek.com".

## Example 1

```
Email:SMTPServer "smtp.tek.com"
```

This command sets the SMTP mail server to smtp.tek.com. This will be the mail server used for mail service requests.

## Example 2

```
Email:SMTPServer?
```

This query might return:

```
:EMAIL:SMTPSERVER "SMTP.TEK.COM"
```

This indicates that the SMTP mail server is set to SMTP.tek.com.

---

# EMail:STATus?

## Description

This query only command returns the status of the last e-mail you attempted to send. The status might be one of the following strings:

- PASS – indicates that the last e-mail message sent was successful or that no email message was attempted.
- FAIL – indicates that the last e-mail message attempted was not sent.
- SEND – indicates that an e-mail message is currently being sent.
- HOST\_FAIL – indicates that the instrument failed to connect to the SMTP server.
- CONNECTION\_FAIL – indicates that the SMTP server refused the connection.

## Group

Email

## Syntax

```
Email:STATus?
```

## Example 1

```
Email:STATus?
```

This query might return

```
:EMAIL:STATUS FAIL
```

indicating that the e-mail failed to be sent.



---

# E`Mail`:T`IME`out

## Description

This command sets or queries the global timeout in seconds. The default is 30 seconds. You use this timeout for socket connections and may need to change it from the default on some networks.

## Group

Email

## Syntax 1

```
EMail:TIMEout <NR1>
```

## Syntax 2

```
EMail:TIMEout?
```

## Argument

- <NR1>

This argument is the global timeout in seconds. The value for NR1 can be 0 through 500.

## Example 1

```
EMail:TIMEout 60
```

This command specifies that 60 seconds will be used for network connection timeouts.

## Example 2

```
EMail:TIMEout?
```

This query might return

```
:EMAIL:TIMEOUT 30
```

This indicates that the default of 30 seconds is set for network connection timeouts.

---

# EMail:TO

## Description

This command sets or queries the address of the recipient(s) of an e-mail.

## Group

Email

## Syntax 1

```
Email:TO <string>
```

## Syntax 2

```
Email:TO?
```

## Argument

- <string>

This argument is the e-mail address of the recipient (or recipients). Multiple addresses are separated with semicolons (;). For example, "johndoe@tek.com;billsmith@tek.com" specifies that both johndoe and billsmith at tek.com will receive e-mail when the event occurs. The length of this string is limited to 252 characters.

## Example 1

```
Email:TO "johndoe@tek.com"
```

This command will send e-mail to johndoe@tek.com when the event occurs.

## Example 2

```
Email:TO?
```

This query might return:

```
:EMAIL:TO "billsmill@tek.com"
```

This indicates that the billsmith@tek.com will receive e-mail when the event occurs.

---

# EMail:TRIGGER

## Description

This command sets or queries whether e-mail is sent when a trigger occurs. The e-mail is sent after the acquisition is complete.

## Group

Email

## Syntax 1

```
EMail:TRIGGER {ON|OFF|NR1}
```

## Syntax 2

```
EMail:TRIGGER?
```

## Arguments

- ON

This argument enables sending e-mail when a trigger occurs.

- OFF

This disables sending e-mail when a trigger occurs.

- <NR1>

A 0 disables sending e-mail when a trigger occurs; any other value enables sending e-mail when a trigger occurs.

## Example 1

```
EMail:TRIGGER 1
```

This command enables sending e-mail when a trigger occurs.

## Example 2

```
EMail:TRIGGER?
```

This query might return

```
:EMAIL:TRIGGER 0
```

This indicates that e-mail will not be sent when a trigger occurs.

---

# EMail:WAVEform

## Description

This command sets or queries whether waveform data is included in e-mail. The waveforms to be included, the start and stop points (and, or, frames for Fast Frame mode), and the data formats are specified by the DATA: and WFMOupre: commands within the Waveform Transfer Command group.

## Group

Email

## Syntax 1

```
EMail:WAVEform {ON|OFF|NR1}
```

## Syntax 2

```
EMail:WAVEform?
```

## Arguments

- ON

This enables the inclusion of waveform data in e-mail.

- OFF

This disables the inclusion of waveform data in e-mail.

- <NR1>

A 0 disables the inclusion of waveform data in the e-mail; any other value enables including waveform data in the e-mail.

## Example 1

```
EMail:WAVEform 1
```

This command enables the inclusion of waveform data in e-mail.

## Example 2

```
EMail:WAVEform?
```

This query might return

```
:EMAIL:WAVEFORM 0
```

This indicates that waveform data is not included in the e-mail.

---

# EVENT?

## Description

This query-only command returns an event code from the Event Queue that provides information about the results of the last \*ESR? read. EVENT? also removes the returned value from the Event Queue.

## Group

Status and Error

## Related Commands

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

## Syntax

EVENT?

## Example

EVENT?

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

---

# EVMsg?

## Description

This query-only command removes a single event code from the Event Queue that is associated with the results of the last \*ESR? read and returns the event code along with an explanatory message. For more information, see Event Handling.

## Group

Status and Error

## Related Commands

ALLEv?

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

## Syntax

EVMsg?

## Returns

The event code and message in the following format:

```
<Event Code><Comma><Qstring>[<Event Code><Comma><Qstring>...]<Qstring>::= <Message>;[<Command>]
```

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

## Example

EVMsg?

This query might return :EVMSG 110, "Command header error".

---

# EVQty?

## Description

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

## Group

Status and Error

## Related Commands

ALLEv?, EVENT?, EVMsg?

## Syntax

EVQty?

## Example

EVQty?

This query might return `:EVQTY 3`, indicating the number of event codes in the Event Queue.

---

# EXPort

## Description

This command sends a copy of the waveform to the file path specified by EXPort:FILENAME. The EXPort query returns image format and file information.

## Group

Export

## Syntax 1

```
EXPort START
```

## Syntax 2

```
Export?
```

## Arguments

- START

Initiates export.

## Example 1

```
EXPort START
```

This command initiates export to the file specified file by EXPort:FILENAME.

## Example 2

```
EXPort?
```

This query returns the export formatting and file path. The query might return:

```
:EXPORT:FILENAME "untitled";  
FORMAT BMP;IMAGE NORMAL;  
PALETTE COLOR;VIEW FULLSCREEN  
"C:\TekScope\Images\hcFILE.BMP"
```



---

# EXPort:FILENAME

## Description

This command sets or returns the file/path that will be sent export data on the next EXPort command.

## Group

Export

## Related Commands

HARDCopy

## Syntax 1

```
EXPort:FILENAME <file path>
```

## Syntax 2

```
EXPort:FILENAME?
```

## Argument

- <file path>

This specifies that the hard copy is sent to the named file. <file path> is a quoted string that defines the file name and path. Input the file path using the form <drive>:<dir>/<file-name>. <drive> and one or more <dir>s are optional. The file path cannot exceed 128 characters. If you don't specify the path with <drive>: and one or more <dir>s, the default location is "C:\TekScope\Images". While filename extensions are not required, they are highly recommended.

## Example 1

```
EXPort:FILENAME "TEK.BMP"
```

This command sets TEK.IBM as the selected file name. The next export will be sent to this file in the default hard copy directory.

## Example 2

```
EXPort:FILENAME?
```

This query might return

```
EXPORT:FILENAME "C:\TekScope\Images\TEK.BMP"
```

as the selected file.

---

# EXPort:FORMat

## Description

This command sets or returns the image format for exporting waveforms to a file.

## Group

Export

## Syntax 1

```
EXPort:FORMat {BMP|JPEG|PNG|PCX|TIFF}
```

## Syntax 2

```
EXPort:FORMat?
```

## Arguments

- BMP

Specifies BMP image format.

- JPEG

Specifies JPEG image format.

- PNG

Specifies PNG image format.

- PCX

Specifies PCX image format.

- TIFF

Specifies TIFF image format.

## Example 1

```
EXPort:FORMat BMP
```

This command sets the export format to BMP.

## Example 2

```
EXPort:FORMat?
```

This query might return

```
:EXPORT:FORMAT JPEG
```

indicating that waveforms will be exported in the JPEG format.

---

# EXPort:IMAGe

## Description

This command sets or returns the inksaver mode for exporting waveforms (File > Export Setup > Images > Image).

## Group

Export

## Syntax 1

```
EXPort:IMAGe {NORMal|INKSaver|ENHANCedwfm}
```

## Syntax 2

```
EXPort:IMAGe?
```

## Arguments

- **NORMal**

Preserves the screen background when printing waveforms.

- **INKSaver**

Changes the screen background to white. This saves ink when printing waveforms.

- **ENHANCedwfm**

Prints (color) screen images with colors designed to print on a white background.

## Example 1

```
EXPort:IMAGe NORMal
```

This command sets the export mode to Normal.

## Example 2

```
EXPort:IMAGe?
```

This query might return

```
:EXPORT:IMAGe NORMAL
```

indicating that the screen background will be printed when printing waveforms.

---

# EXPort:PALEtte

## Description

This command sets or returns the export color palette.

## Group

Export

## Syntax 1

```
EXPort:PALEtte {COLOr|GRAYscale|BLACKANDWhite}
```

## Syntax 2

```
EXPort:PALEtte?
```

## Arguments

- COLOr

Hardcopy output is color.

- GRAYscale

Hardcopy output is gray.

- BLACKANDwhite

Hardcopy output is black and white.

## Example 1

```
EXPort:PALEtte
```

This command sets the export color palette to Color.

## Example 2

```
EXPort:PALEtte
```

This query might return

```
:EXPORT PALETTE COLOR
```

indicating the hardcopy output will be in color.

---

# EXPort:VIEW

## Description

This command sets or returns the area of the screen to be exported.

## Group

Export

## Syntax 1

```
EXPort:VIEW {FULLSCREEN|GRAticule|FULLNOMenu}
```

## Syntax 2

```
EXPort:VIEW?
```

## Arguments

- FULLSCREEN

Displays both the graticule and menu areas of the screen.

- GRAticule

Displays only the graticule area of the screen.

- FULLNOMenu

Displays the full screen but hides any menus or toolbars.

## Example 1

```
EXPort:VIEW FULLSCREEN
```

This command sets the exported view area to Full Screen.

## Example 2

```
EXPort:VIEW?
```

This query might return

```
:EXPORT:VIEW GRATICULE
```

indicating that only the graticule area of the screen will be exported.

---

# FACTory

## Description

This command (no query form) resets the instrument to its factory default settings.

This command is equivalent to pressing the DEFAULT SETUP button located on the instrument front panel or selecting Recall Default Setup from the File menu.

## Group

Save and Recall

## Related Commands

\*PSC, \*RCL?, RECall:SETUp, \*RST, \*SAV, SAVe:SETUp

## Syntax

FACTory

## Arguments

None

## Example

FACTory

This command resets the instrument settings to factory defaults:

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

This command does not reset the following:

- The state of the GPIB (IEEE Std 488.1-1987) interface.
- The selected GPIB address.
- Calibration data that affects device specifications.
- Protected user data.
- Stored settings.
- The current password (if you set one).

---

# FASTAcq:STATE

## Description

This command sets or queries the state of Fast Acquisitions. This command is equivalent to the FASTACQ button on the front panel.

## Group

Acquisition

## Syntax 1

```
FASTAcq:STATE {ON|OFF|<NR1>}
```

## Syntax 2

```
FASTAcq:STATE?
```

## Arguments

- ON

This enables Fast Acquisitions mode.

- OFF

This disables Fast Acquisitions mode.

- <NR1>

A 0 disables Fast Acquisitions mode; any other value enables Fast Acquisitions mode.

## Example 1

```
FASTAcq:STATE ON
```

This command enables the Fast Acquisitions mode.

## Example 2

This query might return

```
:FASTACQ:STATE 1
```

indicating that Fast Acquisitions mode is currently active.

---

# FASTAcq?

## Description

This query-only command returns the state of Fast Acquisitions. This command is equivalent to pressing the FASTACQ button on the front panel.

## Group

Acquisition

## Syntax

FASTAcq?

## Example

FASTAcq?

This query might return the following string for the current acquisition parameters:

```
:FASTACQ:STATE 0
```



---

# FILESystem:COpy

## Description

This command (no query form) copies a named file to a new file. The new file may be in a totally separate directory than the old file. You can only copy one file at a time using this command. Wild card characters are not allowed.

## Group

File System

## Related Commands

FILESystem:CWD, FILESystem:DELEte

## Syntax

```
FILESystem:COpy {<source file path>,<destination file path>}
```

## Arguments

- <file path>

This is a quoted string that defines the file name and path. If the file path is within the current working directory, you need only specify the file name.

## Example

```
FILESystem:COpy "C:\MYDIR\TEK00001.SET",  
"C:\AnotherDir\copyOfTek001.set"
```

This command copies the file named TEK00001.SET, located in the MYDIR directory on the C drive to a file named copyOfTek001.set in the AnotherDir directory on the C drive.

---

# FILESystem:CWD

## Description

This command sets or queries the current working directory for FILE-System GPIB commands. The default working directory is "C:\TekScope". Anytime that you use this command to change the directory, the directory that you specify is retained as the current working directory until you either change the directory or you delete the directory. If you delete the current working directory, the instrument resets current working directory to the default directory (C:\TekScope) the next time the instrument is powered on or the next time you execute a file system command. The current working directory is retained between power cycles.

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

- Relative path names; for example, ".\Temp"
- Absolute path names; for example, "C:\Windows\System"
- Implied relative path names; for example "newfile.txt" becomes "C:\TekScope\newfile.txt" if the current working directory is "C:\TekScope"

## Group

File System

## Syntax 1

```
FILESystem:CWD {<new working directory path>}
```

## Syntax 2

## Arguments

- <new working directory path>

This is a quoted string that defines the current working; a directory name can be up to 128 characters.

## Example 1

```
FILESystem:CWD "C:\TekScope\images"
```

This command sets the current working directory to images.

## Example 2

```
FILESystem:CWD?
```

This query might return:

```
:FILESYSTEM:CWD "C:\TekScope\Waveforms"
```

This indicates that the current working directory is set to Waveforms.

---

# FILESystem:DELEte

## Description

This command (no query form) deletes a named file. It does not delete directories. Use the FILESystem:RMDir command to delete a directory.

## Group

File System

## Related Commands

FILESystem:COpy, FILESystem:CWD

## Syntax

```
FILESystem:DELEte <file path>
```

## Arguments

- <file path>

This is a quoted string that defines the file name and path. If the file path is within the current working directory, you need only specify the file name.

## Example 1

```
FILESystem:DELEte "NOT_MINE.SET"
```

This command deletes the file named NOT\_MINE.SET from the current working directory.

---

# FILESystem:DIR?

## Description

This query-only command returns a list of quoted strings. Each string contains the name of a file or directory in the current working directory.

## Group

File System

## Related Commands

FILESystem:CWD, FILESystem:MKDir

## Syntax

```
FILESystem:DIR?
```

## Arguments

None

## Example

```
FILESystem:DIR?
```

This query returns a list of files and directories in the current working directory.

---

# FILESystem:MKDir

## Description

This command (no query form) creates a new directory.

## Group

File System

## Related Commands

FILESystem:CWD, FILESystem:DIR?

## Syntax

```
FILESystem:MKDir <directory path>
```

## Arguments

- <directory path>

This is a quoted string that specifies the directory to create

## Example 1

```
FILESystem:MKDir "C:\NewDirectory"
```

This command creates the directory named `NewDirectory` at the root of the C drive.

## Example 2

```
FILESystem:CWD "C:\MyDirectory"  
FILESystem:MKDir "MyNewSubDirectory"
```

These two commands create the directory `MyNewSubDirectory` within the existing directory `MyDirectory` at the root of the C drive. This, of course, assumes that `C:\MyDirectory` already existed and was not a read-only directory.

---

# FILESystem:PRInt

## Description

This command (no query form) prints a named file to a named port. This command is the same as the FILESystem:READFile command.

This command is not IEEE 488.2 compliant for the following reasons:

- The command produces output on the 488.2 bus. Queries, not commands, are the only message units that should produce output on the 488.2 bus. Errors may not be recognized if they occur while data is being output in response to this command. This will be evident if a command is sent to the instrument before all of the data generated by this command has been read. In such cases, you should send a device clear message to the instrument.
- The command does not use a standard 488.2 output format. The closest defined format is the indefinite format of ARBITRARY BLOCK PROGRAM DATA encoding. However, this command output omits the initial '#0' characters needed for that format.
- Newlines, ASCII value 10, can be included in the output data. Output on the bus can be terminated by newlines. The output of this command is only terminated by <EOI>.

## Group

File System

## Related Commands

FILESystem:CWD, FILESystem:READFile

## Syntax

```
FILESystem:PRInt <filepath>, GPIb
```

## Arguments

- <file path>

This is a quoted string that defines the file name and path. If the file path is within the current working directory, you need only specify the file name.

- GPIb

This specifies that the hardcopy is sent out of the GPIB port. GPIb is the only port allowed for this command.

## Example

```
FILESystem:PRInt "C:\TEK00000.IBM", GPIb
```

This command sends the file named TEK00000.IBM out of the GPIB port.

---

# FILESystem:READFile

## Description

This command (no query form) prints a named file to a named port. It is identical to the FILESystem:PRInt command.

This command is not IEEE 488.2 compliant for the following reasons:

- The command produces output on the 488.2 bus. Queries, not commands, are the only message units that should produce output on the 488.2 bus. Errors may not be recognized if they occur while data is being output in response to this command. This will be evident if a command is sent to the instrument before all of the data generated by this command has been read. In such cases, you should send a device clear message to the instrument.
- The command does not use a standard 488.2 output format. The closest defined format is the indefinite format of ARBITRARY BLOCK PROGRAM DATA encoding. However, this command output omits the initial '#0' characters needed for that format.
- Newlines, ASCII value 10, can be included in the output data. Output on the bus can be terminated by newlines. The output of this command is only terminated by <EOI>.

## Group

File System

## Related Commands

FILESystem:CWD, FILESystem:PRInt

## Syntax

```
FILESystem:READFile <filepath>, GPIb
```

## Arguments

- <file path>

This is a quoted string that defines the file name and path. If the file path is within the current working directory, you need only specify the file name.

- GPIb

This specifies that the hardcopy is sent out of the GPIB port. GPIb is the only port allowed for this command.

## Example

```
FILESystem:READFile "C:\TEK00000.IBM", GPIb
```

This command sends the file named TEK00000.IBM out of the GPIB port.

---

# FILESystem:REName

## Description

This command (no query form) assigns a new name to an existing file.

## Group

File System

## Related Command

FILESystem:CWD

## Syntax

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

## Arguments

- <old file path>

This is a quoted string that defines the file name and path. If the file path is within the current working directory, you need only specify the file name.

- <new file path>

This is a quoted string that defines the file name and path. If the file path is within the current working directory, you need only specify the file name.

## Example

```
FILESystem:REName "C:\TEK00000.SET", "C:\MYSETTING.SET"
```

This command gives the file named TEK00000.SET the new name of MYSETTING.SET. The file remains in the root directory on the C drive.



---

# FILESystem:RMDir

## Description

This command (no query form) deletes a named directory. The directory cannot contain any files. If there are files in the directory, they must first be deleted. The directory must not be a read-only directory.

## Group

File System

## Related Command

FILESystem:CWD

## Syntax

```
FILESystem:RMDir <directory path>
```

## Arguments

- <directory path>

This is a quoted string that defines the file name and path. If the file path is within the current working directory, you need only specify the file name.

## Example

```
FILESystem:RMDir "C:\OldDirectory"
```

This command removes the directory named OldDirectory from the root of the C drive.

---

# FILESystem:WRITEFile

## Description

This command (no query form) copies the block data from the GPIB port to a named file.

## Group

File System

## Related Command

FILESystem:CWD

## Syntax

```
FILESystem:WRITEFile <file path>,<data>
```

## Arguments

- <file path>

This is a quoted string that defines the file name and path. If the file path is within the current working directory, you need only specify the file name

- <data>

This data can be either DEFINITE LENGTH or INDEFINITE LENGTH ARBITRARY BLOCK PROGRAM DATA encoding as described in IEEE 488.2.

## Example 1

```
FILESystem:WRITEFile "C:\SomeNewFile.txt",#221  
1.23 2.54 8.22 -1.22<EOI>
```

This command creates a file named "SomeNewFile.txt" on the root of the C drive, containing the string of 21 characters "1.23 2.54 8.22 -1.22". This is an example of using the definite length arbitrary block.

## Example 2

```
FILESystem:WRITEFile "C:\SomeNewFile.txt",#0  
This is a test<NL+EOI>
```

This command creates a file named "SomeNewFile.txt" on the root of the C drive, containing the string "This is a test." This is an example of using the indefinite length arbitrary block. The newline character (0x0a) is required.

---

```
FILESystem:WRITEFile "C:\SomeNewFile.txt",#221  
1.23 2.54 8.22 -1.22<NL+EOI>
```

This command creates a file named "SomeNewFile.txt" on the root of the C drive, containing the string of 21 characters "1.23 2.54 8.22 -1.22". Because the data argument is indefinite format arbitrary block program data, it may contain any 8-bit values, including newline characters (hexadecimal 0a). The fact that newline characters are allowed makes this command deviate from the strictest interpretation of the IEEE488.2 specification.

---

# FILESystem?

## Description

This query-only command returns the directory listing of the current working directory. This query is the same as the FILESystem:DIR? query.

## Group

File System

## Related Commands

FILESystem:COPy, FILESystem:CWD, FILESystem:DELEte, FILESystem:DIR?, FILESystem:PRInt, FILESystem:READFile, FILESystem:REName, FILESystem:WRITEFile

## Syntax

FILESystem?

## Arguments

None.

## Example

FILESystem?

This query might return :

```
:FILESYSTEM:DIR, ".","..","myFile.txt","myWaveform.wfm".
```

---

# HARDCopy

## Description

This command sends a copy of the screen display to the port specified by HARDCopy:PORT. This command is equivalent to pressing the PRINT button on the front panel. When printing to a file, the file format is Windows Bitmap Image (BMP).

The HARDCopy query returns the port and file path.

## Group

Hardcopy

## Related Commands

\*WAI, \*CLS?

## Syntax 1

```
HARDCopy START
```

## Syntax 2

```
HARDCopy?
```

## Arguments

- START

This initiates a screen copy to a file or the default system printer, as specified by the :HARDCopy:PORT selection. The default system printer is set within the Windows operating system. If you need information about how to set the default system printer, refer to Microsoft Windows online help.

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

## Example 1

```
HARDCopy START
```

This command initiates a screen copy to a file or the default system printer, as specified by the :HARDCopy:PORT selection.

## Example 2

```
HARDCopy?
```

This query returns the hardcopy file path and the selected port. The query might return:

```
:HARDCOPY:FILENAME "C:\TekScope\Images\hcFILE.BMP";PORT FILE.
```

---

# HARDCopy:FILENAME

## Description

This command sets or returns the file that will be sent hardcopy data on the next HARDCopy command (if the HARDCopy:PORT is set to FILE).

## Group

Hardcopy

## Related Commands

HARDCopy

## Syntax 1

```
HARDCopy:FILENAME <file path>
```

## Syntax 2

```
HARDCopy:FILENAME?
```

## Argument

- <file path>

This specifies that the hard copy is sent to the named file. <file path> is a quoted string that defines the file name and path. Input the file path using the form <drive>:<dir>/<file-name>. <drive> and one or more <dir>s are optional. The file path cannot exceed 128 characters. If you don't specify the path with <drive>: and one or more <dir>s, the default location is "C:\Tek-Scope\Images". While filename extensions are not required, they are highly recommended.

## Example 1

```
HARDCopy:FILENAME "TEK.BMP"
```

This command sets TEK.BMP as the selected file name. The next hardcopy will be sent to this file in the default hardcopy directory, if the :HARDCopy:PORT selection is FILE.

## Example 2

```
HARDCopy:FILENAME?
```

This query might return :HARDCOPY:FILENAME "C:\Tek-Scope\Images\TEK.BMP" as the selected file.

---

### Example 3

`HARDCopy:FILENAME "TEK?.BMP"`

This command sets TEK as the selected file name with a numeric, two-digit suffix. The instrument might return

`:HARDCOPY:FILENAME "TEK00.BMP";"TEK01.BMP"`

This indicates TEK00.BMP is the first file and TEK01.BMP is the second file.

---

# HARDCopy:IMAGe

## Description

This command sets or returns the inksaver mode for hard copy printing.

## Group

Hardcopy

## Syntax 1

```
HARDCopy:IMAGe {NORMal | INKSaver | ENHANCedwfm}
```

## Syntax 2

```
HARDCopy:IMAGe?
```

## Arguments

- **NORMal**

Preserves the screen background when printing waveforms.

- **INKSaver**

Changes the screen background to white. This saves ink when printing waveforms.

- **ENHANCedwfm**

Prints screen images with colors designed to print on a white background.

## Example 1

```
HARDCopy:IMAGe NORMal
```

This command sets the hardcopy mode to Normal.

## Example 2

```
HARDCopy:IMAGe?
```

This query might return

```
:HARDCOPY:IMAGe INKSAVER
```

This indicates that the inksaver mode is on and the screen background is white.



---

# HARDCopy:LAYout

## Description

This command sets or returns the page orientation for hard copy.

## Group

Hardcopy

## Syntax 1

```
HARDCopy:LAYout {PORTRait|LANDscape}
```

## Syntax 2

```
HARDCopy:LAYout?
```

## Arguments

- PORTRait

Orients the screen image vertically on the printed page.

- LANDscape

Orients the screen image horizontally on the printed page.

## Example 1

```
HARDCopy:LAYout LANDscape
```

This command sets the hardcopy page orientation to Landscape.

## Example 2

```
HARDCopy:LAYout?
```

This query might return

```
:HARDCOPY:LAYOUT PORTRAIT
```

indicating that the hardcopy page orientation is set to portrait.

---

# HARDCopy:PALEtte

## Description

This command sets or returns the hardcopy color palette.

## Group

Hardcopy

## Syntax 1

```
HARDCopy:PALEtte {COLOr|GRAYscale|BLACKAND-  
White}
```

## Syntax 2

```
HARDCopy:PALEtte?
```

## Arguments

- COLOr

This argument sets the hardcopy output to color.

- GRAYscale

This argument sets the hardcopy output to gray scale

- BLACKANDwhite

This argument sets the hardcopy output to black and white.

## Example 1

```
HARDCopy:PALEtte COLOr
```

This command sets the hardcopy output to Color.

## Example 2

```
HARDCopy:PALEtte?
```

This argument might return

```
:HARDCOPY:PALETTE GRAYSCALE
```

indicating that the output for hardcopy will be in gray scale.

---

# HARDCopy:PORT

## Description

This command selects or returns whether the hardcopy data will be sent to a file or printed on the next hardcopy command (for example, the HARDCopy START command). This is equivalent to selecting Print in the File menu and then either choosing Print to file or specifying the default printer.

## Group

Hardcopy

## Related Commands

HARDCopy, HARDCopy:FILENAME

## Syntax 1

```
HARDCopy:PORT {FILE|PRINTER}
```

## Syntax 2

```
HARDCopy:PORT?
```

## Arguments

- FILE

This argument specifies that the hardcopy is stored in the file specified in the HARDCopy:FILENAME command.

- PRINTER

This argument specifies that the hardcopy is sent to the printer specified in the Print dialog box.

## Example 1

```
HARDCopy:PORT FILE
```

This command specifies

That the hardcopy is stored in the file specified in the HARDCopy:FILENAME command.

## Example 2

```
HARDCopy:PORT?
```

This query might return

```
:HARDCOPY:PORT FILE
```

This indicates that hardcopies are sent to the specified file.

---

# HARDCopy:VIEW

## Description

This command sets or returns the area of the screen to be hardcopied.

## Group

Hardcopy

## Syntax 1

```
HARDCopy:VIEW {FULLSCREEN|GRAticule|FULLNOME-  
nu}
```

## Syntax 2

```
HARDCopy:VIEW?
```

## Arguments

- FULLSCREEN

Sets the area to be hardcopied to both the graticule and menu areas of the screen

- GRAticule

Sets the area to be hardcopied to only the graticule area of the screen.

- FULLNOMenu

Sets the area to be hardcopied to full screen but hides any menus or toolbars.

## Example 1

```
HARDCopy:VIEW FULLSCREEN
```

This command sets the hardcopy area to Full Screen.

## Example 2

```
HARDCopy:VIEW?
```

This query might return

```
:HARDCOPY:VIEW GRATICULE
```

indicating that only the graticule area of the screen will be hardcopied.

---

# HDR

## Description

This command is identical to the HEADer query and is included for backward compatibility purposes.

## Group

Miscellaneous

## Related Commands

HEADer, VERBose

## Syntax 1

```
HDR {OFF|ON|<NR1>}
```

## Syntax 2

```
HDR?
```

## Arguments

- OFF

This sets the Response Header Enable State to false. This causes the instrument to omit headers on query responses, so that only the argument is returned.

- ON

This sets the Response Header Enable State to true. This causes the instrument to include headers on applicable query responses. You can then use the query response as a command.

- <NR1>

A 0 sets the Response Header Enable State to false; any other value sets this state to true, which causes the instrument to omit headers on query responses.

## Example 1

```
HDR OFF
```

This command specifies that the instrument omits headers on query responses, so that only the argument is returned.

## Example 2

```
HDR?
```

This query might return `:HEADER 1`, indicating that the instrument is including headers on applicable query responses.

---

# HEADer

## Description

This command sets or queries the Response Header Enable State that causes the instrument to either include or omit headers on query responses.

**Note:** This command does not affect IEEE Std 488.2-1987 Common Commands (those starting with an asterisk); these commands never return headers. This command does make a corresponding change in the Response Header Enable State of the opposite interface (physical or virtual GPIB interface). Refer to Introduction for more information.

## Group

Miscellaneous

## Related Commands

HDR, VERBose

## Syntax 1

```
HEADer {OFF | ON | <NR1>}
```

## Syntax 2

```
HEADer?
```

## Arguments

- OFF

This sets the Response Header Enable State to false. This causes the instrument to omit headers on query responses, so that only the argument is returned.

- ON

This sets the Response Header Enable State to true. This causes the instrument to include headers on applicable query responses. You can then use the query response as a command.

- <NR1>

A 0 sets the Response Header Enable State to false; any other value sets this state to true, which causes the instrument to omit headers on query responses.

## Example 1

```
HEADer OFF
```

This command specifies that the instrument omits headers on query responses, so that only the argument is returned.

---

## Example 2

HEADer?

This query might return

:HEADER 1

This indicates that the instrument is including headers on applicable query responses.

---

# HIStogram?

## Description

This query-only command returns all histogram parameters; it queries the state of all histogram parameters that the user can set. This command is equivalent to selecting Waveform Histograms from the Measure menu.

## Group

Histogram

## Related Commands

MEASUrement:METHOD

## Syntax 1

HIStogram?

## Example 1

HIStogram?

This query might return the histogram parameters as

```
:HISTOGRAM:BOXPCNT  
30.0000,25.1000,70.0000,75.2000;DISPLAY LIN-  
EAR;STATE 1;FUNCTION VERTICAL;SIZE  
2.0000;SOURCE CH1
```



---

# HISistogram:Box

## Description

This command defines or returns the left, top, right, and bottom boundaries of the histogram box, in source waveform coordinates. This command is equivalent to selecting Waveform Histograms from the Measure menu and then setting Limits for Left, Right, Top, and Bottom in the appropriate boxes. The command is similar to the HISistogram:BOXPent command except that command uses percentage coordinates to define the boundaries of the histogram box.

**Note:** This command implicitly clears the histogram count and statistics for the histogram and histogram source data when HISistogram:STATE is turned on.

## Group

Histogram

## Related Commands

HISistogram:BOXPent

## Syntax 1

```
HISistogram:Box <NR3>,<NR3>,<NR3>,<NR3>
```

## Syntax 2

```
HISistogram:Box?
```

## Arguments

- <NR3>

This specifies the left position of the histogram box in source waveform coordinates.

- <NR3>

This specifies the top position of the histogram box in source waveform coordinates.

- <NR3>

This specifies the right position of the histogram box in source waveform coordinates.

- <NR3>

This specifies the bottom position of the histogram box in source waveform coordinates.

## Example 1

```
HISistogram:Box 1E-9, 0.250, 2E-9, 0.500
```

This command defines the coordinates of the histogram box in source waveform coordinates.

---

## Example 2

HISTogram:Box?

This query might return

```
:HISTOGRAM:BOX 1.0000E-09, 0.2500, 2.0000E-09,  
0.5000
```

This identifies the coordinates of the histogram box in source waveform coordinates.

---

# HISistogram:BOXPcnt

## Description

This command defines or queries the left, top, right, and bottom positions of the histogram box, in percentage coordinates. The upper left has the value 0,0 and the lower right has the value 100, 100. Any value outside the range of 0 to 100 is clipped. This command is similar to the HISistogram:Box command except that this command uses percentage coordinates to define the boundaries of the histogram box.

**Note:** This command implicitly clears the histogram count and statistics for the histogram and histogram source data when HISistogram:STATE is turned on.

## Group

HISistogram

## Related Commands

HISistogram:Box

## Syntax 1

```
HISistogram:BOXPcnt <NR3>, <NR3>, <NR3>, <NR3>
```

## Syntax 2

```
HISistogram:BOXPcnt?
```

## Arguments

- <NR3>

This specifies the left position of the histogram box in percentage coordinates. The default value is 30.0.

- <NR3>

This specifies the top position of the histogram box in percentage coordinates. The default value is 25.1.

- <NR3>

This specifies the right position of the histogram box in percentage coordinates. The default value is 70.0.

- <NR3>

This specifies the bottom position of the histogram box in percentage coordinates. The default value is 75.2.

---

### Example 1

```
HISTogram:BOXPcnt 30.0,25.0,70.0,75.0
```

This command sets the left boundary of the histogram box to 30% of the graticule (3 divisions from the left edge); the top boundary to 25% of the graticule ( $0.25 \times 8 = 2.0$  divisions from the top edge); the right boundary to 70% of the graticule (7 divisions from the left edge); and the bottom boundary to 75% of the graticule ( $0.75 \times 8 = 6.0$  divisions from the top edge).

### Example 2

```
HISTogram:BOXPct?
```

This query might return :

```
:HISTOGRAM:BOXPCNT  
30.0000,25.1000,70.0000,75.2000.
```

---

# HIStogram:COUNT

## Description

This command (no query form) clears the count and statistics for the histogram and the histogram source data. If the histogram is on, then counting restarts. This command is equivalent selecting Waveform Histograms from the Measure menu and then clicking the Reset button.

## Group

HIStogram

## Related Commands

HIStogram:Box, HIStogram:BOXPct

## Syntax

```
HIStogram:COUNT RESET
```

- RESET

This zeros the histogram count.

## Example

```
HIStogram:COUNT RESET
```

This command clears the count and statistics for the histogram and the histogram source data.



---

# HIStogram:DISplay

## Description

This command sets the scaling of the histogram data display to be the count of each histogram bin or the log of that count. The default scaling is Linear.

The query form of this command returns either the histogram display scaling or the display status "disabled."

This command is equivalent to selecting Waveform Histogram from the Measure menu, and then selecting Linear or Log in the Histogram Scaling group.

## Group

HIStogram

## Related Commands

HIStogram:MODE, HIStogram:STATE

## Syntax 1

```
HIStogram:DISplay {OFF|LOG|LINEAr}
```

## Syntax 2

```
HIStogram:DISplay?
```

## Arguments

- <OFF>

Disables the histogram display but allows histogram data to accumulate.

**Note:** This argument provides for backward compatibility with older TDS Series instruments; it does not affect a change in the display scaling of the histogram. When queried, HIStogram:DISplay? will return OFF if the histogram state or mode has been disabled using the HIStogram:MODE or HIStogram:STATE commands.

- <LOG>

Sets the histogram display to logarithmic scaling.

- <LINEAr>

Sets the histogram display to linear scaling. This is the default setting.

## Example 1

```
HIStogram:DISplay LOG
```

This command sets the histogram scaling to be the log count of each histogram bin.

---

## Example 2

HISTogram:DISplay?

This query might return

:HISTOGRAM:DISPLAY LOG

indicating that the current histogram display scaling is the log count of each histogram bin.



---

# HIStogram:FUNcTion

## Description

This command either selects the type of histogram to create or returns the current histogram type. This command is equivalent to selecting Waveform Histogram from the Measure menu and then choosing either Horizontal or Vertical from the Histogram Mode group box.

## Group

HIStogram

## Related Commands

HIStogram:MODE, HIStogram:STATE

## Syntax 1

```
HIStogram:FUNcTion {HORizontal|VERTical}
```

## Syntax 2

```
HIStogram:FUNcTion?
```

## Arguments

- **HORizontal**

This displays a horizontally positioned histogram that shows time distribution.

- **VERTical**

This displays a vertically positioned histogram that shows a voltage distribution (or another distribution such as amperes).

## Example 1

```
HIStogram:FUNcTion HORizontal
```

This command sets the histogram mode to the horizontal display mode.

## Example 2

```
HIStogram:FUNcTion?
```

This query might return

```
:HISTOGRAM:FUNCTION VERTICAL
```

This indicates that the vertical display mode is selected for the histogram.

---

# HIStogram:MODe

## Description

This command selects the type of histogram to create or disables the histogram display. The query form either returns the current histogram type or that histogram display is disabled. This command is equivalent to selecting Waveform Histogram from the Measure menu and then choosing from the Histogram Mode group box.

HIStogram:MODe is functionally equivalent to the HIStogram:FUNCTION and HIStogram:STATE commands and is included for backward compatibility purposes.

## Group

HIStogram

## Related Commands

HIStogram:FUNCTION, HIStogram:STATE, HIStogram:DISplay

## Syntax 1

```
HIStogram:MODe {HORizontal|VERTical|OFF}
```

## Syntax 2

```
HIStogram:MODe?
```

## Arguments

- **HORizontal**

This enables a horizontally positioned histogram that shows time distribution.

- **VERTical**

This enables a vertically positioned histogram that shows a voltage distribution (or another distribution such as amperes).

- **OFF**

This disables collection of the histogram measurement.

## Example 1

```
HIStogram:MODe HORizontal
```

This command sets the type of histogram created to horizontal, which displays a horizontally positioned histogram that shows time distribution.

---

## Example 2

HISTogram:MODE?

This query might return

:HISTOGRAM:MODE OFF

This indicates that histogram display is disabled.

---

# HIStogram:SIze

## Description

This command sets or queries the width or height of the histogram on the screen. This is equivalent to selecting Waveform Histograms from the Measure menu and then entering a value in the Histogram Size box.

## Group

HIStogram

## Related Commands

HIStogram:DISplay

## Syntax 1

```
HIStogram:SIze <NR3>
```

## Syntax 2

```
HIStogram:SIze?
```

## Arguments

- <NR3>

This specifies the histogram size. The value can vary from 0.1 to 8.0 divisions in HORizontal mode and from 0.1 to 10.0 divisions in VERtical mode. Resolution is to the nearest pixel.

## Example 1

```
HIStogram:SIze 2.0
```

This command sets the size of the histogram to 2.0 divisions.

## Example 2

```
HIStogram:SIze?
```

This query might return

```
:HISTOGRAM:SIze 2.0000E+01
```

This indicates the histogram size is set to 2.0 divisions.

---

# HIStogram:SOUrce

## Description

This command sets or queries which source will be compared against the histogram box when the histogram testing is enabled. This is equivalent to selecting Waveform Histograms from the Measure menu and then choosing the desired waveform source. The waveform need not be displayed for histograms to run. You might want the channel displays disabled so you can see a full-screen histogram and not have waveform data confuse the display.

## Group

HIStogram

## Related Commands

HIStogram:DISplay,

## Syntax 1

```
HIStogram:SOUrce {CH<x>|MATH<x>|REF<x>}
```

## Syntax 2

```
HIStogram:SOUrce?
```

## Arguments

### ■ CH<x>

This selects a channel waveform as the source for the histogram. The x variable can be expressed as an integer ranging from 1 through 4 for 4-channel models and 1 to 2 for 2-channel models.

### ■ MATH<x>

This selects a math waveform as the source for the histogram. The x variable can be expressed as an integer ranging from 1 through 4 for 4-channel models and 1 to 2 for 2-channel models.

### ■ REF<x>

This selects a reference waveform as the source for the histogram. The x variable can be expressed as an integer ranging from 1 through 4 for 4-channel models and 1 to 2 for 2-channel models.

## Example 1

```
HIStogram:SOUrce CH1
```

This command enables the channel 1 waveform to be compared against the histogram box. The default time base, Main, is used for the histogram display.

---

## Example 2

HISTogram:SOUrce?

This query might return

:HISTOGRAM:SOURCE CH1

This indicating that the waveform for channel 1 is the source for the histogram.

---

# HIStogram:STATE

## Description

This command sets or queries whether the histogram calculations are enabled. This is equivalent to selecting Waveform Histograms from the Measure menu.

## Group

HIStogram

## Related Commands

HIStogram:MODE, MEASUrement:METHod

## Syntax 1

```
HIStogram:STATE {ON|OFF|NR1}
```

## Syntax 2

```
HIStogram:STATE?
```

## Arguments

### ■ <ON>

This enables histogram calculations.

### ■ <OFF>

This disables the histogram calculations.

### ■ <NR1>

A 0 disables histogram calculations; any other value enables histogram calculations.

## Example 1

```
HIStogram:STATE ON
```

This command enables histogram calculations.

## Example 2

```
HIStogram:STATE?
```

This query might return

```
:HISTOGRAM:STATE 0
```

This indicates that histogram calculations are currently disabled.

---

# HORizontal:DIVisions?

## Description

This query-only command returns the number of graticule divisions over which the waveform is displayed.

## Group

Horizontal

## Syntax

```
HORizontal:DIVisions?
```

## Arguments

None.

## Example

```
HORizontal:DIVisions?
```

This query might return

```
:HORIZONTAL:DIVISIONS 10.0000
```

indicating that the waveform is displayed across ten division.



---

# HORizontal:FASTframe:COUNT

## Description

This command sets or queries FastFrame frame count. This command is equivalent to selecting Fast Frame Setup from the Horiz/Acq menu and entering a value in the Frame Count box. FastFrame captures a series of triggered acquisitions with minimal intervening time.

## Group

Horizontal

## Related Commands

HORizontal:RECOrdlength

## Syntax 1

```
HORizontal:FASTframe:COUNT <NR1>
```

## Syntax 2

```
HORizontal:FASTframe:COUNT?
```

## Arguments

- <NR1>

This indicates the number of frames to acquire.

## Example 1

```
HORIZONTAL:FASTFRAME:COUNT 2
```

This command sets up FastFrame mode to acquire two frames (segments) of data.

## Example 2

```
HORIZONTAL:FASTFRAME:COUNT?
```

This query might return

```
:HORIZONTAL:FASTFRAME:COUNT 4
```

indicating that FastFrame mode is currently set to acquire 4 frames of data.

---

# HORizontal:FASTframe:LENgth

## Description

This command sets or queries the horizontal record length to the number of sample points in each frame. This command is equivalent to selecting FastFrame Setup from the Horiz/Acq menu and entering a value in the Rec Length box. FastFrame captures a series of triggered acquisitions with minimal intervening time between them.

## Group

Horizontal

## Syntax 1

```
HORizontal:FASTframe:LENgth <NR1>
```

## Syntax 2

```
HORizontal:FASTframe:LENgth?
```

## Argument

- <NR1>

<NR1> represents the supported value for horizontal record length in Fast Frame mode. In addition, to the allowable record length in normal (single frame) mode, 50 and 250 point record lengths are permitted in Fast Frame mode.

For instruments with standard memory configuration, you can have a maximum Fast Frame record length of 8M, 4M, or 2M for 1, 2, or 4 channel operation, respectively. For instruments with 3M memory option, the maximum Fast Frame record lengths are 16M, 8M, or 4M for 1, 2, or 4 channel operation, respectively.

For more information about valid data point ranges, select Specifications from the Help menu; then select the Horizontal & Acquisition tab.

## Example 1

```
HORIZONTAL:FASTFRAME:LENGTH 5000
```

This command sets the horizontal record length to 5000 sample points in each frame.

## Example 2

```
HORIZONTAL:FASTFRAME:LENGTH?
```

This query might return

```
:HORIZONTAL:FASTFRAME:LENGTH 25000
```

This indicates that the FastFrame record length is set to 25000 sample points in each frame.

---

# HORizontal:FASTframe:MULTipleframes: FRAMEStart:<wfm>

## Description

This command sets or queries the start frame number on the specified waveform for the FastFrame multiple frames feature. Valid waveforms include CH<x>, MATH<x>, and REF<x>. The value of x can be an integer ranging from 1 through 4 in four-channel instruments and 1 through 2 in two-channel instruments. The multiple frames feature supports displaying multiple frames in an overlaid manner.

## Group

Horizontal

## Related Commands

HORizontal:FASTframe:MULTipleframes:NUMFRames:<wfm>  
<NR1>,  
HORizontal:FASTframe:MULTipleframes:MODE,  
HORizontal:FASTframe:COUNt

## Syntax 1

```
HORizontal:FASTframe:MULTipleframes:FRAMES-  
Tart:<wfm> <NR1>
```

## Syntax 2

```
HORizontal:FASTframe:MULTipleframes:FRAMES-  
Tart:<wfm>?
```

## Argument

- <NR1>

This specifies the start frame number on the specified waveform.

## Example 1

```
HORIZONTAL:FASTFRAME:MULTIPLEFRAMES:FRAMES-  
TART:CH1 3
```

This command sets the start frame number to frame 3 on channel 1.

## Example 2

```
HORIZONTAL:FASTFRAME:MULTIPLEFRAMES:FRAMES-  
TART?
```

This query might return

```
:HORIZONTAL:FASTFRAME:MULTIFRAMES:FRAMES-  
TART:CH1 3
```

This indicates that the start frame number is frame 3 for channel 1.

---

# HORizontal:FASTframe:MULtipleframes:MODE

## Description

This command sets or queries the mode for the FastFrame multiple frames feature. This feature displays multiple frames in an overlaid manner.

## Group

Horizontal

## Related Commands

HORizontal:FASTframe:MULtipleframes:FRAMEStart:<wfm>  
<NR1>,  
HORizontal:FASTframe:MULtipleframes:NUMFRAMES:<wfm>  
<NR1>,  
HORizontal:FASTframe:COUNT

## Syntax 1

```
HORizontal:FASTframe:MULtipleframes:MODE  
{OFF|OVERlay}
```

## Syntax 2

```
HORizontal:FASTframe:MULtipleframes:MODE?
```

## Argument

- OFF

This turns off the multiple frames mode.

- OVERlay

This sets the multiple frames mode to overlay.

## Example 1

```
HORizontal:FASTframe:MULtipleframes:MODE OFF
```

This command turns off the multiple frames mode.

## Example 2

```
HORizontal:FASTframe:MULtipleframes:MODE?
```

This query might return:

```
:HORIZONTAL:FASTFRAME:MULTIPLEFRAMES:MODE OFF
```

This indicates that the multiple frames mode is off.

---

# HORizontal:FASTframe:MULTipleframes: NUMFRames:<wfm>

## Description

This command sets or queries the number of frames on the specified waveform for the FastFrame multiple frames feature. The multiple frames feature supports displaying multiple frames in an overlaid manner.

## Group

Horizontal

## Related Commands

HORizontal:FASTframe:MULTipleframes:FRAMEStart:<wfm>  
<NR1>,  
HORizontal:FASTframe:MULTipleframes:MODE,  
HORizontal:FASTframe:COUNt

## Syntax 1

```
HORizontal:FASTframe:MULTipleframes:NUM-  
FRAMES:<wfm> <NR1>
```

## Syntax 2

```
HORizontal:FASTframe:MULTipleframes:NUM-  
FRAMES:<wfm>?
```

## Argument

- <wfm>

This specifies the waveform source. Valid waveforms include CH<x>, MATH<x>, and REF<x>. Express the x variable as an integer ranging from 1 through 4 in four-channel instruments and 1 through 2 in two-channel instruments.

- <NR1>

This represents the number of frames on the specified waveform.

## Example 1

```
HORIZONTAL:FASTFRAME:MULTIPLEFRAMES:NUM-  
FRAMES:CH1 3
```

This command sets the number of frames to 3 on channel 1.

---

## Example 2

HORIZONTAL:FASTFRAME:MULTIPLEFRAMES:NUM-  
FRAMES:CH1?

This query might return:

:HORIZONTAL:FASTFRAME:MULTIPLEFRAMES:NUM-  
FRAMES:CH1 1

This indicates that the number of frames on channel 1 is 3.

---

# HORizontal:FASTframe:REF:FRAMe

## Description

This command sets or queries the FastFrame reference frame number. The Reference Frame number is used to calculate time differences for each frame and displays those differences in the graticule. This command is equivalent to selecting FastFrame Setup in the Horiz/Acq menu and entering a value in the Time Stamps/Frame box.

## Group

Horizontal

## Related Commands

HORizontal:FASTframe:REF:SOUrce, HORizontal:FASTframe:TIMEstamp:DELTA:<wfm>?, HORizontal:FASTframe:TIMEstamp:REF?, HORizontal:FASTframe:TRACk

## Syntax 1

```
HORizontal:FASTframe:REF:FRAMe <NR1>
```

## Syntax 2

```
HORizontal:FASTframe:REF:FRAMe?
```

## Argument

- <NR1>

This specifies the FastFrame reference frame.

## Example 1

```
HORizontal:FASTframe:REF:FRAMe 2
```

This command sets the FastFrame reference frame to 2.

## Example 2

```
HORizontal:FASTframe:REF:FRAMe?
```

This query might return

```
:HORIZONTAL:FASTFRAME:REF:FRAMe 3
```

indicating that the FastFrame reference frame is currently set to 3.

---

# HORizontal:FASTframe:REF:SOURce

## Description

This command sets or queries FastFrame Reference waveform source. This is equivalent to selecting FastFrame Setup from the Horiz/Acq menu and choosing the reference source.

## Group

Horizontal

## Related Commands

HORizontal:FASTframe:REF:FRAME, HORizontal:FASTframe:TIMEstamp:DELTA:<wfm>?, HORizontal:FASTframe:TIMEstamp:REF?, HORizontal:FASTframe:TRACk

## Syntax 1

```
HORizontal:FASTframe:REF:SOURce <wfm>
```

## Syntax 2

```
HORizontal:FASTframe:REF:SOURce?
```

## Argument

- <wfm>

This specifies the FastFrame Reference waveform source. Valid waveforms include CH<x>, MATH<x>, and REF<x>. The x variable can be expressed as an integer ranging from 1 through 4 for 4-channel models and 1 to 2 for 2-channel models.

## Example 1

```
HORizontal:FASTframe:REF:SOURce CH2
```

This command sets the horizontal FastFrame waveform reference source to CH2.

## Example 2

```
HORizontal:FASTframe:REF:SOURce?
```

This query might return

```
:HORIZONTAL:FASTFRAME:REF:SOURCE CH2
```

This indicates that the FastFrame waveform source is currently set to CH2.



---

# HORizontal:FASTframe:SELECTED:<wfm>

## Description

This command sets or returns the FastFrame Selected frame number on the specified waveform. This is equivalent to selecting FastFrame Setup from the Horiz/Acq menu, and then choosing the waveform source and frame. Valid waveforms include CH<x>, MATH<x> and REF<x>. The x variable can be expressed as an integer ranging from 1 through 4 for 4-channel models and 1 to 2 for 2-channel models.

## Group

Horizontal

## Related Commands

HORizontal:FASTframe:TRACk, HORizontal:FASTframe:TIME-stamp:DELTA:<wfm>?, HORizontal:FASTframe:TIMEstamp:REF?

## Syntax 1

```
HORizontal:FASTframe:SELECTED:<wfm> <NR1>
```

## Syntax 2

```
HORizontal:FASTframe:SELECTED?
```

## Argument

- <NR1>

This specifies the selected frame number on the specified waveform.

## Example 1

```
HORIZONTAL:FASTFRAME:SELECTED:CH2 33
```

This command sets the FastFrame Selected frame number on channel 2 to 33.

## Example 2

```
HORIZONTAL:FASTFRAME:SELECTED:CH4?
```

This query might return

```
:HORIZONTAL:FASTFRAME:SELECTED:CH4 25
```

This indicates that the FastFrame frame number on channel 4 is set to 25.

## Example 3

```
HORIZONTAL:FASTFRAME:SELECTED?
```

This query might return

```
:HORIZONTAL:FASTFRAME:SELECTED:CH1 2; CH2 2;  
CH3 2; CH4 2; MATH1 2; MATH2 2; MATH3 2; MATH4  
2; REF1 8; REF2 8; REF3 8; REF4 8
```

This indicates all waveforms' Fastframe selected frame numbers.

---

# HORizontal:FASTframe:STATE

## Description

This command sets or returns the state of FastFrame acquisition. This is equivalent to setting FastFrame to On in the FastFrame Setup menu. FastFrame lets users capture a series of triggered acquisitions with minimal time between them.

The digitizing oscilloscope in FastFrame mode is ready to accept a continuous burst of triggers 400 ms after the controller sends the ACQuire:STATE RUN command.

## Group

Horizontal

## Related Commands

ACQuire:STATE RUN

## Syntax 1

```
HORizontal:FASTframe:STATE {ON|OFF|<NR1>}
```

## Syntax 2

```
HORizontal:FASTframe:STATE?
```

## Arguments

- <ON>

This turns on horizontal FastFrame.

- <OFF>

This turns off horizontal FastFrame.

- <NR1>

A 0 turns off horizontal FastFrame; any other value turns on horizontal FastFrame.

## Example 1

```
HORIZONTAL:FASTFRAME:STATE ON
```

This command turns on horizontal FastFrame.

## Example 2

```
HORIZONTAL:FASTFRAME:STATE?
```

This query might return

```
:HORIZONTAL:FASTFRAME:STATE 0
```

This indicates that the current state of FastFrame is off.

---

# HORizontal:FASTframe:SUMFrame

## Description

This command sets or queries the summary frame mode. When ENvelope is selected, the last frame in a FastFrame acquisition is an envelope of all the prior frames in the acquisition. When AVERAGE is selected, the last frame is replaced with a frame that is the computed average of all the prior frames in the acquisition. For the summary frame control to be active, the number of frames must be two or greater.

## Group

Horizontal

## Syntax 1

```
HORizontal:FASTframe:SUMFrame {NONE|ENVELOPE|AVERAGE}
```

## Syntax 2

```
HORizontal:FASTframe:SUMFrame?
```

## Arguments

- NONE

This turns off the summary mode for FastFrame. This is the default setting.

- ENVELOPE

This argument displays the last frame in a FastFrame acquisition as an envelope of all the prior frames in the acquisition.

- AVERAGE

This argument displays the last frame in a FastFrame acquisition as a frame that is the computed average of all the prior frames in the acquisition.

## Example 1

```
HORizontal:FASTframe:SUMframe NONE
```

This command turns off the summary frame feature.

## Example 2

```
HORizontal:FASTframe:SUMframe?
```

This query might return

```
:HORIZONTAL:FASTFRAME:SUMFRAME AVERAGE
```

This indicates that the summary mode for FastFrame is set to AVERAGE and will display the last frame in the acquisition as the average of all the prior frames in the acquisition.

---

# HORizontal:FASTframe:TIMESTAMP:ALL:<wfm>?

## Description

This query-only command returns the frame number and time stamp for each frame between requested frames, inclusive, within the specified waveform. Argument order is unimportant. Valid waveforms include CH<x>, MATH<x> and REF<x>. The x variable can be expressed as an integer ranging from 1 through 4 for 4-channel models and 1 to 2 for 2-channel models.

The format of the output is as follows:

DD MON YYYY HH:MM:SS.xxx xxx xxx xxx

- DD MON YYYY is day, month, and year
- HH:MM:SS is hours, minutes, and seconds
- .xxx xxx xxx xxx is fractions of a second to picoseconds

## Group

Horizontal

## Syntax

```
HORizontal:FASTframe:TIMESTAMP:ALL:<wfm>?  
<NR1>, <NR1>
```

## Arguments

- <NR1>

This specifies the selected "from" frame number within the specified waveform.

- <NR1>

This specifies the selected "to" frame number within the specified waveform.

## Example

```
HORizontal:FASTframe:TIMESTAMP:ALL:CH2? 4,1
```

This query might return

```
:HORIZONTAL:FASTFRAME:TIMESTAMP:ALL:CH1  
4,1,"02 Mar 2000 20:10:54.542 037 272 620"
```

This is the list of time stamps from frame 1 through frame 4 on channel 1.

---

# HORizontal:FASTframe:TIMEStamp:BETween:<wfm>?

## Description

This query-only command returns the relative trigger for the delta time between the specified frames, within the specified waveform. Valid waveforms include CH<x>, MATH<x> and REF<x>. The x variable can be expressed as an integer ranging from 1 through 4 for 4-channel models and 1 to 2 for 2-channel models.

The format of the delta time output is as follows:

- HH:MM:SS is hours, minutes, and seconds
- .xxx xxx xxx xxx is fractions of a second to picoseconds

## Group

Horizontal

## Syntax

```
HORizontal:FASTframe:TIMEStamp:BETween:<wfm>?
<NR1>, <NR1>
```

## Arguments

- <NR1>

This specifies the selected "from" frame number within the specified waveform.

- <NR1>

This specifies the selected "to" frame number within the specified waveform.

## Example

```
HORIZONTAL:FASTFRAME:TIMESTAMP:BETWEEN:CH1?
5,2
```

This query might return

```
:HORIZONTAL:FASTFRAME:TIMESTAMP:BETWEEN:CH1
5,2,"00:00:00.010 000 540 624"
```

This is the delta of channel 1's Time Stamp frame 5 - Time Stamp frame 2 (TS[5]-TS[2]).

---

# HORizontal:FASTframe:TIMEStamp:DELTA:<wfm>?

## Description

This query-only command returns the relative time between the triggers of the FastFrame Selected and the FastFrame Reference, within the specified waveform. Valid waveforms include CH<x>, MATH<x>, and REF<x>. The x variable can be expressed as an integer ranging from 1 through 4 for 4-channel models and 1 to 2 for 2-channel models.

The format of the output is as follows:

HH:MM:SS.xxx xxx xxx xxx

- HH:MM:SS is hours, minutes, and seconds
- .xxx xxx xxx xxx is fractions of a second to picoseconds

## Group

Horizontal

## Syntax

HORizontal:FASTframe:TIMEStamp:DELTA:<wfm>?

## Example

HORIZONTAL:FASTRAME:TIMESTAMP:DELTA:CH1?

This query might return

```
:HORIZONTAL:FASTFRAME:TIMESTAMP:DELTA:CH1  
"00:00:00.006 000 306 556"
```

This is the delta time stamp of FastFrame Selected Frame on channel 1 - FastFrame Reference Frame (TS[C1 Selected]-TS[Reference]).

---

# HORizontal:FASTframe:TIMEStamp:FRAME:<wfm>?

## Description

This query-only command returns the absolute trigger date and time for the specified frame and waveform. Valid waveforms include CH<x>, MATH<x> and REF<x>. The x variable can be expressed as an integer ranging from 1 through 4 for 4-channel models and 1 to 2 for 2-channel models.

The format of the output is as follows:

DD MON YYYY HH:MM:SS.xxx xxx xxx xxx

- DD MON YYYY is day, month, and year
- HH:MM:SS is hours, minutes, and seconds
- .xxx xxx xxx xxx is fractions of a second to picoseconds

## Group

Horizontal

## Syntax

```
HORizontal:FASTframe:TIMEStamp:FRAME:<wfm>?
<NR1>
```

## Argument

- <NR1>

The frame number for which the timestamp will be returned on the specified waveform.

## Example

```
HORizontal:FASTframe:TIMEStamp:FRAME:CH1? 10
```

This query might return

```
:HORIZONTAL:FASTFRAME:TIMESTAMP:FRAME:CH1
10,"02 Mar 2000 20:10:54.536 036 928 432"
```

which is the time stamp of channel 1, frame 10.

---

# HORizontal:FASTframe:TIMESTAMP:REF?

## Description

This query-only command returns the absolute trigger date and time for FastFrame reference.

The format of the output is as follows:

DD MON YYYY HH:MM:SS.xxx xxx xxx xxx

- DD MON YYYY is day, month, and year
- HH:MM:SS is hours, minutes, and seconds
- .xxx xxx xxx xxx is fractions of a second to picoseconds

## Group

Horizontal

## Related Commands

HORizontal:FASTframe:REF:FRAME, HORizontal:FASTframe:REF:SOURce

## Syntax

HORizontal:FASTframe:TIMESTAMP:REF?

## Example

HORIZONTAL:FASTFRAME:TIMESTAMP:REF?

This query might return

```
:HORIZONTAL:FASTFRAME:TIMESTAMP:REF "02 Mar  
2000 20:10:54.533 036 838 784"
```

This is the time stamp of FastFrame Reference.



---

# HORizontal:FASTframe:TIMESamp: SELECTED:<wfm>?

## Description

This query-only command returns the absolute trigger date and time for the FastFrame Selected, within the specified waveform. Valid waveforms include CH<x>, MATH<x> and REF<x>. The x variable can be expressed as an integer ranging from 1 through 4 for 4-channel models and 1 to 2 for 2-channel models.

The format of the output is as follows:

DD MON YYYY HH:MM:SS.xxx xxx xxx xxx

- DD MON YYYY is day, month, and year
- HH:MM:SS is hours, minutes, and seconds
- .xxx xxx xxx xxx is fractions of a second to picoseconds

## Group

Horizontal

## Related Commands

HORizontal:FASTframe:SELECTED:<wfm>?

## Syntax

HORizontal:FASTframe:TIMESamp:SELECTED:<wfm>?

## Example

HORIZONTAL:FASTFRAME:TIMESTAMP:SELECTED:REF1?

This query might return

```
:HORIZONTAL:FASTFRAME:TIMESTAMP:SELECTED:REF1  
"02 MAR 2000 20:10:54:539 037 145 340"
```

This is the time stamp of FastFrame Selected on reference 1.

---

# HORizontal:FASTframe:TRACk

## Description

This command sets up or returns the state of FastFrame tracking feature. This command is equivalent to selecting FastFrame Setup from the Horiz/Acq menu and then clicking the desired Frame Tracking state.

When FastFrame Track is set to "live", the channel and math waveforms are locked together. Adjusting a channel waveform also adjusts a related math waveform. All reference waveforms are also locked together but they are separate from channel and math waveforms.

For example, when you set the Selected Frame Source Ch1 to Frame 3, then Selected Frame Ch2, Ch3, Ch4, Math1, Math2, Math3 and Math4 are also set to Frame 3.

When you set the Selected Frame Source Ref1 to Frame 2, then Selected Frame Ref2, Ref3 and Ref4 are also set to Frame 2. If the Frame Tracking is set to Live, changing Selected Frame Ch1 will not affect the Selected Frame Ref1 frame of the Reference Frame setting.

When FastFrame Track is set to "all", the channel, math and reference waveforms are locked together. Adjusting a channel waveform also adjusts the related math and reference waveforms.

For example, when you set the Selected Frame Source Ch1 to Frame 3, then Selected Frame Ch2, Ch3, Ch4, Math1, Math2, Math3, Math4, Ref1, Ref2, Ref3 and Ref4 are also set to Frame 3.

## Group

Horizontal

## Related Commands

HORizontal:FASTframe:REF:FRAMe, HORizontal:FASTframe:REF:SOURce, HORizontal:FASTframe:SELECTED:<wfm>?

## Syntax 1

```
HORizontal:FASTframe:TRACk {LIVE|ALL}
```

## Syntax 2

```
HORizontal:FASTframe:TRACk?
```

## Arguments

- LIVE

This sets FastFrame Track to Live.

- ALL

This sets FastFrame Track to All.

---

### **Example 1**

`HORIZONTAL:FASTFRAME:TRACK LIVE`

This command sets FastFrame Track to live.

### **Example 2**

`HORIZONTAL:FASTFRAME:TRACK?`

This query might return

`:HORIZONTAL:FASTFRAME:TRACK ALL`

This indicates that all FastFrame Selected Frames and the FastFrame Reference will be locked together.

---

# HORizontal:FASTframe:XZEro:ALL:CH<x>?

## Description

This query-only command returns the frame number and xzero for each frame between requested frames, inclusive, within the specified waveform. Argument order is unimportant. The channel is specified by x. The value of x can range from 1 through 4 for four-channel instruments or 1 through 2 for two-channel instruments.

Xzero is the subsample time between the trigger sample (designated by WFMOutputpre:PT\_Off?) and the occurrence of the actual trigger for the waveform specified.

**Note:** This query uses an asynchronous method to obtain and write its output data, which may result in an arbitrarily large volume. Until all output data has been generated and read by your program, or a device clear message is sent, the instrument will not respond to its user controls.

It is recommended that you follow this query with a query of the event status register (\*ESR?) to determine if the original query completed successfully.

If the timeout is set to less than the time required to generate and transfer all of the output data, the \*ESR? query will indicate that the error bit is set. A subsequent query of the event queue (EVMsg? or ALLEV?) will indicate that the query was interrupted by \*ESR?. In such cases, increase the timeout value and send the query again.

## Group

Horizontal

## Related Commands

WFMOutputpre:XZEro?, WFMOutputpre:PT\_Off?,  
HORizontal:FASTframe:XZEro:ALL:REF<x>?,  
HORizontal:FASTframe:XZEro:FRame:CH<x>?,  
HORizontal:FASTframe:XZEro:FRame:REF<x>?,  
HORizontal:FASTframe:XZEro:REF?,  
HORizontal:FASTframe:XZEro:SELECTED:CH<x>?,  
HORizontal:FASTframe:XZEro:SELECTED:REF<x>?

## Syntax

```
HORizontal:FASTframe:XZEro:ALL:CH<x>? <NR1>,  
<NR1>
```

## Arguments

- <NR1>

This specifies the selected “from” frame number within the specified waveform.

- <NR1>

This specifies the selected “to” frame number within the specified waveform.

---

## Example

```
HORizontal:FASTframe:XZEro:ALL:CH1? 1,10
```

This query might return:

```
:HORIZONTAL:FASTFRAME:XZERO:ALL:CH1 1,10,"1:  
3.3352E-09","2: 1.4647E-09","3:  
2.4770E-09","4: 1.4395E-09","5:  
1.0857E-09","6: 1.0410E-09","7:  
1.4172E-09","8: 1.4437E-09","9:  
1.7050E-09","10: 2.9050E-10"\n'
```

This lists xzeros from frame 1 through frame 10 for the channel 1 waveform.

---

# HORizontal:FASTframe:XZEro:ALL:REF<x>?

## Description

This query-only command returns the frame number and xzero for each frame between requested frames, inclusive, within the specified waveform. Argument order is unimportant. The reference waveform is specified by x. The value of x can range from 1 through 4 for four-channel instruments or 1 through 2 for two-channel instruments.

Xzero is the subsample time between the trigger sample (designated by WFMOutputpre:PT\_Off?) and the occurrence of the actual trigger for the waveform specified.

**Note:** This query uses an asynchronous method to obtain and write its output data, which may result in an arbitrarily large volume. Until all output data has been generated and read by your program, or a device clear message is sent, the instrument will not respond to its user controls.

It is recommended that you follow this query with a query of the event status register (\*ESR?) to determine if the original query completed successfully.

If the timeout is set to less than the time required to generate and transfer all of the output data, the \*ESR? query will indicate that the error bit is set. A subsequent query of the event queue (EVMsg? or ALLEV?) will indicate that the query was interrupted by \*ESR?. In such cases, increase the timeout value and send the query again.

## Group

Horizontal

## Related Commands

WFMOutputpre:XZEro? WFMOutputpre:PT\_Off?,  
HORizontal:FASTframe:XZEro:ALL:CH<x>?,  
HORizontal:FASTframe:XZEro:FRame:CH<x>?,  
HORizontal:FASTframe:XZEro:FRame:REF<x>?,  
HORizontal:FASTframe:XZEro:REF?,  
HORizontal:FASTframe:XZEro:SELECTED:CH<x>?,  
HORizontal:FASTframe:XZEro:SELECTED:REF<x>?

## Syntax

HORizontal:FASTframe:XZEro:ALL:REF<x>? <NR1>,  
<NR1>

## Arguments

- <NR1>

This specifies the selected “from” frame number within the specified waveform.

- <NR1>

This specifies the selected “to” frame number within the specified waveform.

---

## Example

```
HORizontal:FASTframe:XZEro:ALL:REF1? 1,10
```

This query might return:

```
:HORIZONTAL:FASTFRAME:XZERO:ALL:REF1 1,10,"1:  
3.3352E-09","2: 1.4647E-09","3:  
2.4770E-09","4: 1.4395E-09","5:  
1.0857E-09","6: 1.0410E-09","7:  
1.4172E-09","8: 1.4437E-09","9:  
1.7050E-09","10: 2.9050E-10"\n'
```

This lists xzeros from frame 1 through frame 10 for the reference 1 waveform.

---

# HORizontal:FASTframe:XZEro:FRame:CH<x>?

## Description

This query-only command returns xzero for the specified frame. The channel is specified by x. The value of x can range from 1 through 4 for four-channel instruments or 1 through 2 for two-channel instruments.

Xzero is the subsample time between the trigger sample (designated by WFMOutputpre:PT\_Off?) and the occurrence of the actual trigger for the waveform specified.

## Group

Horizontal

## Related Commands

WFMOutputpre:XZEro?, WFMOutputpre:PT\_Off?,  
HORizontal:FASTframe:XZEro:ALL:CH<x>?,  
HORizontal:FASTframe:XZEro:ALL:REF<x>?,  
HORizontal:FASTframe:XZEro:FRame:REF<x>?,  
HORizontal:FASTframe:XZEro:REF?,  
HORizontal:FASTframe:XZEro:SELECTED:CH<x>?,  
HORizontal:FASTframe:XZEro:SELECTED:REF<x>?

## Syntax

```
HORizontal:FASTframe:XZEro:FRame:CH<x>? <NR1>
```

## Argument

<NR1>

The frame number for which the xzero will be returned on the specified waveform.

## Example

```
HORizontal:FASTframe:XZEro:FRame:CH<1>? 1
```

This query might return:

```
:HORIZONTAL:FASTFRAME:XZERO:FRAME:CH1  
1,3.3352E-9\n'
```

This is xzero for the channel 1 waveform, frame 1.



---

# HORizontal:FASTframe:XZEro:FRame:REF<x>?

## Description

This query-only command returns xzero for the specified frame. The reference waveform is specified by x. The value of x can range from 1 through 4 for four-channel instruments or 1 through 2 for two-channel instruments.

Xzero is the subsample time between the trigger sample (designated by WFMOupre:PT\_Off?) and the occurrence of the actual trigger for the waveform specified.

## Group

Horizontal

## Related Commands

WFMOupre:XZEro?, WFMOupre:PT\_Off?,  
HORizontal:FASTframe:XZEro:ALL:CH<x>?,  
HORizontal:FASTframe:XZEro:ALL:REF<x>?,  
HORizontal:FASTframe:XZEro:FRame:CH<x>?,  
HORizontal:FASTframe:XZEro:REF?,  
HORizontal:FASTframe:XZEro:SELECTED:CH<x>?,  
HORizontal:FASTframe:XZEro:SELECTED:REF<x>?

## Syntax

```
HORizontal:FASTframe:XZEro:FRame:REF<x>? <NR1>
```

## Argument

<NR1>

The frame number for which the xzero will be returned on the specified waveform.

## Example

```
HORizontal:FASTframe:XZEro:FRame:REF1? 2
```

This query might return:

```
:HORIZONTAL:FASTFRAME:XZERO:FRAME:REF1  
2,1.4647E-9\n'
```

This is xzero for the reference 1 waveform, frame 2.

---

# HORizontal:FASTframe:XZEro:REF?

## Description

This query-only command returns xzero for the FastFrame reference waveform's frame.

Xzero is the subsample time between the trigger sample (designated by WFMOutputpre:PT\_Off?) and the occurrence of the actual trigger for the waveform specified.

## Group

Horizontal

## Related Commands

WFMOutputpre:XZEro?, WFMOutputpre:PT\_Off?,  
HORizontal:FASTframe:XZEro:ALL:CH<x>?,  
HORizontal:FASTframe:XZEro:ALL:REF<x>?,  
HORizontal:FASTframe:XZEro:FRame:CH<x>?,  
HORizontal:FASTframe:XZEro:FRame:REF<x>?,  
HORizontal:FASTframe:XZEro:SELECTED:CH<x>?,  
HORizontal:FASTframe:XZEro:SELECTED:REF<x>?

## Syntax

```
HORizontal:FASTframe:XZEro:REF?
```

## Example

```
HORizontal:FASTframe:XZEro:REF?
```

This query might return:

```
:HORIZONTAL:FASTFRAME:XZERO:REF 3.3352E-9\n'
```

This is xzero for the current fastFrame reference waveform.

---

# HORizontal:FASTframe:XZEro:SELECTED:CH<x>?

## Description

This query-only command returns xzero for FastFrame Selected, within the specified waveform. The channel is specified by x. The value of x can range from 1 through 4 for four-channel instruments or 1 through 2 for two-channel instruments.

Xzero is the subsample time between the trigger sample (designated by WFMOupre:PT\_Off?) and the occurrence of the actual trigger for the waveform specified.

## Group

Horizontal

## Related Commands

WFMOupre:XZEro?, WFMOupre:PT\_Off?,  
HORizontal:FASTframe:XZEro:ALL:CH<x>?,  
HORizontal:FASTframe:XZEro:ALL:REF<x>?,  
HORizontal:FASTframe:XZEro:FRame:CH<x>?,  
HORizontal:FASTframe:XZEro:FRame:REF<x>?,  
HORizontal:FASTframe:XZEro:REF?,  
HORizontal:FASTframe:XZEro:SELECTED:REF<x>?

## Syntax

```
HORizontal:FASTframe:XZEro:SELECTED:CH<x>?
```

## Example

```
HORizontal:FASTframe:XZEro:SELECTED:CH1?
```

This query might return:

```
:HORIZONTAL:FASTFRAME:XZERO:SELECTED:CH1  
1.4647E-9\n'
```

This is the xzero of FastFrame Selected for the channel 1 waveform.

---

# HORizontal:FASTframe:XZEro:SELECTED:REF<x>?

## Description

This query-only command returns xzero for FastFrame Selected within the specified waveform. The reference waveform is specified by x. The value of x can range from 1 through 4 for four-channel instruments or 1 through 2 for two-channel instruments.

Xzero is the subsample time between the trigger sample (designated by WFMOupre:PT\_Off?) and the occurrence of the actual trigger for the waveform specified.

## Group

Horizontal

## Related Commands

WFMOupre:XZEro?, WFMOupre:PT\_Off?,  
HORizontal:FASTframe:XZEro:ALL:CH<x>?,  
HORizontal:FASTframe:XZEro:ALL:REF<x>?,  
HORizontal:FASTframe:XZEro:FRame:CH<x>?,  
HORizontal:FASTframe:XZEro:FRame:REF<x>?,  
HORizontal:FASTframe:XZEro:REF?,  
HORizontal:FASTframe:XZEro:SELECTED:CH<x>?

## Syntax

```
HORizontal:FASTframe:XZEro:SELECTED:REF<x>?
```

## Example

```
HORizontal:FASTframe:XZEro:SELECTED:REF1?
```

This query might return:

```
:HORIZONTAL:FASTFRAME:XZERO:SELECTED:REF1  
1.4647E-9\n'
```

This is the xzero of FastFrame Selected for the reference 1 waveform.

---

# HORizontal:FASTframe?

## Description

This query-only command returns all of the settings for the horizontal FastFrame commands.

## Group

Horizontal

## Syntax

HORizontal:FASTframe?

## Example

HORizontal:FASTframe?

This query might return

```
:HORIZONTAL:FASTFRAME:STATE 1;READOUTS 0;SUM-  
FRAME NONE;MULTIPLEFRAMES:MODE OVERLAY;FRAMES-  
TART:CH1 1;CH2 1;CH3 1;CH4 1;MATH1 1;MATH2  
1;MATH3 1;MATH4 1;REF1 1;REF2 1;REF3 1;REF4  
1;:HORIZONTAL:FASTFRAME:MULTIPLEFRAMES:NUM-  
FRAMES:CH1 2;CH2 2;CH3 2;CH4 2;MATH1 2;MATH2  
2;MATH3 2;MATH4 2;REF1 2;REF2 2;REF3 2;REF4  
2;:HORIZONTAL:FASTFRAME:LENGTH 500;COUNT 2;SE-  
LECTED:CH1 2;CH2 2;CH3 2;CH4 2;MATH1 2;MATH2  
2;MATH3 2;MATH4 2;REF1 2;REF2 2;REF3 2;REF4  
2;:HORIZONTAL:FASTFRAME:REF:SOURCE CH4;FRAME  
1;:HORIZONTAL:FASTFRAME:TRACK ALL
```

---

# HORizontal[:MAIn]:DELay:MODE

## Description

This command sets or queries the time base trigger delay mode. This command is equivalent to choosing Delay Mode On from the Horiz/Acq menu.

## Group

Horizontal

## Related Commands

HORizontal[:MAIn]:DELay:TIME

## Syntax 1

HORizontal[:MAIn]:DELay:MODE [ON|OFF|<NR1>]

## Syntax 2

HORizontal[:MAIn]:DELay:MODE?

## Arguments

- ON

This enables the time base trigger delay mode.

- OFF

This disables the time base trigger delay mode.

- <NR1>

A 0 disables the time base trigger delay mode, any other value enables the time base trigger delay mode.

## Example 1

```
HORizontal:DELay:MODE ON
```

This command enables the time base trigger delay mode.

## Example 2

```
HORizontal:DELay:MODE?
```

This query might return

```
:HORIZONTAL:DELAY:MODE 1
```

This indicates that the time base trigger delay mode is currently enabled.

---

# HORizontal[:MAIn]:DELay:POSition

## Description

This command sets or queries the time base position when Horizontal Delay Mode is turned on. This command is equivalent to selecting Horizontal/Acquisition Setup from the Horiz/Acq menu and then entering a Ref Point value.

## Group

Horizontal

## Related Commands

HORizontal[:MAIn}:DELay:TIME

## Syntax 1

HORizontal[:MAIn]:DELay:POSition <NR3>

## Syntax 2

HORizontal[:MAIn]:DELay:POSition?

## Arguments

- <NR3>

This is from 0 to  $\approx 100$  and is the percentage of the waveform that is displayed left of the center graticule.

**Note:** The upper limit of the waveform position is slightly limited by a value that is determined from the record length (upper limit =  $100 - 1/\text{record length}$ ).

## Example 1

```
HORizontal:DELay:POSition 5E+1
```

This command sets the time base trigger delay position to 50 percent.

## Example 2

```
HORizontal:DELay:POSition?
```

This query might return

```
:HORIZONTAL:MAIN:DELAY:POSITION 5.0000E+01
```

This indicates that the time base trigger delay position is currently set to 50 percent.

---

# HORizontal[:MAIn]:DELay:TIME

## Description

This command sets or queries the time base trigger delay time. This command is equivalent to selecting Position/Scale from the Horiz/Acq menu and choosing a value for Horiz Delay.

## Group

Horizontal

## Related Commands

HORizontal[:MAIn]:DELay:MODE, HORizontal[:MAIn]:DELay:POSition

## Syntax 1

```
HORizontal[:MAIn]:DELay:TIME <NR3>
```

## Syntax 2

```
HORizontal[:MAIn]:DELay:TIME?
```

## Argument

- <NR3>

This specifies the time base trigger delay time setting, typically represented in seconds.

## Example 1

```
HORizontal:DELay:TIME 5.0E-3
```

This command sets the time base trigger delay time to 5 ms.

## Example 2

```
HORizontal:DELay:TIME?
```

This query might return

```
:HORIZONTAL:MAIN:DELAY 5.0000E-05
```

This indicates that the time delay setting is 5  $\mu$ s.



---

# HORizontal:MAIn:INTERPRatio?

## Description

This query-only command returns the Horizontal Main time base interpolation ratio. The ratio is useful for calculating the maximum sampling rate when users request sampling rates that are faster than the maximum rate specified by the acquisition hardware.

The query HORizontal:MAIn:SAMPLERate? returns the sampling rate (real and interpolated). The real maximum sampling rate, as specified by the acquisition hardware, is:

Max sampling rate =

`HORizontal:MAIn:SAMPLERate?/HORizontal:MAIn:INTERPRatio?`

The default value is 1.

An interpolation ratio other than 1 means that displayed and retrieved waveforms contain additional points that are derived through interpolation. At such times, the retrieved waveform data is actually 16-bit data. You should set the data width accordingly.

## Group

Horizontal

## Syntax

`HORizontal:MAIn:INTERPRatio?`

## Example

`HORIZONTAL:MAIN:INTERPRATIO?`

This query might return

`:HORIZONTAL:MAIN:INTERPRATIO 10.0`

---

# HORizontal[:MAIn]:POSition

This command either positions the waveform horizontally on the display or returns the horizontal position on the display and is identical to the HORizontal:POSition command. When Horizontal Delay mode is turned off, this command is equivalent to adjusting the HORIZONTAL POSITION knob on the front-panel. When Horizontal Delay mode is turned on, this command is equivalent to selecting Horizontal/Acquisition from the Horiz/Acq menu and then choosing a Horizontal Ref Point value.

## Group

Horizontal

## Related Commands

HORizontal[:MAIn]:DELay:MODE

## Syntax 1

```
HORizontal:MAIn:POSition <NR3>
```

## Syntax 2

```
HORizontal:MAIn:POSition?
```

## Arguments

- <NR3>

This argument can range from 0 to  $\approx 100$  and is the percentage of the waveform that is displayed left of the center graticule.

**Note:** The upper limit of the waveform position is slightly limited by a value that is determined from the record length (upper limit =  $100 - 1/\text{record length}$ ).

## Example 1

```
HORizontal:MAIn:POSition 5.000E+01
```

This command sets the horizontal position of the waveform such that 10 percent of the waveform is to the left of center of the screen.

## Example 2

```
HORizontal:MAIn:POSition?
```

This query might return

```
:HORIZONTAL:MAIN:POSITION 5.0000E+01
```

This indicates that the horizontal position of the waveform on the screen is currently set to 50 percent.

---

# HORizontal:MAIn:SAMPLERate

## Description

This command sets the horizontal sample rate to the desired number of samples per second. The record length is automatically adjusted at the same time to maintain a constant number of data points in each frame. The query form of this command returns the current horizontal sample rate. This command is equivalent to selecting Horizontal/Acquisition Setup from the Horiz/Acq menu and then entering the desired Sample Rate.

## Group

Horizontal

## Related Commands

HORizontal:RESOlution

## Syntax 1

```
HORizontal:MAIn:SAMPLERate <NR3>
```

## Syntax 2

```
HORizontal:MAIn:SAMPLERate?
```

## Arguments

- <NR3>

This represents the size of the sample rate, which ranges from 5 S/s to 250 GS/s.

## Example 1

```
HORizontal:MAIn:SAMPLERate 125E6
```

This command sets the sample rate to the sample rate increment that is closest to 125 MS/s.

## Example 2

```
HORizontal:MAIn:SAMPLERate?
```

This query might return

```
:HORIZONTAL:MAIN:SAMPLERATE 2.5000E+09
```

This indicates that the sample rate is currently set to 2.5 GS/s.

---

# HORizontal[:MAIn]:SCAle

## Description

This command sets the time per division for the time base or returns its horizontal scale on the display and is identical to the HORizontal:SCAle command. The specified scale value is rounded to a valid scale setting. This command is equivalent to selecting Position/Scale from the Horiz/Acq menu and then choosing a Scale value.

## Group

Horizontal

## Related Commands

HORizontal:SCAle

## Syntax 1

```
HORizontal:MAIn:SCAle <NR3>
```

## Syntax 2

```
HORizontal:MAIn:SCAle?
```

## Argument

- <NR3>

This is the time per division. The range is from 200 ps through 40 s.

## Example 1

```
HORizontal:MAIn:SCAle 2E-6
```

This command sets the main scale to 2 $\mu$ s per division.

## Example 2

```
HORizontal:MAIn:SCAle?
```

This query might return:

```
:HORIZONTAL:MAIN:SCALE 2.0000E-06
```

This indicates that the main scale is currently set to 2  $\mu$ s per division.

---

# HORizontal:MAIn:UNIts

## Description

This command sets or returns the units for the horizontal time base. It is equivalent to setting the HORizontal:MAIn:UNIts:STRing

## Group

Horizontal

## Related Commands

HORizontal:MAIn:UNIts:STRing

## Syntax 1

```
HORizontal:MAIn:UNIts <STRing>
```

## Syntax 2

```
HORizontal:MAIn:UNIts?
```

## Arguments

- <STRing>

This is the time base units string.

## Example 1

```
HORizontal:MAIn:UNIts "lum"
```

This command sets the time base multiplier to 5.5, the units to lumens and the time base units offset to 2.9.

## Example 2

```
HORizontal:MAIn:UNIts?
```

This query might return:

```
:HORIZONTAL:MAIN:UNITS STRING "lum".
```

---

# HORizontal:MAIn:UNIts:STRing

## Description

This command sets or returns the units string for the horizontal time base trigger delay.

## Group

Horizontal

## Related Commands

HORizontal:MAIn:UNIts

## Syntax 1

```
HORizontal:MAIn:UNIts:STRing <string>
```

## Syntax 2

```
HORizontal:MAIn:UNIts:STRing?
```

## Argument

- <string>

This is the time base units string.

## Example 1

```
HORizontal:MAIn:UNIts:STRing lum
```

This command sets the units string for the time base trigger delay to lumens.

## Example 2

```
HORizontal:MAIn:UNIts:STRing?
```

This query might return

```
:HORIZONTAL:MAIN:UNITS:STRING "lum"
```

This indicates that the units string for the time base trigger delay is set to lumens.

---

# HORizontal:MAIn?

## Description

This query-only command returns the time per division of the time base. This command is equivalent to selecting Position/Scale from the Horiz/Acq menu.

## Group

Horizontal

## Related Commands

HORizontal:SCAlE

## Syntax

HORizontal:MAIn?

## Example

HORizontal:MAIn?

This query might return:

```
:HORIZONTAL:MAIN:INTERPRATIO 1.0000;SCALE  
40.0000E-9;POSITION 50.0000;SAMPLERATE  
1.2500E+9;UNITS:STRING "s"
```

---

# HORizontal:POSition

## Description

This command either positions the waveform horizontally on the display or returns its horizontal position on the display. When Horizontal Delay Mode is turned off, this command is equivalent to adjusting the HORIZONTAL POSITION knob on the front panel.

## Group

Horizontal

## Related Commands

HORizontal[:MAIn]:DELay:MODE

## Syntax 1

HORizontal:POSition <NR3>

## Syntax 2

HORizontal:POSition?

## Argument

- <NR3>

This is from 0 to  $\approx 100$  and is the percentage of the waveform that is displayed left of the center graticule.

**Note:** The upper limit of the waveform position is slightly limited by a value that is determined from the record length (upper limit =  $100 - 1/\text{record length}$ ).

## Example 1

```
HORizontal:POSition 10
```

This command sets the horizontal position of the waveform such that 10% of the waveform is to the left of screen center.

## Example 2

```
HORizontal:POSition?
```

This query might return:

```
:HORIZONTAL:POSITION 5.0000E+01
```

This indicates the horizontal position of the waveform on the screen.



---

# HORizontal:RECOrdlength

## Description

This command sets the horizontal record length to the number of data points in each frame. The query form of this command returns the current horizontal record length. This command is equivalent to selecting Resolution from the Horiz/Acq menu and then entering the desired Record Length.

## Group

Horizontal

## Syntax 1

```
HORizontal:RECOrdlength <NR1>
```

## Syntax 2

```
HORizontal:RECOrdlength?
```

## Argument

- <NR1>

This represents the supported values for horizontal record lengths, which range from 500 through 400K data points. For more information about valid data point ranges, select Specifications from the Help menu and choose the Horizontal & Acquisition tab.

## Example 1

```
HORizontal:RECOrdlength 50000
```

This command specifies that 50000 data points will be acquired for each record.

## Example 2

```
HORizontal:RECOrdlength?
```

This query might return:

```
:HORIZONTAL:RECOrdlength 5000
```

This indicates that the horizontal record length is equal to 5000 data points.

---

# HORizontal:RESOLution

## Description

This command sets the horizontal record length to the number of data points in each frame. The sample rate is automatically adjusted at the same time to maintain a constant time per division. The query form of this command returns the current horizontal record length. This command is equivalent to adjusting the RESOLUTION knob on the front panel.

## Group

Horizontal

## Syntax 1

```
HORizontal:RESOLution <NR1>
```

## Syntax 2

```
HORizontal:RESOLution?
```

## Argument

- <NR1>

This represents the supported values for horizontal record lengths, which range from 500 through 400K data points. For more information about valid data point ranges, select Specifications from the Help menu and choose the Horizontal & Acquisition tab.

## Example 1

```
HORizontal:RESOLution 50000
```

This command specifies that 50000 data points will be acquired for each record.

## Example 2

```
HORizontal:RESOLution?
```

This query might return

```
:HORIZONTAL:RESOLUTION 5000
```

This indicates that the horizontal record length is equal to 5000 data points.

---

# HORizontal:ROLL

## Description

This command sets or queries the Roll Mode status. Use Roll Mode when you want to view data at very slow sweep speeds. It is useful for observing data samples on the screen as they occur. This command is equivalent to selecting Horizontal/Acquisition Setup from the Horiz/Acq menu, selecting the Acquisition tab, and setting the Roll Mode to Auto or Off.

## Group

Horizontal

## Syntax 1

```
HORizontal:ROLL {AUTO|OFF|ON}
```

## Syntax 2

```
HORizontal:ROLL?
```

## Arguments

- AUTO

This enables Roll Mode, if the time/division is set appropriately.

- OFF

This disables Roll Mode.

- ON

This enables Roll Mode, if the time/division is set appropriately.

## Example 1

```
HORizontal:ROLL ON
```

This command enables Roll Mode.

## Example 2

```
HORizontal:RECOrdlength?
```

This query might return:

```
:HORIZONTAL:ROLL OFF
```

This indicates that the Roll Mode is disabled.

---

# HORizontal:SCAlE

## Description

This command sets or queries the time per division for the time base and is identical to the HORizontal:MAIn:SCAlE command. It is provided to maintain program compatibility with some older models of Tektronix instruments. This command is equivalent to selecting Position/Scale from the Horiz/Acq menu and entering the desired Scale value.

## Group

Horizontal

## Related Commands

HORizontal:MAIn:SCAlE

## Syntax 1

```
HORizontal:SCAlE <NR3>
```

## Syntax 2

```
HORizontal:SCAlE?
```

## Argument

- <NR3>

This is the time per division. The range is from 200 ps to 40 s.

## Example 1

```
HORizontal:SCAlE 2E-6
```

This command sets the main scale to 2  $\mu$ s per division.

## Example 2

```
HORizontal:SCAlE?
```

This query might return

```
:HORIZONTAL:SCALE 2.0000E-06
```

This indicates that the time per division is currently set to 2  $\mu$ s.

---

# HORizontal:TRIGger:POSition

## Description

This command sets or queries the position of the trigger and is identical to the HORizontal:MAIn:POSition command. It is provided to maintain program compatibility with some older models of Tektronix instruments. This command is equivalent to selecting Position/Scale from the Horiz/Acq menu and entering the desired Position.

## Group

Horizontal

## Related Commands

HORizontal:MAIn:POSition

## Syntax 1

```
HORizontal:TRIGger:POSition <NR1>
```

## Syntax 2

```
HORizontal:TRIGger:POSition?
```

## Argument

- <NR1>

This is the amount of pretrigger information in the waveform. The range is from 0 to 100%.

## Example 1

```
HORizontal:TRIGger:POSition 50
```

This command sets the pretrigger information in the waveform to 50%.

## Example 2

```
HORizontal:TRIGger:POSition?
```

This query might return:

```
:HORIZONTAL:TRIGGER:POSITION 50
```

This indicates that 50% of the waveform is pretrigger information.

---

# HORizontal?

## Description

This query-only command returns all settings for the horizontal commands.

## Group

HORizontal

## Syntax

HORizontal?

## Example

HORizontal?

This query might return the following horizontal settings

```
:HORIZONTAL:DELAY:MODE 0;POSITION 50.0000;TIME
0.000000000000;:HORIZONTAL:MAIN:INTERPRATIO
1.0000;SCALE 40.0000E-9;POSITION 50.0000;SAM-
PLERATE 1.2500E+9;UNITS:STRING "s";:HORIZON-
TAL:RECORDLENGTH 500;RESOLUTION 500;ROLL
AUTO;FASTFRAME:STATE 1;READOUTS 0;SUMFRAME
NONE;MULTIPLEFRAMES:MODE OVERLAY;FRAMES-
TART:CH1 1;CH2 1;CH3 1;CH4 1;MATH1 1;MATH2
1;MATH3 1;MATH4 1;REF1 1;REF2 1;REF3 1;REF4
1;:HORIZONTAL:FASTFRAME:MULTIPLEFRAMES:NUM-
FRAMES:CH1 2;CH2 2;CH3 2;CH4 2;MATH1 2;MATH2
2;MATH3 2;MATH4 2;REF1 2;REF2 2;REF3 2;REF4
2;:HORIZONTAL:FASTFRAME:LENGTH 500;COUNT 2;SE-
LECTED:CH1 2;CH2 2;CH3 2;CH4 2;MATH1 2;MATH2
2;MATH3 2;MATH4 2;REF1 2;REF2 2;REF3 2;REF4
2;:HORIZONTAL:FASTFRAME:REF:SOURCE CH4;FRAME
1;:HORIZONTAL:FASTFRAME:TRACK ALL
```

---

# ID?

## Description

This query-only command returns identifying information about the instrument and related firmware.

## Group

Miscellaneous

## Related Commands

\*IDN?

## Syntax

ID?

## Example

ID?

This query might return

```
:TEK/TDS5404,CF:91.1CT,FV:01.00.912
```

This indicates the instrument model number, configured format, and firmware version number.

---

# LIMit:Beep

## Description

This command causes the instrument to beep when the waveform data exceeds the limits set in the limit test (LIMit:STATE must be on).

## Group

Limit Test

## Related Commands

LIMit:STATE

## Syntax 1

```
LIMit:BEEP {OFF|ON|<NR1>}
```

## Syntax 2

```
LIMit:BEEP?
```

## Arguments

- OFF

This argument disables the beep.

- ON

This argument enables the beep.

- <NR1>

A 0 disables the beep, any other value enables the beep.

## Example 1

```
LIMit:BEEP ON
```

This command causes a beep when any waveform data exceeds the limits specified in the limit test.

## Example 2

```
LIMit:BEEP?
```

This query might return:

```
:LIMIT:BEEP 0
```

This indicates that the instrument will not beep when waveform data exceeds the limits specified in the limit test.



---

# LIMit:COMpare

## Description

This command resets the Limit Test comparison template.

## Group

Limit Test

## Related Commands

## Syntax

```
LIMit:COMpare {RESet}
```

## Arguments

- RESet

Resets the Limit template to "None."

## Example

```
LIMit:COMpare RESet
```

This command sets the Limit template to None.

---

# LIMit:COMpare:CH<x>

## Description

This command sets or queries the template against which to compare the waveform acquired from the channel specified by x. The template can be a waveform saved in any of the reference locations (REF1 through REF4 for four-channel instruments or REF1 through REF2 for two-channel instruments) or none.

## Group

Limit Test

## Related Commands

### Syntax 1

```
LIMit:COMpare:CH<x> {NONE|REF<x>}
```

### Syntax 2

```
LIMit:COMpare:CH<x>?
```

## Arguments

- NONE

This argument turns off template testing for the channel specified by CH<x>.

- REF<x>

This argument selects which reference waveform to use as the template against which to compare the waveforms acquired by the specified channel (CH<x>).

## Example 1

```
LIMit:COMpare:CH1 REF1
```

This command specifies REF1 as the template waveform against which to compare waveforms acquired using CH1.

## Example 2

```
LIMit:COMpare:CH2?
```

This query might return:

```
:LIMIT:COMPARE:CH2 REF4
```

This indicates that the waveform acquired by CH2 is compared to the template waveform stored in REF4.

---

# LIMit:COMpare:MATH<x>

## Description

This command sets or queries the template against which to compare the math waveform specified by x, which can be 1 through 4. The template can be a waveform saved in any of the reference locations (REF1 through REF4 for four-channel instruments or REF1 through REF2 for two-channel instruments) or none.

## Group

Limit Test

## Related Commands

### Syntax 1

```
LIMit:COMpare:MATH<x> {NONE|REF<x>}
```

### Syntax 2

```
LIMit:COMpare:MATH<x>?
```

## Arguments

- NONE

This argument turns off template testing for the math waveform specified by MATH<x>.

- REF<x>

This argument selects which reference waveform to use as the template against which to compare the math waveform specified by MATH<x>.

## Example 1

```
LIMit:COMpare:MATH1 REF1
```

This command specifies REF1 as the template waveform against which to compare math waveform MATH1.

## Example 2

```
LIMit:COMpare:MATH2?
```

This query might return:

```
:LIMIT:COMPARE:MATH2 REF4
```

This indicates that the MATH2 waveform is compared to the template waveform stored in REF4.

---

# LIMit:COMpare:REF<x>

## Description

This command sets or queries the template against which to compare the reference waveform specified by x, which can be 1 through 4. The template can be a waveform saved in any of the reference locations (REF1 through REF4 for four-channel instruments or REF1 through REF2 for two-channel instruments) or none.

## Group

Limit Test

## Related Commands

### Syntax 1

```
LIMit:COMpare:REF<x> {NONE|REF<x>}
```

### Syntax 2

```
LIMit:COMpare:REF<x>?
```

## Arguments

- NONE

This argument turns off template testing for the math waveform specified by MATH<x>.

- REF<x>

This argument selects which reference waveform to use as the template against which to compare the reference waveform specified by REF<x>.

## Example 1

```
LIMit:COMpare:REF2 REF1
```

This command specifies REF2 as the template waveform against which to compare reference waveform REF1.

## Example 2

```
LIMit:COMpare:REF2?
```

This query might return:

```
:LIMIT:COMPARE:REF2 REF4
```

This indicates that the REF4 waveform is compared to the template waveform stored in REF2.

---

# LIMit:EMail

## Description

This command sets or queries whether an e-mail is generated when the source waveform data exceeds the limits specified for the limit test.

## Group

Limit Test

## Related Commands

## Syntax 1

```
LIMit:EMail {OFF|ON|<NR1>}
```

## Syntax 2

```
LIMit:EMail?
```

## Arguments

- OFF

This argument disables generating an e-mail when the source waveform exceeds the test limits.

- ON

This argument enables generating an e-mail when the source waveform exceeds the test limits.

- NR1

A 0 disables the e-mail feature so that when the source waveform data exceeds the limits set by the limit test, an e-mail is not generated; any other value enables the e-mail feature so that when the source waveform data exceeds the limits set by the limit test, an e-mail is generated. For queries, a 0 is returned if the e-mail feature is off; a 1 is returned if the e-mail feature is on.

## Example 1

```
LIMit:EMail 1
```

This command enables the e-mail feature so that when the source waveform data exceeds the limits set by the limit test, an e-mail is generated.

## Example 2

```
LIMit:EMail?
```

This query might return:

```
:LIMIT:EMAIL 0
```

This indicates that the e-mail feature is disabled and that an e-mail will not be generated when the source waveform exceeds the limits of the limit test.

---

# LIMit:HARDCopy

## Description

This command sets or queries whether a hardcopy operation is executed on the waveform when any waveform data exceeds the limit set in the limit test. LIMit:STATE must be set to ON for the hardcopy operation to execute. The hardcopy operation uses the port and prints in the format and layout specified by the HARDCopy commands.

## Group

Limit Test

## Related Commands

HARDCopy:FILENAME, HARDCopy:PORT, LIMit:STATE

## Syntax 1

```
LIMit:HARDCopy {OFF|ON|<NR1>}
```

## Syntax 2

```
LIMit:HARDCopy?
```

## Arguments

- OFF

This argument disables the hardcopy operation.

- ON

This argument turns on the hardcopy operation for the waveform when any waveform data exceeds the limits set by the limit test.

- NR1

A 0 disables the hardcopy operation; any other value turns on the hardcopy operation for the waveform when any waveform data exceeds the limits set by the limit test. For queries, a 0 is returned if the hardcopy operation is off; a 1 is returned if the hardcopy operation is on.

## Example 1

```
LIMit:HARDCopy ON
```

This command specifies that the hardcopy operation will occur for the waveform when any waveform data exceeds the limits specified in the limit test.

## Example 2

```
LIMit:HARDCopy?
```

This query might return:

```
:LIMIT:HARDCOPY 0
```

This indicates that the harcopy operation is disabled.

---

# LIMit:HIGHLIGHTHits

## Description

This command sets or queries whether violation highlighting occurs when limit testing is active, and, if the RESET argument is set, clears the highlighting.

## Group

Limit Test

## Related Commands

### Syntax 1

```
LIMit:HIGHLIGHTHits {OFF|ON|RESET|<NR1>}
```

### Syntax 2

```
LIMit:HIGHLIGHTHits?
```

## Arguments

#### ■ OFF

This argument disables violation highlighting when limit testing is active.

#### ■ ON

This argument enables violation highlighting when limit testing is active.

#### ■ RESET

This argument clears the violation highlighting.

#### ■ NR1

A 0 disables the violation highlighting when limit testing is active; any other value enables the violation highlighting feature when limit testing is active. For queries, a 0 is returned if the violation highlighting feature is off; a 1 is returned if the violation highlighting feature is on.

## Example 1

```
LIMit:HIGHLIGHTHits 1
```

This command enables the violation highlighting when the limit testing is active.

## Example 2

```
LIMit:HIGHLIGHTHits?
```

This query might return:

```
:LIMIT:HIGHLIGHTHITS 0
```

This indicates that the violation highlighting feature is disabled.

---

# LIMit:LOCK

## Description

This command sets or queries whether vertical scaling and positioning affect both source and template for template comparison pairs.

## Group

Limit Test

## Related Commands

### Syntax 1

```
LIMit:LOCK {OFF|ON|<NR1>}
```

### Syntax 2

```
LIMit:LOCK?
```

## Arguments

- OFF

This argument disables the lock feature.

- ON

This argument enables the lock feature.

- NR1

A 0 disables the lock feature; any other value enables the lock feature. For queries, a 0 is returned if the lock feature is off; a 1 is returned if the lock feature is on.

## Example 1

```
LIMit:LOCK ON
```

This command enables the lock feature so that vertical scaling and positioning affect both the source and the template for template comparison pairs.

## Example 2

```
LIMit:LOCK?
```

This query might return:

```
:LIMIT:LOCK 0
```

This indicates that the lock feature is disabled.



---

# LIMit:LOG

## Description

This command sets or queries whether a log file is saved when the source waveform data exceeds the test limits.

## Group

Limit Test

## Related Commands

### Syntax 1

```
LIMit:LOG {OFF|ON|NR1}
```

### Syntax 2

```
LIMit:LOG?
```

## Arguments

- OFF

This argument disables saving a log file when the source waveform exceeds the test limits.

- ON

This argument enables saving a log file when the source waveform exceeds the test limits.

- NR1

A 0 disables the save log file feature so that when the source waveform data exceeds the limits set by the limit test, a log file is not saved; any other value enables the save log file feature so that when the source waveform data exceeds the limits set by the limit test, a log file is saved. For queries, a 0 is returned if the save log file feature is off; a 1 is returned if the save log file feature is on.

## Example 1

```
LIMit:LOG 1
```

This command enables the save log file feature so that when the source waveform data exceeds the limits set by the limit test, a log file is saved.

## Example 2

```
LIMit:LOG?
```

This query might return:

```
:LIMIT:LOG 0
```

This indicates that the save log file feature is disabled and that a log file will not be saved when the source waveform exceeds the limits of the limit test.

---

# LIMit:SAVEWFM

## Description

This command sets or queries whether the source waveform is saved when the source waveform data exceeds the test limits.

## Group

Limit Test

## Related Commands

## Syntax 1

```
LIMit:SAVEWFM {OFF|ON|NR1}
```

## Syntax 2

```
LIMit:SAVEWFM?
```

## Arguments

- OFF

This argument disables saving the source waveform when it exceeds the test limits.

- ON

This argument enables saving the source waveform when it exceeds the test limits.

- NR1

A 0 disables the save waveform feature so that when the source waveform data exceeds the limits set by the limit test, the source waveform is not saved; any other value enables the save waveform feature so that when the source waveform data exceeds the limits set by the limit test, the source waveform is saved. For queries, a 0 is returned if the save waveform feature is off; a 1 is returned if the save waveform feature is on.

## Example 1

```
LIMit:SAVEWFM 1
```

This command enables the save waveform feature so that when the source waveform data exceeds the limits set by the limit test, the source waveform is saved.

## Example 2

```
LIMit:SAVEWFM?
```

This query might return:

```
:LIMIT:SAVEWFM 0
```

This indicates that the save waveform feature is disabled and that the source waveform will not be saved when it exceeds the limits of the limit test.

---

# LIMit:SAVWFM:FILEName

## Description

This command sets or queries the path where waveforms or log files will be saved when the waveform data exceeds the limits set by the limit test.

## Group

Limit Test

## Related Commands

### Syntax 1

```
LIMit:SAVEWFM:FILEName {<string>}
```

### Syntax 2

```
LIMit:SAVEWFM:FILEName?
```

## Argument

- <string>

This argument is a string containing the path of the waveform to be saved.

## Example 1

```
LIMit:SAVEWFM:FILEName MyLimitTestWaveform
```

This command sets the path where a waveform will be saved when it exceeds the limits of the limit test.

## Example 2

```
LIMit:SAVEWFM:FILEName?
```

This query might return:

```
:LIMIT:SAVEWFM:FILENAME "MyLimitTestWave-  
form.wfm"
```

This indicates the path where the waveform will be saved when it exceeds the limits of the limit test.

---

# LIMit:SRQ

## Description

This command sets or queries whether a Service Request Interrupt (SRQ) is generated when the waveform data falls outside of the test limits.

## Group

Limit Test

## Related Commands

### Syntax 1

```
LIMit:SRQ {OFF|ON|NR1}
```

### Syntax 2

```
LIMit:SRQ?
```

## Arguments

- OFF

This argument disables generation of an SRQ when the waveform data falls outside of the limits set by the limit test.

- ON

This argument enables generation of an SRQ when the waveform data falls outside of the limits set by the limit test. If an SRQ is generated, the instrument sends the status event "Limit testing failed."

- NR1

A 0 disables sending an SRQ when the waveform data falls outside of the limits set by the limit test; any other value enables generation of an SRQ when the waveform data falls outside of the limits set by the limit test. For queries, a 0 is returned if SRQ is off; a 1 is returned if the SRQ is on.

### Example 1

```
LIMit:SRQ 1
```

This command specifies that an SRQ will be generated when the waveform data falls outside of the limits set by the limit test. If an SRQ is generated, an status event of "Limit testing failed" is output.

### Example 2

```
LIMit:SRQ?
```

This query might return:

```
:LIMIT:SRQ 0
```

This indicates that an SRQ will not be generated when the waveform data falls outside of the limits set by the limit test.

---

# LIMit:STATE

## Description

This command sets limit testing on or off or queries whether limit testing is in effect.

## Group

Limit Test

## Related Commands

## Syntax 1

```
LIMit:STATE {OFF | ON | <NR1>}
```

## Syntax 2

```
LIMit:STATE?
```

## Arguments

- OFF

This argument disables limit testing.

- ON

This argument turns on limit testing of waveforms.

- NR1

A 0 disables the limit testing operation; any other value turns on limit testing of waveforms. For queries, a 0 is returned if limit testing is off; a 1 is returned if the limit testing is on.

## Example 1

```
LIMit:STATE ON
```

This command specifies that limit testing of waveforms is on.

## Example 2

```
LIMit:STATE?
```

This query might return:

```
:LIMIT:STATE 0
```

This indicates that the limit testing of waveforms is off.

---

# LIMit:STOPOnviolation

## Description

This command sets or queries whether acquisitions are stopped when the waveform data exceeds the test limits.

## Group

Limit Test

## Related Commands

## Syntax 1

```
LIMit:STOPOnviolation {OFF|ON|NR1}
```

## Syntax 2

```
LIMit:STOPOnviolation?
```

## Arguments

- OFF

This argument disables the stop on violation feature.

- ON

This argument enables the stop on violation feature so that when the waveform data exceeds the limits set by the limit test, acquisitions are stopped.

- NR1

A 0 disables the stop on violation feature; any other value enables the stop on violation feature so that when the waveform data exceeds the limits set by the limit test, acquisitions are stopped. For queries, a 0 is returned if the stop on violation feature is off; a 1 is returned if the stop on violation feature is on.

## Example 1

```
LIMit:STOPOnviolation 1
```

This command enables the stop on violation feature so that when the waveform data exceeds the limits set by the limit test, acquisitions are stopped.

## Example 2

```
LIMit:STOPOnviolation?
```

This query might return:

```
:LIMIT:STOPONVIOLATION 0
```

This indicates that the stop on violation feature is disabled.

---

# LIMit:TEMPLate:STORe

## Description

This command (no query form) saves the specified source waveform to the specified reference or file name.

## Group

Limit Test

## Related Commands

## Syntax

```
LIMit:TEMPLate:STORe <wfm>,{REF<x>|<file path>}
```

## Arguments

### ■ <wfm>

This argument specifies the waveform that will be saved as the template. The source of the waveform can be CH<x> (where x is 1 through 4 for four channel instruments or 1 through 2 for two channel instruments), MATH<x>, or Ref<x> (where x is 1 through 4).

### ■ REF<x>

This argument specifies a reference location in which the template waveform will be stored. The reference waveform specified by x, which can be 1 through 4

### ■ <file path>

This argument specifies a file path where the template waveform will be stored. This argument is a quoted string.

## Example 1

```
LIMit:TEMPLate:STORe CH2,REF1
```

This command creates a template reference waveform, REF1, using CH2 as its source

## Example 2

```
LIMit:TEMPLate:STORe CH1,"template.wfm"
```

This command creates a template reference waveform, template.wfm, using CH1 as the source.

---

# LIMit:TEMPLate:TOLerance:HORizontal

## Description

This command sets or queries the amount, in units of horizontal divisions, by which the source waveform is varied horizontally when creating the destination waveform.

## Group

Limit Test

## Related Commands

### Syntax 1

```
LIMit:TEMPLate:TOLerance:HORizontal <tolerance>
```

### Syntax 2

```
LIMit:TEMPLate:TOLerance:HORizontal?
```

## Argument

- <tolerance>

This argument is the amount, in horizontal divisions, by which the current source waveform is allowed to deviate from the template waveform without exceeding the limits set in the limit test. The range is 0 to 5 divisions. The format is floating point with exponent.

## Example 1

```
LIMit:TEMPLate:TOLerance:HORizontal 1.0
```

This command specifies that the current waveform is close enough to the template waveform if it is within  $\pm 1.0$  horizontal division.

## Example 2

```
LIMit:TEMPLate:TOLerance:HORizontal?
```

This query might return:

```
:LIMit:TEMPLate:TOLerance:HORizontal 2.0
```

This indicates that the current waveform is close enough to the template waveform if it is within  $\pm 2.0$  horizontal divisions.



---

# LIMit:TEMPLate:TOLerance:VERTical

## Description

This command sets or queries the amount, in units of vertical divisions, by which the source waveform is varied vertically when creating the destination waveform.

## Group

Limit Test

## Related Commands

### Syntax 1

```
LIMit:TEMPLate:TOLerance:VERTical <tolerance>
```

### Syntax 2

```
LIMit:TEMPLate:TOLerance:VERTical?
```

## Argument

- <tolerance>

This argument is the amount, in vertical divisions, by which the current source waveform is allowed to deviate from the template waveform without exceeding the limits set in the limit test. The range is 0 to 5 divisions.

## Example 1

```
LIMit:TEMPLate:TOLerance:VERTical 1.0
```

This command specifies that the current waveform is close enough to the template waveform if it is within  $\pm 1.0$  vertical division.

## Example 2

```
LIMit:TEMPLate:TOLerance:VERTical?
```

This query might return:

```
:LIMit:TEMPLate:TOLerance:VERTical 2.0
```

This indicates that the current waveform is close enough to the template waveform if it is within  $\pm 2.0$  vertical divisions.

---

# LIMit?

## Description

This query-only command returns all settings for the Limit commands.

## Group

Limit Test

## Related Commands

## Syntax

LIMit?

## Example

LIMit?

This query might return:

```
:LIMIT:BEEP 0;COMPARE:CH1 NONE;CH2 NONE;CH3  
NONE;CH4 NONE;REF1 NONE;REF2 NONE;REF3  
NONE;REF4 NONE;MATH1 NONE;MATH2 NONE;MATH3  
NONE;MATH4 NONE;:LIMIT:HARDCOPY 0;STATE 0;TEM-  
PLATE:DESTINATION REF1;SOURCE CH2;TOLER-  
ANCE:HORIZONTAL 40.0000E-3;VERTICAL  
40.0000E-3;:LIMIT:SRQ 0;STOPONVIOLATION 0;SA-  
VEWFM 0;SAVEWFM:FILENAME "C:/TekScope/wave-  
forms/";:LIMIT:LOG 0;EMAIL 0;HIGHLIGHTHITS  
1;LOCK 1
```

---

# LOCK

## Description

This command enables or disables all front-panel buttons and knobs, including the touch screen. There is no front panel equivalent.

## Group

Miscellaneous

## Related Commands

UNLock

## Syntax 1

LOCK {ALL|NONE}

## Syntax 2

LOCK?

## Arguments

- ALL

This disables all front panel controls.

- NONE

This enables all front panel controls. This is equivalent to the UNLock ALL command.

**Note:** If the instrument is in the Remote With Lockout State (RWLS), the LOCK NONE command has no effect. For more information, see the ANSI/IEEE Std 488.1-1987 Standard Digital Interface for Programmable Instrumentation, section 2.8.3 on RL State Descriptions.

## Example 1

```
LOCK ALL
```

This command locks the front panel controls.

## Example 2

```
LOCK?
```

This query might return

```
:LOCK NONE
```

This indicates that the front panel controls are enabled by this command.

---

# MASK?

## Description

This query-only command returns the states of all settable mask parameters

## Group

Mask

## Syntax

MASK?

## Example

MASK?

---

# MASK:AUTOAdjust

## Description

This command optimizes signal position within the mask to minimize hits. It sets a mode so that the MASK:SOURCE waveform is compared against the mask and is shifted up, down, left, or right to minimize the hits.

## Group

Mask

## Syntax

```
MASK:AUTOAdjust {ON|OFF|<NR1>}
```

## Arguments

- ON

This enables the autoadjust function.

- OFF

This disables the autoadjust function

- <NR1>

A 0 disables the autoadjust function; any other value enables it.

## Example

```
MASK:AUTOAdjust ON
```

Enables the autoadjust function

---

# MASK:AUTOAdjust:HDELTA

## Description

This command sets or returns how far autofit searches horizontally.

Autofit moves the waveform vertically and/or horizontally to reduce the number of hits within a given mask. If infinite or variable persistence is enabled, these movements will clear any persisted data. If Autofit makes frequent adjustments, there can be very little or even no persisted data displayed.

## Group

Mask

## Syntax

```
MASK:AUTOAdjust:HDELTA {<NR3>}
```

## Argument

- <NR3>

Is a floating point number which represents a percent of a division.

## Example

```
MASK:AUTOAdjust:HDELTA 0.6
```

Sets the horizontal autofit search to 0.6 division.

---

# MASK:AUTOAdjust:VDELTA

## Description

This command sets or returns how far autofit searches vertically.

Autofit moves the waveform vertically and/or horizontally to reduce the number of hits within a given mask. If infinite or variable persistence is enabled, these movements will clear any persisted data. If Autofit makes frequent adjustments, there can be very little or even no persisted data displayed.

## Group

Mask

## Syntax 1

```
MASK:AUTOAdjust:VDELTA {<NR3>}
```

## Syntax 2

```
MASK:AUTOAdjust:VDELTA?
```

## Argument

- <NR3>

Is a floating point number which represents a percent of a division.

## Example

```
MASK:AUTOAdjust:VDELTA 0.6
```

Sets the vertical autofit search to 0.6 of a division.

---

# MASK:AUTOSet:AUTOAdjust

## Description

This command sets or returns whether the instrument settings are automatically matched to signal characteristics and specific mask requirements. It controls what happens at the end of a mask autoset. If ON, the horz/vert auto adjustment is run ONCE to potentially improve the result of mask autoset.

This command, like all the MASK:AUTOSET commands, affects only an autoset on a standard mask, not the general oscilloscope autoset function.

## Group

Mask

## Syntax 1

```
MASK:AUTOSet:AUTOAdjust {ON|OFF|<NR1>}
```

## Syntax 2

```
MASK:AUTOSet:AUTOAdjust?
```

## Arguments

- ON

This enables the autoset:autoadjust function.

- OFF

This disables the autoset:autoadjust function.

- <NR1>

A 0 disables the autoset:autoadjust function; any other value enables it.

## Example

```
MASK:AUTOSet:AUTOAdjust ON
```

Enables the autoset autoadjust function



---

# MASK:AUTOSet:HPOS

## Description

This command controls whether the autosegment algorithm will attempt to change the horizontal position of the signal while attempting to autosegment. The default is ON.

This command, like all the MASK:AUTOSET commands, affects only an autosegment on a standard mask, not the general oscilloscope autosegment function.

## Group

Mask

## Syntax

```
MASK:AUTOSet:HPOS {ON|OFF|<NR1>}
```

## Arguments

- ON

This enables the autosegment:hpos function, causing ... to occur.

- OFF

This disables the autosegment:hpos function, causing ... to occur.

- <NR1>

A 0 disables the autosegment:hpos function; any other value enables it.

## Example

```
MASK:AUTOSet:HPOS ON
```

Enables the autosegment:hpos function

---

# MASK:AUTOSet:HSCALE

## Description

This command controls whether the autosegment algorithm will attempt to change the horizontal scale while attempting to autosegment.

This command, like all the MASK:AUTOSET commands, affects only an autosegment on a standard mask, not the general oscilloscope autosegment function.

## Group

Mask

## Syntax

```
MASK:AUTOSet:HSCALE {ON|OFF|<NR1>}
```

## Arguments

- ON

This enables the autosegment:hscale function.

- OFF

This disables the autosegment:hscale function.

- <NR1>

A 0 disables the autosegment:hscale function; any other value enables it.

## Example

```
MASK:AUTOSet:HSCALE ON
```

Enables the autosegment:hscale function.

---

# MASK:AUTOSet:MODE

## Description

This command controls whether or not an autoset will be automatically done after a standard mask is selected. The autoset will never happen if the standard mask is selected from GPIB, since a "SET?" sent back to the scope should not do a mask autoset.

The default value is: manual.

This command, like all the MASK:AUTOSET commands, affects only an autoset on a standard mask, not the general oscilloscope autoset function.

## Group

Mask

## Syntax

```
MASK:AUTOSet:MODE {MANual|AUTO}
```

## Argument

- MANual

This disables autoset from occurring automatically after a standard mask is selected.

- AUTO

This enables autoset to automatically occur after a standard mask is selected.

## Example

```
MASK:AUTOSet:MODE AUTO
```

Enables autoset to occur automatically after a standard mask is selected.

---

# MASK:AUTOSet:OFFSETAdj

## Description

This command sets autoset to not enforce the rule that, for pulse standards, require 0V to be in a certain place in the mask. Instead, autoset will attempt to measure the DC offset in the signal and use scope offset or position controls to compensate for the DC offset in the signal.

This command, like all the MASK:AUTOSET commands, affects only an autoset on a standard mask, not the general oscilloscope autosest function.

## Group

Mask

## Syntax 1

```
MASK:AUTOSet:OFFSETAdj {ON|OFF|<NR1>}
```

## Syntax 2

```
MASK:AUTOSet:OFFSETAdj?
```

## Arguments

- ON

This enables the autosest:offsetadj function.

- OFF

This disables the autosest:offsetadj function.

- <NR1>

A 0 disables the autosest:offsetadj function; any other value enables it.

## Example

```
MASK:AUTOSET:OFFSETADJ ON
```

Enables the autosest:offsetadj function

---

# MASK:AUTOSet:STANdard

## (TDS5000B Series Option SM)

### Description

This command selects the standard mask in autoselect.

This command, like all the MASK:AUTOSET commands, affects only an autoselect on a standard mask, not the general oscilloscope autoselect function.

### Group

Mask

### Syntax 1

```
MASK:AUTOSet:STANdard {G703D1|E1Symmetri-
cal|E1Coax
|DS2RATESymmetrical|DS2RATE-
Coax|E2|RATE32Mbit|E3
|G703DS3|RATE97Mbit|E4_0|E4_1|STM1E_0|STM1E_1|
DS1
|DS1A|DS1C|DS2|DS3|DS4NA|DS4NA_Max|STS1Pulse
|STS1Eye|STS3|STS3_Max|ENET100STP|ENET100UTP
|ENET1250|FC133|FC266|FC531|FC1063|FC1063Draft
|FC2125|FC133E|FC266E|FC531E|FC1063E|FC1063E_N
BT
|FC1063E_NDT|FC1063E_NGT|FC1063E_ABT|FC1063E_A
DT
|FC1063E_AGT|FC1063E_ABR|FC1063E_ADR|FC1063E_A
GR
|FC2125E_NBT|FC2125E_NDT|FC2125E_NGT|FC2125E_A
BT
|FC2125E_ADT|FC2125E_AGT|FC2125E_ABR|FC2125E_A
DR
|FC2125E_AGR|OC1|OC3|OC12|OC48|OC48_FEC|INF2_5
G
|INF2_5GE|ATATXG1|ATATXG2|ATATXG3|ATARXG1|ATAR
XG2
|ATARXG3|FST1|FST2|FST3|FST4|FST5|FST6|HST1|HS
T2
|HST3|HST4|HST5|HST6|FW1394BS400BT1|FW1394BS40
0BT2
|FW1394BS800BT1|FW1394BS800BT2|FW1394BS1600BT1
|FW1394BS1600BT2|FW1394BS400B|FW1394BS800B
|FW1394BS1600B|USERMask|ENET100BCX_NTP2
|ENET100BCX_ATP2|ENET100BCX_ATP3|ENET-
XAUI_Near
|ENET-
XAUI_Far|RIO_DRV500M|RIO_DRV750M|RIO_DRV1G
|RIO_DRV1_5G|RIO_DRV2G|RIO_EDRV500M|RIO_EDRV75
0M
|RIO_EDRV1G|RIO_EDRV1_5G|RIO_EDRV2G
```

```

|RIO_RCV500M|RIO_RCV750M|RIO_RCV1G|RIO_RCV1_5G
|RIO_RCV2G
|RIO_SERIAL_1G|RIO_SERIAL_2G|RIO_SERIAL_3G
|SFI5_XMITADATA2|SFI5_XMITCDATA2|SFI5_XMI-
TACLK2
|SFI5_XMITCCLK2|SFI5_RCVBDATA2|SFI5_RCVDDATA2
|SFI5_RCVBCLK2|SFI5_RCVDCLK2|SFI5_XMITADATA3
|SFI5_XMITCDATA3|SFI5_XMITACLK3|SFI5_XMITCCLK3
|SFI5_RCVBDATA3|SFI5_RCVDDATA3|SFI5_RCVBCLK3
|SFI5_RCVDCLK3|VSROC192|TFI15_2|TFI5_3
|PCIEXPRESS_Xmit|PCIEXPRESS_Rcv}

```

## Syntax 2

MASK:AUTOSet:STANdard?

### Arguments

- G703D1 (ITU-T, G703 (10/98), DS1 Rate, 1.544 Mb/s)
- E1Symmetrical (ITU-T, G703 (10/98), E1 Sym Pair, 2.048 Mb/s)
- E1Coax (ITU-T, G703 (10/98), E1 Coax Pair, 2.048 Mb/s)
- DS2RATESymmetrical (ITU-T, G703 (10/98), D2 Rate Sym, 6.312 Mb/s)
- DS2RATECoax (ITU-T, G703 (10/98), D2 Rate Coax, 6.312 Mb/s)
- E2 (ITU-T, G703 (10/98), E2, 8.448 Mb/s)
- RATE32Mbit (ITU-T, G703 (10/98), 32.064 Mb/s)
- E3 (ITU-T, G703 (10/98), E3, 34.368 Mb/s)
- G703DS3 (ITU-T, G703 (10/98))
- RATE97Mbit (ITU-T, G703 (10/98), 97 Mbit, 97.728 Mb/s)
- E4\_0 (ITU-T, G703 (10/98), E4 Binary 0)
- E4\_1 (ITU-T, G703 (10/98), E4 Binary 1)
- STM1E\_0 (ITU-T, G703 (10/98), STM1E Binary 0)
- STM1E\_1 (ITU-T, G703 (10/98), STM1E Binary 1)
- DS1 (ANSI T1.102-1993 (R1999), DS1, 1.544 Mb/s)
- DS1A (ANSI T1.102-1993 (R1999), DS1A, 2.048 Mb/s)
- DS1C (ANSI T1.102-1993 (R1999), DS1C, 3.152 Mb/s)

- 
- DS2 (ANSI T1.102-1993 (R1999), DS2, 6.312 Mb/s)
  - DS3 (ANSI T1.102-1993 (R1999), DS3, 44.736 Mb/s)
  - DS4NA (ANSI T1.102-1993 (R1999), DS4NA, 139.26 Mb/s)
  - DS4NA\_Max (ANSI T1.102-1993 (R1999), DSNA Max Output, 139.26 Mb/s)
  - STS1Pulse (ANSI T1.102-1993 (R1999), STS-1 Pulse, 51.84 Mb/s)
  - STS1Eye (ANSI T1.102-1993 (R1999), STS-1 Eye, 51.84 Mb/s)
  - STS3 (ANSI T1.102-1993 (R1999), STS-3, 155.52 Mb/s)
  - STS3\_Max (ANSI T1.102-1993 (R1999), STS-3 Max Output, 155.52 Mb/s)
  - ENET100STP (IEEE Std 802.3 and ANSI X3.263-1995, 100 Base-Tx, STP, 125 Mb/s )
  - ENET100UTP (IEEE Std 802.3 and ANSI X3.263-1995, 100 Base-Tx, UTP, 125 Mb/s)
  - ENET1250 (IEEE Std 802.3 and ANSI X3.263-1995, GB Ethernet, 1.25 Gb/s)
  - FC133 (ANSI X3.230-1999 NCITS 1235D/Rev 11, Optical, 132.8 Mb/s)
  - FC266 (ANSI X3.230-1999 NCITS 1235D/Rev 11, Optical, 265.6 Mb/s)
  - FC531 (ANSI X3.230-1999 NCITS 1235D/Rev 11, Optical, 531.2 Mb/s)
  - FC1063 (ANSI X3.230-1999 NCITS 1235D/Rev 11, Optical, 1.065 Gb/s)
  - FC1063Draft (ANSI X3.230-1999 NCITS 1235D/Rev 11, Optical, Draft Rev 11)
  - FC2125 (ANSI X3.230-1999 NCITS 1235D/Rev 11, Optical, 2.125)
  - FC133E (ANSI X3.230-1999 NCITS 1235D/Rev 11, Electrical 132.8 Mb/s)
  - FC266E (ANSI X3.230-1999 NCITS 1235D/Rev 11, Electrical, 132.8 Mb/s)
  - FC531E (ANSI X3.230-1999 NCITS 1235D/Rev 11, Electrical, 531.2 Mb/s)
  - FC1063E (ANSI X3.230-1999 NCITS 1235D/Rev 11, Electrical, 1.0625 Gb/s)

- 
- FC1063E\_NBT (ANS1 X3.230-1999 NCITS 1235D/Rev 11, Norm, Beta, Transm)
  - FC1063E\_NDT (ANS1 X3.230-1999 NCITS 1235D/Rev 11, Norm, Delta, Transm)
  - FC1063E\_NGT (ANS1 X3.230-1999 NCITS 1235D/Rev 11, Norm, Gamma, Transm)
  - FC1063E\_ABT (ANS1 X3.230-1999 NCITS 1235D/Rev 11, Abs, Beta, Transm)
  - FC1063E\_ADT (ANS1 X3.230-1999 NCITS 1235D/Rev 11, Abs, Beta, Transm)
  - FC1063E\_AGT (ANS1 X3.230-1999 NCITS 1235D/Rev 11, Abs, Gamma, Transm)
  - FC1063E\_ABR (ANS1 X3.230-1999 NCITS 1235D/Rev 11, Abs, Beta, Recv)
  - FC1063E\_ADR (ANS1 X3.230-1999 NCITS 1235D/Rev 11, Abs, Delta, Recv)
  - FC1063E\_AGR (ANS1 X3.230-1999 NCITS 1235D/Rev 11, Abs, Gamma, Recv)
  - FC2125E\_NBT (ANS1 X3.230-1999 NCITS 1235D/Rev 11, Norm, Beta, Transm)
  - FC2125E\_NDT (ANS1 X3.230-1999 NCITS 1235D/Rev 11, Norm, Delta, Transm)
  - FC2125E\_NGT (ANS1 X3.230-1999 NCITS 1235D/Rev 11, Norm, Gamma, Transm)
  - FC2125E\_ABT (ANS1 X3.230-1999 NCITS 1235D/Rev 11, Abs, Beta, Transm)
  - FC2125E\_ADT (ANS1 X3.230-1999 NCITS 1235D/Rev 11, Abs, Delta, Transm)
  - FC2125E\_AGT (ANS1 X3.230-1999 NCITS 1235D/Rev 11, Abs, Gamma, Transm)
  - FC2125E\_ABR (ANS1 X3.230-1999 NCITS 1235D/Rev 11, Abs, Beta, Recv)
  - FC2125E\_ADR (ANS1 X3.230-1999 NCITS 1235D/Rev 11, Abs, Delta, Recv)
  - FC2125E\_AGR (ANS1 X3.230-1999 NCITS 1235D/Rev 11, Abs, Gamma, Recv)
  - OC1 (GR 253-CORE Issue 3 9/21/2000 OC1/STM0, 51.84 Mb/s)
  - OC3 (GR 253-CORE Issue 3 9/21/2000 OC1/STM1, 155.52, Mb/s)
  - OC12 (GR 253-CORE Issue 3 9/21/2000 OC1/STM4, 622.08 Mb/s)



- 
- OC48 (GR 253-CORE Issue 3 9/21/2000  
OC1/STM16, 2.4883 Gb/s)
  - OC48\_FEC (Forward Error Correction -  
CSA8000 mask, 2.666 Gb/s)
  - INF2\_5G (InfiniBand, IBTA Spec 1.0a, 2.5  
Optical, 2.5 Gb/s)
  - INF2\_5GE (InfiniBand, IBTA Spec 1.0a,  
2.5 Electrical, 2.5 Gb/s)
  - ATATXG1 (Serial ATA, G1 Tx, 1.5 Gb/s)
  - ATATXG2 (Serial ATA, G2 Tx, 3.0 Gb/s)
  - ATATXG3 (Serial ATA, G3 Tx)
  - ATARXG1 (Serial ATA, G1 Rx 1.5 Gb/s)
  - ATARXG2 (Serial ATA, G2, Rx, 3.0 Gb/s)
  - ATARXG3 (Serial ATA, G3, Rx)
  - FST1 (USB, FS:T1, 12 Mb/s)
  - FST2 (USB, FS:T2, 12 Mb/s)
  - FST3 (USB, FS:T3, 12 Mb/s)
  - FST4 (USB, FS:T4: 12 Mb/s)
  - FST5 (USB, FS:T5, 12 Mb/s)
  - FST6 (USB, FS:T6, 12 Mb/s)
  - HST1 (USB, HS:T1, 480 Mb/s)
  - HST2 (USB, HS:T2, 480 Mb/s)
  - HST3 (USB, HS:T3, 480 Mb/s)
  - HST4 (USB, HS:T4, 480 Mb/s)
  - HST5 (USB, HS:T5, 480 Mb/s)
  - HST6 (USB, HS:T6, 480 Mb/s)
  - FW1394BS400BT1 (IEEE 1394b, S400b T1,  
491.5 Mb/s)
  - FW1394BS400BT2 (IEEE 1394b, S400b T2,  
491.5 Mb/s)
  - FW1394BS800BT1 (IEEE 1394b, S800b T1,  
983.0 Mb/s)
  - FW1394BS800BT2 (IEEE 1394b, S800b T2,  
983.0 Mb/s)
  - FW1394BS1600BT1 (IEEE 1394b, S1600b T1,  
1.966 Gb/s)
  - FW1394BS1600BT2 (IEEE 1394b, S1600b T2,  
1.966 Gb/s)
  - FW1394BS400B (IEEE 1394b, S400 Optical,  
491.5 Mb/s)

- 
- FW1394BS800B (IEEE 1394b, S800 Optical, 988.0 Mb/s)
  - FW1394BS1600B (IEEE 1394b, S1600 Optical, 1.966 Gb/s)
  - USERMask
  - ENET1000BCX\_NTP2 (1000B-CX Norm, TP2, 1.25 Gb/s)
  - ENET1000BCX\_ATP2 (1000B-CX Abs, TP2, 1.25 Gb/s)
  - ENET1000BCX\_ATP3 (1000B-CX Abs, TP3, 1.25 Gb/s)
  - ENETXAUI\_Near (10 Gigabit Attachment Unit Interface (XAUI), Near, 3,125 Gb/s)
  - ENETXAUI\_Far (10 Gigabit Attachment Unit Interface (XAUI), Far, 3.125 Gb/s)
  - RIO\_DRV500M (Rapid IO Driver, 500 Mb/s)
  - RIO\_DRV750M (Rapid IO Driver, 750 Mb/s)
  - RIO\_DRV1G (Rapid IO Driver, 1 Gb/s)
  - RIO\_DRV1\_5G (Rapid IO Driver, 5 Gb/s)
  - RIO\_DRV2G (Rapid IO Driver, 2 Gb/s)
  - RIO\_EDRV500M (Rapid IO Extended Driver, 500 Mb/s)
  - RIO\_EDRV750M (Rapid IO Extended Driver, 750 Mb/s)
  - RIO\_EDRV1G (Rapid IO Extended Driver, 1 Gb/s)
  - RIO\_EDRV1\_5G (Rapid IO Extended Driver, 1.5 Gb/s)
  - RIO\_EDRV2G (Rapid IO Extended Driver, 2 Gb/s)
  - RIO\_RCV500M (Rapid IO Receiver, 500 Mb/s)
  - RIO\_RCV750M (Rapid IO Receiver, 750 Mb/s)
  - RIO\_RCV1G (Rapid IO Receiver, 1 Gb/s)
  - RIO\_RCV1\_5G (Rapid IO Receiver, 1.5 Gb/s)
  - RIO\_RCV2G (Rapid IO Receiver, 2 Gb/s)
  - RIO\_SERIAL\_1G (Rapid IO Serial, 1.25 Gb/s)
  - RIO\_SERIAL\_2G (Rapid IO Serial, 2.5 Gb/s)

- 
- RIO\_SERIAL\_3G (Rapid IO Serial, 3.25 Gb/s)
  - SFI5\_XMITADATA2 (SFI15 Transmit: Test Point A Data Signal 2, 2.488 Gb/s)
  - SFI5\_XMITCDATA2 (SFI15 Transmit: Test Point C Data Signal 2, 2.488 Gb/s)
  - SFI5\_XMITACLK2 (SFI15 Transmit: Test Point A Clock Signal 2, 2.488 Gb/s)
  - SFI5\_XMITCCLK2 (SFI15 Transmit: Test Point C Clock Signal 2, 2.488 Gb/s)
  - SFI5\_RCVBDATA2 (SFI15 Receive: Test Point B Data Signal 2, 2.488 Gb/s)
  - SFI5\_RCVDDATA2 (SFI15 Receive: Test Point D Data Signal 2, 2.488 Gb/s)
  - SFI5\_RCVBCLK2 (SFI15 Receive: Test Point B Clock Signal 2, 2.488 Gb/s)
  - SFI5\_RCVCLK2 (SFI15 Receive: Test Point D Clock Signal 2, 2.488 Gb/s)
  - SFI5\_XMITADATA3 (SFI15 Transmit: Test Point A Data Signal 3, 3.125 Gb/s)
  - SFI5\_XMITCDATA3 (SFI15 Transmit: Test Point C Data Signal 3, 3.125 Gb/s)
  - SFI5\_XMITACLK3 (SFI15 Transmit: Test Point A Clock Signal 3, 3.125 Gb/s)
  - SFI5\_XMITCCLK3 (SFI15 Transmit: Test Point C Clock Signal 3, 3.125 Gb/s)
  - SFI5\_RCVBDATA3 (SFI15 Receive: Test Point B Data Signal 3, 3.125 Gb/s)
  - SFI5\_RCVDDATA3 (SFI15 Receive: Test Point D Data Signal 3, 3.125 Gb/s)
  - SFI5\_RCVBCLK3 (SFI15 Receive: Test Point B Clock Signal 3, 3.125 Gb/s)
  - SFI5\_RCVCLK3 (SFI15 Receive: Test Point D Clock Signal 3, 3.125 Gb/s)
  - VSROC192 (VSR OC192/STM64, 1.24416 Gb/s)
  - TFI15\_2 (TFI-5, 2.488 Gb/s)
  - TFI15\_3 (TFI-5, 3.1104 Gb/s)
  - PCIEXPRESS\_Xmit (PCI Express Transmitter, 2.5 Gb/s)
  - PCIEXPRESS\_Rcv (PCI Express Receiver, 2.5 Gb/s)

---

### **Example 1**

`MASK:AUTOSET:STANDARD E1COAX`

Selects the E1 Coax standard mask.

### **Example 2**

`MASK:AUTOSET:STANDARD?`

Might return

`:MASK:AUTOSET:STANDARD E1COAX`

Indicating that the E1 Coax mask is selected as the current mask standard.

---

# MASK:AUTOSet:TRIGger

## Description

This command controls if a MASK:AUTOSet affects the trigger level. Other trigger settings such as type of trigger are not changed by MASK:AUTOSet. The default is ON.

This command, like all the MASK:AUTOSET commands, affects only an autoset on a standard mask, not the general oscilloscope autosest function.

## Group

Mask

## Syntax

```
MASK:AUTOSet:TRIGger {ON|OFF|<NR1>}
```

## Arguments

- ON

This enables the autosest:hpos function.

- OFF

This disables the autosest:hpos function.

- <NR1>

A 0 disables the autosest:trigger function; any other value enables it.

## Example

```
MASK:AUTOSet:HPOS ON
```

Enables the autosest:hpos function

---

# MASK:AUTOSet:VPOS

## Description

This command controls whether the autosegment algorithm will affect the vertical position (or offset) of the signal while attempting to autosegment. The default is ON.

This command, like all the MASK:AUTOSET commands, affects only an autosegment on a standard mask, not the general oscilloscope autosegment function.

## Group

Mask

## Syntax

```
MASK:AUTOSet:VPOS {ON|OFF|<NR1>}
```

## Arguments

- ON

This enables the autosegment:vpos function.

- OFF

This disables the autosegment:vpos function.

- <NR1>

A 0 disables the autosegment:vpos function; any other value enables it.

## Example

```
MASK:AUTOSet:VPOS ON
```

Enables the autosegment:vpos function

---

# MASK:AUTOSet:VSCALE

## Description

This command controls whether the autosegment algorithm will affect the vertical scale while attempting to autosegment.

This command, like all the MASK:AUTOSET commands, affects only an autosegment on a standard mask, not the general oscilloscope autosegment function.

## Group

Mask

## Syntax

```
MASK:AUTOSet:VSCALE {ON|OFF|<NR1>}
```

## Arguments

- ON

This enables the autosegment:vscale function, causing ... to occur.

- OFF

This disables the autosegment:vscale function, causing ... to occur.

- <NR1>

A 0 disables the autosegment:vscale function; any other value enables it.

## Example

```
MASK:AUTOSet:VSCALE ON
```

Enables the autosegment:vscale function.

---

# MASK:COpy:USER

## Description

This command copies the currently displayed mask to the User mask.

## Group

Mask

## Syntax

MASK:COpy:USER

## Example

MASK:COpy:USER

Copies the current mask to the user mask.



---

# MASK:COUNT RESET

## Description

This command resets to zero the number of hits for all mask segments on the specified source.

## Group

Mask

## Syntax

```
MASK:COUNT RESET
```

## Example

```
MASK:COUNT RESET
```

Resets to zero the number of hits for all mask segments on the specified source.

---

# MASK:COUNT:FAILURES?

## Description

This command returns the number of pass/fail tests that have failed. The default is 0. To work pass/fail testing must be enabled.

## Group

Mask

## Syntax

MASK:COUNT:FAILURES?

## Returns

<NR1>

## Example

MASK:COUNT:FAILURES?

Might return 0 showing no tests have failed.

---

# MASK:COUNT:HITS?

## Description

This command returns the number of hits on all mask segments. To work, hit counting must be turned on.

## Group

Mask

## Syntax

MASK:COUNT:HITS?

## Returns

<NR1>

## Example

MASK:COUNT:HITS?

Might return: 0

Showing there were no hits from any source on all mask segments.

---

# MASK:COUNT:SEG<m>:HITS?

## Description

This command returns the number of hits in mask segment <m>. To work, hit counting must be turned on.

## Group

Mask

## Syntax

```
MASK:COUNT:SEG<m>:HITS?
```

## Argument

- <m>

Is the mask segment number.

## Returns

<NR1>

## Example

```
MASK:COUNT:SEG1:HITS?
```

Might return 0

showing there have been no hits.

---

# MASK:COUNT:STATE

## Description

Sets or returns the mask hits count state. In other words, it controls whether or not mask counting is being done. MASK:DISplay must be ON to enable MASK:COUNT:STATE to count mask violations.

## Group

Mask

## Syntax

```
MASK:COUNT:STATE {OFF|ON|<NR1>}
```

## Arguments

- ON

Turns on mask counting.

- OFF

Turns off mask counting. This is the default state.

- <NR1>

A 0 turns off mask hit counting, and other values turn on mask hit counting.

## Example

```
MASK:COUNT:STATE ON
```

Turns on mask hit counting

---

# MASK:COUNT:TESTS?

## Description

Returns the number of pass/fail tests that have occurred.

## Group

Mask

## Syntax

MASK:COUNT:TESTS?

## Returns

<NR1>

## Example

MASK:COUNT:TESTS?

Might return: 50

showing 50 tests have occurred.

---

# MASK:COUNT:TOTAl?

## Description

Returns the sum of all hits in all mask segments. This command is the same as MASK:COUNT:HITS? And is kept for compatibility with other Tektronix oscilloscopes.

## Group

Mask

## Syntax

```
MASK:COUNT:TOTAl?
```

## Returns

<NR1>

## Example

```
MASK:COUNT:TOTAL?
```

Might return 50

Showing 50 hits have occurred.

---

# MASK:COUNT:VIOLATIONS?

## Description

This command returns the number of test violations that have occurred in the current pass/fail test. A test violation occurs when any part of a waveform falls within any mask segment. The default is 0.

## Group

Mask

## Related Commands

MASK:COUNT:TESTS?, MASK:COUNT:TOTAL?,  
MASK:COUNT:WAVEFORMS?, MASK:TEST:REPEAT

## Syntax,

MASK:COUNT:VIOLATIONS?

## Returns

<NR1>

## Example

MASK:COUNT:VIOLATIONS?

Might return: 50

Showing 50 test violations have occurred.



---

# MASK:COUNT:WAVEFORMS?

## Description

This command returns the number of waveforms that have been acquired and processed during pass/fail testing.

## Group

Mask

## Related Commands

MASK:COUNT:TESTS?, MASK:COUNT:TOTAL?,  
MASK:COUNT:VIOLATIONS?

## Syntax

MASK:COUNT:WAVEFORMS?

## Returns

<NR1>

## Example

MASK:COUNT:WAVEFORMS?

Might return: 1000

Showing 1,000 waveforms have been acquired and processed.

---

# MASK:DISplay

## Description

Sets or returns whether or not defined masks are displayed on the screen. This is useful for temporarily turning off user defined masks without deleting them. It is also useful for removing a standard mask from the screen, but leaving it as the selected standard. Mask counting, mask testing, and mask autoselect are not available if the mask display is Off. The default is On.

## Group

Mask

## Syntax 1

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

## Syntax 2

```
MASK:DISplay?
```

## Arguments

- ON

Shows the masks on the display. This is the default value.

- OFF

Removes the masks from the display

- <NR1>

A 0 removes the masks from the display; any other value shows the masks on the display.

## Example

```
MASK:DISPLAY ON
```

Sets the display to show the defined masks

## Example 2

```
MASK:DISPLAY?
```

Might return 1, indicating that the display shows masks

---

# MASK:FILTER

## Description

This command sets or returns whether the selected digital filter will be run on the waveform data. The filter simulates optical hardware. That is, it simulates different hardware for each of several different optical standards.

The digital filter runs on OC1, OC3, OC12, OC48, FC133, FC266, FC531, FC1063, FC2125Draft, Gigabit Ethernet, Infiniband 2.5Gb, 1394b, 393Mb, 786.4 3Mb, 1.572 Gb

## Group

Mask

## Related Commands

ACQUIRE:MODE

## Syntax

```
MASK:FILTER {OFF|ON|<NR1>}
```

## Arguments

- ON

This enables the digital filter.

- OFF

This disables the digital filter.

- <NR1>

A 0 disables the digital filter; any other value enables it.

## Example

```
MASK:FILTER ON
```

Enables the digital filter.

---

# MASK:HIGHLIGHTHits

## Description

This command sets or returns whether or not hits in a mask are highlighted in different colors than other waveform data. The default is On.

## Group

Mask

## Syntax 1

```
MASK:HIGHLIGHTHits {OFF|ON|<NR1>}
```

## Syntax 2

```
MASK:HIGHLIGHTHits?
```

## Arguments

- ON

This enables the mask:highlighthis function.

- OFF

This disables the mask:highlighthis function.

- <NR1>

A 0 disables the mask:highlighthis function; any other value enables it.

## Example

```
MASK:HIGHLIGHTHITS ON
```

Enables the mask highlighthis function.

---

# MASK:INVert

## Description

This command controls whether the mask is drawn inverted. It has no effect if this mask cannot be inverted. The default is Off (Positive).

## Group

Mask

## Syntax

```
MASK:INVert {OFF|ON|<NR1>}
```

## Arguments

- ON

Negative

- OFF

Positive

- <NR1>

A 0 disables the mask:invert function; any other value enables it.

## Example

```
MASK:INVERT ON
```

Enables the mask invert function.

---

# MASK:LOCK

## Description

This command sets or returns the mask lock to waveform mode. It locks the mask to the waveform such that changes to the horizontal and/or vertical scale settings redraws the mask segments in proportion. This feature allows for expanding horizontal and/or vertical settings to zoom in on waveforms and masks to visually examine violation areas in more detail.

## Group

Mask

## Syntax 1

```
MASK:LOCK {OFF|ON|<NR1>}
```

## Syntax 2

```
MASK:LOCK?
```

## Arguments

- ON

turns on mask lock to waveform.

- OFF

turns off mask lock to waveform.

- <NR1>

A 0 turns off mask lock to waveform and any other value turns on mask lock to waveform.

## Example

```
MASK:LOCK ON
```

Turns on mask lock to waveform

---

# MASK:MARgin:PERCent

## Description

This command sets or queries the mask margin percentage. If you set a positive mask margin percentage, the polygon expands and the mask test will be more difficult to pass due to the expanded test area. If you set a negative mask margin percentage, the polygon shrinks and the mask test will be easier to pass due to the smaller area.

## Group

Mask

## Syntax 1

```
MASK:MARgin:PERCent <NR3>
```

## Syntax 2

```
MASK:MARgin:PERCent?
```

## Argument

- <NR3>

This values range from -50.0 to +50.0. The default is 5.

## Example 1

```
MASK:MARGIN:PERCENT 25
```

Sets the mask margin percentage to 25.

## Example 2

```
MASK:MARGIN:PERCENT?
```

Might return

```
:MASK:MARGIN:PERCENT 1.0000000000E+001,
```

indicating that the mask margin is set to 10%.

---

# MASK:MARgin:STATE

## Description

This command sets or returns the state of the mask margins.

## Group

Mask

## Syntax

```
MASK:MARgin:STATE {OFF|ON|<NR1>}
```

## Arguments

- ON

Turns on mask margins.

- OFF

Turns off mask margins. The currently displayed margined mask is erased and the original mask is displayed.

- <NR1>

A 0 turns off mask margins; any other integer turns on the selected mask margins.

## Example 1

```
MASK:MARGIN:STATE ON
```

Turns on mask margins

## Example 2

```
MASK:MARGIN:STATE?
```

Might return

```
:MASK:MARGIN:STATE 1
```

indicating that mask margins are on.



---

# MASK:MASKPRE:AMPlitude

## Description

This command sets or returns the current mask's nominal pulse amplitude in volts. The query form of this command returns the nominal pulse amplitude of the displayed mask. The set form of this command only affects the current (displayed) mask.

## Group

Mask

## Related Commands

MASK:MASKPRE:VSCALE

## Syntax 1

MASK:MASKPRE:AMPlitude <NR3>

## Syntax 2

MASK:MASKPRE:AMPlitude?

## Argument

- <NR3>

This is a floating number that sets the nominal pulse amplitude in volts

## Example

MASK:MASKPRE:AMPLITUDE 2.4

Sets the current mask nominal pulse amplitude to 2.4 volts.

---

# MASK:MASKPRE:HSCALE

## Description

This command sets or returns the nominal timing resolution used to draw the mask, in time/division. The query form of this command returns the nominal timing resolution of the displayed mask. The set form of this command only affects the current (displayed) mask.

## Group

Mask

## Related Commands

MASK:MASKPRE:HTRIGPOS

## Syntax 1

MASK:MASKPRE:HSCALE <NR3>

## Syntax 2

MASK:MASKPRE:HSCALE?

## Argument

- <NR3>

This is a floating point number that sets the mask drawing timing resolution.

## Example

MASK:MASKPRE:HSCALE 40e-9

Sets the nominal timing resolution used to draw the mask pulse shape to 40 nanoseconds per division.

---

# MASK:MASKPRE:HTRIGPOS

## Description

This command sets or returns the nominal trigger position.(pulse leading edge) used to draw the mask as a fraction of the display width. The query form of this command returns the nominal trigger position of the displayed mask. The set form of this command only affects the current (displayed) mask.

## Group

Mask

## Related Commands

MASK:MASKPRE:HSCALE?

## Syntax 1

MASK:MASKPRE:HTRIGPOS <NR3>

## Syntax 2

MASK:MASKPRE:HTRIGPOS?

## Argument

- <NR3>

Is a floating point number in the range of 0.0 to 1.0 that sets the trigger points as a fraction of the display width, referenced from the left edge of the graticule.

0.0 represents the left edge.

## Example

MASK:MASKPRE:HTRIGPOS 0.6

Sets the current mask nominal pulse leading edge position to the 6<sup>th</sup> horizontal graticule division.

---

# MASK:MASKPRE:PATTERNBITS

## Description

This command sets or returns the number of bits used for serial trigger for the User mask standard. For example, DS1 requires six bits,; four leading zeros, a one, and a trailing zero. The query form of this command returns the serial bit value of the displayed mask. The set form of this command only affects the current (displayed) mask.

## Group

Mask

## Related Commands

MASK:MASKPRE:PRESAMPBITS

## Syntax 1

MASK:MASKPRE:PATTERNBITS <NR1>

## Syntax 2

MASK:MASKPRE:PATTERNBITS?

## Argument

- <NR1>

Is an integer that sets the number of bits.

## Example

MASK:MASKPRE:PATTERNBITS 6

Sets the current mask number of serial pattern bits to 6.

---

# MASK:MASKPRE:PRESAMPBITS

## Description

This command sets or returns the number of bits before the (isolated one) pulse leading edge in the serial trigger pass/fail testing. For example, DS1 has four leading zeros. The query form of this command returns the presample bit value of the displayed mask. The set form of this command only affects the current (displayed) mask.

## Group

Mask

## Related Commands

MASK:MASKPRE:PATTERNBITS

## Syntax 1

MASK:MASKPRE:PRESAMPBITS <NR1>

## Syntax 2

MASK:MASKPRE:PRESAMPBITS?

## Argument

- <NR1>

Is an integer that sets the number of bits before the trigger pulse.

## Example

MASK:MASKPRE:PRESAMPBITS 4

Sets the current mask number of bits before the trigger pulse to 4.

---

# MASK:MASKPRE:RECOrdlength

## Description

This command sets or returns the nominal record length for pulse mask testing. The query form of this command returns the record length value of the displayed mask. The set form of this command only affects the current (displayed) mask.

## Group

Mask

## Syntax 1

```
MASK:MASKPRE:RECOrdlength <NR1>
```

## Syntax 2

```
MASK:MASKPRE:RECOrdlength?
```

## Argument

- <NR1>

Is an integer number that sets the record length.

## Example

```
MASK:MASKPRE:RECORDLENGTH 500
```

Sets the current mask record length to 500 points.

---

# MASK:MASKPRE:SERIALTRIG

## Description

This command sets or returns the type of triggering used in pass/fail testing of the current mask.

## Group

Mask

## Related Commands

MASK:MASKPRE:PATTERNBITS, MASK:MASKPRE:PRESAMP-BITS

## Syntax 1

```
MASK:MASKPRE:SERIALTRIG  
{AMI | HDB3 | B3ZS | B6ZS | B8ZS | CMI | NRZ | MLT3 | EDGE}
```

## Syntax 2

```
MASK:MASKPRE:SERIALTRIG?
```

## Argument

- AMI

Alternate Mark Inversion

- HDB3

High-Density Bipolar Three-Bit substitution

- B3ZS

Bipolar 3 Zero Substitution

- B6ZS

Bipolar 6 Zero Substitution

- B8ZS

Bipolar 8 Zero Substitution

- CMI

Coded Mark Inversion

- NRZ

Non-Return to Zero

- MLT3

Multi-Level Transition

## Example

```
MASK:MASKPRE:SERIALTRIG AMI
```

Sets AMI as the type of triggering used in testing the current mask.

---

# MASK:MASKPRE:TRIGTOSAMP

## Description

This command sets or returns the time, in seconds, from the (leading edge) trigger position to the pulse bit sampling position. The query form of this command returns the time value of the displayed mask. The set form of this command only affects the current (displayed) mask.

## Group

Mask

## Related Commands

MASK:MASKPRE:WIDTH

## Syntax 1

MASK:MASKPRE:TRIGTOSAMP <NR3>

## Syntax 2

MASK:MASKPRE:TRIGTOSAMP?

## Argument

- <NR3>

Is the floating point number that sets the time to the pulse bit sampling position.

## Example

MASK:MASKPRE:TRIGTOSAMP 162e-9

Sets the current mask time from the (leading edge) trigger position to the pulse bit sampling position to 162 nanoseconds.



---

# MASK:MASKPRE:VOFFSet

## Description

This command sets or returns the nominal vertical offset, in volts, used to vertically offset the input channels. The query form of this command returns the offset value of the displayed mask. The set form of this command only affects the current (displayed) mask.

## Group

Mask

## Related Commands

MASK:MASKPRE:VPOS , MASK:MASKPRE:VSCALE

## Syntax 1

MASK:MASKPRE:VOFFSet <NR3>

## Syntax 2

MASK:MASKPRE:VOFFSet?

## Argument

- <NR3>

Is a floating point number that sets the nominal vertical offset in volts.

## Example

MASK:MASKPRE:VOFFSET 0.0

Sets the current mask nominal vertical offset to 0.0 volts.

---

# MASK:MASKPRE:VPOS

## Description

This command sets or returns the nominal vertical position, in divisions, used to vertically position the input channels. The query form of this command returns the vertical position value of the displayed mask. The set form of this command only affects the current (displayed) mask.

## Group

Mask

## Related Commands

MASK:MASKPRE:VOFFSet, MASK:MASKPRE:VSCALE

## Syntax 1

MASK:MASKPRE:VPOS <NR3>

## Syntax 2

MASK:MASKPRE:VPOS?

## Argument

- <NR3>

Is a floating point number that sets the nominal vertical position in divisions.

## Example

MASK:MASKPRE:VPOS -0.8

Sets the current mask nominal vertical position to -0.8 divisions.

---

# MASK:MASKPRE:VSCALE

## Description

This command sets or returns the nominal vertical scale, in volts per division, used to vertically scale the input channels. The query form of this command returns the vertical scale value of the displayed mask. The set form of this command only affects the current (displayed) mask.

## Group

Mask

## Related Commands

MASK:MASKPRE:VOFFSet, MASK:MASKPRE:VPOS

## Syntax 1

MASK:MASKPRE:VSCALE <NR3>

## Syntax 2

MASK:MASKPRE:VSCALE?

## Argument

- <NR3>

Is a floating point number that sets the nominal bit width in seconds.

## Example

MASK:MASKPRE:VSCALE 200e-3

Sets the current mask nominal vertical scale to 200 millivolts per division

---

# MASK:MASKPRE:WIDTH

## Description

This command sets or returns the nominal bit width in seconds. The query form of this command returns the bit width value of the displayed mask. The set form of this command only affects the current (displayed) mask.

## Group

Mask

## Related Commands

MASK:MASKPRE:TRIGTOSAMP

## Syntax 1

MASK:MASKPRE:WIDTH <NR3>

## Syntax 2

MASK:MASKPRE:WIDTH?

## Argument

- <NR3>

Is a floating point number that sets the nominal bit width in seconds.

This is the time of one bit of data where  $\text{bit width} = 1 / (\text{data rate of the signal})$ .

## Example

MASK:MASKPRE:WIDTH 648e-9

Sets the current mask nominal bit width to 648 nanoseconds (1.544 Mbits/second).

---

# MASK:POLarity

## Description

This command sets or returns the input waveform polarity for the pass/fail test. It controls whether to test positive pulse, negative pulse, or both during pass/fail testing. This command only applies when `MASK:TEST:STATE` is on.

## Group

Mask

## Related Commands

`MASK:TEST:STATE`

## Syntax 1

```
MASK:POLarity {BOTH|NEGAtive|POSITIVe}
```

## Syntax 2

```
MASK:POLarity?
```

## Arguments

- `BOTH`

This enables testing for both positive and negative pulses. The oscilloscope tests positive pulses on the `mask:source` waveform until  $\frac{1}{2}$  of the waveform has been tested. Then the oscilloscope inverts the mask and performs the remaining tests.

- `NEGAtive`

This enables testing on negative pulses

- `POSITIVe`

This enables testing on positive pulses. This is the default.

## Example

```
MASK:POLARITY BOTH
```

Sets pass/fail to test both positive (normal) and negative (inverted) waveforms.

---

# MASK:SEG<m> DELEte

## Description

This command deletes the specified mask segment from the Current mask.

## Group

Mask

## Syntax

```
MASK:SEG<m> DELEte
```

## Arguments

- <m>

Is an integer that specifies the mask segment number to delete from the Current mask.

## Example

```
MASK:SEG3 DELETE
```

Deletes mask segment three from the Current mask.

---

## **MASK:SEG<m>:NR\_Pt?**

### **Description**

This command returns the number of points that make up the specified mask segment of the Current mask. Each mask point consists of an X-Y pair of coordinates.

### **Group**

Mask

### **Syntax**

`MASK:SEG<m>:NR_Pt?`

### **Arguments**

- `<m>`

Is an integer number that specifies a mask segment number of the Current mask.

### **Returns**

`<NR1>` is 0 to 50.

### **Example**

`MASK:SEG2:NR_PT?`

Might return `MASK:SEG2:NR_PT 5`

---

# MASK:SEG<m>:POINTS

## Description

This command sets or returns the X-Y user coordinates of all points in the Current mask segment. The set form defines new points in the Current mask, replacing any existing points in the Current mask segment. In other words, it sets or returns the vertices for a particular segment in the Current mask.

## Group

Mask

## Syntax 1

```
MASK:SEG<m>:POINTS <NR3>, <NR3> [ , <NR3>,  
<NR3> ]
```

## Syntax 2

```
MASK:SEG<m>:POINTS?
```

## Arguments

- <m>

Is an integer that specifies the Current mask segment number.

- <NR3>

Refers to the coordinates of one of the vertices in the Current mask. Each pair of numbers is the horizontal and vertical coordinates of a mask segment vertex. The pairs must be listed in a counterclockwise order. If the vertical or horizontal scale or position is changed after this command and then the query form of this command is issued, then the value returned from the oscilloscope will not be the same. If just one pair is input then they are ignored and the Current mask segment is marked as undefined. The default is no points in the Current mask segment.

## Example

```
MASK:SEG2:POINTS -2.3e-9, 44e-3, -2.5e-9,  
47e-3, 1.2e-9, 44e-3
```

Defines the points in Current mask segment 2.



---

# MASK:SOURce

## Description

This command sets or reports which source will be compared against the mask(s) when counting is turned on. In other words, it controls which trace to use in mask counting. It also affects mask autoselect and how triggering is set up when you select the mask.

## Group

Mask

## Syntax 1

```
MASK:SOURce {CH[1-4] | MATH[1-4] | REF[1-4]}
```

## Syntax 2

```
MASK:SOURce?
```

## Arguments

- CH<1-4>

Selects a channel waveform to be compared against the specified mask. The range for is 1 through 4.

- MATH<1-4>

Selects a math waveform to be compared against the specified mask. The range for is 1 through 4.

- REF<1-4>

Selects a reference waveform to be compared against the specified mask. The range is 1 through 4.

## Example 1

```
MASK:SOURCE CH1
```

Enables CH1 to be compared against the mask. Since a timebase is not specified, the default, Main, is used.

## Example 2

```
MASK:SOURCE?
```

Might return

```
:MASK:SOURCE CH1,MAIN
```

indicated that the Channel 1 waveform will be compared against the mask using the Main timebase.

---

# MASK:STANdard

## (TDS5000B Series Option SM)

### Description

This command deletes the existing mask (if any) and sets the selected standard mask. If MASK:COUNT:STATE is ON, mask counting starts immediately. The query form of this command returns the current mask standard.

**Note:** There are several available mask standards that require a higher bandwidth than permitted for TDS5000B Series instruments. In such cases, a warning is posted to the standard event system upon selection of any of these standards. To ensure that the mask you select is fully supported by the instrument, always check the event status register after selecting a mask standard.

The following warning event is posted if the mask exceeds the instrument bandwidth: 2318, "Consider system bandwidth when testing at this bit rate."

### Group

Mask

### Related Commands

ACQuire:MODE

### Syntax 1

```
MASK:STANdard
{ ATARXG1 | ATARXG2 | ATARXG3 | ATATXG1 | ATATXG2 |
ATATXG3 | CLOCKCoax | CLOCKSymmet-
rical | D1 | D2 | DS0Contra | DS0Double |
DS0Single | DS0Timing | DS1 | DS1A | DS1C | DS2 | DS2RATE-
Coax |
DS2RATESymmet-
rical | DS3 | DS4NA | DS4NA_Max | E1Coax | E1Symmetri-
cal |
E2 | E3 | E4_1 | E4_0 | ENET1000BCX_ATP2 | ENET1000BCX_A
TP3 |
ENET1000BCX_NTP2 | ENET100FX | ENET100STP | ENET100U
TP | ENET1250 |
ENETXAUI_Far | ENET-
XAUI_Near | FC1063 | FC1063Draft | FC1063E |
FC1063E_ABR | FC1063E_ABT | FC1063E_ADR | FC1063E_AD
T | FC1063E_AGR |
FC1063E_AGT | FC1063E_NBT | FC1063E_NDT | FC1063E_NG
T | FC133 |
FC133E | FC2125 | FC2125E_ABR | FC2125E_ABT | FC2125E_
ADR | FC2125E_ADT |
FC2125E_AGR | FC2125E_AGT | FC2125E_NBT | FC2125E_ND
T | FC2125E_NGT |
FC266 | FC266E | FC4250E_ABR | FC4250E_ABT
```

```

| FC4250E_ADR | FC4250E_ADT | FC4250E_AGR
| FC4250E_AGT | FC4250E_NBT | FC4250E_NDT
| FC4250E_NGT | FC531 | FC531E | FST1 | FST2 | FST3
| FST4 | FST5 | FST6 | FW1394BS1600B |
FW1394BS1600BT1 | FW1394BS1600BT2 | FW1394BS400B | F
W1394BS400BT1 |
FW1394BS400BT2 | FW1394BS800B | FW1394BS800BT1 | FW1
394BS800BT2 |
G703DS1 | G703DS3 | HST1 | HST2 | HST3 | HST4 | HST5 | HST6 |
INF2_5G |
INF2_5GE | NONE | OC1 | OC12 | OC3 | OC48 | OC48_FEC | PCIEX-
PRESS_Rcv |
PCIEX-
PRESS_Xmit | RATE32Mbit | RATE97Mbit | RIO_DRV1G | RIO
_DRV1_5G |
RIO_DRV2G | RIO_DRV500M | RIO_DRV500M | RIO_DRV750M |
RIO_EDRV1G |
RIO_EDRV1_5G | RIO_EDRV2G | RIO_EDRV500M | RIO_EDRV5
00M |
RIO_EDRV750M | RIO_RCV1G | RIO_RCV1_5G | RIO_RCV2G | R
IO_RCV500M |
RIO_RCV500M | RIO_RCV750M | RIO_SERIAL_1G | RIO_SE-
RIAL_2G | RIO_SERIAL_3G |
SFI5_RCVBCLK2 | SFI5_RCVBCLK3 | SFI5_RCVBDA-
TA2 | SFI5_RCVBDATA3 |
SFI5_RCVDCLK2 | SFI5_RCVDCLK3 | SFI5_RCVDDA-
TA2 | SFI5_RCVDDATA3 |
SFI5_XMITACLK2 | SFI5_XMITACLK3 | SFI5_XMITADA-
TA2 | SFI5_XMITADATA3 |
SFI5_XMITCCLK2 | SFI5_XMITCCLK3 | SFI5_XMITCDA-
TA2 | SFI5_XMITCDATA3 |
STM0_0 | STM0_1 | STM0_HDBX | STM1E_1 | STM1E_0 | STS1Ey
e | STS1Pulse |
STS3 | STS3_Max | TFI5_2 | TFI5_3 | USERMask | VID-
EO270 | VIDEO360 | VSROC192}

```

## Syntax 2

MASK:STANdard?

### Arguments

- ATARXG1 (Serial ATA, G1 Rx 1.5 Gb/s)
- ATARXG2 (Serial ATA, G2, Rx, 3.0 Gb/s)
- ATARXG3 (Serial ATA, G3, Rx)
- ATATXG1 (Serial ATA, G1 Tx, 1.5 Gb/s)
- ATATXG2 (Serial ATA, G2 Tx, 3.0 Gb/s)
- ATATXG3 (Serial ATA, G3 Tx)
- CLOCKCoax
- CLOCKSymmetrical
- D1
- D2

- 
- DS0Contra
  - DS0Double
  - DS0Single
  - DS0Timing
  - DS1 (ANSI T1.102-1993 (R1999), DS1, 1.544 Mb/s)
  - DS1A (ANSI T1.102-1993 (R1999), DS1A, 2.048 Mb/s)
  - DS1C (ANSI T1.102-1993 (R1999), DS1C, 3.152 Mb/s)
  - DS2 (ANSI T1.102-1993 (R1999), DS2, 6.312 Mb/s)
  - DS2RATECoax (ITU-T, G703 (10/98), D2 Rate Coax, 6.312 Mb/s)
  - DS2RATESymmetrical (ITU-T, G703 (10/98), D2 Rate Sym, 6.312 Mb/s)
  - DS3 (ANSI T1.102-1993 (R1999), DS3, 44.736 Mb/s)
  - DS4NA (ANSI T1.102-1993 (R1999), DS4NA, 139.26 Mb/s)
  - DS4NA\_Max (ANSI T1.102-1993 (R1999), DSNA Max Output, 139.26 Mb)
  - E1Coax (ITU-T, G703 (10/98), E1 Coax Pair, 2.048 Mb/s)
  - E1Symmetrical (ITU-T, G703 (10/98), E1 Sym Pair, 2.048 Mb/s)
  - E2 (ITU-T, G703 (10/98), E2, 8.448 Mb/s)
  - E3 (ITU-T, G703 (10/98), E3, 34.368 Mb/s)
  - E4\_0 (ITU-T, G703 (10/98), E4 Binary 0)
  - E4\_1 (ITU-T, G703 (10/98), E4 Binary 1)
  - ENET100FX
  - ENET100STP (IEEE Std 802.3 and ANSI X3.263-1995, 100 Base-Tx, STP, 125 Mb/s )
  - ENET100UTP (IEEE Std 802.3 and ANSI X3.263-1995, 100 Base-Tx, UTP, 125 Mb/s)
  - ENET1000BCX\_ATP2 (1000B-CX Abs, TP2, 1.25 Gb/s)
  - ENET1000BCX\_ATP3 (1000B-CX Abs, TP3, 1.25 Gb/s)
  - ENET1000BCX\_NTP2 (1000B-CX Norm, TP2, 1.25 Gb/s)

- 
- ENET1250 (IEEE Std 802.3 and ANSI X3.263-1995, GB Ethernet, 1.25 Gb/s)
  - ENETXAUI\_Far (10 Gigabit Attachment Unit Interface (XAUI), Far, 3.125 Gb/s)
  - ENETXAUI\_Near (10 Gigabit Attachment Unit Interface (XAUI), Near, 3,125 Gb/s)
  - FC133 (ANSI X3.230-1999 NCITS 1235D/Rev 11, Optical, 132.8 Mb/s)
  - FC133E (ANSI X3.230-1999 NCITS 1235D/Rev 11, Electrical 132.8 Mb/s)
  - FC266 (ANSI X3.230-1999 NCITS 1235D/Rev 11, Optical, 265.6 Mb/s)
  - FC266E (ANSI X3.230-1999 NCITS 1235D/Rev 11, Electrical, 132.8 Mb/s)
  - FC531 (ANSI X3.230-1999 NCITS 1235D/Rev 11, Optical, 531.2 Mb/s)
  - FC531E (ANSI X3.230-1999 NCITS 1235D/Rev 11, Electrical, 531.2 Mb/s)
  - FC1063 (ANSI X3.230-1999 NCITS 1235D/Rev 11, Optical, 1.065 Gb/s)
  - FC1063E (ANSI X3.230-1999 NCITS 1235D/Rev 11, Electrical, 1.0625 Gb/s)
  - FC1063E\_ABT (ANSI X3.230-1999 NCITS 1235D/Rev 11, Abs, Beta, Transm)
  - FC1063E\_ADT (ANSI X3.230-1999 NCITS 1235D/Rev 11, Abs, Beta, Transm)
  - FC1063E\_AGT (ANSI X3.230-1999 NCITS 1235D/Rev 11, Abs, Gamma, Transm)
  - FC1063E\_NBT (ANSI X3.230-1999 NCITS 1235D/Rev 11, Norm, Beta, Transm)
  - FC1063E\_NDT (ANSI X3.230-1999 NCITS 1235D/Rev 11, Norm, Delta, Transm)
  - FC1063E\_NGT (ANSI X3.230-1999 NCITS 1235D/Rev 11, Norm, Gamma, Transm)
  - FC1063E\_ABR (ANSI X3.230-1999 NCITS 1235D/Rev 11, Abs, Beta, Recv)
  - FC1063E\_ADR (ANSI X3.230-1999 NCITS 1235D/Rev 11, Abs, Delta, Recv)
  - FC1063E\_AGR (ANSI X3.230-1999 NCITS 1235D/Rev 11, Abs, Gamma, Recv)
  - FC1063Draft (ANSI X3.230-1999 NCITS 1235D/Rev 11, Optical, Draft Rev 11)

- 
- FC2125 (ANSI X3.230-1999 NCITS 1235D/Rev 11, Optical, 2.125)
  - FC2125E\_ABT (ANSI X3.230-1999 NCITS 1235D/Rev 11, Abs, Beta, Transm)
  - FC2125E\_ADT (ANSI X3.230-1999 NCITS 1235D/Rev 11, Abs, Delta, Transm)
  - FC2125E\_AGT (ANSI X3.230-1999 NCITS 1235D/Rev 11, Abs, Gamma, Transm)
  - FC2125E\_NBT (ANSI X3.230-1999 NCITS 1235D/Rev 11, Norm, Beta, Transm)
  - FC2125E\_NDT (ANSI X3.230-1999 NCITS 1235D/Rev 11, Norm, Delta, Transm)
  - FC2125E\_NGT (ANSI X3.230-1999 NCITS 1235D/Rev 11, Norm, Gamma, Transm)
  - FC2125E\_ABR (ANSI X3.230-1999 NCITS 1235D/Rev 11, Abs, Beta, Recv)
  - FC2125E\_ADR (ANSI X3.230-1999 NCITS 1235D/Rev 11, Abs, Delta, Recv)
  - FC2125E\_AGR (ANSI X3.230-1999 NCITS 1235D/Rev 11, Abs, Gamma, Recv)
  - FC4250E\_AB T
  - FC4250E\_ A D T
  - FC4250E\_ A G T
  - FC4250E\_ N B T
  - FC4250E\_ N D T
  - FC4250E\_ N G T
  - FC4250E\_ A B R
  - FC4250E\_ A D R
  - FC4250E\_ A G R
  - FST1 (USB, FS:T1, 12 Mb/s)
  - FST2 (USB, FS:T2, 12 Mb/s)
  - FST3 (USB, FS:T3, 12 Mb/s)
  - FST4 (USB, FS:T4: 12 Mb/s)
  - FST5 (USB, FS:T5, 12 Mb/s)
  - FST6 (USB, FS:T6, 12 Mb/s)
  - FW1394BS400B (IEEE 1394b, S400 Optical, 491.5 Mb/s)
  - FW1394BS400BT1 (IEEE 1394b, S400b T1, 491.5 Mb/s)

- 
- FW1394BS400BT2 (IEEE 1394b, S400b T2, 491.5 Mb/s)
  - FW1394BS800B (IEEE 1394b, S800 Optical, 988.0 Mb/s)
  - FW1394BS800BT1 (IEEE 1394b, S800b T1, 983.0 Mb/s)
  - FW1394BS800BT2 (IEEE 1394b, S800b T2, 983.0 Mb/s)
  - FW1394BS1600B (IEEE 1394b, S1600 Optical, 1.966 Gb/s)
  - FW1394BS1600BT1 (IEEE 1394b, S1600b T1, 1.966 Gb/s)
  - FW1394BS1600BT2 (IEEE 1394b, S1600b T2, 1.966 Gb/s)
  - G703D1 (ITU-T, G703 (10/98), DS1 Rate, 1.544 Mb/s)
  - G703DS3 (ITU-T, G703 (10/98))
  - HST1 (USB, HS:T1, 480 Mb/s) G703DS3 (ITU-T, G703 (10/98))
  - HST2 (USB, HS:T2, 480 Mb/s) G703DS3 (ITU-T, G703 (10/98))
  - HST3 (USB, HS:T3, 480 Mb/s)
  - HST4 (USB, HS:T4, 480 Mb/s)
  - HST5 (USB, HS:T5, 480 Mb/s)
  - HST6 (USB, HS:T6, 480 Mb/s)
  - INF2\_5G (InfiniBand, IBTA Spec 1.0a, 2.5 Optical, 2.5 Gb/s)
  - INF2\_5GE (InfiniBand, IBTA Spec 1.0a, 2.5 Electrical, 2.5 Gb/s)
  - NONE
  - OC1 (GR 253-CORE Issue 3 9/21/2000 OC1/STM0, 51.84 Mb/s)
  - OC3 (GR 253-CORE Issue 3 9/21/2000 OC1/STM1, 155.52, Mb/s)
  - OC12 (GR 253-CORE Issue 3 9/21/2000 OC1/STM4, 622.08 Mb/s)
  - OC48 (GR 253-CORE Issue 3 9/21/2000 OC1/STM16, 2.4883 Gb/s)
  - OC48\_FEC (Forward Error Correction - CSA8000 mask, 2.666 Gb/s)
  - PCIEXPRESS\_Xmit (PCI Express Transmitter, 2.5 Gb/s)

- 
- **PCIEXPRESS\_Rcv** (PCI Express Receiver, 2.5 Gb/s)
  - **RATE32Mbit** (ITU-T, G703 (10/98), 32.064 Mb/s)
  - **RATE97Mbit** (ITU-T, G703 (10/98), 97 Mbit, 97.728 Mb/s)
  - **RIO\_DRV1G** (Rapid IO Driver, 1 Gb/s)
  - **RIO\_DRV1\_5G** (Rapid IO Driver, 5 Gb/s)
  - **RIO\_DRV2G** (Rapid IO Driver, 2 Gb/s)
  - **RIO\_DRV500M** (Rapid IO Driver, 500 Mb/s)
  - **RIO\_DRV750M** (Rapid IO Driver, 750 Mb/s)
  - **RIO\_EDRV1G** (Rapid IO Extended Driver, 1 Gb/s)
  - **RIO\_EDRV1\_5G** (Rapid IO Extended Driver, 1.5 Gb/s)
  - **RIO\_EDRV2G** (Rapid IO Extended Driver, 2 Gb/s)
  - **RIO\_EDRV500M** (Rapid IO Extended Driver, 500 Mb/s)
  - **RIO\_EDRV750M** (Rapid IO Extended Driver, 750 Mb/s)
  - **RIO\_RCV500M** (Rapid IO Receiver, 500 Mb/s)
  - **RIO\_RCV750M** (Rapid IO Receiver, 750 Mb/s)
  - **RIO\_RCV1G** (Rapid IO Receiver, 1 Gb/s)
  - **RIO\_RCV1\_5G** (Rapid IO Receiver, 1.5 Gb/s)
  - **RIO\_RCV2G** (Rapid IO Receiver, 2 Gb/s)
  - **RIO\_SERIAL\_1G** (Rapid IO Serial, 1.25 Gb/s)
  - **RIO\_SERIAL\_2G** (Rapid IO Serial, 2.5 Gb/s)
  - **RIO\_SERIAL\_3G** (Rapid IO Serial, 3.25 Gb/s)
  - **SFI5\_XMITADATA2** (SFI15 Transmit: Test Point A Data Signal 2, 2.488 Gb/s)
  - **SFI5\_XMITCDATA2** (SFI15 Transmit: Test Point C Data Signal 2, 2.488 Gb/s)
  - **SFI5\_XMITACLK2** (SFI15 Transmit: Test Point A Clock Signal 2, 2.488 Gb/s)
  - **SFI5\_XMITCCLK2** (SFI15 Transmit: Test Point C Clock Signal 2, 2.488 Gb/s)
  - **SFI5\_RCVBDATA2** (SFI15 Receive: Test Point B Data Signal 2, 2.488 Gb/s)
  - **SFI5\_RCVDDATA2** (SFI15 Receive: Test Point D Data Signal 2, 2.488 Gb/s)
  - **SFI5\_RCVBCLK2** (SFI15 Receive: Test Point B Clock Signal 2, 2.488 Gb/s)



- 
- SFI5\_RCVCLK2 (SFI15 Receive: Test Point D Clock Signal 2, 2.488 Gb/s)
  - SFI5\_XMITADATA3 (SFI15 Transmit: Test Point A Data Signal 3, 3.125 Gb/s)
  - SFI5\_XMITCDATA3 (SFI15 Transmit: Test Point C Data Signal 3, 3.125 Gb/s)
  - SFI5\_XMITACLK3 (SFI15 Transmit: Test Point A Clock Signal 3, 3.125 Gb/s)
  - SFI5\_XMITCCLK3 (SFI15 Transmit: Test Point C Clock Signal 3, 3.125 Gb/s)
  - SFI5\_RCVBDATA3 (SFI15 Receive: Test Point B Data Signal 3, 3.125 Gb/s)
  - SFI5\_RCVDDATA3 (SFI15 Receive: Test Point D Data Signal 3, 3.125 Gb/s)
  - SFI5\_RCVBCLK3 (SFI15 Receive: Test Point B Clock Signal 3, 3.125 Gb/s)
  - SFI5\_RCVCLK3 (SFI15 Receive: Test Point D Clock Signal 3, 3.125 Gb/s)
  - STM0\_0 (ITU-T, G703 (10/98), STM1E Binary 0)
  - STM0\_1 (ITU-T, G703 (10/98), STM1E Binary 1)
  - STM0\_HDBX
  - STS1Eye (ANSI T1.102-1993 (R1999), STS-1 Eye, 51.84 Mb/s)
  - STS1Pulse (ANSI T1.102-1993 (R1999), STS-1 Pulse, 51.84 Mb/s)
  - STS3 (ANSI T1.102-1993 (R1999), STS-3, 155.52 Mb/s)
  - STS3\_Max (ANSI T1.102-1993 (R1999), STS-3 Max Output, 155.52 Mb/s)
  - TFI15\_2 (TFI-5, 2.488 Gb/s)
  - TFI15\_3 (TFI-5, 3.1104 Gb/s)
  - USERMask
  - VIDEO270
  - VIDEO360
  - VSROC192 (VSR OC192/STM64, 1.24416 Gb/s)

---

### **Example 1**

`MASK:STANDARD E1COAX`

Selects the E1 Coax standard mask.

### **Example 2**

`MASK:STANDARD?`

Might return

`:MASK:STANDARD E1COAX`

Indicating that the E1 Coax mask is selected as the current mask standard.

---

# MASK:STOPOnviolation

## Description

This command sets or returns the stop on violation setting. When enabled, this command stops waveform acquisitions on the first occurrence of a waveform violation. MASK:DISplay must be ON for the oscilloscope to detect violations. In other words, it controls whether the pass/fail test stops running when a violation is detected.

## Group

Mask

## Syntax 1

```
MASK:STOPOnviolation {OFF|ON|<NR1>}
```

## Syntax 2

```
MASK:STOPOnviolation?
```

## Arguments

- ON

Stops waveform acquisition on the first occurrence of a mask violation.

- OFF

Turns off mask test stop on violation.

- <NR1>

A 0 is an integer, which turns off stop on violation, and any other value turns on stop on violation.

## Example

```
MASK:STOPONVIOLATION ON
```

Sets the oscilloscope to stop waveform acquisition on the first occurrence of a waveform violation.

---

# MASK:TEST:AUX:COMPLetion

## Description

This command sets or returns the test:aux on pass/fail test completion mode. When enabled, this command causes the oscilloscope to send a TTL signal out the Aux:Out port when the mask pass/fail status changes to 'Failing'. In other words, this command controls whether or not a TTL signal is sent out the AuxOut port on completion of the pass/fail test.

## Group

Mask

## Syntax 1

```
MASK:TEST:AUX:COMPLetion {OFF|ON|<NR1>}
```

## Syntax 2

```
MASK:TEST:AUX:COMPLetion?
```

## Arguments

- ON

Turns on the pass/fail mask:test:aux on completion.

- OFF

Turns off the pass/fail mask:test:aux on completion.

- <NR1>

A 0 disables the mask:test:aux:completion function; any other value enables it.

## Example

```
MASK:TEST:AUX:COMPLetION ON
```

Turns on the pass/fail test on completion

---

# MASK:TEST:AUX:FAILure

## Description

This command sets or returns the test:aux status on pass/fail test failure mode. When enabled, this command causes the oscilloscope to send a TTL signal out the AuxOut port when the pass/fail status changes to 'Failing'. In other words, it sends the signal when a failure occurs during the mask pass/fail test.

## Group

Mask

## Syntax 1

```
MASK:TEST:AUX:FAILure {OFF|ON|<NR1>}
```

## Syntax 2

```
MASK:TEST:AUX:FAILure?
```

## Arguments

- ON

Turns on the pass/fail mask:test:aux on failure.

- OFF

Turns off the pass/fail mask:test:aux on failure. This is the default.

- <NR1>

A 0 disables the mask:test:aux:failure function; any other value enables it.

## Example

```
MASK:TEST:AUX:FAILURE ON
```

Turns on the pass/fail test on failure

---

# MASK:TEST:BEEP:COMPLetion

## Description

This command sets or returns the beep on pass/fail test completion mode. When enabled, this command causes the oscilloscope to emit a tone when the mask pass/fail test completes.

## Group

Mask

## Syntax 1

```
MASK:TEST:BEEP:COMPLetion {OFF|ON|<NR1>}
```

## Syntax 2

```
MASK:TEST:BEEP:COMPLetion?
```

## Arguments

- ON

Turns on the pass/fail beep on completion.

- OFF

Turns off the pass/fail beep on completion.

- <NR1>

A 0 turns off the pass/fail beep on completion, any other integer turns on the pass/fail beep on completion.

## Example

```
MASK:TEST:BEEP:COMPLETION ON
```

Turns on pas/fail beep on completion

---

# MASK:TEST:BEEP:FAILure

## Description

This command sets or returns the beep status on pass/fail test failure mode. When enabled, this command causes the oscilloscope to emit a tone when the pass/fail status changes to 'Failed'. In other words, it sounds the bell when a failure occurs during the mask pass/fail test.

## Group

Mask

## Syntax 1

```
MASK:TEST:BEEP:FAILure {OFF|ON|<NR1>}
```

## Syntax 2

```
MASK:TEST:BEEP:FAILure?
```

## Arguments

- ON

Turns on the pass/fail beep on failure.

- OFF

Turns off the pass/fail beep on failure. This is the default.

- <NR1>

A 0 turns off the pass/fail beep on failure, and any other integer turns on the pass/fail beep on failure.

## Example

```
MASK:TEST:BEEP:FAILURE OFF
```

Turns off the pass/fail beep on failure

---

# MASK:TEST:DElay

## Description

This command sets or returns the amount of time, in seconds, to wait after starting pass/fail testing to evaluate waveforms. This is useful if the test system requires 'settling' time prior to evaluating waveforms.

## Group

Mask

## Syntax 1

MASK:TEST:DElay <NR3>

## Syntax 2

MASK:TEST:DElay?

## Argument

- <NR3>

Is a floating point number that specifies the amount of time, in seconds, to delay the start of the pass/fail test.

## Example

MASK:TEST:DELAY 1

Delays waveform evaluation by 1 second after the start of pass/fail testing.



---

# MASK:TEST:HARDCopy

## Description

This command sets or returns the hard copy in pass/fail failure mode. When enabled, this command causes the oscilloscope to generate a screen hard copy, using the current oscilloscope hard copy settings, as soon as the pass/fail status changes to 'Failing'. In other words, it prints the screen to the default printer.

## Group

Mask

## Syntax 1

```
MASK:TEST:HARDCOPY {OFF|ON|<NR1>}
```

## Syntax 2

```
MASK:TEST:HARDCOPY?
```

## Arguments

- ON

Turns on generate hard copy on failure.

- OFF

Turns off generate hard copy on failure.

- <NR1>

A 0 turns off generate hard copy on failure, and any other value turns on generate hard copy on failure.

## Example

```
MASK:TEST:HARDCOPY ON
```

Turns on generate hard copy on failure.

---

# MASK:TEST:LOG:FAILure

## Description

This command sets or returns the log status on pass/fail test failure mode. When enabled, this command causes the oscilloscope to log the current date and time to a file when the pass/fail status changes to 'Failing'. The file name is determined from the current date and time. The user cannot change this name.

## Group

Mask

## Related Commands

MASK:TEST:SAVEWFM:FILENAME

## Syntax 1

```
MASK:TEST:LOG:FAILure {OFF|ON|<NR1>}
```

## Syntax 2

```
MASK:TEST:LOG:FAILure?
```

## Arguments

- ON

Turns on the pass/fail log on failure.

- OFF

Turns off the pass/fail log on failure. This is the default.

- <NR1>

A 0 turns off the pass/fail log on failure, and any other integer turns on the pass/fail log on failure.

## Example

```
MASK:TEST:LOG:FAILURE OFF
```

Turns off the pass/fail log on failure.

---

# MASK:TEST:REPeat

## Description

This command sets or returns the repeat pass/fail test on completion mode. When enabled, this command repeats the pass/fail test cycle at the completion of the previous test cycle.

## Group

Mask

## Related Commands

MASK:COUNT:VIOLATIONS?

## Syntax 1

```
MASK:TEST:REPeat {OFF|ON|<NR1>}
```

## Syntax 2

```
MASK:TEST:REPeat?
```

## Arguments

- ON

Turns on repeat on completion.

- OFF

Turns off repeat on completion

- <NR1>

A 0 turns off repeat on completion, and any other integer turns on repeat on completion.

## Example

```
MASK:TEST:REPEAT ON
```

Causes the oscilloscope to continuously repeat the pass/fail test.

---

# MASK:TEST:SAMple

## Description

This command sets or returns the minimum number of waveform database (WfmDB) points the oscilloscope can acquire before it stops a single sequence acquisition or stops running a mask test.

Hint: a bigger sample size may allow a greater throughput (more waveforms to get acquired at a time). However, a very large sample size may slow down the display update. So if you find yourself wanting to speed up the display rate, consider reducing the sample size.

This command works the same as the ACQUIRE:NUMSAMPLES command.

## Group

Mask

## Syntax 1

```
MASK:TEST:SAMple {<NR1>}
```

## Syntax 2

```
MASK:TEST:SAMple?
```

## Arguments

- <NR1>

The number of points to sample.

## Example

```
MASK:TEST:SAMPLE 100000
```

Sets the oscilloscope to acquire 100,000 WfmDB points.

---

# MASK:TEST:SAMple:THReshold

## Description

This command sets or returns the minimum number of hits in mask regions needed to cause the pass/fail status to change from PASSING to FAILING. This affects the mask test when the oscilloscope is acquiring in waveform database (WfmDB) mode.

## Group

Mask

## Syntax 1

```
MASK:TEST:SAMple:THReshold {<NR1>}
```

## Syntax 2

```
MASK:TEST:SAMple:THReshold?
```

## Arguments

- <NR1>

The number of hits that can happen when the test fails.

## Example

```
MASK:TEST:SAMPLE:THRESHOLD 100
```

The test will fail when 100 or more hits happen.

---

# MASK:TEST:SAVEWFM

## Description

This command sets or returns the save waveform to file on failure setting. When enabled, this command copies waveform data from active channels to files when the pass/fail test status changes to 'Failing'.

**Note:** This oscilloscope saves files only on the first violation that causes the pass/fail test to fail. To generate a waveform file for every violation occurrence, set MASK:TEST:REPEAT to ON and MASK:TEST:WAVEFORM to 1. Use this technique only when the expected number of violations is small and the violation occurs intermittently.

## Group

Mask

## Syntax 1

```
MASK:TEST:SAVEWFM {OFF|ON|<NR1>}
```

## Syntax 2

```
MASK:TEST:SAVEWFM?
```

## Arguments

- ON

Turns on copying all active channel waveforms to files.

- OFF

Turns off copying all active channel waveforms to files.

- <NR1>

A 0 turns off copying all active channel waveforms to files, and any other integer turns on copying all active channel waveforms to files.

## Example

```
MASK:TEST:SAVEWFM ON
```

Sets the oscilloscope to save the failing waveform data to files.

---

# MASK:TEST:SAVEWFM:FILENAME

## Description

This command sets or returns the name of the directory and file to use with the MASK:TEST:SAVEWFM command. It defines the directory the files will be put in (name comes from date and time).

## Group

Mask

## Related Commands

MASK:TEST:SAVEWFM, MASK:TEST:LOG:FAILURE

## Syntax 1

```
MASK:TEST:SAVEWFM:FILENAME <Qstring>
```

## Syntax 2

```
MASK:TEST:SAVEWFM:FILENAME?
```

## Argument

- <Qstring>

Is a string representing the name of the file to save waveform data to

## Example

```
MASK:TEST:SAVEWFM:FILENAME "TEST"
```

Sets the file "TEST" as the file to save waveform data to on failure.

---

# MASK:TEST:SRQ:COMPLetion

## Description

This command sets or returns an SRQ on pass/fail test completion mode. When enabled, this command causes the oscilloscope to send a GPIB SRQ command when the mask pass/fail test completes.

## Group

Mask

## Syntax 1

```
MASK:TEST:SRQ:COMPLetion {OFF|ON|<NR1>}
```

## Syntax 2

```
MASK:TEST:SRQ:COMPLetion?
```

## Arguments

- ON

Turns on the pass/fail SRQ on completion.

- OFF

Turns off the pass/fail SRQ on completion.

- <NR1>

A 0 turns off the pass/fail SR on completion, any other integer turns on the pass/fail SRQ on completion.

## Example

```
MASK:TEST:SRQ:COMPLetION ON
```

Turns on pass/fail SRQ on completion



---

# MASK:TEST:SRQ:FAILure

## Description

This command sets or returns the SRQ status on pass/fail test failure mode. When enabled, this command causes the oscilloscope to send a GPIB SRQ command when the pass/fail status changes to 'Failing'. In other words, it sends an SRQ when a failure occurs during the mask pass/fail test.

## Group

Mask

## Syntax 1

```
MASK:TEST:SRQ:FAILure {OFF|ON|<NR1>}
```

## Syntax 2

```
MASK:TEST:SRQ:FAILure?
```

## Arguments

- ON

Turns on the pass/fail SRQ on failure.

- OFF

Turns off the pass/fail SRQ on failure. This is the default.

- <NR1>

A 0 turns off the pass/fail SRQ on failure, and any other integer turns on the pass/fail SRQ on failure.

## Example

```
MASK:TEST:SRQ:FAILURE OFF
```

Turns off the pass/fail SRQ on failure

---

# MASK:TEST:STATE

## Description

This command sets or returns the state of mask pass/fail testing. When enabled, this command resets the pass/fail status information and starts pass/fail mask testing. This command also generates an 'Operation Complete' message when pass/fail testing completes. Refer to BUSY? And \*OPC commands for information on synchronizing a controller with pass/fail testing.

## Group

Mask

## Syntax 1

```
MASK:TEST:STATE {OFF|ON|<NR1>}
```

## Syntax 2

```
MASK:TEST:STATE?
```

## Arguments

- ON

Turns on mask pass/fail testing.

- OFF

Turns off mask pass/fail testing.

- <NR1>

A 0 turns off mask pass/fail testing; any other integer turns on mask pass/fail testing.

## Example

```
MASK:TEST:STATE ON
```

Turns on mask pass/fail testing.

---

# MASK:TEST:STATUS?

## Description

This command returns the pass/fail test status. This command returns one of: OFF, DELAY, PASSING, FAILING, PASSED, FAILED, and VIOLATION. In other words, it indicates the result of the pass/fail test. When the violation count exceeds the violation threshold, the status changes from Passing to Failed.

## Group

Mask

## Syntax

MASK:TEST:STATUS?

## Example

MASK:TEST:STATUS?

Might return: MASK:TEST:STATUS PASSING

---

# MASK:TEST:STOP:FAILure

## Description

This command sets or returns the stop status on pass/fail test failure mode. When enabled, this command causes the oscilloscope to stop acquiring data when the pass/fail status changes to 'Failing'. In other words, it stops when a failure occurs during the mask pass/fail test. Repeat-on-completion mode has no effect.

## Group

Mask

## Syntax 1

```
MASK:TEST:STOP:FAILure {OFF|ON|<NR1>}
```

## Syntax 2

```
MASK:TEST:STOP:FAILure?
```

## Arguments

- ON

Turns on the pass/fail stop on failure.

- OFF

Turns off the pass/fail stop on failure. This is the default.

- <NR1>

A 0 turns off the pass/fail stop on failure, and any other integer turns on the pass/fail SRQ on failure.

## Example

```
MASK:TEST:STOP:FAILURE OFF
```

Turns off the pass/fail stop on failure

---

# MASK:TEST:THReshold

## Description

This command sets or returns the number of failed tested waveforms needed in each pass/fail test to cause the pass/fail status to change from PASSING to FAILING.

## Group

Mask

## Syntax 1

```
MASK:TEST:THReshold <NR1>
```

## Syntax 2

```
MASK:TEST:THReshold?
```

## Argument

- <NR1>

Is an integer that represents the number of tested waveform violations occurring in each pass/fail test needed to change the pass/fail status to FAILING.

This means that if the threshold is set to the default of 1 then 1 failure would cause a test failure. If the threshold is set to 2 then 2 failures would cause a test failure.

## Example

```
MASK:TEST:THRESHOLD?
```

Might return: MASK:TEST:THRESHOLD 32

---

# MASK:TEST:WAVEform

## Description

This command sets or returns the number of waveforms to test during pass/fail testing. If Averaging is on, this is the number of averaged waveforms to test during pass/fail testing. In other words, it specifies the number of waveforms to be acquired in counting the mask hits in the pass/fail test. The default is 20.

## Group

Mask

## Syntax 1

```
MASK:TEST:WAVEform <NR1>
```

## Syntax 2

?

## Argument

- <NR1>

Is an integer that sets the number of tested (sampled or averaged) waveforms to test for each pass/fail test.

## Example

```
MASK:TEST:WAVEform?
```

Might return

```
:MASK:TEST:WAVEFORM 500
```

---

# MASK:USER:AMPLitude

## Description

This command sets or returns the current mask's nominal pulse amplitude in volts. The query form of this command returns the nominal pulse amplitude of the displayed mask. The set form of this command only affects the User mask, regardless of the current (displayed) mask.

## Group

Mask

## Related Commands

MASK:USER:VSCAle

## Syntax 1

MASK:USER:AMPLitude <NR3>

## Syntax 2

MASK:USER:AMPLitude?

## Argument

- <NR3>

This is a floating number that sets the nominal pulse amplitude in volts

## Example

MASK:USER:AMPLITUDE 2.4

Sets the User mask nominal pulse amplitude to 2.4 volts.

---

# MASK:USER:BITRate

## Description

This command sets or returns the bit rate for the user mask.

## Group

Mask

## Syntax 1

```
MASK:USER:BITRate <NR1>
```

## Syntax 2

```
MASK:USER:BITRate?
```

## Argument

- <NR1>

This is a number that sets the bit rate of the user mask in bits per second.

## Example 1

```
MASK:USER:BITRATE 156e6
```

Sets the bit rate.

## Example 2

```
MASK:USER:BITRATE?
```

Might return: 1560000



---

# MASK:USER:HSCALE

## Description

This command sets or returns the nominal timing resolution used to draw the mask, in time/division. The query form of this command returns the nominal timing resolution of the displayed mask. The set form of this command only affects the User mask, regardless of the current (displayed) mask.

## Group

Mask

## Related Commands

MASK:USER:HTRIGPOS

## Syntax 1

MASK:USER:HSCALE <NR3>

## Syntax 2

MASK:USER:HSCALE?

## Argument

- <NR3>

This is a floating point number that sets the mask drawing timing resolution.

## Example

MASK:USER:HSCALE 40e-9

Sets the nominal timing resolution used to draw the mask pulse shape to 40 nanoseconds per division.

---

# MASK:USER:HTRIGPOS

## Description

This command sets or returns the nominal trigger position.(pulse leading edge) used to draw the mask as a fraction of the display width. The query form of this command returns the nominal trigger position of the displayed mask. The set form of this command only affects the User mask, regardless of the current (displayed) mask.

## Group

Mask

## Related Commands

MASK:USER:HSCALE?

## Syntax 1

MASK:USER:HTRIGPOS <NR3>

## Syntax 2

MASK:USER:HTRIGPOS?

## Argument

- <NR3>

Is a floating point number in the range of 0.0 to 1.0 that sets the trigger points as a fraction of the display width, referenced from the left edge of the graticule.

0.0 represents the left edge.

## Example

MASK:USER:HTRIGPOS 0.6

Sets the User mask nominal pulse leading edge position to the 6<sup>th</sup> horizontal graticule division.

---

# MASK:USER:LABel

## Description

This command sets or queries the label assigned to a user mask. This label is displayed in the upper left-hand corner of the graticule when a user mask is selected.

## Group

Mask

## Syntax 1

```
MASK:USER:LABel <string>
```

## Syntax 2

```
MASK:USER:LABel?
```

## Arguments

- <string>

This is an alphanumeric character string used to annotate a user mask label.

## Example 1

```
MASK:USER:LABel "mikesdata"
```

This command assigns "mikesdata" as the user mask label.

## Example 2

```
MASK:USER:LABel?
```

This query might return MASK:USER:LABel "mikesdata" as the user mask label.

---

# MASK:USER:PATTERNBITS

## Description

This command sets or returns the number of bits used for serial trigger for the User mask standard. For example, DS1 requires six bits,; four leading zeros, a one, and a trailing zero. The query form of this command returns the serial bit value of the displayed mask. The set form of this command only affects the User mask, regardless of the current (displayed) mask.

## Group

Mask

## Related Commands

MASK:USER:PRESAMPBITS

## Syntax 1

MASK:USER:PATTERNBITS <NR1>

## Syntax 2

MASK:USER:PATTERNBITS?

## Argument

- <NR1>

Is an integer that sets the number of bits.

## Example

MASK:USER:PATTERNBITS 6

Sets the User mask number of serial pattern bits to 6.

---

# MASK:USER:PRESAMPBITS

## Description

This command sets or returns the number of bits before the (isolated one) pulse leading edge in the serial trigger pass/fail testing. For example, DS1 has four leading zeros. The query form of this command returns the presample bit value of the displayed mask. The set form of this command only affects the User mask, regardless of the current (displayed) mask.

## Group

Mask

## Related Commands

MASK:USER:PATTERNBITS

## Syntax 1

MASK:USER:PRESAMPBITS <NR1>

## Syntax 2

MASK:USER:PRESAMPBITS?

## Argument

- <NR1>

Is an integer that sets the number of bits before the trigger pulse.

## Example

MASK:USER:PRESAMPBITS 4

Sets the User mask number of bits before the trigger pulse to 4.

---

# MASK:USER:RECOrdlength

## Description

This command sets or returns the nominal record length for pulse mask testing. The query form of this command returns the record length value of the displayed mask. The set form of this command only affects the User mask, regardless of the current (displayed) mask.

## Group

Mask

## Syntax 1

```
MASK:USER:RECOrdlength <NR1>
```

## Syntax 2

```
MASK:USER:RECOrdlength?
```

## Argument

- <NR1>

Is an integer number that sets the record length.

## Example

```
MASK:USER:RECORDLENGTH 500
```

Sets the User mask record length to 500 points.

---

# MASK:USER:SEG<m> DELEte

## Description

This command deletes the specified mask segment from the User mask, whether or not the User mask is the current mask.

## Group

Mask

## Syntax

```
MASK:USER:SEG<m> DELEte
```

## Arguments

- <m>

Is an integer that specifies the User mask segment number to delete from the User mask.

## Example

```
MASK:USER:SEG3 DELETE
```

Deletes mask segment three from the User mask.

---

# MASK:USER:SEG<m>:NR\_Pt?

## Description

This command returns the number of points that make up the specified User mask segment. Each mask point consists of an X-Y pair of coordinates.

## Group

Mask

## Syntax

MASK:USER:SEG<m>:NR\_Pt?

## Arguments

- <m>

Is an integer number that specifies a User mask segment number.

## Returns

<NR1> is 0 to 50.

## Example

MASK:USER:SEG<m>:NR\_PT?

Might return MASK:USER:SEG2:NR\_PT 5



---

# MASK:USER:SEG<m>:POINTS

## Description

This command sets or returns the X-Y user coordinates of all points in the specified User mask segment. The set form defines new points in the User mask, replacing any existing points in the specified User mask segment. In other words, it sets or returns the vertices for a particular segment in the selected mask.

## Group

Mask

## Syntax 1

```
MASK:USER:SEG<m>:POINTS <NR3>, <NR3> [ , <NR3>, <NR3> ]
```

## Syntax 2

```
MASK:USER:SEG<m>:POINTS?
```

## Arguments

- <m>

Is an integer that specifies the User mask segment number.

- <NR3>

Refers to the coordinates of one of the vertices in the User mask. Each pair of numbers is the horizontal and vertical coordinates of a mask segment vertex. The pairs must be listed in a counterclockwise order. If the vertical or horizontal scale or position is changed after this command and then the query form of this command is issued, then the value returned from the oscilloscope will not be the same. If just one pair is input then they are ignored and the User mask segment is marked as undefined. The default is not points in the User mask segment.

## Example

```
MASK:USER:SEG2:POINTS -2.3e-9, 44e-3, -2.5e-9, 47e-3, 1.2e-9, 44e-3
```

Defines the points in User mask segment 2.

---

# MASK:USER:TRIGTOSAMP

## Description

This command sets or returns the time, in seconds, from the (leading edge) trigger position to the pulse bit sampling position. The query form of this command returns the time value of the displayed mask. The set form of this command only affects the User mask, regardless of the current (displayed) mask.

## Group

Mask

## Related Commands

MASK:USER:WIDTH

## Syntax 1

MASK:USER:TRIGTOSAMP <NR3>

## Syntax 2

MASK:USER:TRIGTOSAMP?

## Argument

- <NR3>

Is the floating point number that sets the time to the pulse bit sampling position.

## Example

MASK:USER:TRIGTOSAMP 162e-9

Sets the User mask time from the (leading edge) trigger position to the pulse bit sampling position to 162 nanoseconds.

---

# MASK:USER:VOFFSet

## Description

This command sets or returns the nominal vertical offset, in volts, used to vertically offset the input channels. The query form of this command returns the offset value of the displayed mask. The set form of this command only affects the User mask, regardless of the current (displayed) mask.

## Group

Mask

## Related Commands

MASK:USER:VPOS, MASK:USER:VSCale

## Syntax 1

MASK:USER:VOFFSet <NR3>

## Syntax 2

MASK:USER:VOFFSet?

## Argument

- <NR3>

Is a floating point number that sets the nominal vertical offset in volts.

## Example

MASK:USER:VOFFSET 0.0

Sets the User mask nominal vertical offset to 0.0 volts.

---

# MASK:USER:VPOS

## Description

This command sets or returns the nominal vertical position, in divisions, used to vertically position the input channels. The query form of this command returns the vertical position value of the displayed mask. The set form of this command only affects the User mask, regardless of the current (displayed) mask.

## Group

Mask

## Related Commands

MASK:USER:VOFFSet, MASK:USER:VSCALE

## Syntax 1

MASK:USER:VPOS <NR3>

## Syntax 2

MASK:USER:VPOS?

## Argument

- <NR3>

Is a floating point number that sets the nominal vertical position in divisions.

## Example

MASK:USER:VPOS -0.8

Sets the User mask nominal vertical position to -0.8 divisions.

---

# MASK:USER:VSCALE

## Description

This command sets or returns the nominal vertical scale, in volts per division, used to vertically scale the input channels. The query form of this command returns the vertical scale value of the displayed mask. The set form of this command only affects the User mask, regardless of the current (displayed) mask.

## Group

Mask

## Related Commands

MASK:USER:VOFFSet , MASK:USER:VPOS

## Syntax 1

MASK:USER:VSCALE <NR3>

## Syntax 2

MASK:USER:VSCALE?

## Argument

- <NR3>

Is a floating point number that sets the nominal bit width in seconds.

## Example

MASK:USER:VSCALE 200e-3

Sets the User mask nominal vertical scale to 200 millivolts per division

---

# MASK:USER:WIDTH

## Description

This command sets or returns the nominal bit width in seconds. The query form of this command returns the bit width value of the displayed mask. The set form of this command only affects the User mask, regardless of the current (displayed) mask.

## Group

Mask

## Related Commands

MASK:USER:TRIGTOSAMP

## Syntax 1

MASK:USER:WIDTH <NR3>

## Syntax 2

MASK:USER:WIDTH?

## Argument

- <NR3>

Is a floating point number that sets the nominal bit width in seconds.

This is the time of one bit of data where  $\text{bit width} = 1 / (\text{data rate of the signal})$ .

## Example

MASK:USER:WIDTH 648e-9

Sets the User mask nominal bit width to 648 nanoseconds (1.544 Mbits/second).

---

# MATH<x>:DEFIne

## Description

This command allows you to define new waveforms using mathematical expressions. Sending this command is equivalent to selecting Math Setup from the Math menu, selecting a math waveform (Math 1 through Math 4 for 4-channel models or Math 1 through Math 2 for 2-channel models), and then entering a math expression in the Math<x> box. The query form of this command returns the math definition for the specified math waveform.

You can specify a math expression from waveforms, measurements and scalar sources, functions, operands, and numerical constants. You can define and display up to four math waveforms simultaneously on 4-channel models or up to 2 math waveforms on 2-channel models.

Math expressions can be simple, such as Ch1, which specifies that a waveform should show the signal source of Channel 1 with no mathematical computation. Math expressions can also be complex, consisting of 100 plus characters and comprising many sources (including other math waveforms), functions, and operands. As an example, you can enter the expression  $\text{Log}(\text{Ch1}+\text{Ch2})$ , which specifies that the signals from channels 1 and 2 are to be algebraically added, and the base 10 log of the sum is to be shown as the final math waveform.

For more information about constructing mathematical expressions, see *Creating and Using Math Waveforms* in the user online help for this instrument.

## Group

Math

## Related Commands

MATHVAR:VAR<x>

## Syntax 1

MATH<x>:DEFIne <QString>

## Syntax 2

MATH<x>:DEFIne?

## Argument

- <QString>

This quoted string argument is the mathematical expression that defines the waveform.

## Example 1

```
MATH2:DEFIne "Ch1+Ch2"
```

This command adds the Ch 1 waveform and Ch 2 waveform, storing the results in Math 2.

---

## Example 2

`MATH2:DEFIne?`

This query might return

`:MATH1:DEFINE "CH2*REF2"`

as the expression that defines Math 1.



---

# MATH<x>:LABel:NAME

## Description

This command sets or returns the label string, which is used for annotating the math waveform on the screen. The math waveform to which the label is attached is specified by x, which ranges in value from 1 through 4 for four-channel instruments or 1 through 2 for two-channel instruments. This command is equivalent to selecting Math Setup from the Math menu and entering a label in the Label box.

## Group

Math

## Syntax 1

```
MATH<x>:LABel:NAME <string>
```

## Syntax 2

```
MATH<x>:LABel:NAME?
```

## Argument

- <string>

This specifies the label to annotate the math waveform.

## Example 1

```
MATH2:LABel:NAME "Probe point7"
```

This command assigns "Probe point7" Math 2.

## Example 2

```
MATH2:LABel:NAME?
```

This query might return

```
:MATH2:LABEL:NAME "Probe point7"
```

indicating that Probe point 7 is the label for Math 2.

---

# MATH<x>:LABel:XPOS

## Description

This command sets or queries the X screen offset at which the label attached to a math waveform is displayed, relative to the left edge of the screen. Channels are specified by x, which ranges from 1 through 4 for four-channel instruments or 1 through 2 for two-channel instruments. This command is equivalent to selecting Math Label from the Math menu and entering a value in the X Position box.

## Group

Math

## Related Commands

MATH<x>:LABel:YPOS

## Syntax 1

MATH<x>:LABel:XPOS <NR1>

## Syntax 2

MATH<x>:LABel:XPOS?

## Argument

- <NR1>

This is the location (in pixels) where the label for the selected math waveform is displayed, relative to the left edge of the screen. Arguments should be integers ranging from 0 to 500.

## Example 1

MATH2:LABel:XPOS 50

This command moves the waveform label for the MATH2 waveform (on a four-channel instrument) so that it begins 50 pixels to the right of the left edge of the screen.

## Example 2

MATH2:LABel:XPOS?

This query might return

:MATH2:LABEL:XPOS 50

indicating that the waveform label for the MATH2 waveform is currently 50 pixels to the right of the left edge of the screen.

---

# MATH<x>:LABel:YPOS

## Description

This command sets or queries the Y screen offset at which the label attached to a math waveform is displayed, relative to the top edge of the screen. The Math waveform is specified by x, which ranges from 1 through 4 for four-channel instruments or 1 through 2 for two-channel instruments. This command is equivalent to selecting Math Label from the Math menu and entering a value in the Y Position box.

## Group

Math

## Related Commands

MATH<x>:LABel:XPOS

## Syntax 1

MATH<x>:LABel:YPOS <NR1>

## Syntax 2

MATH<x>:LABel:YPOS?

## Argument

- <NR1>

This is the location (in pixels) where the label for the selected math waveform is displayed, relative to the top edge of the screen. Arguments should be integers ranging from 0 to 400.

## Example 1

MATH2:LABel:YPOS -25

This command moves the waveform label for the MATH2 waveform to just beneath (25 pixels below) the top of the screen.

## Example 2

MATH2:LABel:YPOS?

This query might return

:MATH2:LABEL:YPOS 0

indicating that the waveform label for the MATH2 waveform is currently located just beneath the top of the screen.

---

# MATH<x>:NUMAVg

## Description

This command sets or returns the acquisition number at which the averaging algorithm will begin exponential averaging. Prior to that acquisition number, the algorithm uses stable averaging. This has no effect unless the AVG() function is used in the specified math expression. If so, it affects all AVG() functions in this math expression. The Math waveform is specified by x, which ranges from 1 through 4 for four-channel instruments or 1 through 2 for two-channel instruments. This command is equivalent to selecting Set Math Averages from the Math menu and then entering an averaging value for the math waveform.

## Group

Math

## Related Commands

ACQuire:NUMAVg

## Syntax 1

MATH<x>:NUMAVg <NR1>

## Syntax 2

MATH<x>:NUMAVg?

## Argument

- <NR1>

This specifies the number of acquisitions over which exponential averaging is performed.

## Example 1

MATH2:NUMAVg 10

This command averages the Math 2 waveform ten times.

## Example 2

MATH2:NUMAVg?

This query might return

:MATH2:NUMAVG 10

indicating that ten Math 2 waveforms are averaged before a single acquisition occurs.

---

# MATH<x>:SPECTral?

## Description

This query-only command returns the current spectral setups for the specified math waveform. The Math waveform is specified by x, which ranges from 1 through 4 for four-channel instruments or 1 through 2 for two-channel instruments. This command is equivalent to selecting Spectral Setup from the Math menu and viewing the current spectral setup values.

## Group

Math

## Syntax

MATH<x>:SPECTral?

## Example

MATH1:SPECTral?

This query might return the spectral setup values for MATH1 as follows:

```
:MATH1:SPECTRAL:MAG DB;  
PHASE DEGREES;  
GATEPOS 0.0000E+00;  
GATEWIDTH 1.9996E-06;  
REFLEVEL 4.4587+01;  
REFLEVELOFFSET2.2360E-01;  
SPAN 1.2500E+09;  
CENTER 6.2500E+08;  
RESBW 1.0002E=06;  
WINDOW GAUSSIAN;  
SUPPRESS -3.5000E+01;  
UNWRAP 0;LOCK 0
```

---

# MATH<x>:SPECTral:CENTER

## Description

This command specifies or returns the center frequency of the spectral analyzer output data span for the specified math waveform. The Math waveform is specified by x, which ranges from 1 through 4 for four-channel instruments or 1 through 2 for two-channel instruments. This command is equivalent to selecting Spectral Setup from the Math menu and then entering a Center Freq value.

## Group

Math

## Related Commands

MATH<x>:SPECTral:SPAN

## Syntax 1

MATH<x>:SPECTral:CENTER <NR3>

## Syntax 2

MATH<x>:SPECTral:CENTER?

## Argument

- <NR3>

This is the desired frequency of the spectral analyzer output data span in hertz.

## Example 1

MATH3:SPECTral:CENTER 10.09E6

This command sets the center frequency to the closest value it can attain to 10.09 MHz.

## Example 2

MATH2:SPECTral:CENTER?

This query might return

:MATH2:SPECTral:CENTER 10.0900E+06

indicating that the center frequency is currently set at 10.09 MHz.

---

# MATH<x>:SPECTral:GATEPOS

## Description

This command sets or returns the position of the center of the gate, which is used as the data input to the spectral analyzer for the specified math waveform. The math waveform is specified by x, which ranges from 1 through 4 for four-channel instruments or 1 through 2 for two-channel instruments. This command is equivalent to selecting Spectral Setup from the Math menu and then entering a Gate Pos value.

## Group

Math

## Related Commands

MATH<x>:SPECTral:GATEWIDTH

## Syntax 1

```
MATH<x>:SPECTral:GATEPOS <NR3>
```

## Syntax 2

```
MATH<x>:SPECTral:GATEPOS?
```

## Argument

- <NR3>

This is the gate position. Units are represented in seconds, with respect to trigger position.

## Example 1

```
MATH1:SPECTral:GATEPOS 0
```

This command specifies the position of the center edge of the gate used as the data input to the spectral analyzer.

## Example 2

```
MATH2:SPECTral:GATEPOS?
```

This query might return

```
:MATH2:SPECTRAL:GATEPOS 0.0000E+00
```

indicating that the gate position is set to zero (the trigger position).

---

# MATH<x>:SPECTral:GATEWIDTH

## Description

This command sets or returns the gate width input, in seconds, to the spectral analyzer for the specified math waveform. The math waveform is specified by x, which ranges from 1 through 4 for four-channel instruments or 1 through 2 for two-channel instruments. This command is equivalent to selecting Spectral Setup from the Math menu and entering a duration value in the Gate Dur box.

## Group

Math

## Related Commands

MATH<x>:SPECTral:GATEPOS

## Syntax 1

MATH<x>:SPECTral:GATEWIDTH <NR3>

## Syntax 2

MATH<x>:SPECTral:GATEWIDTH?

## Argument

- <NR3>

This is the time across the 10-division screen in seconds.

## Example 1

```
MATH1:SPECTral:GATEWIDTH 1.0E-3
```

This command sets the gate width input to the spectral analyzer at 1 ms.

## Example 2

```
MATH3:SPECTral:GATEWIDTH?
```

This query might return

```
:MATH3:SPECTRAL:GATEWIDTH 1.0000E-03
```

indicating that the gate width to the spectral analyzer is set to 1 ms.



---

# MATH<x>:SPECTral:LOCK

## Description

This command locks menus for two or more math waveforms together as a group. The query form of this command returns an ON (1) or OFF (0), indicating whether spectral locking is turned on. This command is equal to selecting Spectral Setup from the Math menu, choosing the Control tab and then clicking the Time/Track Frequency Domain Controls button associated with the math waveforms that you want to lock. However, applying spectral locking functionality from the interface is limited to locking Math1 and Math2 (on two-channel and four-channel instruments) or Math 3 and Math4 (on four channel instruments).

MATH<x> Lock Combinations			
MATH1	MATH2	MATH3	Locked Math Waveforms
Off	Off	Off	None
Off	Off	On	Math3 and Math4 locked
Off	On	Off	Math2 and Math3 locked
Off	On	On	Math2, Math3 and Math4 locked
On	Off	Off	Math1 and Math2 locked
On	Off	On	Math1 and Math 2 locked, Math3 and Math4 locked
On	On	Off	Math1, Math2 and Math3 locked
On	On	On	Math1, Math2, Math3 and Math4 locked

**Note:** Executing MATH4:SPECTral:LOCK via the GPIB interface has no affect since there is no Math5 to which it can be locked.

## Group

Math

## Syntax 1

MATH<x>:SPECTral:LOCK {ON|OFF}

## Syntax 2

MATH<x>:SPECTral:LOCK?

## Arguments

- ON

This turns on the parameter lock for the specified math waveform.

- OFF

This turns off the parameter lock for the specified math waveform.

- <NR1>

A 0 disables the parameter lock for the specified math waveform; any other value enables the parameter lock.

---

### **Example 1**

`MATH1:SPECTral:LOCK ON`

This command turns on the parameter lock for Math1, which causes the parameters for Math1 and Math2 to be locked together.

### **Example 2**

`MATH1:SPECTral:LOCK?`

This query might return

`:MATH1:SPECTRAL:LOCK 0`

indicating that the parameter lock for Math1 is turned off.

---

# MATH<x>:SPECTral:MAG

## Description

This command sets or returns the units of the SpectralMag function in the math string. The Math waveform is specified by x, which ranges from 1 through 4 for four-channel instruments or 1 through 2 for two-channel instruments. If you are using the standard math, this command is equivalent to selecting Magnitude Spectrum from the Math menu and then selecting the units that you want from the Scale button drop-down menu. If you are using the Advanced Analysis functions, this command is equivalent to selecting Spectral Setup from the Math menu, choosing the Mag tab, and then clicking the desired Scale button.

## Group

Math

## Related Commands

MATH<x>:SPECTral:PHASE

## Syntax 1

MATH<x>:SPECTral:MAG {LINEAR|DB|DBM}

## Syntax 2

MATH<x>:SPECTral:MAG?

## Arguments

- LINEAR

This sets the SpectralMag units to linear.

- DB

This sets the SpectralMag units to decibels.

- DBM

This sets the SpectralMag units to decibels. It also sets the Ref Level Offset to a value that is the equivalent of 1mW into 50Ω.

## Example 1

```
MATH2:SPECTral:MAG DB
```

This command sets the SpectralMag units for Math2 to decibels.

## Example 2

```
MATH2:SPECTral:MAG?
```

This query might return

```
:MATH2:SPECTRAL:MAG DB
```

indicating that the SpectralMag units for Math2 are set to decibels.

---

# MATH<x>:SPECTral:PHASE

## Description

This command sets or returns the units of a SpectralPhase function in the math string. The Math waveform is specified by x, which ranges from 1 through 4 for four-channel instruments or 1 through 2 for two-channel instruments. This command is equal to selecting Spectral Setup from the Math menu, choosing the Phase tab, and then clicking the desired Scale button.

## Group

Math

## Related Commands

MATH<x>:SPECTral:MAG

## Syntax 1

```
MATH<x>:SPECTral:PHASE {DEGREES|RADIANS|GROUPDELAY}
```

## Syntax 2

```
MATH<x>:SPECTral:PHASE?
```

## Arguments

- DEGREES

This sets the SpectralPhase units to degrees.

- RADIANS

This sets the SpectralPhase units to radians.

- GROUPDELAY

This sets the SpectralPhase units to groupdelay, which commutes the derivative of unwrapped phase spectrum. Units are expressed in seconds.

## Example 1

```
MATH2:SPECTral:PHASE DEGREES
```

This command sets the SpectralPhase units for Math2 to degrees.

## Example 2

```
MATH2:SPECTral:PHASE?
```

This query might return

```
:MATH2:SPECTRAL:PHASE RADIANS
```

indicating that the SpectralPhase units for Math2 are set to radians.

---

# MATH<x>:SPECTral:REFLevel

## Description

This command specifies the vertical position of the output data from the spectral analyzer on the display screen for the specified math waveform. The numerical value represents the position at the top of the display graticule. The Math waveform is specified by x, which ranges from 1 through 4 for four-channel instruments or 1 through 2 for two-channel instruments. This command is equal to selecting Spectral Setup from the Math menu, choosing the Mag tab and then entering a value in the Reference Level box.

## Group

Math

## Related Commands

MATH<x>:SPECTral:REFLEVELOffset

## Syntax 1

MATH<x>:SPECTral:REFLevel <NR3>

## Syntax 2

MATH<x>:SPECTral:REFLevel?

## Argument

- <NR3>

This is the position that represents the top of the display screen graticule. The range depends on the units and both the MATH<x>:VERTical:SCALE and MATH<x>:VERTical:POSITION settings.

## Example 1

```
MATH1:SPECTral:REFLevel -10
```

This sets the top of the display screen to be a reference level of -10 dB. If the vertical scale is LINEAR, then the vertical units will be the same as the input waveform.

## Example 2

```
MATH1:SPECTral:REFLevel?
```

This query might return

```
:MATH1:SPECTRAL:REFLEVEL 2.0000E+01
```

indicating that the top of the display screen is set to a reference level of 20 dB.

---

# MATH<x>:SPECTral:REFLEVELOffset

## Description

This command sets or returns the spectral level offset used for calculating the dB value for the specified math waveform. The Math waveform is specified by x, which ranges from 1 through 4 for four-channel instruments or 1 through 2 for two-channel instruments. Changing the reference level offset causes the spectral waveform to move vertically, with respect to zero dB. This command is equal to selecting Spectral Setup from the Math menu, choosing the Mag tab and then entering a value in the Reference Level Offset box.

## Group

Math

## Related Commands

MATH<x>:SPECTral:REFLevel

## Syntax 1

MATH<x>:SPECTral:REFLEVELOffset {DBM|<NR3>}

## Syntax 2

MATH<x>:SPECTral:REFLEVELOffset?

## Arguments

- DBM

This specifies the reference level used for calculation to be equivalent to 1 mW into 50 Ω (Zero dB will occur at this level).

- <NR3>

This specifies the reference level used for calculation of the decibel value when the output units are Log.

## Example 1

```
MATH1:SPECTral:REFLEVELOFFset 0.5
```

This sets the reference level for the Log calculation for decibel to be 0.5.  $\text{dB} = A \times \text{Log}(y/\text{<NR3>})$  where A is 10 if the input units are watts and A is 20 if the input units are otherwise.

## Example 2

```
MATH1:SPECTral:REFLEVELOffset DBM
```

This sets the decibel reference to be equivalent to 1 mW into 50 Ω. The reference level numerical value will depend on the input units. If the units are volts, the value is set to 0.2236 V; if the units are amperes, the value is set to 40 μA; if the units are watts, the value is set to 0.001 W.

---

### Example 3

`MATH1:SPECTral:REFLEVELOffset?`

This query might return

`:MATH1:SPECTRAL:REFLEVELOFFSET 2.23360E-01`

indicating that the spectral reference level offset is 223.6 mV.

---

# MATH<x>:SPECTral:RESBw

## Description

This command sets or returns the resolution bandwidth of the spectral analyzer for the specified math waveform. The Math waveform is specified by x, which ranges from 1 through 4 for four-channel instruments or 1 through 2 for two-channel instruments. This command is equivalent to selecting Spectral Setup from the Math menu and then entering a value in the Res BW box.

## Group

Math

## Related Commands

MATH<x>:SPECTral:GATEWIDTH, MATH<x>:SPECTral:CENTER, MATH<x>:SPECTral:SPAN, MATH<x>:SPECTral:WINDOW

## Syntax 1

```
MATH<x>:SPECTral:RESBw <NR3>
```

## Syntax 2

```
MATH<x>:SPECTral:RESBw?
```

## Argument

- <NR3>

This is the desired resolution bandwidth value. Units are represented in hertz.

## Example 1

```
MATH1:SPECTral:RESBw 250E3
```

This command sets the resolution bandwidth to the attainable value that is close to 250 KHz.

## Example 2

```
MATH1:SPECTral:RESBw?
```

This query might return

```
:MATH1:SPECTRAL:RESBW 1.0002E+06
```

indicating the actual resolution bandwidth value obtained from the spectral analyzer.



---

# MATH<x>:SPECTral:SPAN

## Description

This command sets the ceiling of the span to a value that is closest to the specified value. The query form of this command returns the current span value for specified math waveform. The Math waveform is specified by x, which ranges from 1 through 4 for four-channel instruments or 1 through 2 for two-channel instruments. This command is equal to selecting Spectral Setup from the Math menu and then entering a value in the Freq Span box.

## Group

Math

## Related Commands

MATH<x>:SPECTral:CENTER

## Syntax 1

MATH<x>:SPECTral:SPAN {<NR3>|FULL}

## Syntax 2

MATH<x>:SPECTral:SPAN?

## Arguments

- <NR3>

This specifies the frequency span of the output data vector from the spectral analyzer.

- FULL

This sets the top of the span to 1/2 the sample rate and sets the center frequency to 1/2 the span.

## Example 1

```
MATH1:SPECTral:SPAN FULL
```

This command sets the top of the span to 1/2 the sample rate and sets the center frequency to 1/2 the span.

## Example 2

```
MATH1:SPECTral:SPAN 2.56E6
```

This command sets the top of the span to a value that is closest to 2.56 MHz.

---

### **Example 3**

`MATH1:SPECTral:SPAN?`

This query might return

`:MATH1:SPECTRAL:SPAN 1.2500E+09`

indicating the actual span value obtained by the spectral analyzer.

---

# MATH<x>:SPECTral:SUPPress

## Description

This command sets or returns the phase suppression threshold for the specified math waveform. The Math waveform is specified by x, which ranges from 1 through 4 for four-channel instruments or 1 through 2 for two-channel instruments. This command is equal to selecting Spectral Setup from the Math menu, choosing the Phase tab and then entering a value in the Suppression Threshold box.

## Group

Math

## Related Commands

MATH<x>:SPECTral:UNWRap

## Syntax 1

```
MATH<x>:SPECTral:SUPPress <NR3>
```

## Syntax 2

```
MATH<x>:SPECTral:SUPPress?
```

## Argument

- <NR3>

This is the level under which all data with magnitudes are displayed as zero phase.

## Example 1

```
MATH1:SPECTral:SUPPress -62
```

This command specifies that any magnitude values less than -62 dB will have their phase output set to zero.

## Example 2

```
MATH1:SPECTral:SUPPress?
```

This query might return

```
:MATH1:SPECTRAL:SUPPRESS -3.5000E+01
```

indicating that the phase suppression threshold is currently set to -35 dB.

---

# MATH<x>:SPECTral:UNWRap

## Description

This command sets or returns whether phase unwrap of the spectral analyzer output data is enabled for the specified math waveform. The Math waveform is specified by x, which ranges from 1 through 4 for four-channel instruments or 1 through 2 for two-channel instruments. This command is equal to selecting Spectral Setup from the Math menu, choosing the Phase tab and then clicking the Unwrap button.

## Group

Math

## Related Commands

MATH<x>:SPECTral:SUPPress

## Syntax 1

MATH<x>:SPECTral:UNWRap <ON|OFF>

## Syntax 2

MATH<x>:SPECTral:UNWRap?

## Arguments

- ON

This enables phase unwrap.

- OFF

This disables phase wrap.

- <NR1>

A 0 disables phase wrap; any other value enables phase wrap.

## Example 1

```
MATH1:SPECTral:UNWRap ON
```

This command enables phase wrap of the spectral analyzer output data.

## Example 2

```
MATH1:SPECTral:UNWRap?
```

This query might return

```
:MATH1:SPECTRAL:UNWRAP 0
```

indicating that phase unwrap of the spectral analyzer output data is disabled.

---

# MATH<x>:SPECTral:WINDow

## Description

This command sets or returns the window function used to multiply the spectral analyzer input data for the specified math waveform. The Math waveform is specified by x, which ranges from 1 through 4 for four-channel instruments or 1 through 2 for two-channel instruments. A spectral window determines what the filter shape of the spectral analyzer will be in the frequency domain. It can be described by a mathematical function that is multiplied point-by-point times the input data to the spectral analyzer. This command is equal to selecting Spectral Setup from the Math menu, and choosing from the Window Type drop-down list.

Following is a list of arguments that specify the window function used to multiply the spectral analyzer input data. The windows are listed in the order of their ability to resolve frequencies (resolution bandwidth). For additional information about spectral windows, see *Selecting a Spectral Window* in the online help for this instrument.

## Group

Math

## Related Commands

MATH<x>:SPECTral:RESBw

## Syntax 1

```
MATH<x>:SPECTral:WINDow {RECTANGULAR |  
HAMMING | HANNING | KAISERBESSEL | BLACKMANHAR-  
RIS | FLATTOP2 | GAUSSIAN | TEKEXPONENTIAL}
```

## Syntax 2

```
MATH<x>:SPECTral:WINDow?
```

## Arguments

### ■ RECTANGULAR

This type of window function is equivalent to multiplying all gate data by one.

### ■ HAMMING

This type of window function is based on a cosine series.

### ■ HANNING

This type of window function is based on a cosine series.

### ■ KAISERBESSEL

This type of window function is based on a cosine series.

### ■ BLACKMANHARRIS

This type of window function is based on a cosine series.

---

- GAUSSIAN

This type of window function has the best localization characteristics in the joint time/frequency plane.

- TEKEXPONENTIAL

This type of window has an exponential nonsymmetrical shape in the time domain and a triangular shape in the frequency domain.

- FLATTOP2

This type of window function is a cosine series window with a flattened frequency response lobe.

### Example 1

```
MATH2:SPECTral:WINDow HANNING
```

This command applies a Hanning window to the spectral analyzer input data.

### Example 2

```
MATH2:SPECTral:WINDow?
```

This query might return

```
:MATH2:SPECTRAL:WINDOW TEKEXPONENTIAL
```

indicating that the window function used to multiply the spectral analyzer input data is the Tek exponential function.

---

# MATH<x>:UNITString

## Description

This command sets or queries the string to use for units for the math waveform specified by x, which can be 1 through 4. This command will override the default unit string with the one that you specify.

## Group

Math

## Syntax 1

```
MATH<x>:UNITString <QString>
```

## Syntax 2

```
MATH<x>:UNITString?
```

## Argument

- <QString>

This quoted string argument is the units to be used for the specified math waveform.

## Example 1

```
MATH2:UNITString "deg"
```

This command specifies that the unit of the Math2 operation will be called deg.

## Example 2

```
MATH1:UNITString?
```

This query might return `:MATH1:UNITSTRING "deg"` as the unit for the Math1 operation.

---

# MATH<x>:VERTical:POSition

## Description

This command sets or queries the vertical position of the specified Math waveform. The Math waveform is specified by x, which ranges from 1 through 4 for four-channel instruments or 1 through 2 for two-channel instruments. The position value is usually applied to the signal before it is digitized. The highest three units/div scale ranges of a given math are implemented by changing the way the acquired data is displayed. When the instrument is operating in any of these highest three scale ranges, the position control operates only on the signal after it is digitized. Note that if a signal that exceeds the range of the digitizer in one of these three scale ranges is repositioned, the displayed waveform will contain clipped values on-screen. This command is equivalent to selecting Position/Scale from the Math menu and then entering a Vert Pos value or adjusting the front-panel Vertical POSITION knob.

Increasing the position value of a waveform causes the waveform to move up, and decreasing the position value causes the waveform to move down. Position adjusts only the display position of a waveform, whether a channel, math, or reference waveform. The position value determines the vertical graticule coordinate at which input signal values, equal to the present offset setting for that reference, are displayed. For example, if the position for Math 3 is set to 2.0 and the offset is set to 3.0, then the input signals equal to 3.0 are displayed 2.0 divisions above the center of the screen.

Be aware that autoscaling occurs when a math waveform is first defined and enabled, or when a math string changes. After the math waveform is computed for the first time, the instrument determines the min + max of that waveform data. Then, the instrument sets the math position so that  $(\text{min} + \text{max})/2$  is in the center of the screen. In addition, the instrument sets the math scale so that the range of the min and max cover 6 divisions. This autoscaling process can take up to 1/2 second to complete and will override any vertical scale or position commands for that math waveform received during this time. You should insert an appropriate pause in your program after defining and enabling a math waveform before changing its position or scale.

## Group

Math, Vertical

## Related Commands

CH<x>POSition, REF<x>:VERTical:POSition

## Syntax 1

MATH<x>:VERTical:POSition <NR3>

## Syntax 2

MATH<x>:VERTical:POSition?



---

## Argument

- <NR3>

This is the desired position in divisions from the center graticule.

### Example 1

```
MATH2:VERTical:POSition 1.3E+00
```

This command positions the Math 2 input signal 1.3 divisions higher than a position of 0.

### Example 2

```
MATH1:VERTical:POSition?
```

This query might return

```
:MATH1:VERTICAL:POSITION -1.3000E+00
```

This indicates that the current position of Math 1 is 1.3 divisions below the center graticule.

---

# MATH<x>:VERTical:SCALE

## Description

This command sets or queries the vertical scale of the specified math waveform. The Math waveform is specified by x, which ranges from 1 through 4 for four-channel instruments or 1 through 2 for two-channel instruments. This command is equivalent to selecting Position/Scale from the Math menu and then entering a Vert Scale value or adjusting the front-panel Vertical SCALE knob.

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

Scale affects all waveforms. For reference and math waveforms, the scale setting controls the display only, graphically scaling these waveforms and having no effect on the acquisition hardware.

Be aware that autoscaling occurs when a math waveform is first defined and enabled, or when a math string changes. After the math waveform is computed for the first time, the instrument determines the min + max of that waveform data. Then, the instrument sets the math position so that  $(\text{min} + \text{max})/2$  is in the center of the screen. In addition, the instrument sets the math scale so that the range of the min and max covers 6 divisions. This autoscaling process can take up to 1/2 second to complete and will override any vertical scale or position commands for that math waveform received during this time. You should insert an appropriate pause in your program after defining and enabling a math waveform before changing its position or scale.

## Group

Math, Vertical

## Related Commands

CH<x>:SCALE, REF<x>:VERTical:SCALE

## Syntax 1

MATH<x>:VERTical:SCALE <NR3>

## Syntax 2

MATH<x>:VERTical:SCALE?

## Argument

- <NR3>

This is the scale, in volts, amps or watts per division. The range is from 100.0E-36 through 100.0E+36.

---

### **Example 1**

```
MATH4:VERTical:SCAle 100E-03
```

This command sets the Math 4 scale to 100 mV per division.

### **Example 2**

```
CH2:VERTical:SCAle?
```

This query might return

```
:MATH2:VERTICAL:SCALE 1.0000E+00
```

This indicates that the current scale setting of Math 2 is 1 V per division.

---

# MATHVAR:VAR<x>

## Description

This command sets or queries up to 8 different numerical values you can use within math expressions. These values can range from  $-9.9e37$  to  $9.9e37$ ; the default values are 0.0. <x> specifies the location, 1 through 8, in which you can store values. Stored math variables can be reference within math expressions as VAR1, VAR2,...VAR8.

For example, the following command defines MATH1 as the product of Channel 1 and math variable 1: MATH1:DEFINE "CH1 \* VAR1".

## Group

Math

## Related Commands

MATH<x>:DEFine

## Syntax 1

MATHVAR:VAR<x> <NR3>

## Argument

- <NR3>

Specifies the numerical value to be stored in location x <1 through 8>.

## Syntax 2

MATHVAR:VAR<x>?

Returns the value of the math variable store in location <x>.

## Syntax 3

MATHVAR?

Returns the values of all variables stored in locations 1 through 8.

## Example 1

MATHVAR:VAR3 -2.43e-5

Stores the value  $-2.43e-5$  in the 3<sup>rd</sup> math variable location.

## Example 2

MATHVAR:VAR3?

This query might return :MATHVAR:VAR3 24.3000E-6 for the expression stored in location 3.

---

# MATH<x>?

## Description

This query-only command returns the definition for the math waveform specified by <x>, which ranges from 1 through 4 in four-channel instruments and 1 through 2 in two-channel instruments.

## Group

Math

## Related Commands

SElect:<wfm>

## Syntax

MATH<x>?

## Example

MATH1?

This query might return the following definition of a math waveform:

```
:MATH1:DEFINE "Ch1-Ch2";UNITSTRING "V";  
NUMAVG 2;VERTICAL:SCALE 112.0000E-3;  
POSITION 1.6400;:MATH1:LABEL:NAME "";XPOS  
1.0000;YPOS 600.0000E-3;:MATH1:SPECTRAL:MAG  
DB;PHASE DEGREES;GATEPOS 0.0000;GATEWIDTH  
4.0000E-6;REFLEVEL 20.0000;REFLEVELOFFSET  
223.6000E-3;SPAN 625.0000E+6;CENTER  
312.5000E+6;RESBW 500.0000E+3;WINDOW GAUS-  
SIAN;SUPPRESS -200.0000;UNWRAP 0;LOCK 0
```

---

# MEASUREMENT:GATING

## Description

This command specifies or returns the measurement gating setting. This command is equivalent to selecting Gating from the Measure menu and then clicking the desired Measurement Gating setting.

## Group

Measurement

## Syntax 1

```
MEASUREMENT:GATING {ON|OFF|<NR1>|ZOOM|CURSOR}
```

## Syntax 2

```
MEASUREMENT:GATING?
```

## Arguments

- ON

This turns on measurement gating.

- OFF

This turns off measurement gating.

- <NR1>

A 0 turns off measurement gating; any other value turns on measurement gating.

- ZOOM<x>

This turns on gating, using the left and right edges of the zoom box. <x> specifies the zoom window, which ranges from 1 through 4.

- CURSOR

This limits measurements to the portion of the waveform between the vertical bar cursors, even if they are off screen.

## Example 1

```
MEASUREMENT:GATING ON
```

This command turns on measurement gating.

## Example 2

```
MEASUREMENT:GATING?
```

This query might return

```
:MEASUREMENT:GATING CURSOR
```

indicating that measurements are limited to the portion of the waveform between the vertical bar cursors.

---

# MEASUrement:IMMed?

## Description

This query-only command returns all immediate measurement setup parameters.

## Group

Measurement

## Syntax

```
MEASUrement:IMMed?
```

## Example

```
MEASUrement:IMMed?
```

This query might return

```
:MEASUREMENT:IMMED:TYPE AMPLITUDE;  
UNITS "V";SOURCE1 CH1;SOURCE2 CH1;  
SOURCE1:SIGTYPE PULSE;  
:MEASUREMENT:IMMED:SOURCE2:SIGTYPE PULSE;  
:MEASUREMENT:IMMED:DELAY:EDGE1 RISE;  
EDGE2 RISE;DIRECTION FORWARDS;  
:MEASUREMENT:IMMED:REFLEVEL:METHOD PERCENT;  
ABSOLUTE:HIGH 0.0000;LOW 0.0000;MID1  
0.0000;MID2 0.0000;:MEASUREMENT:IMMED:REFLE-  
VEL:PERCENT:HIGH 90.0000;LOW 10.0000;MID1  
50.0000;MID2 50.0000;:MEASUREMENT:IMMED:METHOD  
HISTOGRAM;NOISE HIGH
```

---

# MEASUrement:IMMed:DELaY?

## Description

This query-only command returns information about the immediate delay measurement. This command is equivalent to selecting Measurement Setup from the Measure menu, choosing the Time tab and then clicking the Delay button.

## Group

Measurement

## Syntax

```
MEASUrement:IMMed:DELaY?
```

## Example

```
MEASUrement:IMMed:DELaY?
```

This query might return

```
:MEASUREMENT:IMMED:DELAY:EDGE1 RISE;EDGE2  
RISE; DIRECTION FORWARDS
```



---

# MEASUREMENT:IMMED:DELAY:DIRECTION

## Description

This command sets or returns the starting point and direction that determines the delay "to" edge when taking an immediate delay measurement. Use the MEASUREMENT:IMMED:SOURCE2 command to specify the delay "to" waveform. This command is equivalent to selecting Measurement Setup from the Measure menu, choosing the Time tab, clicking the Delay button to display the delay settings and then clicking the desired Search Direction setting.

## Group

Measurement

## Related Commands

MEASUREMENT:IMMED:SOURCE2

## Syntax 1

```
MEASUREMENT:IMMED:DELAY:DIRECTION {BACK-  
WARDS|FORWARDS}
```

## Syntax 2

```
MEASUREMENT:IMMED:DELAY:DIRECTION?
```

## Arguments

### ■ BACKWARDS

This starts the search at the end of the waveform and looks for the last rising or falling edge in the waveform.

### ■ FORWARDS

This starts the search at the beginning of the waveform and looks for the first rising or falling edge in the waveform.

## Example 1

```
MEASUREMENT:IMMED:DELAY:DIRECTION FORWARDS
```

This command starts searching from the beginning of the waveform record and looks for the first rising or falling edge.

## Example 2

```
MEASUREMENT:IMMED:DELAY:DIRECTION?
```

This query might return

```
:MEASUREMENT:IMMED:DELAY:DIRECTION BACKWARDS
```

indicating that searching begins at the end of the waveform record and looks for the last rising or falling edge.

---

# MEASUrement:IMMed:DELay:EDGE[1]

## Description

This command sets or queries the slope of the edge that is used for the delay "from" waveform when taking an immediate delay measurement. Use the MEASUrement:IMMed:SOURCE1 command to specify the waveform. This command is equivalent to selecting Measurement Set-up from the Measure menu, choosing the Time tab, clicking the Delay button to display the delay settings and then clicking the desired Delay Edge1 setting.

**Note:** If you do not specify a numerical suffix, the source is assumed to be SOURCE 1.

## Group

Measurement

## Related Commands

MEASUrement:IMMed:SOURCE1

## Syntax 1

```
MEASUrement:IMMed:DELay:EDGE[1] {FALL|RISe}
```

## Syntax 2

```
MEASUrement:IMMed:DELay:EDGE[1]?
```

## Arguments

- FALL

This specifies the falling edge.

- RISe

This specifies the rising edge.

## Example 1

```
MEASUrement:IMMed:DELay:EDGE[1] RISe
```

This command specifies that the rising edge be used for the immediate delay measurement.

## Example 2

```
MEASUrement:IMMed:DELay:EDGE[1]?
```

This query might return

```
:MEASUREMENT:IMMED:DELAY:EDGE1 FALL
```

indicating that the falling edge or negative edge of the waveform is used for the immediate delay measurement.

---

# MEASUREMENT:IMMED:DELAY:EDGE2

## Description

This command sets or queries the slope of the edge that is used for the delay "to" waveform when taking an immediate delay measurement. Use the MEASUREMENT:IMMED:SOURCE2 command to specify the waveform. This command is equivalent to selecting Measurement Set-up from the Measure menu, choosing the Time tab, clicking the Delay button to display the delay settings and then clicking the desired Delay Edge2 setting.

## Group

Measurement

## Related Commands

MEASUREMENT:IMMED:SOURCE2

## Syntax 1

```
MEASUREMENT:IMMED:DELAY:EDGE2 {FALL|RISe}
```

## Syntax 2

```
MEASUREMENT:IMMED:DELAY:EDGE2?
```

## Arguments

- FALL

This specifies the falling edge.

- RISe

This specifies the rising edge.

## Example 1

```
MEASUREMENT:IMMED:DELAY:EDGE2 RISe
```

This command specifies that the rising edge be used for the immediate delay measurement.

## Example 2

```
MEASUREMENT:IMMED:DELAY:EDGE2?
```

This query might return

```
:MEASUREMENT:IMMED:DELAY:EDGE2 FALL
```

indicating that the falling edge or negative edge of the waveform is used for the immediate delay measurement.

---

# MEASUREMENT:IMMED:NOISE

## (TDS5000B Series option SM)

### Description

This command sets or queries whether the noise measurement is made on the high or low level of the waveform. Sending this command is equivalent to selecting Ref Levs > Eye > Top Level or Base Level in the Comm tab of the Measurement Setup dialog box. The Eye section is displayed only if you have an eye-pattern or optical measurement defined.

### Group

Measurement

### Syntax 1

```
MEASUREMENT:IMMED:NOISE {HIGH|LOW}
```

### Syntax 2

```
MEASUREMENT:IMMED:NOISE?
```

### Arguments

#### ■ HIGH

This argument causes the measurement for noise to be taken at the high level of the waveform.

#### ■ LOW

This argument causes the measurement for noise to be taken at the low level of the waveform.

### Example 1

```
MEASUREMENT:IMMED:NOISE HIGH
```

This command causes the noise measurement to be made at the high level of the waveform.

### Example 2

```
MEASUREMENT:IMMED:NOISE?
```

This query might return

```
:MEASUREMENT:IMMED:NOISE LOW
```

indicating that the noise measurement will be made at the low level of the waveform

---

# MEASUrement:IMMed:REFLevel:ABSolute:HIGH

## Description

This command sets or queries the high reference level, and is the upper reference level when MEASUrement:IMMed:REFLevel:METHOD is set to Absolute. Note that this command affects the results of rise and fall measurements.

## Group

Measurement

## Related Commands

MEASUrement:IMMed:REFLevel:METHOD

## Syntax 1

MEASUrement:IMMed:REFLevel:ABSolute:HIGH <NR3>

## Syntax 2

MEASUrement:IMMed:REFLevel:ABSolute:HIGH?

## Arguments

- <NRf>

This is the high reference level, in volts. The default is 0.0 V.

## Example 1

```
MEASUREMENT:IMMED:REFLEVEL:ABSOLUTE:HIGH 1.71
```

This command sets the high reference level to 1.71 V.

## Example 2

```
MEASUREMENT:IMMED:REFLEVEL:ABSOLUTE:HIGH?
```

This query might return

```
:MEASUREMENT:IMMED:REFLEVEL:ABSOLUTE:HIGH  
1.7100E+00
```

indicating that the absolute high reference level is set to 1.71 V.

---

# MEASUrement:IMMed:REFLevel:ABSolute:LOW

## Description

This command sets or queries the low reference level, and is the zero percent level when MEASUrement:IMMed:REFLevel:METHOD is set to Absolute. Note that this command affects the results of rise and fall measurements.

## Related Commands

MEASUrement:IMMed:REFLevel:METHOD

## Group

Measurement

## Syntax 1

```
MEASUrement:IMMed:REFLevel:ABSolute:LOW <NR3>
```

## Syntax 2

```
MEASUrement:IMMed:REFLevel:ABSolute:LOW?
```

## Arguments

- <NR3>

This is the low reference level, in volts. The default is 0.0 V.

## Example 1

```
MEASUREMENT:IMMED:REFLEVEL:ABSolute:LOW 0.0 V.
```

This command sets the low reference level to 0.0 V.

## Example 2

```
MEASUREMENT:IMMED:REFLEVEL:ABSolute:LOW?
```

This query might return

```
:MEASUREMENT:MEAS1:REFLEVEL:ABSOLUTE:LOW  
0.0000E+00
```

indicating that the absolute low reference level is set to 0.0 V.

---

# MEASUrement:IMMed:REFLevel:ABSolute:MID[1]

## Description

This command sets or queries the mid reference level, and is the 50% reference level when MEASUrement:IMMed:REFLevel:MEthod is set to Absolute. Note that this command affects the results of period, frequency, delay, and all cyclic measurements.

## Group

Measurement

## Related Commands

MEASUrement:IMMed:REFLevel:MEthod

## Syntax 1

```
MEASUrement:IMMed:REFLevel:ABSolute:MID[1]
<NR3>
```

## Syntax 2

```
MEASUrement:IMMed:REFLevel:ABSolute:MID[1]?
```

## Arguments

- <NR3>

This is the mid reference level, in volts. The default is 0.0 V.

## Example 1

```
MEASUREMENT:IMMED:REFLEVEL:ABSOLUTE:MID[1] .71
```

This command sets the mid reference level to .71 V.

## Example 2

```
MEASUREMENT:IMMED:REFLEVEL:ABSOLUTE:MID[1]?
```

This query might return

```
:MEASUREMENT:IMMED:REFLEVEL:ABSOLUTE:MID
0.7100E+00
```

indicating that the absolute mid1 reference level is set to .71 V.

---

# MEASUrement:IMMed:REFLevel:ABSolute:MID2

## Description

This command sets or queries the mid reference level for the "to" waveform when taking a delay measurement, and is the 50% reference level when MEASUrement:IMMed:REFLevel:METhod is set to Absolute. Note that this command affects the results of delay measurements.

## Group

Measurement

## Related Commands

MEASUrement:IMMed:REFLevel:METhod

## Syntax 1

```
MEASUrement:IMMed:REFLevel:ABSolute:MID2 <NR3>
```

## Syntax 2

```
MEASUrement:IMMed:REFLevel:ABSolute:MID2?
```

## Arguments

- <NR3>

This is the mid2 reference level, in volts. The default is 0.0 V.

## Example 1

```
MEASUREMENT:IMMED:REFLEVEL:ABSOLUTE:MID2 0.5
```

This command sets the mid reference level for the delay waveform to 0.5 V.

## Example 2

```
MEASUREMENT:IMMED:REFLEVEL:ABSOLUTE:MID2?
```

This query might return

```
:MEASUREMENT:IMMED:REFLEVEL:ABSOLUTE:MID2  
0.5000E+00
```

indicating that the absolute mid2 reference level is set to 0.5 V.



---

# MEASUREMENT:IMMED:REFLEVEL:METHOD

## Description

This command specifies or queries the reference level units used for measurement calculations. This command is equivalent to selecting Reference Levels from the Measure menu and then choosing the desired reference level from the Units group box.

## Group

Measurement

## Syntax 1

```
MEASUREMENT:IMMED:REFLEVEL:METHOD {ABSolute|PERCent}
```

## Syntax 2

```
MEASUREMENT:IMMED:REFLEVEL:METHOD?
```

## Arguments

### ■ ABSolute

This specifies that the reference levels are set explicitly using the MEASUREMENT:IMMED:REFLEVEL:ABSolute commands. This method is useful when precise values are required. For instance, when designing to published interface specifications, such as RS-232-C.

### ■ PERCent

This specifies that the reference levels are calculated as a percent relative to HIGH and LOW. The percentages are defined using the MEASUREMENT:IMMED:REFLEVEL:PERCent commands.

## Example 1

```
MEASUREMENT:IMMED:REFLEVEL:METHOD ABSOLUTE
```

This command specifies that explicit user-defined values are used for the reference levels.

## Example 2

```
MEASUREMENT:IMMED:REFLEVEL:METHOD?
```

This query might return

```
:MEASUREMENT:IMMED:REFLEVEL:METHOD PERCENT
```

indicating that the reference level units used are calculated as a percent relative to HIGH and LOW.

---

# MEASUrement:IMMed:REFLevel:PERCent:HIGH

## Description

This command sets or queries the percentage (where 100% is equal to HIGH) used to calculate the high reference level when MEASUrement:IMMed:REFLevel:METHOD is set to Percent. Note that this command affects the results of rise and fall measurements.

## Group

Measurement

## Related Commands

MEASUrement:IMMed:REFLevel:METHOD

## Syntax 1

```
MEASUrement:IMMed:REFLevel:PERCent:HIGH <NR1>
```

## Syntax 2

```
MEASUrement:IMMed:REFLevel:PERCent:HIGH?
```

## Arguments

- <NR1>

This is the high reference level, ranging from 0 to 100%. The default high reference level is 90%.

## Example 1

```
MEASUREMENT:IMMED:REFLEVEL:PERCENT:HIGH 95
```

This command sets the high reference level to 95% of HIGH.

## Example 2

```
MEASUREMENT:IMMED:REFLEVEL:PERCENT:HIGH?
```

This query might return

```
:MEASUREMENT:IMMED:REFLEVEL:PERCENT:HIGH 90
```

indicating that the percentage high reference level is set to 90% of HIGH.

---

# MEASUrement:IMMed:REFLevel:PERCent:LOW

## Description

This command sets or queries the percentage (where 100% is equal to HIGH) used to calculate the low reference level when MEASUrement:IMMed:REFLevel:METHOD is set to Percent. Note that this command affects the results of rise and fall measurements. This command is equivalent to selecting Reference Levels from the Measure menu and then entering the Percentage Low Ref value.

## Group

Measurement

## Related Commands

MEASUrement:IMMed:REFLevel:METHOD

## Syntax 1

```
MEASUrement:IMMed:REFLevel:PERCent:LOW <NR1>
```

## Syntax 2

```
MEASUrement:IMMed:REFLevel:PERCent:LOW?
```

## Arguments

- <NR1>

This is the low reference level, ranging from 0 to 100%. The default low reference level is 10%.

## Example 1

```
MEASUREMENT:IMMED:REFLEVEL:PERCENT:LOW 15
```

This command sets the low reference level to 15% of HIGH.

## Example 2

```
MEASUREMENT:IMMED:REFLEVEL:PERCENT:LOW?
```

This query might return

```
:MEASUREMENT:IMMED:REFLEVEL:PERCENT:LOW 10
```

indicating that the percentage low reference level is set to 10% of HIGH.

---

# MEASUrement:IMMed:REFLevel:PERCent:MID[1]

## Description

This command sets or queries the percentage (where 100% is equal to HIGH) used to calculate the mid reference level when MEASUrement:IMMed:REFLevel:METHOD is set to Percent. Note that this command affects the results of period, frequency, delay, and all cyclic measurements.

## Group

Measurement

## Related Commands

MEASUrement:IMMed:REFLevel:METHOD

## Syntax 1

```
MEASUrement:IMMed:REFLevel:PERCent:MID[1]
<NR1>
```

## Syntax 2

```
MEASUrement:IMMed:REFLevel:PERCent:MID[1]?
```

## Arguments

- <NR1>

This is the mid reference level, ranging from 0 to 100%. The default mid reference level is 50%.

## Example 1

```
MEASUREMENT:IMMED:REFLEVEL:PERCENT:MID[1] 60
```

This command sets the mid reference level to 60% of HIGH.

## Example 2

```
MEASUREMENT:IMMED:REFLEVEL:PERCENT:MID[1]?
```

This query might return

```
:MEASUREMENT:IMMED:REFLEVEL:PERCENT:MID 65
```

Indicating that the percentage mid reference level is set to 65% of HIGH.

---

# MEASUrement:IMMed:REFLevel:PERCent:MID2

## Description

This command sets or queries the percentage (where 100% is equal to HIGH) used to calculate the mid reference level for the second waveform specified when MEASUrement:IMMed:REFLevel:MEthod is set to Percent. Note that this command affects the results of delay measurements.

## Group

Measurement

## Related Commands

MEASUrement:IMMed:REFLevel:MEthod

## Syntax 1

```
MEASUrement:IMMed:REFLevel:PERCent:MID2 <NR1>
```

## Syntax 2

```
MEASUrement:IMMed:REFLevel:PERCent:MID2?
```

## Arguments

- <NR1>

This is the mid reference level, ranging from 0 to 100%. The default mid reference level is 50%.

## Example 1

```
MEASUREMENT:IMMED:REFLEVEL:PERCENT:MID2 40
```

This command sets the mid2 reference level to 40% of HIGH.

## Example 2

```
MEASUREMENT:IMMED:REFLEVEL:PERCENT:MID2?
```

This query might return

```
:MEASUREMENT:IMMED:REFLEVEL:PERCENT:MID2 45
```

Indicating that the percentage mid2 reference level is set to 45% of HIGH.

---

# MEASUrement:IMMed:REFLevel?

## Description

This query-only command returns the reference level settings for the immediate measurement. It returns them in the following order: ABSOLUTE and then PERCENT for individual user measurements.

## Group

Measurement

## Syntax

```
MEASUrement:IMMed:REFLevel?
```

## Example

```
MEASUREMENT:IMMed:REFLEVEL?
```

---

# MEASUREMENT:SOURCE<x>:SIGTYPE

## (TDS5000B Series option SM)

### Description

This command sets or queries the type of input signal used for measurement SOURCE<x>, 1 or 2.

To ensure accurate measurements, use this command to specify the input-signal type for the measurement source.

### Group

Measurement

### Syntax 1

```
MEASUREMENT:SOURCe<x>:SIGType {PULse|EYE}
```

### Syntax 2

```
MEASUREMENT:SOURce<x>:SIGType?
```

### Arguments

- PULSE

Use this signal type for generic signals that are not associated with synchronous communications standards.

- EYE

Use this signal type for synchronous-communication signals with NRZ-like characteristics (nonreturn to zero).

### Example 1

```
MEASUREMENT:SOURce1:SIGType PULse
```

This command selects Pulse as the signal type for source 1.

### Example 2

```
MEASUREMENT:SOURce<x>:SIGType?
```

This query might return

```
:MEASUREMENT:SOURce1:SIGType EYE
```

indicating that the signal type for source 1 is EYE.

---

# MEASUrement:IMMed:SOURCE[1]

## Description

This command sets or queries the source for all single channel immediate measurements and specifies the source to measure "from" when taking an immediate delay measurement or phase measurement. This command is equivalent to selecting Measurement Setup from the Measure menu, choosing the Time tab, clicking the Delay button to display the delay settings and then clicking the desired Source1 (From) setting.

**Note:** If you do not specify a numerical suffix, the source is assumed to be SOURCE 1.

## Group

Measurement

## Related Commands

MEASUrement:IMMed:SOURCE2

## Syntax 1

```
MEASUrement:IMMed:SOURCE[ 1 ]  
{CH<x> | MATH<y> | REF<x> | HISTogram}
```

## Syntax 2

```
MEASUrement:IMMed:SOURCE[ 1 ]?
```

## Arguments

### ■ CH<x>

This is an input channel waveform. The x variable can be expressed as an integer ranging from 1 through 4 for 4-channel models and 1 to 2 for 2-channel models.

### ■ MATH<y>

This is a math waveform. The y variable can be expressed as an integer ranging from 1 through 4 for 4-channel models and 1 to 2 for 2-channel models.

### ■ REF<X>

This is a reference waveform. The x variable can be expressed as an integer ranging from 1 through 4 for 4-channel models and 1 to 2 for 2-channel models.

### ■ HISTogram

This indicates histogram as the object to be measured.

## Example 1

```
MEASUrement:IMMed:SOURCE[ 1 ] MATH1
```

This command specifies Math1 as the immediate measurement source .



---

## Example 2

`MEASUrement:IMMed:SOURCE[1]?`

This query might return

`:MEASUREMENT:IMMED:SOURCE1 CH3`

indicating that channel 3 is the immediate measurement source.

---

# MEASUrement:IMMed:SOURCE2

## Description

This command sets or queries the source to measure "to" for phase or delay immediate measurements. This command is equivalent to selecting Measurement Setup from the Measure menu, choosing the Time tab, clicking the Delay button to display the delay settings and then clicking the desired Source2 (To) setting.

Tip: Source2 measurements only apply to phase and delay measurement types, which require both a target (Source1) and reference (Source2) source.

## Group

Measurement

## Related Commands

MEASUrement:IMMed:SOURCE[1]

## Syntax 1

```
MEASUrement:IMMed:SOURCE2  
{CH<x>|MATH<y>|REF<x>|HISTogram}
```

## Syntax 2

```
MEASUrement:IMMed:SOURCE2?
```

## Arguments

- CH<x>

This is an input channel waveform. The x variable can be expressed as an integer ranging from 1 through 4 for 4-channel models and 1 to 2 for 2-channel models.

- MATH<y>

This is a math waveform. The y variable can be expressed as an integer ranging from 1 through 4 for 4-channel models and 1 to 2 for 2-channel models.

- REF<x>

This is a reference waveform. The x variable can be expressed as an integer ranging from 1 through 4 for 4-channel models and 1 to 2 for 2-channel models.

- HISTogram

This indicates histogram as the object to be measured.

## Example 1

```
MEASUrement:IMMed:SOURCE2 REF3
```

This command sets the waveform in reference memory location 3 as the delay "to" source when making delay measurements.

---

## Example 2

MEASUrement:IMMed:SOURCE2?

This query might return

:MEASUREMENT:IMMED:SOURCE2 MATH1

indicating that Math1 is the immediate measurement source.

---

# MEASurement:IMMed:SOUrce<x>:SIGType

## (TDS5000B Series option SM)

### Description

This command sets or queries the type of input signal used for measurement SOURCE<x>, 1 or 2.

To ensure accurate measurements, use this command to specify the input-signal type for the measurement source.

### Group

Measurement

### Syntax 1

```
MEASurement:IMMed:SOUrce<x>:SIGType  
{PULse|EYE}
```

### Syntax 2

```
MEASurement:IMMed:SOUrce<x>:SIGType?
```

### Arguments

- PULSE

Use this signal type for generic signals that are not associated with synchronous communications standards.

- EYE

Use this signal type for synchronous-communication signals with NRZ-like characteristics (nonreturn to zero).

### Example 1

```
MEASurement:IMMed:SOUrce1:SIGType PULse
```

This command selects Pulse as the signal type for source 1.

### Example 2

```
MEASurement:IMMed:SOUrce<x>:SIGType?
```

This query might return

```
:MEASUREMENT:IMMED:SOURCE1:SIGTYPE EYE
```

indicating that the signal type for source 1 is EYE.

---

# MEASUREMENT:IMMED:TYPE

## Description

This command sets or queries the immediate measurement type.

## Group

Measurement

## Syntax 1

```
MEASUREMENT:IMMED:TYPE {AMPLITUDE|AREa|  
BURst|CAREa|CMEan|CRMs|DELay|DISTDUty|  
EXTINCTDB|EXTINCTPCT|EXTINCTRATIO|  
EYEHeight|EYEWidth|FALL|FREQuency|HIGH|  
HITs|LOW|MAXimum|MEAN|MEDIan|MINIMUM|  
NCROss|NDUty|NOVershoot|NWIdth|PBASe|  
PCROss|PCTCROss|PDUty|PEAKHits|PERIOd|  
PHASe|PK2Pk|PKPKJitter|PKPKNoise|  
POVershoot|PTOT|PWidth|QFACTOR|RISe|  
RMS|RMSJitter|PMSNoise|SIGMA1|SIGMA2|  
SIGMA3|SIXSigmajit|SNRatio|STDdev|  
UNDEFINED|WAVEFORMS}
```

## Syntax 2

```
MEASUREMENT:IMMED:TYPE?
```

## Arguments

### ■ AMPLITUDE

This measures the amplitude of the selected waveform. In other words, it measures the high value less the low value measured over the entire waveform or gated region.

Amplitude = High - Low

### ■ AREa

This measures the voltage over time. The area is over the entire waveform or gated region and is measured in volt-seconds. The area measured above the ground is positive, while the area below ground is negative.

### ■ BURst

The burst width measures the duration of a burst. The measurement is made over the entire waveform or gated region.

### ■ CAREa

The cycle area measures the voltage over time. In other words, it measures, in volt-seconds, the area over the first cycle in the waveform or the first cycle in the gated region. The area measured above the common reference point is positive, while the area below the common reference point is negative.

### ■ CMEan

This measures the arithmetic mean over the first cycle in the waveform or the first cycle in the gated region.

---

- **CRMs**

This measures the true Root Mean Square voltage over the first cycle in the waveform or the first cycle in the gated region.

- **DELaY**

This measures the time between the middle reference (default = 50%) amplitude point of the source waveform and the destination waveform.

- **DISTDUtY**

The duty cycle distortion measures the time between the falling edge and the rising edge of the eye pattern at the mid reference level.

It is the peak-to-peak time variation of the first eye crossing measured at the mid-reference as a percent of the eye period.

- **EXTINCTDB**

This (optical waveform (eye diagram) measurement) measures its extinction ratio. Extinction Ratio (dB) measures the ratio of the average power levels for the logic High to the logic Low of an optical waveform and expresses the result in dB.

This measurement only works for fast acquisition signals or a reference waveform saved in fast acquisition mode.

Extinction dB = 10 X (log 10 (High / Low))

- **EXTINCTPCT**

This (optical measurement) measures the extinction ratio of the selected optical waveform. Extinction Ratio (%) measures the ratio of the average power levels for the logic Low (off) to the logic (High) (on) of an optical waveform and expresses the result in percent.

This measurement only works for fast acquisition signals or a reference waveform saved in fast acquisition mode.

Extinction % = 100.0 x (Low / High)

- **EXTINCTRATIO**

This (optical measurement) measures the extinction ratio of the selected optical waveform. Extinction Ratio measures the ratio of the average power levels for the logic High to the logic Low of an optical waveform and expresses the result without units.

This measurement only works for fast acquisition signals or a reference waveform saved in fast acquisition mode.

Extinction Ratio = (High / Low), ratios greater than 100 or less than 1 generate errors; low must be greater than or equal to 1 microwatt.

- **EYEHeight**

This measures the vertical opening of an eye diagram in volts.

- **EYEWidth**

This measures the width of an eye diagram in seconds.

---

- **FALL**

This measures the time taken for the falling edge of the first pulse in the waveform or gated region to fall from a high reference value (default is 90%) to a low reference value (default is 10%).

- **FREQuency**

This measures the first cycle in the waveform or gated region. Frequency is the reciprocal of the period and is measured in hertz (Hz), where 1 Hz = 1 cycle per second.

- **HIGH**

This measures the High reference (100% level, sometimes called Top-line) of a waveform.

You can also limit the High measurement (normally taken over the entire waveform record) to a gated region on the waveform.

- **HITs**

The hits-in-box histogram measurement measures the number of points in or on the histogram box.

- **LOW**

This measures the Low reference (0% level, sometimes called Baseline) of a waveform.

- **MAXimum**

This finds the maximum amplitude. This value is the most positive peak voltage found. It is measured over the entire waveform or gated region.

When histogram is selected with the MEASUREMENT:METHOD command, the maximum measurement measures the voltage of the highest nonzero bin in vertical histograms or the time of the rightmost bin in horizontal histograms.

- **MEAN**

This amplitude measurement finds the arithmetic mean over the entire waveform or gated region.

When histogram is selected with the MEASUREMENT:METHOD command, the mean measurement measures the average of all acquired points within or on the histogram.

- **MEDian**

This (histogram measurement) measures the middle point of the histogram box. Half of all acquired points within or on the histogram box are less than this value and half are greater than this value.

- **MINimum**

This amplitude measurement finds the minimum amplitude. This value is typically the most negative peak voltage. It is measured over the entire waveform or gated region.

When histogram is selected with the MEASUREMENT:METHOD command, the minimum measurement measures the lowest nonzero bin in vertical histograms or the time of the leftmost nonzero bin in the horizontal histograms.

---

- **NCROss**

This (timing measurement) measures the time from the trigger point to the first falling edge of the waveform or gated region. The distance (time) is measured at the middle reference amplitude point of the signal.

- **NDUty**

The negative duty cycle is the ratio of the negative pulse width to the signal period, expressed as a percentage. The duty cycle is measured on the first cycle in the waveform or gated region.

Negative Duty Cycle = (Negative Width) / Period \* 100%

- **NOVershoot**

The negative overshoot amplitude measurement finds the negative overshoot value over the entire waveform or gated region.

Negative Overshoot = (Low – Minimum) / Amplitude \* 100%

- **NWIdth**

The negative width measurement is the distance (time) between the middle reference (default = 50%) amplitude points of a negative pulse. The measurement is made on the first pulse in the waveform or gated region.

- **PBASe**

This measures the base value used in extinction ratio measurements.

- **PCROss**

This (timing measurement) measures the time from the trigger point to the first positive edge of the waveform or gated region. The distance (time) is measured at the middle reference amplitude point of the signal.

- **PCTCROss**

The crossing percent (optical measurement) measures the location of the eye crossing point expressed as a percentage of EYEHeight.

Crossing percent = 100 \* [(eye-crossing-point – PBASe)/(PTOP – PBASe)]

- **PDUty**

The positive duty cycle is the ratio of the positive pulse width to the signal period, expressed as a percentage. It is measured on the first cycle in the waveform or gated region.

Positive Duty Cycle = (Positive Width)/Period \* 100%

- **PEAKHits**

This (histogram measurement) measures the number of points in the largest bin of the histogram.

- **PERIod**

This timing measurement is the time required to complete the first cycle in a waveform or gated region. Period is the reciprocal of frequency and is measured in seconds.



---

#### ■ PHase

This measures the phase difference (amount of time a waveform leads or lags the reference waveform) between two waveforms. The measurement is made between the middle reference points of the two waveforms and is expressed in degrees, where 360° represents one waveform cycle.

#### ■ PK2Pk

This peak-to-peak amplitude measurement finds the absolute difference between the maximum and minimum amplitude in the entire waveform or gated region.

When histogram is selected with the MEASUREMENT:METHoD command, the PK2Pk measurement measures the histogram's peak to peak difference.

#### ■ PKPKJitter

This measures the variance (minimum and maximum values) in the time locations of the cross point.

#### ■ PKPKNoise

This measures the peak-to-peak noise on a waveform at the mid reference level.

#### ■ POvershoot

The positive overshoot amplitude measurement finds the positive overshoot value over the entire waveform or gated region.

Positive Overshoot = (Maximum – High) / Amplitude \* 100%

#### ■ PTOT

This measures the top value used in extinction ratio measurements.

#### ■ PWidth

The positive width is the distance (time) between the middle reference (default = 50%) amplitude points of a positive pulse. The measurement is made on the first pulse in the waveform or gated region.

#### ■ QFACTOR

This measures the quality factor. The Q factor is a figure of merit for an eye diagram, which indicates the vertical eye opening relative to the noise at the low and high logic levels. It is the ratio of the eye size to noise.

#### ■ RISE

This timing measurement finds the rise time of the waveform. The rise time is the time it takes for the leading edge of the first pulse encountered to rise from a low reference value (default is 10%) to a high reference value (default is 90%).

#### ■ RMS

This amplitude measurement finds the true Root Mean Square voltage in the entire waveform or gated region.

---

- **RMSJitter**

This measures the variance in the time locations of the cross point. The RMS jitter is defined as one standard deviation at the cross point.

- **RMSNoise**

This measures the Root Mean Square noise amplitude on a waveform at the mid reference level.

- **SIGMA1**

This (histogram measurement) measures the percentage of points in the histogram that are within one standard deviation of the histogram mean.

- **SIGMA2**

This (histogram measurement) measures the percentage of points in the histogram that are within two standard deviations of the histogram mean.

- **SIGMA3**

This (histogram measurement) measures the percentage of points in the histogram that are within three standard deviations of the histogram mean.

- **SIXSigmajit**

This (histogram measurement) is six \* RMSJitter.

- **SNRatio**

This measures the signal-to-noise ratio. The signal-to-noise ratio is the amplitude of a noise rejection band centered on the mid level.

- **STDdev**

This measures the standard deviation (Root Mean Square (RMS) deviation) of all acquired points within or on the histogram box.

- **UNDEFINED**

This is the default measurement type, which indicates that no measurement type is specified. Once a measurement type is chosen, it can be cleared using this argument.

- **WAVEFORMS**

The waveform count (histogram measurement) measures the number of waveforms used to calculate the histogram.

### **Example 1**

```
MEASUREMENT:IMMED:TYPE FREQUENCY
```

This defines the immediate measurement to be a frequency measurement.

### **Example 2**

```
MEASUREMENT:IMMED:TYPE?
```

---

This query might return

`:MEASUREMENT:IMMED:TYPE RMS`

indicating that the immediate measurement is the true Root Mean Square voltage.

---

# MEASUrement:IMMed:UNIts?

## Description

This query-only command returns the units of the immediate measurement.

## Group

Measurement

## Syntax

```
MEASUrement:IMMed:UNIts?
```

## Example

```
MEASUrement:IMMed:UNIts?
```

This query might return

```
:MEASUREMENT:IMMED:UNIts "s"
```

indicating that units for the immediate measurement are in seconds.

---

# MEASUrement:IMMed:VALue?

## Description

This query-only command returns the value of the measurement specified by the MEASUrement:IMMed:TYPe command. The measurement is immediately taken on the source(s) specified by a MEASUrement:IMMed:SOURCE command.

**Note:** A change to HORizontal:MAIn:SCALe or CH<x>:SCALe will not necessarily have taken affect if followed by this command.

## Group

Measurement

## Related Commands

MEASUrement:IMMed:TYPe, MEASUrement:IMMed:SOURCE, \*ESR?, ALLEV?

## Syntax

MEASUrement:IMMed:VALue?

## Example

MEASUrement:IMMed:VALue?

This query might return

:MEASUREMENT:IMMED:VALUE 9.9000E+37

If the measurement has an error or warning associated with it, then an item is added to the error queue. The error can be checked for with the \*ESR? and ALLEV? commands.

---

# MEASUrement:MEAS<x>:COUNT?

## Description

This query-only command returns the number of values accumulated for this measurement since the last statistical reset. Some values may have been ignored because they generated an error. Measurements are specified by x, which ranges from 1 through 8.

## Group

Measurement

## Syntax

```
MEASUrement:MEAS<x>:COUNT?
```

## Example

```
MEASUrement:MEAS3:COUNT?
```

This query might return

```
:MEASUREMENT:MEAS3:COUNT 3.247000000E+03.
```

---

# MEASUREMENT:MEAS<x>:MAXIMUM?

## Description

This query-only command returns the maximum value found for this measurement since the last statistical reset. Measurements are specified by x, which ranges from 1 through 8.

## Group

Measurement

## Syntax

```
MEASUREMENT:MEAS<x>:MAXIMUM?
```

## Example

```
MEASUREMENT:MEAS3:MAXIMUM?
```

This query might return

```
:MEASUREMENT:MEAS3:MAXIMUM 4.18E-9.
```

---

# MEASUREMENT:MEAS<x>:MEAN?

## Description

This query-only command returns the mean value accumulated for this measurement since the last statistical reset. Measurements are specified by x, which ranges from 1 through 8.

## Group

Measurement

## Syntax

```
MEASUREMENT:MEAS<x>:MEAN?
```

## Example

```
MEASUREMENT:MEAS1:MEAN?
```

This query might return

```
:MEASUREMENT:MEAS1:MEAN 514.71E-09.
```



---

# MEASUREMENT:MEAS<x>:MINIMUM?

## Description

This query-only command returns the minimum value found for this measurement since the last statistical reset. Measurements are specified by x, which ranges from 1 through 8.

## Group

Measurement

## Syntax

```
MEASUREMENT:MEAS<x>:MINIMUM?
```

## Example

```
MEASUREMENT:MEAS1:MINIMUM?
```

This query might return

```
:MEASUREMENT:MEAS1:MINIMUM 1.75E-09.
```

---

# MEASUREMENT:MEAS<x>:NOISE

## (TDS5000B Series option SM)

### Description

This command sets or queries whether the noise measurement is made on the high or low level of the waveform. Sending this command is equivalent to selecting Ref Levs > Eye > Top Level or Base Level in the Comm tab of the Measurement Setup dialog box. The Eye section is displayed only if you have an eye-pattern or optical measurement defined. Measurements are specified by x, which ranges from 1 through 8.

### Group

Measurement

### Syntax 1

```
MEASUREMENT:MEAS<x>:NOISE {HIGH|LOW}
```

### Syntax 2

```
MEASUREMENT:MEAS<x>:NOISE?
```

### Argument

#### ■ HIGH

This argument causes the measurement for noise to be taken at the high level of the waveform.

#### ■ LOW

This argument causes the measurement for noise to be taken at the low level of the waveform.

### Example 1

```
MEASUREMENT:MEAS1:NOISE HIGH
```

This command causes the noise measurement to be made at the high level of the waveform for Measurement 1.

### Example 2

```
MEASUREMENT:MEAS2:NOISE?
```

This query might return

```
:MEASUREMENT:MEAS2:NOISE LOW
```

indicating that Measurement 2 will measure noise at the low level of the waveform.

---

# MEASUrement:MEAS<x>:STDdev?

## Description

This query-only command returns the standard deviation of values accumulated for this measurement since the last statistical reset. Measurements are specified by x, which ranges from 1 through 8.

## Group

Measurement

## Syntax

```
MEASUrement:MEAS<x>:STDdev?
```

## Example

```
MEASUrement:MEAS1:STDdev?
```

This query might return

```
:MEASUREMENT:MEAS1:STDDEV 21.0E-12.
```

---

# MEASUrement:MEAS<x>:DELay:DIREction

## Description

This command sets or queries the starting point and direction that determines the delay "to" edge when taking a delay measurement. Use the MEASUrement:MEAS<x>:SOURCE2 command to specify the waveform. This command is equivalent to selecting Time from the Measure menu, choosing Delay from the drop-down list and then clicking the desired Search Direction setting. Measurements are specified by x, which ranges from 1 through 8.

## Group

Measurement

## Related Commands

MEASUrement:MEAS<x>:SOURCE2

## Syntax 1

```
MEASUrement:MEAS<x>:DELay:DIREction {BACK-  
Wards|FORWards}
```

## Syntax 2

```
MEASUrement:MEAS<x>:DELay:DIREction?
```

## Arguments

### ■ BACKWards

This means that the search starts at the end of the waveform and looks for the last rising or falling edge in the waveform. Use the MEASUrement:MEAS<x>:DELay:EDGE2 command to specify the slope of the edge.

### ■ FORWards

This means that the search starts at the beginning of the waveform and looks for the first rising or falling edge in the waveform. Use the MEASUrement:MEAS<x>:DELay:EDGE2 command to specify the slope of the edge.

## Example 1

```
MEASUrement:MEAS3:DELay:DIREction BACKWards
```

This command starts searching from the end of the waveform record.

## Example 2

```
MEASUrement:MEAS3DELay:DIREction?
```

This query might return

```
:MEASUREMENT:MEAS3:DELAY:DIRECTION BACKWARDS
```

indicating that the current search direction is backwards.

---

# MEASUREMENT:MEAS<x>:DELAY:EDGE[1]

## Description

This command sets or queries the slope of the edge that is used for the delay "from" waveform when taking a delay measurement. Use the MEASUREMENT:MEAS<x>:SOURCE[1] command to specify the waveform. This command is equivalent to selecting Time from the Measure menu, choosing Delay from the drop-down list and clicking the desired Delay Edge1 setting. Measurements are specified by x, which ranges from 1 through 8.

## Group

Measurement

## Related Commands

MEASUREMENT:MEAS<x>:SOURCE[1]

## Syntax 1

MEASUREMENT:MEAS<x>:DELAY:EDGE[1] {FALL|RISe}

## Syntax 2

MEASUREMENT:MEAS<x>:DELAY:EDGE[1]?

## Arguments

- FALL

This specifies the falling edge of the waveform.

- RISe

This specifies the rising edge of the waveform.

## Example 1

MEASUREMENT:MEAS3:DELAY:EDGE[1] RISe

This command specifies that the rising edge of the waveform be used for measurement 3.

## Example 2

MEASUREMENT:MEAS1:DELAY:EDGE[1]?

This query might return

:MEASUREMENT:MEAS1:DELAY:EDGE[1] FALL

indicating that the falling edge of the waveform is being used for measurement 1.

---

# MEASUrement:MEAS<x>:DELay:EDGE2

## Description

This command sets or queries the slope of the edge that is used for the delay "to" waveform when taking a delay measurement. Use the MEASUrement:MEAS<x>:SOURCE2 command to specify the waveform. This command is equivalent to selecting Time from the Measure menu, choosing Delay from the drop-down list and then clicking the desired Delay Edge2 setting. Measurements are specified by x, which ranges from 1 through 8.

## Group

Measurement

## Related Commands

MEASUrement:MEAS<x>:SOURCE2

## Syntax 1

MEASUrement:MEAS<x>:DELay:EDGE2 {FALL|RISe}

## Syntax 2

MEASUrement:MEAS<x>:DELay:EDGE2?

## Arguments

- FALL

This specifies the falling edge of the waveform.

- RISe

This specifies the rising edge of the waveform.

## Example 1

MEASUrement:MEAS3:DELay:EDGE2 RISe

This command specifies that the rising edge of the waveform be used for measurement 3.

## Example 2

MEASUrement:MEAS1:DELay:EDGE2?

This query might return

:MEASUREMENT:MEAS1:DELAY:EDGE2 FALL

indicating that the falling edge of the waveform is being used for measurement 1.

---

# MEASUREMENT:MEAS<x>:DELAY?

## Description

This query-only command returns the delay measurement parameters for the measurement specified by <x>, which ranges from 1 through 8. Measurement parameters are presented in the following order: Edge1, Edge2, and Direction. This query command is equivalent to selecting Time from the Measure menu, choosing Delay and viewing the Delay Edge and Search Direction settings.

## Group

Measurement

## Syntax 1

```
MEASUREMENT:MEAS<x>:DELAY?
```

## Example

```
MEASUREMENT:MEAS3?
```

This query might return

```
:MEASUREMENT:MEAS3:DELAY:EDGE1 RISE;  
EDGE2 RISE;DIRECTION FORWARDS.
```

---

# MEASUrement:MEAS<x>:REFLevel:ABSolute:HIGH

## Description

This command sets or queries the high reference level, and is the upper reference level when MEASUrement:MEAS<x>:REFLevel:METHOD is set to Absolute. Measurements are specified by x, which ranges from 1 through 8.

**Note:** This command affects the results of rise and fall measurements

## Group

Measurement

## Related Commands

MEASUrement:MEAS<x>:REFLevel:METHOD

## Syntax 1

```
MEASUrement:MEAS<x>:REFLevel:ABSolute:HIGH  
<NR3>
```

## Syntax 2

```
MEASUrement:MEAS<x>:REFLevel:ABSolute:HIGH?
```

## Arguments

- <NR3>

This is the high reference level, in volts. The default is 0.0 V.

## Example 1

```
MEASUREMENT:MEAS1:REFLEVEL:ABSOLUTE:HIGH 1.71
```

This command sets the high reference level to 1.71 V.

## Example 2

```
MEASUREMENT:MEAS1:REFLEVEL:ABSOLUTE:HIGH?
```

This query might return

```
:MEASUREMENT:MEAS1:REFLEVEL:ABSOLUTE:HIGH  
1.7100E+00
```

indicating that the absolute high reference level is set to 1.71 V.



---

# MEASUREMENT:MEAS<x>:REFLEVEL:ABSOLUTE:LOW

## Description

This command sets or queries the low reference level, and is the lower reference level when MEASUREMENT:MEAS<x>:REFLEVEL:METHOD is set to Absolute. Measurements are specified by x, which ranges from 1 through 8.

**Note:** This command affects the results of rise and fall measurements.

## Group

Measurement

## Related Commands

MEASUREMENT:MEAS<x>:REFLEVEL:METHOD

## Syntax 1

```
MEASUREMENT:MEAS<x>:REFLEVEL:ABSOLUTE:LOW  
<NR3>
```

## Syntax 2

```
MEASUREMENT:MEAS<x>:REFLEVEL:ABSOLUTE:LOW?
```

## Arguments

- <NR3>

This is the low reference level, in volts. The default is 0.0 V.

## Example 1

```
MEASUREMENT:REFLEVEL:ABSOLUTE:LOW 0.0 V.
```

This command sets the low reference level to 0.0 V.

## Example 2

```
MEASUREMENT:MEAS1:REFLEVEL:ABSOLUTE:LOW?
```

This query might return

```
:MEASUREMENT:MEAS1:REFLEVEL:ABSOLUTE:LOW  
0.0000E+00
```

indicating that the absolute low reference level is set to 0.0 V.

---

# MEASUrement:MEAS<x>:REFLevel:ABSolute:MID[1]

## Description

This command sets or queries the mid reference level, and is the 50% reference level when MEASUrement:MEAS<x>:REFLevel:METHOD is set to Absolute. Measurements are specified by x, which ranges from 1 through 8.

**Note:** This command affects the results of period, frequency, delay, and all cyclic measurements.

## Group

Measurement

## Related Commands

MEASUrement:MEAS<x>:REFLevel:METHOD

## Syntax 1

```
MEASUrement:MEAS<x>:REFLevel:ABSolute:MID[1]
<NR3>
```

## Syntax 2

```
MEASUrement:MEAS<x>:REFLevel:ABSolute:MID[1]?
```

## Arguments

- <NR3>

This is the mid reference level, in volts. The default is 0.0 V.

## Example 1

```
MEASUREMENT:MEAS1:REFLEVEL:ABSOLUTE:MID[1] .71
```

This command sets the mid reference level to .71 V.

## Example 2

```
MEASUREMENT:MEAS1:REFLEVEL:ABSOLUTE:MID[1]?
```

This query might return

```
:MEASUREMENT:MEAS1:REFLEVEL:ABSOLUTE:MID
0.7100E+00
```

indicating that the absolute mid1 reference level is set to .71 V.

---

# MEASUREMENT:MEAS<x>:REFLevel:ABSolute:MID2

## Description

This command sets or queries the mid reference level for the "to" waveform when taking a delay measurement, and is the 50% reference level when MEASUREMENT:MEAS<x>:REFLevel:METHOD is set to Absolute. Measurements are specified by x, which ranges from 1 through 8.

**Note:** This command affects the results of delay measurements.

## Group

Measurement

## Related Commands

MEASUREMENT:MEAS<x>:REFLevel:METHOD

## Syntax 1

```
MEASUREMENT:MEAS<x>:REFLevel:ABSolute:MID2  
<NR3>
```

## Syntax 2

```
MEASUREMENT:MEAS<x>:REFLevel:ABSolute:MID2?
```

## Arguments

- <NR3>

This is the mid reference level, in volts. The default is 0.0 V.

## Example 1

```
MEASUREMENT:MEAS1:REFLEVEL:ABSOLUTE:MID2 0.5
```

This command sets the mid reference level for the delay waveform to 0.5 V.

## Example 2

```
MEASUREMENT:MEAS1:REFLEVEL:ABSOLUTE:MID2?
```

This query might return

```
:MEASUREMENT:MEAS1:REFLEVEL:ABSOLUTE:MID2  
0.5000E+00
```

indicating that the absolute mid2 reference level is set to 0.5 V.

---

# MEASUREMENT:MEAS<x>:REFLEVEL:METHOd

## Description

This command specifies or queries the reference level units used for measurement calculations. This command is equivalent to selecting Reference Levels from the Measure menu and then choosing the desired reference level from the Units group box. Measurements are specified by x, which ranges from 1 through 8.

## Group

Measurement

## Syntax 1

```
MEASUREMENT:MEAS<x>:REFLEVEL:METHOd {ABSolute|PERCent}
```

## Syntax 2

```
MEASUREMENT:MEAS<x>:REFLEVEL:METHOd?
```

## Arguments

### ■ ABSolute

This specifies that the reference levels are set explicitly using the MEASUREMENT:MEAS<x>:REFLEVEL:ABSolute commands. This method is useful when precise values are required. For instance, when designing to published interface specifications, such as RS-232-C.

### ■ PERCent

This specifies that the reference levels are calculated as a percent relative to HIGH and LOW. The percentages are defined using the MEASUREMENT:MEAS<x>:REFLEVEL:PERCent commands.

## Example 1

```
MEASUREMENT:MEAS1:REFLEVEL:METHOd ABSOLUTE
```

This command specifies that explicit user-defined values are used for the reference levels.

## Example 2

```
MEASUREMENT:MEAS1:REFLEVEL:METHOd?
```

This query might return

```
:MEASUREMENT:MEAS1:REFLEVEL:METHOd PERCENT
```

indicating that the reference level units used are calculated as a percent relative to HIGH and LOW.

---

# MEASUREMENT:MEAS<x>:REFLEVEL:PERCENT:HIGH

## Description

This command sets or queries the percentage (where 100% is equal to HIGH) used to calculate the high reference level when MEASUREMENT:MEAS<x>:REFLEVEL:METHOD is set to Percent. Measurements are specified by x, which ranges from 1 through 8.

**Note:** This command affects the results of rise and fall measurements.

## Group

Measurement

## Related Commands

MEASUREMENT:MEAS<x>:REFLEVEL:METHOD

## Syntax 1

```
MEASUREMENT:MEAS<x>:REFLEVEL:PERCENT:HIGH  
<NR3>
```

## Syntax 2

```
MEASUREMENT:MEAS<x>:REFLEVEL:PERCENT:HIGH?
```

## Arguments

- <NR3>

This is the high reference level, ranging from 0 to 100%. The default high reference level is 90%.

## Example 1

```
MEASUREMENT:MEAS1:REFLEVEL:PERCENT:HIGH 95
```

This command sets the high reference level to 95% of HIGH.

## Example 2

```
MEASUREMENT:MEAS1:REFLEVEL:PERCENT:HIGH?
```

This query might return

```
:MEASUREMENT:MEAS1:REFLEVEL:PERCENT:HIGH 90
```

indicating that the percentage high reference level is set to 90% of HIGH.

---

# MEASUREMENT:MEAS<x>:REFLEVEL:PERCENT:LOW

## Description

This command sets or queries the percentage (where 100% is equal to HIGH) used to calculate the low reference level when MEASUREMENT:MEAS<x>:REFLEVEL:METHOD is set to Percent. This command is equivalent to selecting Reference Levels from the Measure menu, and then entering the Percentage Low Ref value. Measurements are specified by x, which ranges from 1 through 8.

**Note:** This command affects the results of rise and fall measurements.

## Group

Measurement

## Related Commands

MEASUREMENT:MEAS<x>:REFLEVEL:METHOD

## Syntax 1

MEASUREMENT:MEAS<x>:REFLEVEL:PERCENT:LOW <NR3>

## Syntax 2

MEASUREMENT:MEAS<x>:REFLEVEL:PERCENT:LOW?

## Arguments

- <NR3>

This is the low reference level, ranging from 0 to 100%. The default low reference level is 10%.

## Example 1

MEASUREMENT:MEAS<x>:REFLEVEL:PERCENT:LOW 15

This command sets the high reference level to 15% of HIGH.

## Example 2

MEASUREMENT:MEAS<x>:REFLEVEL:PERCENT:LOW?

This query might return

:MEASUREMENT:MEAS1:REFLEVEL:PERCENT:LOW 10

indicating that the percentage high reference level is set to 10% of HIGH.

---

# MEASUrement:MEAS<x>:REFLevel:PERCent:MID[1]

## Description

This command sets or queries the percentage (where 100% is equal to HIGH) used to calculate the mid reference level when MEASUrement:MEAS<x>:REFLevel:METHOD is set to Percent. Measurements are specified by x, which ranges from 1 through 8.

**Note:** This command affects the results of period, frequency, delay, and all cyclic measurements.

## Group

Measurement

## Related Commands

MEASUrement:MEAS<x>:REFLevel:METHOD

## Syntax 1

```
MEASUrement:MEAS<x>:REFLevel:PERCent:MID[1]
<NR3>
```

## Syntax 2

```
MEASUrement:MEAS<x>:REFLevel:PERCent:MID[1]?
```

## Arguments

- <NR3>

This is the mid reference level, ranging from 0 to 100%. The default mid reference level is 50%.

## Example 1

```
MEASUREMENT:MEAS1:REFLEVEL:PERCENT:MID[1] 60
```

This command sets the mid reference level to 60% of HIGH.

## Example 2

```
MEASUREMENT:MEAS1:REFLEVEL:PERCENT:MID[1]?
```

This query might return

```
:MEASUREMENT:MEAS1:REFLEVEL:PERCENT:MID 65
```

indicating that the percentage mid reference level is set to 65% of HIGH.

---

# MEASUREMENT:MEAS<x>:REFLEVEL:PERCENT:MID2

## Description

This command sets or queries the percentage (where 100% is equal to HIGH) used to calculate the mid reference level for the second waveform specified when MEASUREMENT:MEAS<x>:REFLEVEL:METHOD is set to Percent. Measurements are specified by x, which ranges from 1 through 8.

**Note:** This command affects the results of delay measurements.

## Group

Measurement

## Related Commands

MEASUREMENT:MEAS<x>:REFLEVEL:METHOD

## Syntax 1

```
MEASUREMENT:MEAS<x>:REFLEVEL:PERCENT:MID2
<NR3>
```

## Syntax 2

```
MEASUREMENT:MEAS<x>:REFLEVEL:PERCENT:MID2?
```

## Arguments

- <NR3>

This is the mid reference level, ranging from 0 to 100%. The default mid reference level is 50%.

## Example 1

```
MEASUREMENT:MEAS1:REFLEVEL:PERCENT:MID2 40
```

This command sets the mid2 reference level to 40% of HIGH.

## Example 2

```
MEASUREMENT:MEAS1:REFLEVEL:PERCENT:MID2?
```

This query might return

```
:MEASUREMENT:MEAS1:REFLEVEL:PERCENT:MID2 45
```

indicating that the percentage mid2 reference level is set to 45% of HIGH.



---

# MEASUrement:MEAS<x>:REFLevel?

## Description

This query-only command returns the current reference level parameters. It returns them in the following order: ABSOLUTE and then PERCENT for individual user measurements. Measurements are specified by x, which ranges from 1 through 8.

## Group

Measurement

## Syntax

```
MEASUrement:MEAS<x>:REFLevel?
```

## Example

```
MEASUREMENT:MEAS1:REFLEVEL?
```

---

# MEASUREMENT:MEAS<x>:SOURCE[1]

## Description

This command sets or queries the source for all single channel measurements and specifies the source to measure "from" when taking a delay measurement or phase measurement. Measurements are specified by x, which ranges from 1 through 8. This command is equivalent to selecting Measurement Setup from the Measure menu and then choosing the desired measurement source.

## Group

Measurement

## Syntax 1

```
MEASUREMENT:MEAS<x>:SOURCE[1]  
{CH<x>|MATH<y>|REF<x>|HISTogram}
```

## Syntax 2

```
MEASUREMENT:MEAS<x>:SOURCE[1]?
```

## Arguments

### ■ CH<x>

This is an input channel waveform. The x variable can be expressed as an integer ranging from 1 through 4 for 4-channel models and 1 to 2 for 2-channel models.

### ■ MATH<y>

This is a math waveform. The y variable can be expressed as an integer ranging from 1 through 4 for 4-channel models and 1 to 2 for 2-channel models.

### ■ REF<x>

This is a reference waveform. The x variable can be expressed as an integer ranging from 1 through 4 for 4-channel models and 1 to 2 for 2-channel models.

### ■ HISTogram

This is a histogram.

## Example 1

```
MEASUREMENT:MEAS2:SOURCE[1] MATH1
```

This command specifies Math 1 as the measurement 2 source.

---

## Example 2

MEASUrement:MEAS1:SOURCE[1]?

This query might return

:MEASUREMENT:MEAS1:SOURCE[1] MATH1

indicating that Math1 is the measurement 2 source.

---

# MEASUrement:MEAS<x>:SOURCE2

## Description

This command sets or queries the source for all single channel measurements and specifies the reference source to measure "to" when taking a delay measurement or phase measurement. Measurements are specified by x, which ranges from 1 through 8. This command is equivalent to selecting Measurement Setup from the Measure menu, selecting a measurement type of either Phase or Delay, and then choosing the desired measurement source.

Tip: Source2 measurements only apply to phase and delay measurement types, which require both a target (Source1) and reference (Source2) source.

## Group

Measurement

## Related Commands

MEASUrement:MEAS<x>:TYPE

## Syntax 1

```
MEASUrement:MEAS<x>:SOURCE2  
{CH<x>|MATH<y>|REF<x>}
```

## Syntax 2

```
MEASUrement:MEAS<x>:SOURCE2?
```

## Arguments

### ■ CH<x>

This is an input channel waveform. The x variable can be expressed as an integer ranging from 1 through 4 for 4-channel models and 1 to 2 for 2-channel models.

### ■ MATH<y>

This is a math waveform. The y variable can be expressed as an integer ranging from 1 through 4 for 4-channel models and 1 to 2 for 2-channel models.

### ■ REF<x>

This is a reference waveform. The x variable can be expressed as an integer ranging from 1 through 4 for 4-channel models and 1 to 2 for 2-channel models.

## Example 1

```
MEASUrement:MEAS4:SOURCE2 CH1
```

This command specifies CH1 as the delay "to" source when making delay measurement.

---

## Example 2

MEASUrement:MEAS2:SOURCE2?

This query might return

:MEASUREMENT:MEAS2:SOURCE2 MATH1

indicating that Math 1 is the measurement 2 source.

---

# MEASurement:MEAS<x>:SOUrce<x>:SIGType

## (TDS5000B Series option SM)

### Description

This command sets or queries the type of input signal used for the specified measurement slot. MEAS<x> is 1 through 8 for the measurement slot using SOURCE<x>, 1 or 2.

To ensure accurate measurements, use this command to specify the input-signal type for the measurement source.

### Group

Measurement

### Syntax 1

```
MEASurement:MEAS<x>:SOUrce<x>:SIGType  
{PULse|EYE}
```

### Syntax 2

```
MEASurement:MEAS<x>:SOUrce<x>:SIGType?
```

### Arguments

- PULSE

Use this signal type for generic signals that are not associated with synchronous communications standards.

- EYE

Use this signal type for synchronous-communication signals with NRZ-like characteristics (nonreturn to zero).

### Example 1

```
MEASurement:MEAS5:SOUrce1:SIGType PULse
```

This command selects Pulse as the signal type associated with source 1 for measurement slot 5.

### Example 2

```
MEASurement:MEAS<x>:SOUrce<x>:SIGType?
```

This query might return

```
:MEASUREMENT:MEAS2:SOURC1:SIGTYPE EYE
```

indicating that the signal type associated with source 1 of measurement slot 2 is EYE.

---

# MEASUrement:MEAS<x>:STATE

## Description

This command sets or queries whether the specified measurement slot is computed and displayed. The measurement slot is specified by *x*, which ranges from 1 through 8. This command is equivalent to selecting Measurement Setup from the Measure menu and then clicking the Display button.

For a measurement to display, you must have selected a source waveform and defined the measurement you want to take and display. You select the measurement using the MEASUrement:MEAS<x>:SOURCE[1] command. You define the measurement type using the MEASUrement:MEAS<x>:TYPE command.

## Group

Measurement

## Related Commands

MEASUrement:MEAS<x>:SOURCE[1], MEASUrement:MEAS<x>:TYPE

## Syntax 1

```
MEASUrement:MEAS<x>:STATE {OFF|ON|<NR1>}
```

## Syntax 2

```
MEASUrement:MEAS<x>:STATE?
```

## Arguments

- OFF

This disables calculation and display of the specified measurement slot.

- ON

This enables calculation and display of the specified measurement slot.

- <NR1>

A 0 disables calculation and display of the specified measurement slot; any other value enables calculation and display of the specified measurement slot.

## Example 1

```
MEASUrement:MEAS2:STATE ON
```

This command computes and displays the measurement defined as measurement 2.

---

## Example 2

MEASUrement:MEAS1:STATE?

This query might return

:MEASUREMENT:MEAS1:STATE 0

indicating that measurement defined for measurement slot 1 is disabled.



---

# MEASUREMENT:MEAS<x>:TYPE

## Description

This command sets or queries the measurement type defined for the specified measurement slot. The measurement slot is specified by x, which ranges from 1 through 8. This command is equivalent to selecting Measurement Setup from the Measure menu and then choosing the desired measurement type.

## Group

Measurement

## Syntax 1

```
MEASUREMENT:MEAS<x>:TYPE {AMPLITUDE|AREa|BURst|CARea|CMEan|CRMs|DELay|DISTDUty|EXTINCTDB|EXTINCTPCT|EXTINCTRATIO|EYEHeight|EYEWIdth|FALL|FREQuency|HIGH|HITs|LOW|MAXimum|MEAN|MEDIan|MINImum|NCROss|NDUty|NOVershoot|NWIdth|PBASe|PCROss|PCTCROss|PDUty|PEAKHits|PERIOd|PHASe|PK2Pk|PKPKJitter|PKPKNoise|POVershoot|PTOT|PWIdth|QFACTOR|RISe|RMS|RMSJitter|PMSNoise|SIGMA1|SIGMA2|SIGMA3|SIXSigmajit|SNRatio|STDdev|UNDEFINED|WAVEFORMS}
```

## Syntax 2

```
MEASUREMENT:MEAS<x>:TYPE?
```

## Arguments

### ■ AMPLITUDE

This measures the amplitude of the selected waveform. In other words, it measures the high value less the low value measured over the entire waveform or gated region.

Amplitude = High - Low

### ■ AREa

This measures the voltage over time. The area is over the entire waveform or gated region and is measured in volt-seconds. The area measured above the ground is positive, while the area below ground is negative.

### ■ BURst

The burst width measures the duration of a burst. The measurement is made over the entire waveform or gated region.

---

- **CARea**

The cycle area measures the voltage over time. In other words, it measures, in volt-seconds, the area over the first cycle in the waveform or the first cycle in the gated region. The area measured above the common reference point is positive, while the area below the common reference point is negative.

- **CMEan**

This measures the arithmetic mean over the first cycle in the waveform or the first cycle in the gated region.

- **CRMs**

This measures the true Root Mean Square voltage over the first cycle in the waveform or the first cycle in the gated region.

- **DELAy**

This measures the time between the middle reference (default = 50%) amplitude point of the source waveform and the destination waveform.

- **DISTDUty**

The duty cycle distortion measures the time between the falling edge and the rising edge of the eye pattern at the mid reference level.

It is the peak-to-peak time variation of the first eye crossing measured at the mid-reference as a percent of the eye period.

- **EXTINCTDB**

This (optical waveform (eye diagram) measurement) measures its extinction ratio. Extinction Ratio (dB) measures the ratio of the average power levels for the logic High to the logic Low of an optical waveform and expresses the result in dB.

This measurement only works for fast acquisition signals or a reference waveform saved in fast acquisition mode.

Extinction dB = 10 X (log 10 (High / Low))

- **EXTINCTPCT**

This (optical measurement) measures the extinction ratio of the selected optical waveform. Extinction Ratio (%) measures the ratio of the average power levels for the logic Low (off) to the logic (High) (on) of an optical waveform and expresses the result in percent.

This measurement only works for fast acquisition signals or a reference waveform saved in fast acquisition mode.

Extinction % = 100.0 x (Low / High)

- **EXTINCTRATIO**

This (optical measurement) measures the extinction ratio of the selected optical waveform. Extinction Ratio measures the ratio of the average power levels for the logic High to the logic Low of an optical waveform and expresses the result without units.

---

This measurement only works for fast acquisition signals or a reference waveform saved in fast acquisition mode.

Extinction Ratio = (High / Low), ratios greater than 100 or less than 1 generate errors; low must be greater than or equal to 1 microwatt.

■ **EYEHeight**

This measures the vertical opening of an eye diagram in volts.

■ **EYEWidth**

This measures the width of an eye diagram in seconds.

■ **FALL**

This measures the time taken for the falling edge of the first pulse in the waveform or gated region to fall from a high reference value (default is 90%) to a low reference value (default is 10%).

■ **FREQuency**

This measures the first cycle in the waveform or gated region. Frequency is the reciprocal of the period and is measured in hertz (Hz), where 1 Hz = 1 cycle per second.

■ **HIGH**

This measures the High reference (100% level, sometimes called Top-line) of a waveform.

You can also limit the High measurement (normally taken over the entire waveform record) to a gated region on the waveform.

■ **HITs**

The hits-in-box histogram measurement measures the number of points in or on the histogram box.

■ **LOW**

This measures the Low reference (0% level, sometimes called Baseline) of a waveform.

■ **MAXimum**

This finds the maximum amplitude. This value is the most positive peak voltage found. It is measured over the entire waveform or gated region.

When histogram is selected with the MEASUREMENT:METHoD command, the maximum measurement measures the voltage of the highest nonzero bin in vertical histograms or the time of the rightmost bin in horizontal histograms.

■ **MEAN**

This amplitude measurement finds the arithmetic mean over the entire waveform or gated region.

When histogram is selected with the MEASUREMENT:METHoD command, the mean measurement measures the average of all acquired points within or on the histogram.

---

- **MEDian**

This (histogram measurement) measures the middle point of the histogram box. Half of all acquired points within or on the histogram box are less than this value and half are greater than this value.

- **MINImum**

This amplitude measurement finds the minimum amplitude. This value is typically the most negative peak voltage. It is measured over the entire waveform or gated region.

When histogram is selected with the MEASUREMENT:METHOD command, the minimum measurement measures the lowest nonzero bin in vertical histograms or the time of the leftmost nonzero bin in the horizontal histograms.

- **NCROSS**

This (timing measurement) measures the time from the trigger point to the first falling edge of the waveform or gated region. The distance (time) is measured at the middle reference amplitude point of the signal.

- **NDUty**

The negative duty cycle is the ratio of the negative pulse width to the signal period, expressed as a percentage. The duty cycle is measured on the first cycle in the waveform or gated region.

Negative Duty Cycle = (Negative Width) / Period \* 100%

- **NOVershoot**

The negative overshoot amplitude measurement finds the negative overshoot value over the entire waveform or gated region.

Negative Overshoot = (Low – Minimum) / Amplitude \* 100%

- **NWIdth**

The negative width measurement is the distance (time) between the middle reference (default = 50%) amplitude points of a negative pulse. The measurement is made on the first pulse in the waveform or gated region.

- **PBASE**

This measures the base value used in extinction ratio measurements.

- **PCROSS**

This (timing measurement) measures the time from the trigger point to the first positive edge of the waveform or gated region. The distance (time) is measured at the middle reference amplitude point of the signal.

- **PCTCROSS**

The crossing percent (optical measurement) measures the location of the eye crossing point expressed as a percentage of EYEHEIGHT.

Crossing percent = 100 \* [(eye-crossing-point – PBASE)/(PTOP – PBASE)]

---

- **PDuty**

The positive duty cycle is the ratio of the positive pulse width to the signal period, expressed as a percentage. It is measured on the first cycle in the waveform or gated region.

Positive Duty Cycle = (Positive Width)/Period \* 100%

- **PEAKHits**

This (histogram measurement) measures the number of points in the largest bin of the histogram.

- **PERIOD**

This timing measurement is the time required to complete the first cycle in a waveform or gated region. Period is the reciprocal of frequency and is measured in seconds.

- **PHase**

This measures the phase difference (amount of time a waveform leads or lags the reference waveform) between two waveforms. The measurement is made between the middle reference points of the two waveforms and is expressed in degrees, where 360° represents one waveform cycle.

- **PK2Pk**

This peak-to-peak amplitude measurement finds the absolute difference between the maximum and minimum amplitude in the entire waveform or gated region.

When histogram is selected with the MEASUREMENT:METHOd command, the PK2Pk measurement measures the histogram's peak to peak difference.

- **PKPKJitter**

This measures the variance (minimum and maximum values) in the time locations of the cross point.

- **PKPKNoise**

This measures the peak-to-peak noise on a waveform at the mid reference level.

- **POvershoot**

The positive overshoot amplitude measurement finds the positive overshoot value over the entire waveform or gated region.

Positive Overshoot = (Maximum – High) / Amplitude \* 100%

- **PTOT**

This measures the top value used in extinction ratio measurements.

- **PWidth**

The positive width is the distance (time) between the middle reference (default = 50%) amplitude points of a positive pulse. The measurement is made on the first pulse in the waveform or gated region.

---

- **QFACTOR**

This measures the quality factor. The Q factor is a figure of merit for an eye diagram, which indicates the vertical eye opening relative to the noise at the low and high logic levels. It is the ratio of the eye size to noise.

- **RISe**

This timing measurement finds the rise time of the waveform. The rise time is the time it takes for the leading edge of the first pulse encountered to rise from a low reference value (default is 10%) to a high reference value (default is 90%).

- **RMS**

This amplitude measurement finds the true Root Mean Square voltage in the entire waveform or gated region.

- **RMSJitter**

This measures the variance in the time locations of the cross point. The RMS jitter is defined as one standard deviation at the cross point.

- **RMSNoise**

This measures the Root Mean Square noise amplitude on a waveform at the mid reference level.

- **SIGMA1**

This (histogram measurement) measures the percentage of points in the histogram that are within one standard deviation of the histogram mean.

- **SIGMA2**

This (histogram measurement) measures the percentage of points in the histogram that are within two standard deviations of the histogram mean.

- **SIGMA3**

This (histogram measurement) measures the percentage of points in the histogram that are within three standard deviations of the histogram mean.

- **SIXSigmajit**

This (histogram measurement) is six \* RMSJitter.

- **SNRatio**

This measures the signal-to-noise ratio. The signal-to-noise ratio is the amplitude of a noise rejection band centered on the mid level.

- **STDdev**

This measures the standard deviation (Root Mean Square (RMS) deviation) of all acquired points within or on the histogram box.

- **UNDEFINED**

This is the default measurement type, which indicates that no measurement type is specified. Once a measurement type is chosen, it can be cleared using this argument.

---

## ■ WAVEFORMS

The waveform count (histogram measurement) measures the number of waveforms used to calculate the histogram.

### Example 1

```
MEASUREMENT:MEAS2:TYPE FREQUENCY
```

This command defines measurement 2 as a measurement of the frequency of a waveform.

### Example 2

```
MEASUREMENT:MEAS1:TYPE?
```

This query might return

```
:MEASUREMENT:MEAS1:TYPE RMS
```

indicating that measurement 1 is defined to measure the RMS value of a waveform.

---

# MEASUrement:MEAS<x>:UNIts?

## Description

This query-only command returns the units associated with the specified measurement. Measurements are specified by x, which ranges from 1 through 8.

## Group

Measurement

## Related Commands

MEASUrement:MEAS<x>:TYPE

## Syntax

MEASUrement:MEAS<x>:UNIts?

## Example

MEASUrement:MEAS1:UNIts?

This query might return

:MEASUREMENT:MEAS1:UNIts %

indicating units for measurement 1 are set to percent.



---

# MEASUrement:MEAS<x>:VALue?

## Description

This query-only command returns the value that has been calculated for the measurement specified by <x>, which ranges from 1 through 8. This command is equivalent to selecting Display Statistics from the Measure menu and then choosing Value from the drop-down list to display *all* measurement values on-screen.

**Note:** This is the displayed value in the onscreen display. If measurement statistics are enabled, a new value is calculated with every waveform. In addition, this value is updated about every 1/3 second. If you are acquiring a long acquisition record, the instrument may take longer to update.

## Group

Measurement

## Related Commands

MEASUrement:MEAS<x>:UNIts?, \*ESR?, ALLEV?

## Syntax

MEASUrement:MEAS<x>:VALue?

## Example

MEASUrement:MEAS1:VALue?

This query might return

:MEASUREMENT:MEAS1:VALue 2.8740E-06

If the measurement has an error or warning associated with it, then an item is added to the error queue. The error can be checked for with the \*ESR? and ALLEV? commands.

---

# MEASUrement:MEAS<x>?

## Description

This query-only command returns all measurement parameters for the displayed measurement specified by x, which can range from 1 through 8. This query command is equivalent to selecting Measurement Setup from the Measure menu and viewing the Measurements table; then choosing the Time tab, clicking the Delay button and viewing the Delay Edge and Search Direction settings.

## Group

Measurement

## Syntax

MEASUrement:MEAS<x>?

## Example

MEASUrement:MEAS1?

This query might return

```
:MEASUREMENT:MEAS1:STATE 0;TYPE UNDEFINED;UNITS "V";SOURCE1 CH1;SOURCE2 CH1;SOURCE1:SIGTYPE PULSE;:MEASUREMENT:MEAS1:SOURCE2:SIGTYPE PULSE;:MEASUREMENT:MEAS1:DELAY:EDGE1 RISE;EDGE2 RISE;DIRECTION FORWARDS;:MEASUREMENT:MEAS1:REFLEVEL:METHOD PERCENT;ABSOLUTE:HIGH 0.0000;LOW 0.0000;MID1 0.0000;MID2 0.0000;:MEASUREMENT:MEAS1:REFLEVEL:PERCENT:HIGH 90.0000;LOW 10.0000;MID1 50.0000;MID2 50.0000;:MEASUREMENT:MEAS1:METHOD HISTOGRAM;NOISE HIGH.
```

---

# MEASUrement:METhod

## Description

This command sets or queries the method used to calculate the 0% and 100% reference level. This command is equivalent to selecting Reference Levels from the Measure menu and then choosing the desired Determine Base, Top From setting.

## Group

Measurement

## Related Commands

MEASUrement:REFLevel:PERCent:HIGH,  
MEASUrement:REFLevel:PERCent:LOW,  
MEASUrement:REFLevel:PERCent:MID,  
MEASUrement:REFLevel:PERCent:MID2

## Syntax 1

```
MEASUrement:METhod {HISTogram|MEAN|MINMax}
```

## Syntax 2

```
MEASUrement:METhod?
```

## Arguments

### ■ HISTogram

This sets the high and low reference levels to the most common values either above or below the mid point, depending on whether the high reference point or the low reference point is being defined. Because the statistical approach ignores short-term aberrations, such as overshoot or ringing, the histogram method is the best setting for examining pulses.

### ■ MEAN

This sets the high and low reference levels to the mean values using all values either above or below the midpoint, depending of whether it is defining the high or low reference level. The selection is best used for examining eye patterns.

### ■ MINMax

This uses the highest and lowest values of the waveform record. This selection is best for examining waveforms with no large, flat portions of a common value, such as sine waves and triangle waves.

## Example 1

```
MEASUrement:METhod HISTogram
```

This command specifies that the high and low reference levels are set statistically.

---

## Example 2

MEASUrement:METhod?

This query might return

:MEASUREMENT:METHOD MINMAX

indicating that the reference levels are set to MIN and MAX.

---

# MEASUREMENT:NOISE

## (TDS5000B Series option SM)

### Description

This command sets or queries whether the noise measurement is made on the high or low level of the waveform. Sending this command is equivalent to selecting Ref Levs > Eye > Top Level or Base Level in the Comm tab of the Measurement Setup dialog box. The Eye section is displayed only if you have an eye-pattern or optical measurement defined.

### Group

Measurement

### Syntax 1

```
MEASUREMENT:NOISE {HIGH|LOW}
```

### Syntax 2

```
MEASUREMENT:NOISE?
```

### Arguments

- HIGH

This argument causes the measurement for noise to be taken at the high level of the waveform.

- LOW

This argument causes the measurement for noise to be taken at the low level of the waveform.

### Example 1

```
MEASUREMENT:NOISE HIGH
```

This command causes the noise measurement to be made at the high level of the waveform.

### Example 2

```
MEASUREMENT:NOISE?
```

This query might return

```
:MEASUREMENT:NOISE LOW
```

indicating that the noise measurement will be made at the low level of the waveform.

---

# MEASUrement:REFLevel:ABSolute:HIGH

## Description

This command sets or queries the high reference level, and is the upper reference level when MEASUrement:REFLevel:METHod is set to Absolute. This command affects the results of rise and fall measurements and is equivalent to selecting Reference Levels from the Measure menu, and then entering the Absolute High Ref value.

**Note:** this command affects the associated reference level parameter for all MEASurements:IMMed and the eight periodic measurements. To change the parameter for individual measurements, use the MEASUrement:MEAS<x>:REFLevel commands.

## Group

Measurement

## Related Commands

MEASUrement:REFLevel:METHod, MEASUrement:IMMed:TYPE, MEASUrement:MEAS<x>:TYPE

## Syntax 1

```
MEASUrement:REFLevel:ABSolute:HIGH <NR3>
```

## Syntax 2

```
MEASUrement:REFLevel:ABSolute:HIGH?
```

## Argument

- <NR3>

This is the high reference level, in volts. The default is 0.0 V.

## Example 1

```
MEASUrement:REFLevel:ABSolute:HIGH 1.71
```

This command sets the high reference level to 1.71 V.

## Example 2

```
MEASUrement:REFLevel:ABSolute:HIGH?
```

This query might return

```
:MEASUREMENT:REFLEVEL:ABSOLUTE:HIGH 1.7100E+00
```

indicating that the absolute high reference level is set to 1.71 V.

---

# MEASUrement:REFLevel:ABSolute:LOW

## Description

This command sets or queries the low reference level, and is the lower reference level when MEASUrement:REFLevel:METHod is set to Absolute. This command affects the results of rise and fall measurements and is equivalent to selecting Reference Levels from the Measure menu, and then entering the Absolute Low Ref value.

**Note:** this command affects the associated reference level parameter for all MEASurements:IMMed and the eight periodic measurements. To change the parameter for individual measurements, use the MEASUrement:MEAS<x>:REFLevel commands.

## Group

Measurement

## Related Commands

MEASUrement:REFLevel:METHod, MEASUrement:IMMed:TYPE, MEASUrement:MEAS<x>:TYPE

## Syntax 1

```
MEASUrement:REFLevel:ABSolute:LOW <NR3>
```

## Syntax 2

```
MEASUrement:REFLevel:ABSolute:LOW?
```

## Argument

- <NR3>

This is the low reference level, in volts. The default is 0.0 V.

## Example 1

```
MEASUrement:REFLevel:ABSolute:LOW 0.0 V.
```

This command sets the low reference level to 0.0 V.

## Example 2

```
MEASUrement:REFLevel:ABSolute:LOW?
```

This query might return

```
:MEASUREMENT:REFLEVEL:ABSOLUTE:LOW 0.0000E+00
```

indicating that the absolute low reference level is set to 0.0 V.

---

# MEASUREMENT:REFLEVEL:ABSOLUTE:MID[1]

## Description

This command sets or queries the mid reference level, and is the 50% reference level when MEASUREMENT:REFLEVEL:METHOD is set to Absolute. This command affects the results of period, frequency, delay, and all cyclic measurements and is equivalent to selecting Reference Levels from the Measure menu, and then entering the Absolute Mid Ref value.

**Note:** this command affects the associated reference level parameter for all MEASUREMENTS:IMMED and the eight periodic measurements. To change the parameter for individual measurements, use the MEASUREMENT:MEAS<x>:REFLEVEL commands.

## Group

Measurement

## Related Commands

MEASUREMENT:REFLEVEL:METHOD

## Syntax 1

```
MEASUREMENT:REFLEVEL:ABSOLUTE:MID[1] <NR3>
```

## Syntax 2

```
MEASUREMENT:REFLEVEL:ABSOLUTE:MID[1]?
```

## Argument

- <NR3>

This is the mid reference level, in volts. The default is 0.0 V.

## Example 1

```
MEASUREMENT:REFLEVEL:ABSOLUTE:MID[1] .71
```

This command sets the mid reference level to .71 V.

## Example 2

```
MEASUREMENT:REFLEVEL:ABSOLUTE:MID[1]?
```

This query might return

```
:MEASUREMENT:REFLEVEL:ABSOLUTE:MID 0.7100E+00
```

indicating that the absolute mid1 reference level is set to .71 V.



---

# MEASUrement:REFLevel:ABSolute:MID2

## Description

This command sets or queries the mid reference level for the "to" waveform when taking a delay measurement, and is the 50% reference level when MEASUrement:REFLevel:METHOD is set to Absolute. This command affects the results of delay measurements and is equivalent to selecting Reference Levels from the Measure menu, and then entering the Absolute Mid2 Ref value.

**Note:** this command affects the associated reference level parameter for all MEASUrement:IMMed and the eight periodic measurements. To change the parameter for individual measurements, use the MEASUrement:MEAS<x>:REFLevel commands.

## Group

Measurement

## Related Commands

MEASUrement:REFLevel:METHOD

## Syntax 1

```
MEASUrement:REFLevel:ABSolute:MID2 <NR3>
```

## Syntax 2

```
MEASUrement:REFLevel:ABSolute:MID2?
```

## Argument

- <NR3>

This is the mid reference level, in volts. The default is 0.0 V.

## Example 1

```
MEASUrement:REFLevel:ABSolute:MID2 0.5
```

This command sets the mid reference level for the delay waveform to 0.5 V.

## Example 2

```
MEASUrement:REFLevel:ABSolute:MID2?
```

This query might return

```
:MEASUREMENT:REFLEVEL:ABSOLUTE:MID2 0.5000E+00
```

indicating that the absolute mid2 reference level is set to 0.5 V.

---

# MEASUrement:REFLevel:METHOD

## Description

This command specifies or queries the reference level units used for measurement calculations. This command is equivalent to selecting Reference Levels from the Measure menu, and then choosing the desired reference level from the Units group box.

**Note:** this command affects the associated reference level parameter for all MEASurements:IMMed and the eight periodic measurements. To change the parameter for individual measurements, use the MEASUrement:MEAS<x>:REFLevel commands.

## Group

Measurement

## Syntax 1

```
MEASUrement:REFLevel:METHOD {ABSolute|PERCent}
```

## Syntax 2

```
MEASUrement:REFLevel:METHOD?
```

## Arguments

### ■ ABSolute

This specifies that the reference levels are set explicitly using the MEASUrement:REFLevel:ABSolute commands. This method is useful when precise values are required (for example, when designing to published interface specifications, such as RS-232-C).

### ■ PERCent

This specifies that the reference levels are calculated as a percent relative to HIGH and LOW. The percentages are defined using the MEASUrement:REFLevel:PERCent commands.

## Example 1

```
MEASUrement:REFLevel:METHOD ABSolute
```

This command specifies that explicit user-defined values are used for the reference levels.

## Example 2

```
MEASUrement:REFLevel:METHOD?
```

This query might return

```
:MEASUREMENT:REFLEVEL:METHOD PERCENT
```

indicating that the reference level units used are calculated as a percent relative to HIGH and LOW.

---

# MEASUrement:REFLevel:PERCent:HIGH

## Description

This command sets or queries the percent (where 100% is equal to HIGH) that is used to calculate the high reference level when MEASUrement:REFLevel:METHod is set to Percent. This command affects the results of rise and fall measurements and is equivalent to selecting Reference Levels from the Measure menu, and then entering the Percentage High Ref value.

**Note:** this command affects the associated reference level parameter for all MEASUrement:IMMed and the eight periodic measurements. To change the parameter for individual measurements, use the MEASUrement:MEAS<x>:REFLevel commands.

## Group

Measurement

## Related Commands

MEASUrement:REFLevel:METHod, MEASUrement:IMMed:TYPe, MEASUrement:MEAS<x>:TYPe

## Syntax 1

```
MEASUrement:REFLevel:PERCent:HIGH <NR3>
```

## Syntax 2

```
MEASUrement:REFLevel:PERCent:HIGH?
```

## Argument

- <NR3>

This is the high reference level, ranging from 0 to 100%. The default high reference level is 90%.

## Example 1

```
MEASUrement:REFLevel:PERCent:HIGH 95
```

This command sets the high reference level to 95% of HIGH.

## Example 2

```
MEASUrement:REFLevel:PERCent:HIGH?
```

This query might return

```
:MEASUREMENT:REFLEVEL:PERCENT:HIGH 90
```

indicating that the percentage high reference level is set to 90% of HIGH.

---

# MEASUrement:REFLevel:PERCent:LOW

## Description

This command sets or queries the percent (where 100% is equal to HIGH) that is used to calculate the low reference level when MEASUrement:REFLevel:METHOD is set to Percent. This command affects the results of rise and fall measurements and is equivalent to selecting Reference Levels from the Measure menu, and then entering the Percentage Low Ref value.

**Note:** this command affects the associated reference level parameter for all MEASUrement:IMMed and the eight periodic measurements. To change the parameter for individual measurements, use the MEASUrement:MEAS<x>:REFLevel commands.

## Group

Measurement

## Related Commands

MEASUrement:REFLevel:METHOD, MEASUrement:IMMed:TYPE, MEASUrement:MEAS<x>:TYPE

## Syntax 1

```
MEASUrement:REFLevel:PERCent:LOW <NR3>
```

## Syntax 2

```
MEASUrement:REFLevel:PERCent:LOW?
```

## Argument

- <NR3>

This is the low reference level, ranging from 0 to 100%. The default low reference level is 10%.

## Example 1

```
MEASUrement:REFLevel:PERCent:LOW 15
```

This command sets the high reference level to 15% of HIGH.

## Example 2

```
MEASUrement:REFLevel:PERCent:LOW?
```

This query might return

```
:MEASUREMENT:REFLEVEL:PERCENT:LOW 10
```

indicating that the percentage high reference level is set to 10% of HIGH.

---

# MEASUrement:REFLevel:PERCent:MID[1]

## Description

This command sets or queries the percent (where 100% is equal to HIGH) that is used to calculate the mid reference level when MEASUrement:REFLevel:METHod is set to Percent. This command affects the results of period, frequency, delay, and all cyclic measurements and is equivalent to selecting Reference Levels from the Measure menu and, then entering the Percentage Mid Ref value.

**Note:** this command affects the associated reference level parameter for all MEASurements:IMMed and the eight periodic measurements. To change the parameter for individual measurements, use the MEASUrement:MEAS<x>:REFLevel commands.

## Group

Measurement

## Related Commands

MEASUrement:REFLevel:METHod

## Syntax 1

```
MEASUrement:REFLevel:PERCent:MID[1] <NR3>
```

## Syntax 2

```
MEASUrement:REFLevel:PERCent:MID[1]?
```

## Argument

- <NR3>

This is the mid reference level, ranging from 0 to 100%. The default mid reference level is 50%.

## Example 1

```
MEASUrement:REFLevel:PERCent:MID[1] 60
```

This command sets the mid reference level to 60% of HIGH.

## Example 2

```
MEASUrement:REFLevel:PERCent:MID[1]?
```

This query might return

```
:MEASUREMENT:REFLEVEL:PERCENT:MID 65
```

indicating that the percentage mid reference level is set to 65% of HIGH.

---

# MEASUrement:REFLevel:PERCent:MID2

## Description

This command sets or queries the percent (where 100% is equal to HIGH) that is used to calculate the mid reference level for the second waveform specified when MEASUrement:REFLevel:METHOD is set to Percent. This command affects the results of delay measurements and is equivalent to selecting Reference Levels from the Measure menu, and then entering the Percentage Mid2 Ref value.

**Note:** this command affects the associated reference level parameter for all MEASurements:IMMed and the eight periodic measurements. To change the parameter for individual measurements, use the MEASUrement:MEAS<x>:REFLevel commands.

## Group

Measurement

## Related Commands

MEASUrement:REFLevel:METHOD

## Syntax 1

MEASUrement:REFLevel:PERCent:MID2 <NR3>

## Syntax 2

MEASUrement:REFLevel:PERCent:MID2?

## Argument

- <NR3>

This is the mid reference level, ranging from 0 to 100%. The default mid reference level is 50%.

## Example 1

MEASUrement:REFLevel:PERCent:MID2 40

This command sets the mid2 reference level to 40% of HIGH.

## Example 2

MEASUrement:REFLevel:PERCent:MID2?

This query might return

:MEASUREMENT:REFLEVEL:PERCENT:MID2 45

indicating that the percentage mid2 reference level is set to 45% of HIGH.

---

# MEASUrement:REFLevel?

## Description

This query-only command returns the current reference level parameters. This command is equivalent to selecting Reference Levels from the Measure menu, and then viewing the current Reference Levels settings.

## Group

Measurement

## Syntax

```
MEASUrement:REFLevel?
```

## Example

```
MEASUrement:REFLevel?
```

This query might return these reference level settings

```
:MEASUREMENT:REFLEVEL:METHOD ABSOLUTE;  
ABSOLUTE:HIGH 0.0000E+00;LOW 0.0000E+00;  
MID1 0.0000E+00;MID2 0.0000E+00;  
:MEASUREMENT:REFLEVEL:PERCENT:HIGH  
9.0000E+01;LOW 1.0000+01;MID1 5.0000E+01;  
MID2 5.0000E+01
```

---

# MEASUrement:STATIstics:COUNT

## Description

This command (no query form) clears existing measurement statistics from memory. This command is equivalent to selecting Measurement Setup from the Measure menu, selecting Statistics, and clicking the Reset button.

## Group

Measurement

## Related Commands

MEASUrement:STATIstics:MODE

## Syntax

```
MEASUrement:STATIstics:COUNT {RESET}
```

## Argument

- RESET

This clears existing measurement statistics from memory.

## Example

```
MEASUrement:STATIstics:COUNT RESET
```

This command resets the count of the statistics measurement to zero, clears the measurement statistics from memory, and restarts the calculation.



---

# MEASUrement:STATIstics:MODe

## Description

This command controls the operation and display of management statistics. This command is equivalent to selecting Measurement Setup from the Measure menu, clicking the Statistics button and then choosing the desired Measurement Format.

## Group

Measurement

## Related Commands

MEASUrement:STATIstics:WEIghting

## Syntax 1

```
MEASUrement:STATIstics:MODe {OFF|ALL|VALUE-  
Mean}
```

## Syntax 2

```
MEASUrement:STATIstics:MODe?
```

## Arguments

- OFF

This turns all measurements off. This is the default value.

- ALL

This turns on statistics and displays all statistics for each measurement.

- VALUEMean

This turns on statistics and displays the value and the mean ( $\mu$ ) of each measurement.

## Example 1

```
MEASUrement:STATIstics:MODe OFF
```

This command turns statistic measurements off.

## Example 2

```
MEASUrement:STATIstics:MODe?
```

This query might return

```
:MEASUREMENT:STATISTICS:MODE ALL
```

indicating that measurement statistics are turned on and all statistics are being displayed for each measurement.

---

# MEASUrement:STATIstics:WEIghting

## Description

This command sets or queries the time constant for mean and standard deviation statistical accumulations. This command is equivalent to selecting Measurement Setup from the Measure menu, clicking the Statistics button and entering the desired Weight n= value.

## Group

Measurement

## Related Commands

MEASUrement:STATIstics:MODE

## Syntax 1

```
MEASUrement:STATIstics:WEIghting <NR1>
```

## Syntax 2

```
MEASUrement:STATIstics:WEIghting?
```

## Argument

- <NR1>

This is the time constant for the mean and standard deviation statistical accumulations.

## Example 1

```
MEASUrement:STATIstics:WEIghting 4
```

This command sets statistical weighting to 4.

## Example 2

```
MEASUrement:STATIstics:WEIghting?
```

This query might return

```
:MEASUREMENT:STATISTICS:WEIGHTING 4
```

indicating that measurement statistics weighting is currently set to 4.

---

# MEASUrement?

## Description

This query-only command returns all measurement parameters in the following order: MEAS1, MEAS2, MEAS3, MEAS4, MEAS5, MEAS6, MEAS7, MEAS8, and IMMED.

## Group

Measurement

## Syntax

MEASUrement?

## Example

MEASUrement?

This query might return

```
:MEASUREMENT:GATING OFF;IMMED:TYPE UNDEFINED;UNITS "V";SOURCE1 CH1;SOURCE2 CH1;SOURCE1:SIGTYPE PULSE;:MEASUREMENT:IMMED:SOURCE2:SIGTYPE PULSE;:MEASUREMENT:IMMED:DELAY:EDGE1 RISE;EDGE2 RISE;DIRECTION FORWARDS;:MEASUREMENT:IMMED:REFLEVEL:METHOD PERCENT;ABSOLUTE:HIGH 0.0000;LOW 0.0000;MID1 0.0000;MID2 0.0000;:MEASUREMENT:IMMED:REFLEVEL:PERCENT:HIGH 90.0000;LOW 10.0000;MID1 50.0000;MID2 50.0000;:MEASUREMENT:IMMED:METHOD HISTOGRAM;NOISE HIGH;:MEASUREMENT:MEAS1:STATE 0;TYPE UNDEFINED;UNITS "V";SOURCE1 CH1;SOURCE2 CH1;SOURCE1:SIGTYPE PULSE;:MEASUREMENT:MEAS1:SOURCE2:SIGTYPE PULSE;:MEASUREMENT:MEAS1:DELAY:EDGE1 RISE;EDGE2 RISE;DIRECTION FORWARDS;:MEASUREMENT:MEAS1:REFLEVEL:METHOD PERCENT;ABSOLUTE:HIGH 0.0000;LOW 0.0000;MID1 0.0000;MID2 0.0000;:MEASUREMENT:MEAS1:REFLEVEL:PERCENT:HIGH 90.0000;LOW 10.0000;MID1 50.0000;MID2 50.0000;:MEASUREMENT:MEAS1:METHOD HISTOGRAM;NOISE HIGH;:MEASUREMENT:MEAS2:STATE 0;TYPE UNDEFINED;UNITS "V";SOURCE1 CH1;SOURCE2 CH1;SOURCE1:SIGTYPE PULSE;:MEASUREMENT:MEAS2:SOURCE2:SIGTYPE PULSE;:MEASUREMENT:MEAS2:DELAY:EDGE1 RISE;EDGE2 RISE;DIRECTION...
```

---

# NEWpass

## Description

This command (no query form) changes the password that enables access to password protected data. The PASSWord command must be successfully executed before using this command or an execution error will be generated.

## Group

Miscellaneous

## Related Commands

PASSWord, \*PUD

## Syntax

```
NEWpass <Qstring>
```

## Argument

- <Qstring>

This is the new password, which can contain up to 10 characters.

## Example

```
NEWpass "mypassword"
```

This command creates a new password (mypassword) for accessing your protected data.

---

# PASSWord

## Description

This command (no query form) enables the \*PUD and NEWpass set commands. Sending PASSWord without any arguments disables these same commands. Once the password is successfully entered, the \*PUD and NEWpass commands are enabled until the instrument is powered off, or until the FACtory command, the PASSWord command with no arguments, or the \*RST command is issued.

To change the password, you must first enter the valid password with the PASSWord command and then change to your new password with the NEWpass command. Remember that the password is case sensitive.

## Group

Miscellaneous

## Related Commands

NEWpass, \*PUD

## Syntax

```
PASSWord <QString>
```

## Argument

- <QString>

This is the password, which can contain up to 10 characters. The factory default password is "XYZZY" and is always valid.

## Example 1

```
PASSWord "XYZZY"
```

This command enables the \*PUD and NEWPass set commands.

## Example 2

```
PASSWord
```

This command disables the \*PUD and NEWPass set commands. You can still use the query version of \*PUD.

---

# RECALL:MASK

## Description

This command (no query form) recalls a stored mask as the User mask.

## Group

## Mask

## Related Commands

SAVe:MASK

## Syntax

```
RECALL:MASK <Qstring>
```

## Arguments

- <Qstring>

Is a quoted string that defines the file name and path from where the mask will be recalled. Input the file path using the form <drive>/<dir>/<filename>. <drive> and one or more <dir>s are optional. If you do not specify them, the instrument will read the mask from the default directory. <filename> stands for a filename of up to 128 characters (use of wildcard characters in filenames is not supported). Filename extensions are not required but are highly recommended.

The file is stored in ASCII format and is user-editable.

## Example

```
RECALL:MASK "TEST"
```

This command recalls the mask stored in the file named TEST as the User mask.

---

# RECALL:SETUp

## Description

This command (no query form) returns stored or factory settings to the instrument from a copy of the settings stored in memory. If factory is referenced (by specifying '0'), the factory default values will be restored. This command is equivalent to \*RCL and performs the same function as selecting Recall from the File menu, and then choosing the Setup button.

## Group

Save and Recall

## Related Commands

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

## Syntax

```
RECALL:SETUp {FACTory|<NR1>|<file path>}
```

## Arguments

- FACTory

This restores the factory setup.

- <NR1>

This specifies a setup storage location to restore. Setup storage location values range from 0 through 10; using an out of range value causes an error (222, "Data out of range").

- <file path>

The instrument will recall the setup from the <file path> location.

<file path> is a quoted string that defines the file name and path. Input the file path using the form <drive>:<dir>/<filename>. <drive> and one or more <dir>s are optional. If you do not specify them, the instrument will read the file from the default directory (C:\TekScope\setups). <filename> stands for a filename of up to 128 characters (use of wildcard characters in filenames is not supported). Filename extensions are not required, but highly recommended.

## Example 1

```
RECALL:SETUp FACTory
```

This command recalls (and makes current) the instrument setup to its factory defaults.

## Example 2

```
RECALL:SETUp 2
```

This command recalls the instrument setup from setup storage location 2.

---

### Example 3

```
RECALL:SETUp "TEK00000.SET"
```

This command recalls the setup from the file `TEK00000.SET` in the default directory for setups (*C:\TekScope\setups*).



---

# RECALL:WAVEform

## Description

This command (no query form) recalls a stored waveform to a reference location. This command is equivalent to selecting Recall from the File menu, and then pressing the Waveform button.

## Group

Save and Recall

## Related Commands

DELEte:WAVEform, SAVe\_WAVEform

## Syntax

```
RECALL:WAVEform {<file path> , REF<x>}
```

## Arguments

- REF<x>

The instrument will recall the waveform from this location in internal reference memory. Reference memory location values range from 1 through 4.

- <file path>

The instrument will recall the setup from the <file path> location:

<file path> is a quoted string that defines the file name and path. Input the file path using the form <drive>:<dir>/<filename>. <drive> and one or more <dir>s are optional. If you do not specify them, the instrument will read the waveform from the default directory (C:\TekScope\Waveforms). <filename> stands for a filename of up to 128 characters (use of wildcard characters in filenames is not supported). Filename extensions are not required but are highly recommended.

## Example

```
RECALL:WAVEform "TEK00000.WFM", REF1
```

This command recalls the waveform stored in the file named TEK00000.WFM from the default directory for waveforms (C:\TekScope\Waveforms) to the reference location 1.

---

# REF<x>:LABel?

## Description

This query-only command returns a branch query containing the waveform label name and the coordinates at which the label (attached to the displayed waveform of the specified reference) is displayed. The reference waveform is specified by x. The value of x can range from 1 through 4 for four-channel instruments or 1 through 2 for two-channel instruments. This command is equivalent to selecting Reference Waveforms from the File menu and then choosing Label from the drop-down list.

## Group

Vertical

## Related Commands

REF<x>:LABel:NAME?, REF<x>:LABel:XPOS?,  
REF<x>:LABel:YPOS?

## Syntax

REF<x>:LABel?

## Argument

None

## Example

REF1:LABel?

This query might return

```
:REF1:LABel:NAME "Myname";:XPOS-200;:YPOS 50.
```

---

# REF<x>:LABel:NAME

## Description

This command sets or queries the label of the designated waveform. The reference waveform is specified by x. The value of x can range from 1 through 4 for four-channel instruments or 1 through 2 for two-channel instruments. This command is equivalent to selecting Reference Waveforms from the File menu, choosing Label from the drop-down list, selecting the tab associated with the reference for which you want to create a label, and entering a label in the Label box.

## Group

Vertical

## Related Commands

REF<x>:LABel

## Syntax 1

REF<x>:LABel:NAME <String>

## Syntax 2

REF<x>:LABel:NAME?

## Argument

- <String>

This is the character string that will be used for the reference waveform label name.

## Example 1

```
REF3:LABel:NAME?
```

This query might return

```
:REF3:LABEL:NAME "Signal2"
```

indicating that the label name for Reference 3 is currently set to "Signal2".

## Example 2

```
REF4:LABel:NAME "My Reference"
```

This command sets the label name of Reference 4 to "My Reference".

---

# REF<x>:LABel:XPOS

## Description

This command sets or queries the X screen offset at which the label (attached to the displayed waveform of the specified reference) is displayed, relative to the left edge of the screen. The reference waveform is specified by x. The value of x can range from 1 through 4 for four-channel instruments or 1 through 2 for two-channel instruments. This command is equivalent to selecting Reference Waveforms from the File menu, choosing Label from the drop-down list, selecting the tab associated with the reference for which you want to position a label, and entering a value in the X Position box.

## Group

Vertical

## Related Commands

REF<x>:LABel

## Syntax 1

```
REF<x>:LABel:XPOS <NR1>
```

## Syntax 2

```
REF<x>:LABel:XPOS?
```

## Argument

- <NR1>

This is the location (in pixels) where the waveform label for the selected reference is displayed, relative to the left edge of the screen. Arguments should be integers ranging from 0 through 500.

## Example 1

```
REF4:LABel:XPOS 10
```

This command moves the waveform label for the REF3 waveform, so that it begins 10 pixels to the right of the left edge of the screen.

## Example 2

```
REF2:LABel:XPOS?
```

This query might return

```
:REF2:LABEL:XPOS 150
```

indicating that the x-axis for the REF 2 waveform is currently 150 pixels to the right of the left edge of the screen.

---

# REF<x>:LABel:YPOS

## Description

This command sets or queries the Y screen offset at which the label (attached to the displayed waveform of the specified reference) is displayed, relative to the top edge of the screen. The reference waveform is specified by x. The value of x can range from 1 through 4 for four-channel instruments or 1 through 2 for two-channel instruments. This command is equivalent to selecting Reference Waveforms from the File menu, choosing Label from the drop-down list, selecting the tab associated with the reference for which you want to position a label, and entering a value in the Y Position box.

## Group

Vertical

## Related Commands

REF<x>:LABel

## Syntax 1

```
REF<x>:LABel:YPOS <NR1>
```

## Syntax 2

```
REF<x>:LABel:YPOS?
```

## Argument

- <NR1>

This is the location (in pixels) where the waveform label for the selected reference is displayed, relative to the top edge of the screen. Arguments should be integers ranging from 0 to 385.

## Example 1

```
REF3:LABel:YPOS -10
```

This command moves the waveform label for the REF 3 waveform to just beneath (10 pixels) the top of the screen.

## Example 2

```
REF2:LABel:YPOS?
```

This query might return

```
:REF2:LABEL:YPOS 0
```

indicating that the waveform label for the REF 2 waveform is currently located just beneath the top of the screen.

---

# REF<x>:HORizontal:POSition

## Description

This command sets or queries the horizontal display position of the reference waveform. The reference waveform is specified by x. The value of x can range from 1 through 4 for four-channel instruments or 1 through 2 for two-channel instruments. This command is equivalent to selecting Reference Waveforms from the File menu, choosing Reference Setup... from the drop-down list, selecting a reference waveform, and then entering the horizontal position value using the multipurpose knob.

## Group

Vertical

## Syntax 1

```
REF<x>:HORizontal:POSition <NR3>
```

## Syntax 2

```
REF<x>:HORizontal:POSition?
```

## Argument

- <NR3>

This argument specifies the horizontal position of the specified reference waveform in percentage of the waveform that is displayed to the right of the center vertical graticule. The range of this argument is from 0 through 100.

## Example 1

```
REF2:HORizontal:POSition 0.0
```

This command positions the start of the Reference 2 waveform at the center graticule.

## Example 2

```
REF1:HORizontal:POSition?
```

This query might return

```
:REF1:HORIZONTAL:POSITION 50
```

indicating that the Reference 1 waveform is centered on the horizontal graticule.

---

# REF<x>:VERTical:POSition

## Description

This command sets or queries the vertical position of a reference waveform. The reference waveform is specified by x. The value of x can range from 1 through 4 for four-channel instruments or 1 through 2 for two-channel instruments.

This command is equivalent to selecting Reference Waveforms from the File menu, choosing Reference Setup... from the drop-down list, selecting a reference waveform, and then entering the Position value using the multipurpose knob.

Increasing the position value of a waveform causes the waveform to move up, and decreasing the position value causes the waveform to move down. Position adjusts only the display position of a waveform. The position value determines the vertical graticule coordinate at which signal values are displayed. For example, if the position for Reference 3 is set to 2.0, the signal represented by that reference will be displayed at 2.0 divisions above the center of the screen.

**Note:** References are static. All position and scale changes are applied in a post-processing mode.

## Group

Vertical

## Related Commands

CH<x>:POSition, MATH<x>:POSition

## Syntax 1

```
REF<x>:VERTical:POSition <NR3>
```

## Syntax 2

```
REF<x>:VERTical:POSition?
```

## Argument

- <NR3>

This is the desired position, in divisions from the center horizontal graticule. The range is from -5 to 5 divisions.

## Example 1

```
REF2:VERTical:POSition 1.3E+00
```

This command positions the Reference 2 input signal 1.3 divisions above the center horizontal graticule.

---

## Example 2

REF1:VERTical:POSition?

This query might return

:REF1:VERTICAL:POSITION -1.3000E+00

indicating that the current position of Reference 1 is 1.3 divisions below the center horizontal graticule.



---

# REF<x>:VERTical:SCALE

## Description

This command sets or queries the vertical scale of a reference waveform. The reference waveform is specified by x. The value of x can range from 1 through 4 for four-channel instruments or 1 through 2 for two-channel instruments. This command is equivalent to selecting Reference Waveforms from the File menu, choosing Reference Setup... from the drop-down list, selecting a reference waveform, and then entering the Scale value using the multipurpose knob.

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

Scale affects all waveforms, but affects reference and math waveforms differently from channel waveforms:

- For reference and math waveforms, this setting controls the display only, graphically scaling these waveforms and having no affect on the acquisition hardware.
- For channel waveforms, this setting controls the vertical size of the acquisition window as well as the display scale. The range and resolution of scale values depends on the probe attached and any other external factors you have specified.

## Group

Vertical

## Related Commands

CH<x>:SCALE, MATH<x>:VERTical:SCALE

## Syntax 1

```
REF<x>:VERTical:SCALE <NR3>
```

## Syntax 2

```
REF<x>:VERTical:SCALE?
```

## Argument

- <NR3>

This is the gain in user units per division.

## Example 1

```
REF4:VERTical:SCALE 100E-03
```

This command sets the Reference 4 scale to 100 mV per division.

---

## Example 2

CH2:SCALE?

This query might return

:REF2:VERTICAL:SCALE 1.0000e+00

indicating that the current vertical scale setting for Reference 2 is 1 V per division.

---

# REM

## Description

This command (no query form) embeds a comment within GPIB programs as a means of internally documenting the programs. The instrument ignores these embedded comment lines.

## Group

Miscellaneous

## Related Commands

NEWpass, \*PUD

## Syntax

```
REM <QString>
```

## Argument

- <QString>

This is a string that can contain a maximum of 80 characters.

## Example

```
REM "This is a comment"
```

The instrument ignores this comment string.

---

# ROSc:SOUrce

## Description

This command sets or queries the source for the 10 MHz timebase reference. Depending on the command argument that you specify, you can use an external reference or use the internal crystal oscillator as the timebase reference.

This command is also useful for synchronizing multiple instruments.

## Group

Miscellaneous

## Related Commands

ROSc:STATE

## Syntax 1

```
ROSc:SOUrce {INTERnal|EXTernal}
```

## Syntax 2

```
ROSc:SOUrce?
```

## Arguments

- **INTERnal**

This specifies the internal 10 MHz crystal oscillator as the timebase reference.

- **EXTernal**

This specifies the user-supplied external signal as the timebase reference.

## Example 1

```
ROSc:SOUrce INTERNAL
```

This command specifies the internal 10 MHz crystal oscillator as the timebase reference.

## Example 2

```
ROSc:SOUrce?
```

This query might return

```
:ROSC:SOURCE INTERNAL
```

indicating that the 10 MHz crystal oscillator is being used as the timebase reference.

---

# ROSc:STATE?

## Description

This query-only command returns the state of the 10 MHz timebase reference. This command will return either LOCKED or UNLOCKED.

## Group

Miscellaneous

## Related Commands

ROSc:SOUrce

## Syntax

ROSc:STATE?

## Example

ROSc:STATE?

This query might return

:ROSc:STATE LOCKED

indicating that the state of the 10 MHz timebase reference is locked.

---

# SAVe:MASK

## Description

This command (no query form) saves the User mask to a mass storage file.

## Group

## Mask

## Related Commands

## RECAII:MASK

## Syntax

```
SAVe:MASK <Qstring>
```

## Arguments

- <Qstring>

This is the location to where the mask is stored. <file path> is a quoted string that defines the file name and path. Input the file path using the form <drive>/<dir>/<filename>. <drive> and one or more <dir>s are optional.

The file is stored in ASCII format and is user-editable.

## Example

```
SAVe:MASK "TEST"
```

This command saves the User mask to the file TEST in the default directory and on the default drive.

---

# SAVe:SETUp

## Description

This command (no query form) stores the state of the instrument to a specified memory location. You can later use the \*RCL command to restore the instrument to this saved state. This is equivalent to selecting Save As from the File menu, and then pressing the Setup button.

## Group

Save and Recall

## Related Commands

\*RCL, RECALL:SETUp, \*SAV

## Syntax

```
SAVe:SETUp {<file path>|<NR1>}
```

## Arguments

- <file path>

This is the target location for storing the setup file.

<file path> is a quoted string that defines the file name and path. Input the file path using the form <drive>:<dir>/<filename>. <drive> and one or more <dir>s are optional. If you do not specify them, the instrument will store the file in the default directory (C:\TekScope\setups). <filename> stands for a filename of up to 128 characters. (Use of wildcard characters in filenames is not supported.) Filename extensions are not required but are highly recommended. For setups, use the extension ".SET".

- <NR1>

This specifies a location for saving the current front-panel setup. The front-panel setup value ranges from 1 to 10. Using an out-of-range value causes an execution error. Any settings that have been stored previously at this location will be overwritten.

## Example 1

```
SAVe:SETUp 5
```

This command saves the current instrument setup in memory location 5.

## Example 2

```
SAVe:SETUp "TEK00000.SET"
```

This command saves the current instrument setup in the file TEK00000.SET in the default directory for setups (C:\TekScope\setups).

---

# SAVe:WAVEform

## Description

This command (no query form) saves a waveform to one of four reference memory locations or a file. This command is equivalent to selecting Reference Waveforms from the File menu and choosing Save Wfm from the drop-down list.

## Group

Save and Recall

## Related Commands

DELEte:WAVEform, RECALL:WAVEform

## Syntax

```
SAVe:WAVEform <wfm>,{<file path>|REF<x>}
```

## Arguments

- <wfm>

This is the waveform that will be saved. Valid waveforms include CH<x>, MATH<y>, and REF<x>.

- REF<x>

This specifies the location in internal reference memory to where the waveform is stored.

- <file path>

This is the location to where the waveform is stored.

<file path> is a quoted string that defines the file name and path. Input the file path using the form <drive>/<dir>/<filename>. <drive> and one or more <dir>s are optional. If you do not specify them, the instrument will read the waveform from the default directory. <filename> stands for a filename of up to 125 characters, followed by a period (".") and the three-character extension. Internal format waveforms use the .wfm extension for spreadsheet format files or the .dat extension for MathCad format files. The instrument will generate an error if you use any other extension for saving a waveform.

## Example 1

```
SAVe:WAVEform MATH2,REF2
```

This command saves the Math2 waveform in reference memory location2.

## Example 2

```
SAVe:WAVEform MATH1,"TEK0000.WFM"
```

This command saves the Math1 waveform to the file TEK00000.WFM in the default directory and on the default drive.



---

# SAVe:WAVEform:FILEFormat

## Description

This command specifies or returns the file format for saved waveforms. Waveform header and timing information is included in the resulting file of non-internal formats. The instrument saves DPO waveforms as a 500 x 200 matrix, with the first row corresponding to the most recently acquired data. The values specified by DATA:START and DATA:STOP determine the range of waveform data to output. In the event that DATA:STOP value is greater than the current record length, the current record length determines the last output value.

This command is equivalent to selecting Save As from the File menu, clicking the Waveform button, and choosing the desired waveform file format from the Save as type drop-down list.

## Group

Save and Recall

## Related Commands

CURVe , DATA, DATA:START, DATA:STOP, SAvE:WAVEform, WFMInpre:NR\_FR?, WFMInpre:NR\_Pt, WFMOupre:NR\_FR?, WFMOupre:NR\_Pt?

## Syntax 1

```
SAVe:WAVEform:FILEFormat {INTERNAL|MATH-  
Cad|MATLab|SPREADSHEETCsv|SPREADSHEETTxt}
```

## Syntax 2

```
SAVe:WAVEform:FILEFormat?
```

## Arguments

### ■ INTERNAL

This specifies that waveforms are saved in an internal format, using a .wfm filename extension. These files can be recalled as reference waveforms. When this argument is specified, the settings specified via the DATA:START and DATA:STOP commands have no meaning as the entire waveform is saved.

### ■ MATHCad

This specifies that waveforms are saved in MathCad format, using a .dat filename extension. When saving in this format, waveform values are delimited with new lines. MathCad format enables easy import of waveform data into MathCad or MATLAB.

For FastAcq waveforms, data is imported as a matrix. For these formats, waveform header and timing information is saved in a separate header file. MathCad format header files use a \_hdr.dat extension.

---

- **MATLab**

This specifies that waveforms are saved in Matlab format, using a .dat filename extension. When saving in this format, waveform values are delimited with new lines. MATLAB format enables easy import of waveform data into MathCad or MATLAB.

For FastAcq waveforms, data is imported as a matrix. For these formats, waveform header and timing information is saved in a separate header file. MATLAB format header files use a \_hdr.dat extension.

- **SPREADSHEETCsv**

This specifies that waveform data is saved in a format that contains comma delimited values. These waveform data files are named using the .csv filename extension. Saving waveforms in CSV format enables spreadsheet programs to import the data.

- **SPREADSHEETTtxt**

This specifies that waveform data is saved in a format that contains tab delimited values. These waveform data files are named using the .txt filename extension. Saving waveforms in this format enables spreadsheet programs to import the data.

### **Example 1**

```
SAVE:WAVEform:FILEFormat INTERNAL
```

This command specifies that the internal file format is the format used for saving waveforms.

### **Example 2**

```
SAVE:WAVEform:FILEFormat?
```

This query might return

```
:SAVE:WAVEFORM:FILEFORMAT INTERNAL
```

indicating that waveforms are saved using the internal format.

---

# SElect:<wfm>

## Description

This command turns on the display of a specified waveform and also resets the acquisition. The query returns whether the channel is on or off but does not indicate whether it is the selected waveform.

## Group

Vertical

## Syntax 1

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

## Syntax 2

```
SElect:<wfm>?
```

## Arguments

- ON

This turns on the display of the specified waveform. This waveform also becomes the selected waveform.

- OFF

This turns off the display of the specified waveform.

- <NR1>

A 0 turns off the display of the specified waveform; any other value turns on the display of the specified waveform.

## Example

```
SElect:CH2 ON
```

This command turns the channel 2 waveform display on, and selects channel 2.

```
SElect:REF1?
```

This query might return

```
:SELECT:REF1
```

indicating that REF1 is the selected waveform.

---

# SElect:CONTRol <wfm>

## Description

This command sets or queries the waveform that is selected as the implied recipient of channel-related commands that support legacy-style programs. The command form also performs the equivalent of a SELECT:<wfm> ON command. This command is equivalent to selecting Measurement Setup from the Measure menu and either viewing or setting the Source waveform.

## Group

Vertical

## Syntax 1

```
SElect:CONTROL (CH<x> | MATH<x> | REF<x>)
```

## Syntax 2

```
SElect:CONTROL?
```

## Arguments

### ■ CH<x>

This selects the specified channel waveform as the waveform that is affected by the front-panel controls. The x variable can be expressed as an integer ranging from 1 through 4 for 4-channel models and 1 to 2 for 2-channel models.

### ■ MATH<x>

This selects the specified math waveform as the waveform that is affected by the front-panel controls. The x variable can be expressed as an integer ranging from 1 through 4 for 4-channel models and 1 to 2 for 2-channel models.

### ■ REF<x>

This selects the specified reference waveform as the waveform that is affected by the front-panel controls. The x variable can be expressed as an integer ranging from 1 through 4 for 4-channel models and 1 to 2 for 2-channel models.

## Example 1

```
SElect:CONTROL CH2
```

This command resets acquisition displays on channel 2, and causes the selected waveform to be the implied object of waveform commands.

## Example 2

```
SElect:CONTROL?
```

This query might return

```
:SELECT:CONTROL MATH3
```

indicating that math 3 is the implied object of waveform commands.

---

# SElect?

## Description

This query-only command returns the selected waveform that is affected by the front-panel controls and the display status (on or off) of all waveforms. This query command is equivalent to selecting Measurement Setup from the Measure menu and viewing the Source waveform setting.

## Group

Vertical

## Syntax

SElect?

## Example

SElect?

This query might return the waveform and status information as

```
:SELECT:CH1 1;CH2 0;CH3 0;CH4 0;MATH1 0;  
MATH2 0;MATH3 0;MATH4 0;REF1 0;REF2 0;  
REF3 0;REF4 0;CONTROL CH1
```

---

# SETUp:NAME

## Description

This command allows you to create (or query) a name for your saved setups. The default name for all user setups is "User." The default name for factory setups is "Factory."

This command is equivalent to selecting Save As from the File menu, pressing the Setup button, selecting the desired setup location, clicking the Keyboard icon, and entering your setup name.

## Group

Save and Recall

## Related Commands

SAVe:SETUp

## Syntax 1

```
SETUp:NAME <NR1>,<QString>
```

## Syntax 2

```
SETUp:NAME? <NR1>
```

## Arguments

- <NR1>

This specifies a location in which the setup label is stored. Location values range from 1 through 10.

- <QString>

This is a string containing the setup label.

## Example 1

```
SETUp:NAME 1, "My Setup"
```

This command changes the setup label for internal setup location 1 to "My Setup".

## Example 2

```
SETUp:NAME? 2
```

This query might return

```
:SETUP:NAME 2,"My Setup"
```

indicating that "My Setup" is the setup label that you defined for internal setup location 2.

---

# SET?

## Description

This query-only command returns the commands that list the instrument settings, except for configuration information for the calibration values. You can use these commands to return the instrument to the state it was in when you made the SET? query. This command is identical to the \*LRN? command.

## Group

Miscellaneous

## Related Commands

HEADer, \*LRN?, VERBose

## Syntax

SET?

**Note:** The SET? query always returns command headers, regardless of the setting of the HEADer command. This is because the returned commands are intended to be sent back to the instrument as a command string. The VERBose command can still be used to specify whether the returned headers should be abbreviated or full-length.

## Example

SET?

This query might return the following response:

```
:ACQUIRE:STOPAFTER RUNSTOP;STATE 1;MODE SAMPLE;NUMENV 10;NUMAVG 16;REPET 1;;FASTACQ:STATE 0;;APPLICATION:GPKNOB1:ACTIVE 0;;APPLICATION:GPKNOB2:ACTIVE 0;;APPLICATION:WINDOW:HEIGHT 236;WIDTH 640;;APPLICATION:SCOPEAPP:STATE RUNNING;WINDOW FULLSCREEN;;APPLICATION:EXTAPP:STATE NOTRUNNING;;AUXOUT:SOURCE ATRIGGER;EDGE FALLING;;CMDBATCH 1;;HEADER 1;;LOCK NONE;;ROSC:SOURCE INTERNAL;;VERBOSE 1;;ALIAS:STATE 0;;DISPLAY:CLOCK 1;COLOR:PALETTE NORMAL;MATHCOLOR DEFAULT;REFCOLOR DEFAULT;;DISPLAY:FILTER SINX;FORMAT YT;GRATICULE FULL;INTENSITY:WAVEFORM 75.0000;AUTOBRIGHT 1;SCREENSAVER 1;SCREENSAVERDELAY 28800;;DISPLAY:PERSISTENCE OFF;STYLE VECTORS;TRIGBAR SHORT;TRIGT 1;VARPERSIST 500.0000E-3;;HARDCOPY:FILENAME "untitled";PORT FILE;;DIAG:LEVEL SUBSYS;;SAVE:WAVEFORM:FILEFORMAT INTERNAL;;TRIGGER:A:MODE AUTO;TYPE EDGE;LEVEL 0.0000;HOLDOFF:BY DEFAULT;TIME 1.5000E-6;;TRIGGER:A:EDGE:SOURCE CH1;COUPLING DC;SLOPE RISE;...
```

---

# TEKSecure

## Description

This command initializes both waveform and setup memories, overwriting any previously stored data.

TEKSecure deletes all four waveform reference memory slots on the hard drive, if they exist, and puts all setups in the factory-initialized state. External setups that are stored on the hard drive are not affected.

## Group

Miscellaneous

## Syntax

```
TEKSecure
```

## Example

```
TEKSecure
```

This command initializes both waveform and setup memories.



---

# TEST

## Description

This command (no query form) provides the ability to select and execute an item at any level of the test hierarchy (Test, Area or Subsystem). This command is equivalent to selecting Instrument Diagnostics from the Utilities menu, choosing a test and then pressing Run.

**Note:** Turning off both DIAG:CONTROL:HALT and DIAG:CONTROL:LOOP before executing the TEST command is recommended.

## Group

Diagnostics

## Related Commands

TEST:RESults?, TEST:RESults:VERBose?, TEST:STOP

## Syntax

TEST

## Argument

- <NR3>

This sets the test ID, which ranges from 0 through 3 characters. If no test ID is specified, all available diagnostics are executed.

## Example 1

```
TEST "1"
```

This command executes all Acquisition tests (Subsystem:Acquisition).

## Example 2

```
TEST "11"
```

This command executes all Acquisition Memory tests (Subsystem:Acquisition, Area:Memory)

## Example 3

```
TEST "113"
```

This command executes a specific Acquisition Memory test (Subsystem:Acquisition, Area:Memory, Test:Address Lines).

---

# TEST:RESults?

## Description

This query-only command returns an abbreviated status about the results of the last TEST execution. This command is equivalent to selecting Instrument Diagnostics from the Utilities menu and then reviewing the Diagnostic Status.

## Group

Diagnostics

## Related Commands

TEST, TEST:RESults:VERBose?

## Syntax

TEST:RESults?

## Example 1

TEST:RESults?

This query might return

```
:TEST:RESULTS "135"
```

indicating the specific failure code of the first test failure.

## Example 2

TEST:RESults?

This query might return

```
:TEST:RESULTS "PASS"
```

indicating that the previously executed test passed.

## Example 3

TEST:RESults?

This query might return

```
:TEST:RESULTS "*****"
```

indicating that the test was not run.

---

# TEST:RESults:VERBose?

## Description

This query-only command returns a more explanatory message about the results of the last TEST execution than the TEST:RESults query. This command is equivalent to selecting Instrument Diagnostics from the Utilities menu and then clicking the Error Log button.

## Group

Diagnostics

## Related Commands

TEST, TEST:RESults?

## Syntax

TEST:RESults:VERBose?

## Example

TEST:RESults:VERBose?

This query might return

```
:TEST:RESULTS:VERBOSE DIAG ABORTED, ERROR 15  
in A2D-Dmux 1 interconnects".
```

---

# TEST:STOP

## Description

This command (no query form) causes test (or diagnostics) execution to terminate at the end of the next low-level test. This command is equivalent to selecting Instrument Diagnostics from the Utilities menu and then clicking Abort.

## Group

Diagnostics

## Related Commands

DIAG:STOP

## Syntax

TEST:STOP

## Example

TEST:STOP

This command terminates test (or diagnostics) execution at the end of the next low-level test.

---

# TIME

## Description

This command sets or queries the time that the instrument displays. This command is equivalent to selecting Set Time & Date from the Utilities menu and then setting the fields in the Time group box.

## Group

Miscellaneous

## Related Commands

DATE

## Syntax 1

```
TIME <Qstring>
```

## Syntax 2

```
TIME?
```

## Argument

- <QString>

This is a time in the form "hh:mm:ss" where hh refers to a two-digit hour number, mm refers to a two-digit minute number from 01 to 60, and ss refers to a two-digit second number from 01 to 60.

## Example 1

```
TIME "14:00:00"
```

This command sets the time to exactly 2:00 p.m.

## Example 2

```
TIME?
```

This query might return

```
:TIME "14:05:17"
```

indicating the current time is set to 2:05 p.m. and 17 seconds.

---

# TRIGger

## Description

This command forces a trigger event to occur. The query returns the current trigger parameters for the instrument.

## Group

Trigger

## Syntax 1

TRIGger FORCe

## Syntax 2

TRIGger?

## Argument

- FORCe

This creates a trigger event. If TRIGger:STATE is set to READy, the acquisition will complete. Otherwise, this command will be ignored. This is equivalent to selecting A Event (Main) Trigger Setup from the Trig menu and then clicking Force Trigger.

## Example 1

TRIGger FORCe

This command forces a trigger event to occur.

## Example 2

TRIGger?

This query might return these trigger parameters:

```
:TRIGGER:A:MODE AUTO;TYPE EDGE;LEVEL
0.0000;HOLDOFF:BY DEFAULT;TIME
1.5000E-6;:TRIGGER:A:EDGE:SOURCE CH1;COUPLING
DC;SLOPE RISE;:TRIGGER:A:LOGIC:CLASS PAT-
TERN;FUNCTION AND;THRESHOLD:CH1 1.2000;CH2
1.2000;CH3 1.2000;CH4 1.2000;:TRIGGER:A:LOG-
IC:INPUT:CH1 HIGH;CH2 X;CH3 X;:TRIGGER:A:LOG-
IC:PATTERN:INPUT:CH4 X;:TRIGGER:A:LOGIC:PAT-
TERN:WHEN TRUE;WHEN:LESSLIMIT
5.0000E-9;MORELIMIT 5.0000E-9;:TRIGGER:A:LOG-
IC:SETHOLD:CLOCK:EDGE RISE;THRESHOLD
1.2000;SOURCE CH2;:TRIGGER:A:LOGIC:SE-
THOLD:DATA:THRESHOLD 1.2000;SOURCE CH1;:TRIG-
GER:A:LOGIC:SETHOLD:HOLDTIME 2.0000E-9;SETTIME
3.0000E-9;:TRIGGER:A:LOGIC:STATE:INPUT:CH4
RISE;:TRIGGER:A:LOGIC:STATE:WHEN TRUE;:TRIG-
GER:A:PULSE:CLASS GLITCH;SOURCE
CH1;GLITCH:WIDTH 2.0000E-9;TRIGIF ACCEPT;PO-
```

---

```
LARITY POSITIVE;:TRIGGER:A:PULSE:WINDOW:TYPE
INSIDE;WHEN OCCURS;WIDTH 2.0000E-9;LOGIC:IN-
PUT:CH1 HIGH;CH2 HIGH;CH3 HIGH;CH4 HIGH;:TRIG-
GER:A:PULSE:WINDOW:LOGIC:THRESHOLD:CH1
1.2000;CH2 1.2000;CH3 1.2000;CH4 1.2000;:TRIG-
GER:A:PULSE:WINDOW:THRESHOLD:HIGH 1.2000;LOW
800.0000E-3;:TRIGGER:A:PULSE:RUNT:LOGIC:IN-
PUT:CH1 HIGH;CH2 HIGH;CH3 HIGH;CH4 HIGH;:TRIG-
GER:A:PULSE:RUNT:LOGIC:THRESHOLD:CH1
1.2000;CH2 1.2000;CH3 1.2000;CH4 1.2000;:TRIG-
GER:A:PULSE:RUNT:POLARITY POSITIVE;THRESH-
OLD:HIGH 1.2000;LOW 800.0000E-3;:TRIG-
GER:A:PULSE:RUNT:WHEN OCCURS;WIDTH
2.0000E-9;:TRIGGER:A:PULSE:TRANSITION:DELTA-
TIME 2.0000E-9;POLARITY POSITIVE;THRESH-
OLD:HIGH 1.2000;LOW 800.0000E-3;:TRIG-
GER:A:PULSE:TRANSITION:WHEN
FASTER THAN;:TRIGGER:A:PULSE:WIDTH:LOWLIMIT
2.0000E-9;HIGHLIMIT 2.0000E-9;WHEN WITHIN;PO-
LARITY POSITIVE;:TRIGGER:A:PULSE:TIMEOUT:PO-
LARITY STAYSHIGH;TIME 2.0000E-9;:TRIG-
GER:A:VIDEO:CUSTOM:FORMAT INTERLACED;SCAN
RATE1;:TRIGGER:A:VIDEO:FIELD ALLFIELDS;HOLD-
OFF:FIELD 0.0000;:TRIGGER:A:VIDEO:LINE 1;PO-
LARITY NORMAL;SCAN RATE1;SOURCE CH1;STANDARD
NTSC;:TRIGGER:A:COMMUNICATION:STANDARD CUS-
TOM;CODE NRZ;BITRATE 1544000;AMI:PULSEFORM
PLUSONE;THRESHOLD:HIGH 0.0000;LOW
0.0000;:TRIGGER:A:COMMUNICATION:CLOCK:POLARITY
RISE;:TRIGGER:A:COMMUNICATION:CMI:PULSEFORM
PLUSONE;:TRIGGER:A:COMMUNICATION:SOURCE
CH1;SOURCE:TYPE DATA;:TRIGGER:B:STATE 0;TYPE
EDGE;LEVEL 0.0000;BY EVENTS;EDGE:SOURCE
CH1;SLOPE RISE;COUPLING DC;:TRIGGER:B:TIME
16.0000E-9;EVENTS:COUNT 2
```

---

# TRIGger:A

## Description

This command sets the A trigger level automatically to 50% of the range of the minimum and maximum values of the trigger input signal. The query returns current A trigger parameters. The trigger level is the voltage threshold through which the trigger source signal must pass to generate a trigger event. This command is equivalent to pushing the LEVEL knob on the front panel.

## Group

Trigger

## Related Commands

TRIGger:A:EDGE?, TRIGger:A:LOGic?, TRIGger:A:PULSE?

## Syntax 1

```
TRIGger:A SETLevel
```

## Syntax 2

```
TRIGger:A?
```

## Argument

- SETLevel

This sets the A trigger level to 50% of the range of the minimum and maximum values of the trigger input signal.

## Example 1

```
TRIGger:A SETLevel
```

This command sets the A trigger level to 50% of the range of the minimum and maximum values of the trigger input signal.

## Example 2

```
TRIGger:A?
```

This query might return the following A trigger parameters:

```
:TRIGGER:A:MODE AUTO;TYPE EDGE;LEVEL  
0.0000;HOLDOFF:BY DEFAULT;TIME  
1.5000E-6;:TRIGGER:A:EDGE:SOURCE CH1;COUPLING  
DC;SLOPE RISE;:TRIGGER:A:LOGIC:CLASS PAT-  
TERN;FUNCTION AND;THRESHOLD:CH1 1.2000;CH2  
1.2000;CH3 1.2000;CH4 1.2000;:TRIGGER:A:LOG-  
IC:INPUT:CH1 HIGH;CH2 X;CH3 X;:TRIGGER:A:LOG-  
IC:PATTERN:INPUT:CH4 X;:TRIGGER:A:LOGIC:PAT-  
TERN:WHEN TRUE;WHEN:LESSLIMIT  
5.0000E-9;MORELIMIT 5.0000E-9;:TRIGGER:A:LOG-  
IC:SETHOLD:CLOCK:EDGE RISE;THRESHOLD
```



---

```
1.2000;SOURCE CH2;:TRIGGER:A:LOGIC:SE-
THOLD:DATA:THRESHOLD 1.2000;SOURCE CH1;:TRIG-
GER:A:LOGIC:SETHOLD:HOLDTIME 2.0000E-9;SETTIME
3.0000E-9;:TRIGGER:A:LOGIC:STATE:INPUT:CH4
RISE;:TRIGGER:A:LOGIC:STATE:WHEN TRUE;:TRIG-
GER:A:PULSE:CLASS GLITCH;SOURCE
CH1;GLITCH:WIDTH 2.0000E-9;TRIGIF ACCEPT;PO-
LARITY POSITIVE;:TRIGGER:A:PULSE:WINDOW:TYPE
INSIDE;WHEN OCCURS;WIDTH 2.0000E-9;LOGIC:IN-
PUT:CH1 HIGH;CH2 HIGH;CH3 HIGH;CH4 HIGH;:TRIG-
GER:A:PULSE:WINDOW:LOGIC:THRESHOLD:CH1
1.2000;CH2 1.2000;CH3 1.2000;CH4 1.2000;:TRIG-
GER:A:PULSE:WINDOW:THRESHOLD:HIGH 1.2000;LOW
800.0000E-3;:TRIGGER:A:PULSE:RUNT:LOGIC:IN-
PUT:CH1 HIGH;CH2 HIGH;CH3 HIGH;CH4 HIGH;:TRIG-
GER:A:PULSE:RUNT:LOGIC:THRESHOLD:CH1
1.2000;CH2 1.2000;CH3 1.2000;CH4 1.2000;:TRIG-
GER:A:PULSE:RUNT:POLARITY POSITIVE;THRESH-
OLD:HIGH 1.2000;LOW 800.0000E-3;:TRIG-
GER:A:PULSE:RUNT:WHEN OCCURS;WIDTH
2.0000E-9;:TRIGGER:A:PULSE:TRANSITION:DELTA-
TIME 2.0000E-9;POLARITY POSITIVE;THRESH-
OLD:HIGH 1.2000;LOW 800.0000E-3;:TRIG-
GER:A:PULSE:TRANSITION:WHEN
FASTER THAN;:TRIGGER:A:PULSE:WIDTH:LOWLIMIT
2.0000E-9;HIGHLIMIT 2.0000E-9;WHEN WITHIN;PO-
LARITY POSITIVE;:TRIGGER:A:PULSE:TIMEOUT:PO-
LARITY STAYSHIGH;TIME 2.0000E-9;:TRIG-
GER:A:VIDEO:CUSTOM:FORMAT INTERLACED;SCAN
RATE1;:TRIGGER:A:VIDEO:FIELD ALLFIELDS;HOLD-
OFF:FIELD 0.0000;:TRIGGER:A:VIDEO:LINE 1;PO-
LARITY NORMAL;SCAN RATE1;SOURCE CH1;STANDARD
NTSC;:TRIGGER:A:COMMUNICATION:STANDARD CUS-
TOM;CODE NRZ;BITRATE 1544000;AMI:PULSEFORM
PLUSONE;THRESHOLD:HIGH 0.0000;LOW
0.0000;:TRIGGER:A:COMMUNICATION:CLOCK:POLARITY
RISE;:TRIGGER:A:COMMUNICATION:CMI:PULSEFORM
PLUSONE;:TRIGGER:A:COMMUNICATION:SOURCE
CH1;SOURCE:TYPE DATA
```

---

# TRIGger:A:COMMunication:AMI...PULSEForm

## (Option SM Only)

### Description

This entry covers five separate commands, one each for AMI and AMI subtypes. The commands set or query the AMI pulse form to one of three possibilities.

### Group

Trigger

### Syntax 1

```
TRIGger:A:COMMunication:(AMI|HDB3|B3ZS|B6ZS|B8ZS):PULSEForm {PLUSOne|MINUSOne|EYEDiagram}
```

### Syntax 2

```
TRIGger:A:COMMunication:(AMI|HDB3|B3ZS|B6ZS|B8ZS):PULSEForm?
```

### Arguments

- PLUSOne

Corresponds to the Isolated +1 on the front panel menu.

- MINUSOne

Corresponds to the Isolated -1 on the front panel menu.

- EYEDiagram

Positions the trigger point off screen in order to display the eye.

### Example 1

```
TRIGger:A:COMMunication:AMI:PULSEForm PLUSOne
```

This commands sets the pulse form to PLUSONE.

### Example 2

```
TRIGger:A:COMMunication:AMI:PULSEForm?
```

This query might return

```
:TRIGGER:A:COMMUNICATION:AMI:PULSEFORM PLUSONE
```

indicating that the AMI PULSFORM is set to PLUSONE.

---

# TRIGger:A:COMMunication:AMI...THReshold:HIGH

## (Option SM Only)

### Description

This command sets or queries the threshold high level.

### Group

Trigger

### Syntax 1

```
TRIGger:A:COMMunica-  
tion:(AMI|HDB3|B3ZS|B6ZS|B8ZS):THReshold:HIGH  
<NR3>
```

### Syntax 2

```
TRIGger:A:COMMunication:(AMI|HDB3|B3ZS|  
B6ZS|B8ZS):THReshold:HIGH?
```

### Arguments

- <NR3>

The high threshold parameter expressed in volts.

### Example 1

```
TRIGger:A:COMMUNICATION:AMI:THRESHOLD:HIGH  
2.8E-2
```

This command sets the threshold high level to 2.8E-2.

### Example 2

```
TRIGger:A:COMMUNICATION:AMI:THRESHOLD:HIGH?
```

This query might return

```
TRIGGER:A:COMMUNICATION:AMI:THRESHOLD:HIGH  
2.8E-2
```

indicating the threshold high level in volts.

---

# TRIGger:A:COMMunication:AMI...THReshold:LOW

## (Option SM Only)

### Description

This command sets or queries the threshold low level.

### Group

Trigger

### Syntax 1

```
TRIGger:A:COMMunica-  
tion:(AMI|HDB3|B3ZS|B6ZS|B8ZS):THReshold:LOW  
<NR3>
```

### Syntax 2

```
TRIGger:A:COMMunication:(AMI|HDB3|B3ZS|  
B6ZS|B8ZS):THReshold:LOW?
```

### Arguments

- <NR3>

The high threshold parameter expressed in volts.

### Example 1

```
TRIGger:A:COMMUNICATION:AMI:THRESHOLD:LOW  
2.8E-2
```

This command sets the threshold low level to 2.8E-2.

### Example 2

```
TRIGger:A:COMMUNICATION:AMI:THRESHOLD:LOW?
```

This query might return

```
TRIGGER:A:COMMUNICATION:AMI:THRESHOLD:LOW  
2.8E-2
```

indicating the threshold low level in volts.

---

# TRIGger:A:COMMunication:BITRate

## (Option SM Only)

### Description

This command sets or queries the bit rate.

### Group

Trigger

### Syntax 1

```
TRIGger:A:COMMunication:BITRate <NR3>
```

### Syntax 2

```
TRIGger:A:COMMunication:BITRate?
```

### Arguments

- <NR3>

A non-negative number greater than one and expressed as bits per second.

### Example 1

```
TRIGger:A:COMMunication:BITRate 1.544e6
```

This command sets the bit rate to 1.544e6 bits per second.

### Example 2

```
TRIGger:A:COMMunication:BITRate?
```

This query might return

```
:TRIGGER:A:COMMUNICATION:BITRATE 1.544e6
```

indicating a bit rate of 1.544 Mb/s.

---

# TRIGger:A:COMMunication:CLOCK:POLarity

## (Option SM Only)

### Description

This command sets or queries the communication clock polarity

### Group

Trigger

### Syntax 1

```
TRIGger:A:COMMunication:CLOCK:POLarity  
(RISe|FALL)
```

### Syntax 2

```
TRIGger:A:COMMunication:CLOCK:POLarity?
```

### Arguments

- RISE

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

- FALL

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

### Example 1

```
TRIGger:A:COMMunication:CLOCK:POLarity RISE
```

This command sets RISE as the clock polarity.

### Example 2

```
TRIGger:A:COMMunication:CLOCK:POLarity?
```

This query might return

```
:TRIGGER:A:COMMUNICATION:CLOCK:POLARITY RISE
```

indicating that the polarity is to trigger on the rising edge of the clock.

---

# TRIGger:A:COMMunication:CMI:PULSEForm

## (Option SM Only)

### Description

This command sets or queries the CMI pulse form.

### Group

Trigger

### Syntax 1

```
TRIGger:A:COMMunication:CMI:PULSEForm  
(PLUSOne | MINUSOne | ZERO | EYEDiagram)
```

### Syntax 2

```
TRIGger:A:COMMunication:CMI:PULSEForm?
```

### Arguments

- PLUSOne

Triggers on a positive mark.

- MINUSOne

Triggers on a negative mark.

- ZERO

Triggers on the bit representing zero.

- EYEDiagram

Positions the trigger point off screen in order to display the eye.

### Example 1

```
TRIGger:A:COMMunication:CMI:PULSEForm PLUSOne
```

This command sets triggering to occur on a positive mark.

### Example 2

```
TRIGger:A:COMMunication:CMI:PULSEForm?
```

This query might return

```
:TRIGGER:A:COMMUNICATION:CMI:PULSEFORM PLUSONE
```

indicating that the pulse form is PLUSONE.

---

# TRIGger:A:COMMunication:CODe

## (Option SM Only)

### Description

This command sets or queries the signal code that the communications trigger should expect on the incoming signal.

### Group

Trigger

### Syntax 1

```
TRIGger:A:COMMunication:CODe  
{AMI | HDB3 | B3ZS | B6ZS | B8ZS | CMI | NRZ | MLT3 | MANChes-  
ter}
```

### Syntax 2

```
TRIGger:A:COMMunication:CODe?
```

### Arguments

- AMI
- HDB3
- B3ZS
- B6ZS
- B8ZS
- CMI
- NRZ
- MLT3
- MANChester

### Example 1

```
TRIGger:A:COMMunication:CODe AMI
```

This command selects AMI as the line code that the communications trigger expects on the incoming signal.

### Example 2

```
TRIGger:A:COMMunication:CODe?
```

This query might return

```
:TRIGGER:A:COMMUNICATION:CODE AMI
```

indicating that the code is AMI.



---

# TRIGger:A:COMMunication:SOURce

## (Option SM Only)

### Description

This command sets or queries the source channel.

### Group

Trigger

### Syntax 1

```
TRIGger:A:COMMunication:SOURce  
{CH1|CH2|CH3|CH4}
```

### Syntax 2

```
TRIGger:A:COMMunication:SOURce?
```

### Arguments

- CH1

This argument selects CH 1 as the source channel.

- CH2

This argument selects CH 2 as the source channel.

- CH3

This argument selects CH 3 as the source channel.

- CH4

This argument selects CH 4 as the source channel.

### Example 1

```
TRIGger:A:COMM:SOURCE CH1
```

This command sets channel 1 as the source channel.

### Example 2

```
TRIGger:A:COMMunication:SOURce?
```

This query might return

```
:TRIGGER:A:COMMUNICATION:SOURCE CH1
```

indicating that the source is channel 1.

---

# TRIGger:A:COMMunication:SOURce:TYPE

## (Option SM Only)

### Description

This command sets or queries the source type. This command works only when the Eye Diagram pulseform is selected.

TRIGger:MAIN:COMMunication:SOURce:TYPE works in an identical fashion to this command and provides compatibility with previous Tektronix oscilloscopes.

### Group

Trigger

### Syntax 1

```
TRIGger:A:COMMunication:SOURce:TYPE  
{DATA | CLOCK}
```

### Syntax 2

```
TRIGger:A:COMMunication:SOURce:TYPE?
```

### Arguments

- DATA

The oscilloscope triggers and shifts five unit intervals to form the expected eye pattern.

- CLOCK

The oscilloscope triggers but no shift occurs. The clock type causes random triggers with respect to the data channel which must be one of the other three channels.

### Example 1

```
TRIGger:A:COMMunication:SOURce:TYPE CLOCK
```

This command sets the source type to clock..

### Example 2

```
TRIGger:A:COMMunication:SOURce:TYPE?
```

This query might return

```
:TRIGGER:A:COMMUNICATION:SOURCE:TYPE DATA
```

indicating that the source type is data.

---

# TRIGger:A:COMMunication:STANdard

(Option SM Only)

## Description

This command sets or queries the standard that identifies the code and bit rate. The bit rate is used to compute the Unit Interval, which is the inverse of the bit rate. The Unit Interval influences time skew in an Eye Diagram, where you do post processing on AMI isolated pulses, and pulse width settings if CMI.

## Group

Trigger

## Related Commands

MASK:STANdard

## Syntax 1

```
TRIGger:A:COMMunication:STANdard
{ATAG1|ATAG2|ATAG3|DSOContra|
DSODoubleDSOSingle|DSOTiming|
DS1|DS1A|DS1C|DS2|DS2RATESymmetrical|
DS2RATECoax|RATE32Mbit|RATE97Mbit|
DS3|DS4NA|E1|E2|E3|E4|STS1|STS3|STM1E|
FC133|FC266|FC531|1063|FC2125|OC1|OC3|
OC12|OC48|OC48_FEC|STM1E|ENET100|ENET1250|
ENETXAUI|INF2_5G|FST|HST|FW1394BS400B
|FW1394BS1600B|Custom|RIO_500M|RIO_750M|
RIO_1G|RIO_1_5G|RIO_2G|RIO_SERIAL_1G|
RIO_SERIAL_2G|RIO_SERIAL_3G|SFI5_2|
RATE32Mbit|RATE97Mbit|SFI5_3|VSROC192|
TFI5_2|TFI5_3|PCIExpress}
```

## Syntax 2

```
TRIGger:A:COMMunication:STANdard?
```

## Arguments

The following table shows various standards and their corresponding parameters.

Communication Trigger Standards				
Argument	Code	Bit Rate	Mask Type	Models
ATAG1	NRZ	1.5 Gb/s	Serial ATA	
ATAG2	NRZ	3.0 Gb/s	Serial ATA	
ATAG3	NRZ	6.0 Gb/s	Serial ATA	
CLOCK Coax	None	2.048 Mb/s	ITU-T G.703	All
CLOCK Symmetrical	None	2.048 Mb/s	ITU-T G.703	All
D1	NRZ	270 Mb/s	Video	All
D2	NRZ	143.18 Mb/s	Video	All
DS0 Contra				
DS0 Double				
DS0 Single				
DS0 Timing				
DS1	AMI B8ZS	1.544 Mb/s	ANSI T1.102	All
DS1A	AMI HDB3	2.048 Mb/s	ANSI T1.102	All
DS1C	AMI B8ZS	3.152 Mb/s	ANSI T2.102	All
DS2	AMI B6ZS	6.312 Mb/s	ANSI T2.102	All
DS2RAT ECoax	AMI B8ZS	6.312 Mb/s	ITU-T G.703	All
DS2RAT ESymmetrical	AMI B6ZS	6.312 Mb/s	ITU-T G.703	All
DS3	AMI B3ZS	44.736 Mb/s	ANSI T1.102	All
DS4NA	CMI	139.3 Mb/s	ANSI T1.102	All

E1	AMI HDB3	2.048 Mb/s	ITU-T G.703	All
E2	AMI HDB3	8.448 Mb/s	ITU-T G.703	All
E3	AMI HDB3	34.368 Mb/s	ITU-T G.703	All
E4	CMI	139.26	ITU-T G.703	All
ENET100	MLT3	100 Mb/s	Ethernet	All
ENET1250	NRZ	1.25 Gb/s	Ethernet	
ENETXAU	NRZ	3.125 Gb/s	Ethernet	
FC133	NRZ	132.8 Mb/s	Fibre Channel	All
FC266	NRZ	265.6 Mb/s	Fibre Channel	All
FC531	NRZ	531.2 Mb/s	Fibre Channel	TDS5104B
FC1063	NRZ	1.062.5 Gb/s	Fibre Channel	
FC2125	NRZ	2.125 Gb/s	Fibre Channel	
FC4250	NRZ	4.256 Gb/s	Fibre Channel	

FST	NRZ	12.0 Mb/s	USB 1.1/2.0	All
FW1394 BS400B	NRZ	491.5 Mb/s	IEEE139 4b	TDS5104B
FW1394 BS1600 B	NRZ	1.966 Gb/s	IEEE139 4b	
HST	NRZ	480.0 Mb/s	USB 1.1/2.0	TDS5104B
INF2_5G	NRZ	2.5 Gb/s	Infiniband	
OC1	NRZ	51.84 Mb/s	SONET/ SDH	All
OC3	NRZ	155.52 Mb/s	SONET/ SDH	All
OC12	NRZ	622.08 Mb/s	SONET/ SDH	TDS5104B
OC48				
OC48 F EC	NRZ	2.666 Gb/s		
PCIEXP RESS	NRZ	2.5 Gb/s		

RATE32 Mbit	AMI	32.06 Mb/s	ITU-T G.703 (10/98)	
RATE97 Mbit	AMI	97.73 Mb/s	ITU-T G.703 (10/98)	
STM1E	CMI	155.52 Mb/s	ITU-T G.703 (10/98)	All
STS1	AMI B3ZS	51.84 Mb/s	ANSI T1.102	All
STS3	CMI	155.52 Mb/s	ANSI T1.	All
VIDEO 270	NRZ	270 Mb/s	Video	All
VIDEO 360	NRZ	360 Mb/s	Video	TDS5104B
Custom	AMI B3ZS B6ZS B8ZS CMI HDB3 MLT3 NRZ	1.544 Mb/s		All
RIO_500 M	NRZ	500 Mb/s		TDS5104B
RIO_750 M	NRZ	750 Mb/s		

RIO_1G	NRZ	1 Gb/s		
RIO_1_5 G	NRZ	1 Gb/s		
RIO_2G	NRZ	2 Gb/s		
RIO_SE RIAL_1G	NRZ	1.25 Gb/s		
RIO_SE RIAL_2G	NRZ	2.5 Gb/s		
RIO_SE RIAL_3G	NRZ	3.125 Gb/s		
SFI5_2	NRZ	2.488 Gb/s		
SFI5_3	NRZ	3.125 Gb/s		
VSROC 192	NRZ	1.24416 Gb/s		
TFI5_2	NRZ	2.488 Gb/s		
TFI5_3	NRZ	3.1104 Gb/s		

---

### **Example 1**

TRIGger:A:COMMUNICATION:SOURCE:STANDARD E4

This command sets the standard to E4.

### **Example 2**

TRIGger:A:COMMUNICATION:SOURCE:STANDARD?

This query might return

TRIGGER:A:COMMUNICATION:SOURCE:STANDARD DS1

---

# TRIGger:A:EDGE:COUPLing

## Description

This command sets or queries the type of coupling for the A edge trigger. This command is equivalent to selecting A Event (Main) Trigger Setup from the Trig menu, selecting Edge Trigger, and choosing from the Coupling drop-down list.

## Group

Trigger

## Related Commands

TRIGger:A:EDGE:SOUrce, TRIGger:A:EDGE:SLOPe

## Syntax 1

```
TRIGger:A:EDGE:COUPLing  
{AC|DC|HFRej|LFRej|NOISErej}
```

## Syntax 2

```
TRIGger:A:EDGE:COUPLing?
```

## Arguments

### ■ AC

This selects AC trigger coupling, which passes the input signals above 60 Hz to the trigger circuitry.

### ■ DC

This selects DC trigger coupling, which passes all input signals to the trigger circuitry.

### ■ HFRej

This coupling attenuates signals above 50 kHz before passing the signals to the trigger circuitry.

### ■ LFRej

This coupling attenuates signals below 80 kHz before passing the signals to the trigger circuitry.

### ■ NOISErej

This coupling provides stable triggering by increasing the trigger hysteresis. Increased hysteresis reduces the trigger sensitivity to noise but may require greater trigger signal amplitude.

## Example 1

```
TRIGger:A:EDGE:COUPLing DC
```

This command sets the A edge trigger coupling to DC.



---

## Example 2

TRIGger:A:EDGE:COUPling?

This query might return

:TRIGGER:A:EDGE:COUPLING DC

indicating that the A edge trigger coupling is set to DC.

---

# TRIGger:A:EDGE:SLOpe

## Description

This command sets or queries the slope for the A edge trigger. This command is equivalent to selecting A Event (Main) Trigger Setup from the Trig menu and then choosing the desired Slope.

## Group

Trigger

## Related Commands

TRIGger:A:EDGE:SOUrce, TRIGger:A:EDGE:COUPling

## Syntax 1

```
TRIGger:A:EDGE:SLOpe {RISe|FALL}
```

## Syntax 2

```
TRIGger:A:EDGE:SLOpe?
```

## Arguments

- RISE

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

- FALL

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

## Example 1

```
TRIGger:A:EDGE:SLOpe?
```

This query might return

```
:TRIGGER:A:EDGE:SLOPE FALL
```

indicating that the A edge trigger slope is negative.

## Example 2

```
TRIGger:A:EDGE:SLOpe RISE
```

This command sets the A edge trigger slope to positive, which triggers on the rising edge of the signal.

---

# TRIGger:A:EDGE:SOURce

## Description

This command sets or queries the source for the A edge trigger. This command is equivalent to selecting A Event (Main) Trigger Setup from the Trig menu and then choosing from the Source drop-down list.

## Group

Trigger

## Related Commands

TRIGger:A:EDGE:SLOpe, TRIGger:A:EDGE:COUPling

## Syntax 1

```
TRIGger:A:EDGE:SOURce {AUXiliary|CH<x>|LINE}
```

## Syntax 2

```
TRIGger:A:EDGE:SOURce?
```

## Arguments

- **AUXiliary**

This specifies an external trigger using the Auxiliary Trigger Input connector located on the rear panel of the instrument.

- **CH<x>**

This specifies one input channel as the A edge trigger source.

- **LINE**

This specifies AC line voltage.

## Example 1

```
TRIGger:A:EDGE:SOURce?
```

This query might return

```
:TRIGGER:A:EDGE:SOURCE CH1
```

indicating that channel 1 is the A edge trigger source.

## Example 2

```
TRIGger:A:EDGE:SOURce CH1
```

This command sets channel 1 as the A edge trigger source.

---

# TRIGger:A:EDGE?

## Description

This query-only command returns the trigger source, coupling, and slope for the A edge trigger. This command is equivalent to selecting Edge Setup from the Trig menu and viewing the current setups.

## Group

Trigger

## Related Commands

TRIGger:A:PULse, TRIGger:A:LOGic

## Syntax

TRIGger:A:EDGE?

## Example

TRIGger:A:EDGE?

This query might return

```
:TRIGGER:A:EDGE:SOURCE CH1;COUPLING DC;  
SLOPE RISE
```

indicating the trigger source, coupling, and slope for the A edge trigger.

---

# TRIGger:A:HOLDoff:ACTUal?

## Description

This query-only command returns the holdoff time actually used (expressed in seconds) by the A trigger. This command is equivalent to selecting Holdoff from the Trig menu and then viewing the current Trig Holdoff value.

## Group

Trigger

## Related Commands

TRIGger:A:HOLDoff?, TRIGger:A:HOLDoff:BY,  
TRIGger:A:HOLDoff:TIME

## Syntax

TRIGger:A:HOLDoff:ACTUal?

## Example

TRIGger:A:HOLDoff:ACTUal?

This query might return

```
:TRIGGER:A:HOLDOFF:ACTUAL 4.0000E-06
```

showing that the holdoff time is set to 4  $\mu$ s.

---

# TRIGger:A:HOLDoff:BY

## Description

This command sets or queries the type of holdoff for the A trigger. Holdoff types are expressed as either user-specified time (TIME) or by an internally calculated minimum time value (DEFAULT/AUTO). This command is equivalent to selecting Holdoff from the Trig menu and then setting the Holdoff type.

## Group

Trigger

## Related Commands

TRIGger:A:HOLDoff:TIME

## Syntax 1

TRIGger:A:HOLDoff:BY {TIME|DEFAULT}

## Syntax 2

TRIGger:A:HOLDoff:BY {TIME|RANDOM|AUTO}

## Syntax 3

TRIGger:A:HOLDoff:BY?

## Arguments

- TIME

This enables you to set the holdoff time via the TRIGger:A:HOLDoff:TIME command.

- DEFAULT

This automatically calculates a holdoff time to use. This time is typically equivalent to the greater of 1/2 screen (5 divisions) of time or 250 ns. The maximum value is 12 s. For example, if the instrument is set to 1 ms/division then the default holdoff will be 1 ms/division x 25 divisions = 25 ms.

## Example 1

TRIGger:A:HOLDoff:BY?

This query might return

:TRIGGER:A:HOLDOFF:BY TIME

indicating that you will set the holdoff time.

## Example 2

TRIGger:A:HOLDoff:BY TIME

This command sets the holdoff to the "by time" setting. This enables you to set the holdoff time.

---

# TRIGger:A:HOLDoff:TIME

## Description

This command sets or queries the A trigger holdoff time. This command is equivalent to selecting Holdoff from the Trig menu and then choosing the desired Trig Holdoff.

## Group

Trigger

## Related Commands

TRIGger:A:HOLDoff:BY

## Syntax 1

```
TRIGger:A:HOLDoff:TIME <NR3>
```

## Syntax 2

```
TRIGger:A:HOLDoff:TIME?
```

## Argument

- <NR3>

This specifies the holdoff time in seconds. The range is from 250 ns through 12.0 s.

## Example 1

```
TRIGger:A:HOLDoff:TIME?
```

This query might return

```
:TRIGGER:A:HOLDOFFTIME 1.2000E-06
```

indicating that the A trigger holdoff time is set to 1.2  $\mu$ s.

## Example 2

```
TRIGger:A:HOLDoff:TIME 10
```

This command sets the A trigger holdoff time to 10 s.

---

# TRIGger:A:HOLDoff?

## Description

This query-only command returns the A trigger holdoff parameters. These parameters specify the time period during which the trigger circuitry is not looking to generate a trigger event. This command is equivalent to selecting Holdoff from the Trig menu and then viewing the current settings.

## Group

Trigger

## Related Commands

TRIGger:A:HOLDoff:ACTUal?, TRIGger:A:HOLDoff:BY, TRIGger:A:HOLDoff:TIME

## Syntax

```
TRIGger:A:HOLDoff?
```

## Example

```
TRIGger:A:HOLDoff?
```

This query might return

```
:TRIGGER:A:HOLDOFF:TIME 900.0000E-09;  
BY DEFAULT
```

indicating that the A edge trigger holdoff time is set 900 ns.



---

# TRIGger:A:LEVel

## Description

This command sets or queries the level for the A trigger. This command is equivalent to selecting Holdoff from the Trig menu and then viewing or setting the trigger Level.

## Group

Trigger

## Syntax 1

```
TRIGger:A:LEVel {ECL|TTL|<NR3>}
```

## Syntax 2

```
TRIGger:A:LEVel?
```

## Arguments

- ECL

This specifies the ECL high level.

- TTL

This specifies the TTL high level.

- <NR3>

This specifies the trigger level in user units (usually volts).

## Example 1

```
TRIGger:A:LEVel?
```

This query might return

```
:TRIGGER:A:LEVel 1.3000E+00
```

indicating that the A edge trigger is set to 1.3 V.

## Example 2

```
TRIGger:A:LEVel TTL
```

This command sets the A edge trigger to TTL high level, which is 1.4 V.

---

# TRIGger:A:LOGic:CLAss

## Description

This command sets or queries the class of the Logic Trigger. Used in conjunction with the TRIGger:A:TYPe command, this command is equivalent to selecting Logic Pattern, Logic State or Setup/Hold Setup from the Trig menu.

## Group

Trigger

## Related Commands

TRIGger:A:TYPe, TRIGger:A:PULse:CLAss

## Syntax 1

```
TRIGger:A:LOGic:CLAss {PATtern|STATE|SETHold}
```

## Syntax 2

```
TRIGger:A:LOGic:CLAss?
```

## Arguments

### ■ PATtern

In four-channel instruments, this argument sets the instrument to trigger when the specified logical combinations of channels 1, 2, 3, and 4 are met. On two-channel instruments, only channel 1 and channel 2 are available.

### ■ STATE

In four-channel instruments, this argument sets the instrument to trigger when the specified conditions of channels 1, 2, and 3 are met after the channel 4 (clock) condition is met. On two-channel instruments, only channel 1 and channel 2 (clock) are available.

### ■ SETHold

This argument sets the instrument to trigger on setup and hold violations between a data source and a clock source. Use one channel input as the clock signal and a second channel input as the data input. The clocking and data levels are used to determine if a clock or data transition has occurred.

## Example 1

```
TRIGger:A:LOGic:CLAss?
```

This query might return

```
:TRIGGER:A:LOGIC:CLASS PATTERN
```

---

## Example 2

TRIGger:A:LOGic:CLAss PATTERN

This command sets the trigger A logic class to PATtern, which causes the instrument to trigger when the specified logical combinations of channels 1, 2, 3, and 4 are met.

---

# TRIGger:A:LOGIc:FUNcTION

## Description

This command sets or queries the logical combination of the input channels for the A pattern and A state logic triggers. This command is equivalent to selecting A Event (Main) Trigger Setup from the Trig menu, selecting Pattern or State for the Trigger Type, and setting or viewing the Pattern Type.

## Group

Trigger

## Related Commands

TRIGger:A:LOGIc:PATtern:WHEn, TRIGger:A:LOGIc:INPut:CH<x>, TRIGger:A:LOGIc:THReshold:CH<x>

## Syntax 1

TRIGger:A:LOGIc:FUNcTION {AND | NANd | NOR | OR}

## Syntax 2

TRIGger:A:LOGIc:FUNcTION?

## Arguments

- AND

This specifies to trigger if all conditions are true.

- NANd

This specifies to trigger if any of the conditions are false.

- NOR

This specifies to trigger if all conditions are false.

- OR

This specifies to trigger if any of the conditions are true.

## Example 1

TRIGger:A:LOGIc:FUNcTION?

This query might return

:TRIGGER:A:LOGIC:FUNCTION NAND

which indicates that the instrument will trigger if the AND logic conditions are false.

## Example 2

TRIGger:A:LOGIc:FUNcTION AND

This command sets the logical combination of channels to be true when all conditions are true.

---

# TRIGger:A:LOGic:INPut:CH<x>

## Description

This command sets or queries the A logical input for the logic trigger channel specified by x. The value of x ranges from 1 through 3 for four-channel instruments. For two-channel instruments x can only be 1. Note that CH4 on four-channel instruments or CH2 on two-channel instruments cannot be set or queried with this command. For details about setting this channel, see TRIGger:A:LOGic:PATtern:INPut:CH4 or TRIGger:A:LOGic:PATtern:INPut:CH2. This command is equivalent to selecting A Event (Main) Trigger Setup from the Trig menu and then choosing the desired logical input from the Ch<x> drop-down list, which is located in the Input Threshold group box.

## Group

Trigger

## Related Commands

TRIGger:A:LOGic:PATtern:INPut:CH4, TRIGger:A:LOGic:PATtern:INPut:CH2

## Syntax 1

```
TRIGger:A:LOGic:INPut:CH<x> {HIGH|LOW|X}
```

## Syntax 2

```
TRIGger:A:LOGic:INPut:CH<x>?
```

## Arguments

- HIGH

This specifies the logic high.

- LOW

This specifies the logic low.

- X

This specifies a "do not care" state.

## Example 1

```
TRIGger:A:LOGic:INPut:CH1?
```

This query might return

```
:TRIGGER:LOGIC:INPUT:CH1 X
```

indicating that the setting for the A logic trigger input to channel 1 does not matter.

---

## Example 2

```
TRIGger:A:LOGic:INPut:CH2 HIGH
```

This command sets the A logic trigger input to logic HIGH for channel 2.

---

# TRIGger:A:LOGic:INPut?

## Description

On four-channel instruments, this query-only command returns the A logic trigger input expected for channel 1, 2, and 3. Channel 4 is set or queried with the command TRIGger:A:LOGic:PATtern:INPut:CH4.

On two-channel instruments, this query returns the A logic trigger input expected for channel 1. Channel 2 is set or queried with the command TRIGger:A:LOGic:PATtern:INPut:CH2.

This command is equivalent to selecting A Event (Main) Trigger Setup and viewing or setting the Input Threshold for channel 1 through 3 for four-channel instruments or channel 1 for two-channel instruments.

## Group

Trigger

## Related Commands

TRIGger:A:LOGic:PATtern:INPut:CH4

## Syntax

TRIGger:A:LOGic:INPut?

## Example

TRIGger:A:LOGic:INPut?

This query might return

```
:TRIGGER:A:LOGIC:INPUT:CH1 HIGH;CH2 X;CH3 X
```

indicating that a logic high is expected on channel 1 while channel 2 and channel three are Don't Care.

---

# TRIGger:A:LOGic:PATtern:INPut:CH2

## Description

This command sets or queries the A logic trigger input for channel 2 in two-channel instruments only. The command specifies the logic value used when the pattern trigger detects the threshold level. Sending this command is equivalent to selecting Logic Pattern from the Trig menu and then choosing the desired logical input from the Ch2 drop-down list, which is located in the Input Threshold group box.

To set or query the A logic trigger input for a four-channel instrument, see TRIGger:A:LOGic:PATtern:INPut:CH4.

## Group

Trigger

## Related Commands

TRIGger:A:LOGic:FUNCTion, TRIGger:A:LOGic:INPut:CH<x>, TRIGger:A:LOGic:THReshold:CH<x>

## Syntax 1

TRIGger:A:LOGic:PATtern:INPut:CH2 {HIGH|LOW|X}

## Syntax 2

TRIGger:A:LOGic:PATtern:INPut:CH2?

## Arguments

- HIGH

This specifies the logic high.

- LOW

This specifies the logic low.

- X

This specifies a "do not care" state.

## Example 1

TRIGger:A:LOGic:PATtern:INPut:CH2?

This query might return

:TRIGGER:A:LOGIC:PATTERN:INPUT:CH2 HIGH

indicating that the logic input for channel 2 is logic high.

## Example 2

TRIGger:A:LOGic:PATtern:INPut:CH2 HIGH

This command sets the A logic trigger input to logic high for channel 2 when the logic class is set to PATtern. When the threshold level is detected, HIGH places a 1 on the channel 2 input to the selected function.



---

# TRIGger:A:LOGic:PATtern:INPut:CH4

## Description

This command sets or queries the A logic trigger input for channel 4 in four-channel instruments only. The command specifies the logic value used when the pattern trigger detects the threshold level. Sending this command is equivalent to selecting Logic Pattern from the Trig menu and then choosing the desired logical input from the Ch4 drop-down list, which is located in the Input Threshold group box.

To set or query the A logic trigger input for a two-channel instrument, see TRIGger:A:LOGic:PATtern:INPut:CH2.

## Group

Trigger

## Related Commands

TRIGger:A:LOGic:FUNCTion, TRIGger:A:LOGic:INPut:CH<x>, TRIGger:A:LOGic:THReshold:CH<x>

## Syntax 1

TRIGger:A:LOGic:PATtern:INPut:CH4 {HIGH|LOW|X}

## Syntax 2

TRIGger:A:LOGic:PATtern:INPut:CH4?

## Arguments

- HIGH

This specifies the logic high.

- LOW

This specifies the logic low.

- X

This specifies a "do not care" state.

## Example 1

TRIGger:A:LOGic:PATtern:INPut:CH4?

This query might return

:TRIGGER:A:LOGIC:PATTERN:INPUT:CH4 HIGH

indicating that the logic input for channel 4 is logic high.

## Example 2

TRIGger:A:LOGic:PATtern:INPut:CH4 HIGH

This command sets the A logic trigger input to logic high for channel 4 when the logic class is set to PATtern. When the threshold level is detected, HIGH places a 1 on the channel 4 input to the selected function.

---

# TRIGger:A:LOGic:PATtern:WHEn

## Description

This command sets or queries the condition for generating an A logic pattern trigger with respect to the defined input pattern. This command is equivalent to selecting A Event (Main) Trigger Setup from the Trig menu, selecting Pattern for Trigger Type, and choosing a trigger condition from the Pattern drop-down list, which is located in the Trigger When group box.

## Group

Trigger

## Related Commands

TRIGger:A:LOGic:FUNcTION, TRIGger:A:LOGic:INPut, TRIGger:A:LOGic:THReshold, TRIGger:A:LOGic:PATtern:WHEn:LESSLimit, TRIGger:A:LOGic:PATtern:WHEn:MORELimit

## Syntax 1

```
TRIGger:A:LOGic:PATtern:WHEn {TRUE|FALSE|  
LESSThan|MOREThan}
```

## Syntax 2

```
TRIGger:A:LOGic:PATtern:WHEn?
```

## Arguments

### ■ TRUE

This argument sets the instrument to trigger when the pattern becomes true.

### ■ FALSE

This argument sets the instrument to trigger when the pattern becomes false.

### ■ LESSThan

This argument sets the instrument to trigger if the specific pattern is true less than the time set by the TRIGger:A:LOGic:PATtern:WHEn:LESSLimit command.

### ■ MOREThan

This argument sets the instrument to trigger if the specific pattern is true longer than the specified time set by the TRIGger:A:LOGic:PATtern:WHEn:MORELimit command.

---

### **Example 1**

TRIGger:A:LOGIC:PATtern:WHEn?

This query might return

:TRIGGER:A:LOGIC:PATTERN:WHEN TRUE

indicating that the A logic pattern will trigger when the pattern becomes true.

### **Example 2**

TRIGger:A:LOGIC:PATtern:WHEn FALSE

This command specifies to trigger the A logic pattern when the pattern becomes false.

---

# TRIGger:A:LOGic:PATtern:WHEn:LESSLimit

## Description

This command sets or queries the maximum time that the selected pattern may be true and still generate an A logic pattern trigger. This command is equivalent to selecting A Event (Main) Trigger Setup from the Trig menu, selecting Pattern as the Trigger Type, selecting Less Than for the Pattern in the Trigger When settings, and entering a maximum value for Time.

## Group

Trigger

## Related Commands

TRIGger:A:LOGic:PATtern:WHEn:MORELimit

## Syntax 1

```
TRIGger:A:LOGic:PATtern:WHEn:LESSLimit <NR3>
```

## Syntax 2

```
TRIGger:A:LOGic:PATtern:WHEn:LESSLimit?
```

## Argument

- <NR3>

This specifies the amount of time to hold the pattern true.

## Example 1

```
TRIGger:A:LOGic:PATtern:WHEn:LESSLimit?
```

This query might return

```
:TRIGGER:A:LOGIC:PATTERN:WHEN:LESSLIMIT  
5.0000E-09
```

indicating that the selected pattern may hold true for up to 5 ns and still generate an A logic pattern trigger.

## Example 2

```
TRIGger:A:LOGic:PATtern:WHEn:LESSLimit  
10.0E+00
```

This command sets the maximum time that the selected pattern may hold true (and generate an A logic pattern trigger) to 10 s.

---

# TRIGger:A:LOGic:PATtern:WHEn:MORELimit

## Description

This command sets or queries the minimum time that the selected pattern may be true and still generate an A logic pattern trigger. This command is equivalent to selecting A Event (Main) Trigger Setup from the Trig menu, selecting Pattern as the Trigger Type, selecting More Than for the Pattern in the Trigger When settings, and entering a minimum value for Time.

## Group

Trigger

## Related Commands

TRIGger:A:LOGic:PATtern:WHEn:LESSLimit

## Syntax 1

```
TRIGger:A:LOGic:PATtern:WHEn:MORELimit <NR3>
```

## Syntax 2

```
TRIGger:A:LOGic:PATtern:WHEn:MORELimit?
```

## Argument

- <NR3>

This specifies the amount of time to hold the pattern true.

## Example 1

```
TRIGger:A:LOGic:PATtern:WHEn:MORELimit?
```

This query might return

```
:TRIGGER:A:LOGIC:PATTERN:WHEN:MORELIMIT  
5.0000E-09
```

indicating that the selected pattern must hold true for at least 5 ns to generate an A logic pattern trigger.

## Example 2

```
TRIGger:A:LOGic:PATtern:WHEn:MORELimit  
10.0E+00
```

This command sets the minimum time that the selected pattern may hold true (and generate an A logic pattern trigger) to 10 s.

---

# TRIGger:A:LOGic:PATtern?

## Description

This query-only command returns the conditions used for generating an A logic pattern trigger, with respect to the defined input pattern, and identifies the maximum and minimum time that the selected pattern may be true and still generate the trigger. This command is equivalent to selecting Logic Pattern from the Trig menu and then viewing the current setups.

## Group

Trigger

## Related Commands

TRIGger:A:LOGic:PATtern:INPut:CH4,  
TRIGger:A:LOGic:PATtern:WHEn,  
TRIGger:A:LOGic:PATtern:WHEn:LESSLimit,  
TRIGger:A:LOGic:PATtern:WHEn:MORELimit

## Syntax

TRIGger:A:LOGic:PATtern?

## Example

TRIGger:A:LOGic:PATtern?

This query might return

```
:TRIGGER:A:LOGIC:PATTERN:INPUT:CH4 X;  
:TRIGGER:A:LOGIC:PATTERN:WHEN TRUE;  
WHEN:LESSLIMIT 5.0000E-9;MORELIMIT 5.0000E-9.
```

---

# TRIGger:A:LOGic:SETHold:CLOCK:EDGE

## Description

This command sets or queries the clock edge polarity for setup and hold triggering. This is equivalent to selecting Setup/Hold Setup from the Trig menu and then choosing the desired Clock Edge.

## Group

Trigger

## Related Commands

TRIGger:A:LOGic:SETHold:CLOCK:SOURce, TRIGger:A:LOGic:SETHold:CLOCK:THReshold

## Syntax 1

```
TRIGger:A:LOGic:SETHold:CLOCK:EDGE {FALL|RISe}
```

## Syntax 2

```
TRIGger:A:LOGic:SETHold:CLOCK:EDGE?
```

## Arguments

- FALL

This specifies polarity as the clock falling edge.

- RISe

This specifies polarity as the clock rising edge.

## Example 1

```
TRIGger:A:LOGic:SETHold:CLOCK:EDGE?
```

This query might return

```
:TRIGGER:A:LOGIC:SETHOLD:CLOCK:EDGE RISe
```

indicating that polarity is specified as the clock rising edge.

## Example 2

```
TRIGger:A:LOGic:SETHold:CLOCK:EDGE RISe
```

This command specifies the polarity as the clock rising edge.

---

# TRIGger:A:LOGic:SETHold:CLOCK:SOURce

## Description

This command sets or queries the clock source for the A logic trigger setup and hold input. This is equivalent to selecting Setup/Hold Setup from the Trig menu and choosing the desired channel from the Clock Source drop-down list.

## Group

Trigger

## Related Commands

TRIGger:A:LOGic:SETHold:CLOCK:EDGE,  
TRIGger:A:LOGic:SETHold:CLOCK:THReshold

## Syntax 1

```
TRIGger:A:LOGic:SETHold:CLOCK:SOURce CH<x>
```

## Syntax 2

```
TRIGger:A:LOGic:SETHold:CLOCK:SOURce?
```

## Argument

- CH<x>

This specifies the input channel, which ranges from 1 through 4 for four-channel instruments or 1 through 2 for two channel instruments.

## Example 1

```
TRIGger:A:LOGic:SETHold:CLOCK:SOURce?
```

This query might return

```
:TRIGGER:A:LOGIC:SETHOLD:CLOCK:SOURCE CH4
```

indicating that channel 4 is the clock source for the setup and hold input. For the A logic trigger.

## Example 2

```
TRIGger:A:LOGic:SETHold:CLOCK:SOURce CH1
```

This command specifies channel 1 as the A logic setup and hold input.



---

# TRIGger:A:LOGic:SETHold:CLOCK:THReshold

## Description

This command sets or queries the clock voltage threshold for the setup and hold trigger. This command is equivalent to selecting Setup/Hold Setup from the Trig menu and setting the desired Clock Level.

## Group

Trigger

## Related Commands

TRIGger:A:LOGic:SETHold:CLOCK:EDGE, TRIGger:A:LOGic:SETHold:CLOCK:SOUrce

## Syntax 1

```
TRIGger:A:LOGic:SETHold:CLOCK:THReshold  
{ECL | TTL | <NR3>}
```

## Syntax 2

```
TRIGger:A:LOGic:SETHold:CLOCK:THReshold?
```

## Arguments

- ECL

This specifies a preset ECL high level of -1.3 V.

- TTL

This specifies a preset TTL high level of 1.4 V.

- <NR3>

This is the clock level, in volts.

## Example 1

```
TRIGger:A:LOGic:SETHold:CLOCK:THReshold?
```

This query might return

```
:TRIGGER:A:LOGIC:SETHOLD:CLOCK:THRESHOLD  
1.2000E+00
```

indicating that the clock threshold for the setup and hold trigger is 1.2 V.

## Example 2

```
TRIGger:A:LOGic:SETHold:CLOCK:THReshold TTL
```

This command specifies the preset TTL value of 1.4 V as the clock threshold for the setup and hold trigger.

---

# TRIGger:A:LOGic:SETHold:CLOCK?

## Description

This query-only command returns the clock edge polarity, voltage threshold and source input for setup and hold triggering. This command is equivalent to selecting Setup/Hold Setup from the Trig menu and then viewing the current clock setups.

## Group

Trigger

## Related Commands

TRIGger:A:LOGic:SETHold:DATA?,  
TRIGger:A:LOGic:SETHold:CLOCK:EDGE,  
TRIGger:A:LOGic:SETHold:CLOCK:THReshold,  
TRIGger:A:LOGic:SETHold:CLOCK:SOUrce

## Syntax

TRIGger:A:LOGic:SETHold:CLOCK?

## Example

TRIGger:A:LOGic:SETHold:CLOCK?

This query might return

```
:TRIGGER:A:LOGic:SETHold:CLOCK:EDGE RISE;  
THRESHOLD 1.4000;SOURCE CH2
```

indicating the current clock settings for setup and hold triggering.

---

# TRIGger:A:LOGic:SETHold:DATA:SOURce

## Description

This command sets or queries the data source for the setup and hold trigger. This command is equivalent to selecting Setup/Hold Setup from the Trig menu and choosing the desired channel from the Data Source drop-down list.

## Group

Trigger

## Related Commands

TRIGger:A:LOGic:SETHold:DATA:THReshold

## Syntax 1

```
TRIGger:A:LOGic:SETHold:DATA:SOURce CH<x>
```

## Syntax 2

```
TRIGger:A:LOGic:SETHold:DATA:SOURce?
```

## Argument

- CH<x>

This specifies the input channel, which ranges from 1 through 4 for four-channel instruments or 1 through 2 for two-channel instruments..

## Example 1

```
TRIGger:A:LOGic:SETHold:DATA:SOURce?
```

This query might return

```
:TRIGGER:A:LOGIC:SETHOLD:DATA:SOURCE CH2
```

indicating that channel 2 is the current clock source for the setup and hold trigger.

## Example 2

```
TRIGger:A:LOGic:SETHold:DATA:SOURce CH1
```

This command sets channel 1 as the clock source for the setup and hold trigger.

---

# TRIGger:A:LOGic:SETHold:DATA:THReshold

## Description

This command sets or queries the data voltage threshold for setup and hold trigger. This command is equivalent to selecting Setup/Hold Setup from the Trig menu and then setting the desired Data Level.

## Group

Trigger

## Related Commands

TRIGger:A:LOGic:SETHold:DATA:SOUrce

## Syntax 1

```
TRIGger:A:LOGic:SETHold:DATA:THReshold  
{ECL | TTL<NR3>}
```

## Syntax 2

```
TRIGger:A:LOGic:SETHold:DATA:THReshold?
```

## Arguments

- ECL

This specifies the preset ECL high level of -1.3 V.

- TTL

This specifies the preset TTL high level of 1.4 V.

- <NR3>

This is the setup and hold data level, in V.

## Example 1

```
TRIGger:A:LOGic:SETHold:DATA:THReshold?
```

This query might return

```
:TRIGGER:A:LOGIC:SETHOLD:DATA:THRESHOLD  
1.2000E+00
```

indicating that 1.2 V is the current data voltage level for the setup and hold trigger.

## Example 2

```
TRIGger:A:LOGic:SETHold:DATA:THReshold TTL
```

This command specifies the preset ECL high level of 1.4 V as the current data voltage level for the setup and hold trigger.

---

# TRIGger:A:LOGic:SETHold:DATA?

## Description

This query-only command returns the voltage threshold and data source for the setup and hold trigger. This command is equivalent to selecting Setup/Hold Setup from the Trig menu and then viewing the current data setups.

## Group

Trigger

## Related Commands

TRIGger:A:LOGic:SETHold:CLOCK?

## Syntax

TRIGger:A:LOGic:SETHold:DATA?

## Example

TRIGger:A:LOGic:SETHold:DATA?

This query might return

```
:TRIGGER:A:LOGIC:SETHOLD:DATA:THRESHOLD  
1.4000;SOURCE CH1
```

indicating the current trigger data settings.

---

# TRIGger:A:LOGic:SETHold:HOLDTime

## Description

This command sets or queries the hold time for setup and hold violation triggering. This command is equivalent to selecting Setup/Hold Setup from the Trig menu and then setting the desired Hold Time.

## Group

Trigger

## Related Commands

TRIGger:A:LOGic:SETHold:SETTime

## Syntax 1

```
TRIGger:A:LOGic:SETHold:HOLDTime <NR3>
```

## Syntax 2

```
TRIGger:A:LOGic:SETHold:HOLDTime?
```

## Argument

- <NR3>

This specifies the hold time setting in seconds. Positive values for hold time occur after the clock edge. Negative values occur before the clock edge.

## Example 1

```
TRIGger:A:LOGic:SETHold:HOLDTime?
```

This query might return

```
:TRIGGER:A:LOGIC:SETHOLD:HOLDTIME 2.0000E-09
```

indicating that the current hold time for the setup and hold trigger is 2 ns.

## Example 2

```
TRIGger:A:LOGic:SETHold:HOLDTime 3.0E-3
```

This command sets the hold time for the setup and hold trigger to 3 ms.

---

# TRIGger:A:LOGic:SETHold:SETTime

## Description

This command sets or queries the setup time for setup and hold violation triggering. This command is equivalent to selecting Setup/Hold Setup from the Trig menu and then setting the desired Setup Time.

## Group

Trigger

## Related Commands

TRIGger:A:LOGic:SETHold:HOLDTime

## Syntax 1

```
TRIGger:A:LOGic:SETHold:SETTime <NR3>
```

## Syntax 2

```
TRIGger:A:LOGic:SETHold:SETTime?
```

## Argument

- <NR3>

This specifies the setup time for setup and hold violation triggering.

## Example 1

```
TRIGger:A:LOGic:SETHold:SETTime?
```

This query might return

```
:TRIGGER:A:LOGIC:SETHOLD:SETTIME 2.0000E-09
```

indicating that the current setup time for setup and hold trigger is 2 ns.

## Example 2

```
TRIGger:A:LOGic:SETHold:SETTime 3.0E-6
```

This command specifies that the current setup time for setup and hold trigger is 3  $\mu$ s.

---

# TRIGger:A:LOGic:SETHold?

## Description

This query-only command returns the clock edge polarity, voltage threshold and source input; data voltage threshold and source; and both setup and hold times for setup and hold violation triggering. This command is equivalent to selecting Setup/Hold Setup from the Trig menu and then viewing the current setups.

## Group

Trigger

## Related Commands

TRIGger:A:LOGic:PATtern?, TRIGger:A:LOGic:STATE?

## Syntax

TRIGger:A:LOGic:SETHold?

## Example

TRIGger:A:LOGic:SETHold?

This query might return the settings

```
:TRIGGER:A:LOGIC:SETHOLD:CLOCK:EDGE RISE;  
THRESHOLD 1.4000;SOURCE CH2;  
:TRIGGER:A:LOGIC:SETHOLD:DATA:THRESHOLD  
1.4000;SOURCE CH1;:TRIGGER:A:LOGIC:SETHOLD  
:HOLDTIME 2.0000E-9;SETTIME 3.0000E-9
```



---

# TRIGger:A:LOGic:STATE:INPut:CH2

## Description

This command sets or queries the slope for channel 2 when the logic class is set to STATE in two-channel instruments. This command is equivalent to selecting Logic State from the Trig menu and then choosing the desired channel input (NEG or POS) from the Ch2 drop-down list.

To set or query the slope of the logic state in four-channel instruments, see TRIGger:A:LOGic:STATE:INPut:CH4.

## Group

Trigger

## Related Commands

TRIGger:A:LOGic:STATE:WHEn

## Syntax 1

TRIGger:A:LOGic:STATE:INPut:CH2 {FALL|RISe}

## Syntax 2

TRIGger:A:LOGic:STATE:INPut:CH2?

## Arguments

- FALL

This specifies falling edge.

- RISe

This specifies rising edge.

## Example 1

TRIGger:A:LOGic:STATE:INPut:CH2?

This query might return

:TRIGGER:A:LOGIC:STATE:INPUT:CH2 RISe

indicating that the A logic trigger input for channel 2 is the rising edge.

## Example 2

TRIGger:A:LOGic:STATE:INPut:CH2 RISe

This command specifies that the A logic trigger input for channel 2 is the rising edge.

---

# TRIGger:A:LOGic:STATE:INPut:CH4

## Description

This command sets or queries the slope for channel 4 when the logic class is set to State in four-channel instruments. This command is equivalent to selecting Logic State from the Trig menu and then choosing the desired channel input (NEG or POS) from the Ch4 (Clk) drop-down list.

To set or query the slope of the logic state in two-channel instruments, see TRIGger:A:LOGic:STATE:INPut:CH2.

## Group

Trigger

## Related Commands

TRIGger:A:LOGic:STATE:WHEn

## Syntax 1

```
TRIGger:A:LOGic:STATE:INPut:CH4 {FALL|RISe}
```

## Syntax 2

```
TRIGger:A:LOGic:STATE:INPut:CH4?
```

## Arguments

- FALL

This specifies falling edge.

- RISe

This specifies rising edge.

## Example 1

```
TRIGger:A:LOGic:STATE:INPut:CH4?
```

This query might return

```
:TRIGGER:A:LOGIC:STATE:INPUT:CH4 RISE
```

indicating that the A logic trigger input for channel 4 is the rising edge.

## Example 2

```
TRIGger:A:LOGic:STATE:INPut:CH4 RISE
```

This command specifies that the A logic trigger input for channel 4 is the rising edge.

---

# TRIGger:A:LOGic:STATE:WHEn

## Description

This command sets or queries the condition for generating an A logic state trigger. This command is equivalent to selecting Logic State from the Trig menu and choosing the desired condition from the Trigger When Pattern drop-down list.

## Group

Trigger

## Related Commands

TRIGger:A:LOGic:INPut:CH<x>,  
TRIGger:A:LOGic:STATE:INPut:CH4

## Syntax 1

```
TRIGger:A:LOGic:STATE:WHEn {TRUE|FALSE}
```

## Syntax 2

```
TRIGger:A:LOGic:STATE:WHEn?
```

## Arguments

### ■ TRUE

On four-channel instruments, this argument specifies that the trigger occurs when the clock transition on channel 4 occurs and the pattern of channels 1-3 are at the desired logic input states. On two-channel instruments, this specifies that the trigger occurs when the clock transition on channel 2 occurs and the pattern of channel 1 is at the desired logic input state.

### ■ FALSE

On four-channel instruments, this argument specifies that the trigger occurs when the desired clock transition on channel 4 occurs and the desired logic input states on channels 1-3 are not found. On two-channel instruments, this specifies that the trigger occurs when the clock transition on channel 2 occurs and the desired logic input state on channel 1 is not found.

## Example 1

```
TRIGger:A:LOGic:STATE:WHEn?
```

This query might return

```
:TRIGGER:A:LOGIC:STATE:WHEN FALSE
```

indicating that the logic condition to trigger upon is false.

## Example 2

```
TRIGger:A:LOGic:STATE:WHEn TRUE
```

This command specifies that the logic condition to trigger upon is true.

---

# TRIGger:A:LOGic:STATE?

## Description

This query-only command returns the data input and trigger criteria for the A logic trigger. This command is equivalent to selecting Logic State from the Trig menu and then viewing the current logic state settings.

## Group

Trigger

## Related Commands

TRIGger:A:LOGic:STATE:INPut:CH4,  
TRIGger:A:LOGic:STATE:WHEn

## Syntax

TRIGger:A:LOGic:STATE?

## Example

TRIGger:A:LOGic:STATE?

This query might return

```
:TRIGGER:A:LOGIC:STATE:INPUT:CH4 RISE;  
:TRIGGER:A:LOGIC:STATE:WHEN TRUE
```

---

# TRIGger:A:LOGic:THReshold:CH<x>

## Description

This command sets or queries the A logic trigger threshold voltage for the channel, specified by x, which ranges from 1 through 4 for four-channel instruments or 1 through 2 for two-channel instruments. This command is equivalent to selecting A Event (Main) Trigger Setup from the Trig menu, choosing a logic trigger type, such as State or Pattern, and setting the Input Threshold voltage for the desired channel.

## Group

Trigger

## Related Commands

TRIGger:A:LOGic:INPut?

## Syntax 1

TRIGger:A:LOGic:THReshold:CH<x> <NR3>

## Syntax 2

TRIGger:A:LOGic:THReshold:CH<x>?

## Argument

- <NR3>

This specifies the threshold voltage.

## Example 1

TRIGger:A:LOGic:THReshold:CH3?

This query might return

```
:TRIGGER:A:LOGIC:THRESHOLD:CH3 1.2000E+00
```

indicating that the A logic trigger threshold voltage for channel 3 is 1.2 V.

## Example 2

TRIGger:A:LOGic:THReshold:CH2 3.0E-3

This command sets the A logic trigger threshold voltage for channel 2 to 3 mV.

---

# TRIGger:A:LOGic:THReshold?

## Description

This query-only command returns the threshold voltage for all channels in an A logic trigger. This command query is equivalent to selecting A Event (Main) Trigger Setup from the Trig menu, choosing a logic trigger type, such as State or Pattern, and viewing the current Input Threshold voltage settings.

## Group

Trigger

## Related Commands

TRIGger:A:LOGic:THReshold:CH<x>, TRIGger:A:LOGic:INPut, TRIGger:A:LOGic:INPut:CH<x>

## Syntax

TRIGger:A:LOGic:THReshold?

## Example

TRIGger:A:LOGic:THReshold?

This query might return

```
:TRIGGER:A:LOGIC:THRESHOLD CH1 24.0000E-03;  
CH2 1.2000E+00;CH3 1.2000E+00;  
CH4 1.2000E+00
```

indicating the threshold voltages for the channels in an A logic trigger are as follows: channel 1 = 24 mV; channel 2 = 1.2 V; channel 3 = 1.2 V; channel 4 = 1.2 V.

---

# TRIGger:A:LOGic?

## Description

This query-only command returns all of the A logic trigger parameters.

## Group

Trigger

## Related Commands

TRIGger:A:LOGic:CLAss

## Syntax

TRIGger:A:LOGic?

## Example

TRIGger:A:LOGic?

This query might return

```
:TRIGGER:A:LOGIC:CLASS PATTERN;  
FUNCTION AND;WHEN TRUE;  
THRESHOLD:CH1 1.4000;CH2 1.4000;CH3 1.4000;  
CH4 1.4000;:TRIGGER:A:LOGIC:INPUT:CH1 HIGH;  
CH2 X;CH3 X;  
:TRIGGER:A:LOGIC:PATTERN:INPUT:CH4 X;  
:TRIGGER:A:LOGIC:PATTERN:WHEN TRUE;  
WHEN:LESSLIMIT 5.0000E-9;  
MORELIMIT 5.0000E-9;  
:TRIGGER:A:LOGIC:SETHOLD:CLOCK:EDGE RISE;  
THRESHOLD 1.4000;SOURCE CH2;  
:TRIGGER:A:LOGIC:SETHOLD:DATA:THRESHOLD  
1.4000;SOURCE CH1;  
:TRIGGER:A:LOGIC:SETHOLD:HOLDTIME2.0000E-9;  
SETTIME 3.0000E-9;  
:TRIGGER:A:LOGIC:STATE:INPUT:CH4 RISE;  
:TRIGGER:A:LOGIC:STATE:WHEN TRUE
```

---

# TRIGger:A:MODE

## Description

This command sets or queries the A trigger mode. This command is equivalent to selecting Mode from the Trig menu and then choosing the desired Trigger Mode.

## Group

Trigger

## Related Commands

TRIGger:A:LEVel

## Syntax 1

```
TRIGger:A:MODE {AUTO|NORMAl}
```

## Syntax 2

```
TRIGger:A:MODE?
```

## Arguments

- AUTO

This generates a trigger if one is not detected within a specified time period.

- NORMAl

This waits for a valid trigger event.

## Example 1

```
TRIGger:A:MODE?
```

This query might return

```
:TRIGGER:A:MODE NORMAL
```

indicating that a valid trigger event must occur before a trigger is generated.

## Example 2

```
TRIGger:A:MODE NORMAL
```

This command specifies that a valid trigger event must occur before a trigger is generated.



---

# TRIGger:A:PULse:CLAss

## Description

This command sets or queries the type of pulse on which to trigger. This command is equivalent to selecting the setup menu for the pulse type that you want from the Trig menu: Glitch Setup, Width Setup, Runt Setup, Timeout Setup, or Transition Setup.

## Group

Trigger

## Related Commands

TRIGger:A:PULse:GLItch, TRIGger:A:PULse:RUNT, TRIGger:A:PULse:WIDth, TRIGger:A:PULse:TIMEOut, TRIGger:A:PULse:TRANSition?, TRIGger:A:TYPE

## Syntax 1

```
TRIGger:A:PULse:CLAss {GLItch|RUNT|WIDth|  
TRANSition|TIMEOut|WINDow}
```

## Syntax 2

```
TRIGger:A:PULse:CLAss?
```

## Arguments

### ■ GLItch

This triggers when a pulse is found that is of the specified polarity and width. These are set with the commands TRIGger:A:PULse:GLItch:POLarity and TRIGger:A:PULse:GLItch:WIDth.

### ■ RUNT

This triggers when a pulse crosses the first preset voltage threshold but does not cross the second preset threshold before re-crossing the first. The thresholds are set with the TRIGger:A:PULse:RUNT:THReshold:LOW and TRIGger:A:PULse:RUNT:THReshold:HIGH commands.

### ■ WIDth

This triggers when a pulse is found that has the specified polarity and is either inside or outside the limits as specified by TRIGger:A:PULse:WIDth:LOWLimit and TRIGger:A:PULse:WIDth:HIGHLimit. The polarity is selected using the TRIGger:A:PULse:WIDth:POLarity command.

### ■ TRANSition

This triggers when a pulse crosses both thresholds in the same direction as the specified polarity and the transition time between the two threshold crossings is greater or less than the specified time delta.

### ■ TIMEOut

This triggers when the pulse train stops in the selected state for longer than the specified time.

---

## ■ WINDow

This triggers when a pulse is found that meets the conditions set by the A Event window trigger type, specified by the following commands:

TRIGger:A:PULse:WINDow:THReshold:HIGH

TRIGger:A:PULse:WINDow:THReshold:LOW

TRIGger:A:PULse:WINDow:THReshold:BOTH

TRIGger:A:PULse:WINDow:TYPe

TRIGger:A:PULse:WINDow:WHEn

TRIGger:A:PULse:WINDow:WIDth

### Example 1

TRIGger:A:PULse:CLAss?

This query might return

:TRIGGER:A:PULSE:CLASS GLITCH

indicating that a pulse was found that is of the specified polarity and width.

### Example 2

TRIGger:A:PULse:CLAss WIDth

This command specifies a width pulse for the A trigger.

---

# TRIGger:A:PULse:GLItch:POLarity

## Description

This command sets or queries the polarity for A pulse glitch trigger. This command is equivalent to selecting Glitch Setup from the Trig menu and then choosing the desired Polarity.

## Group

Trigger

## Related Commands

TRIGger:A:LEVel, TRIGger:A:PULse:GLItch:WIDth

## Syntax 1

```
TRIGger:A:PULse:GLItch:POLarity {POSITIVE|NEG-  
ative|EITHer}
```

## Syntax 2

```
TRIGger:A:PULse:GLItch:POLarity?
```

## Arguments

- POSITIVE

This specifies that the instrument will only trigger when the polarity of the glitch is positive.

- NEGative

This specifies that the instrument will only trigger when the polarity of the glitch is negative.

- EITHer

This specifies that the instrument will trigger when the polarity of the glitch is either positive or negative.

## Example 1

```
TRIGger:A:PULse:GLItch:POLarity?
```

This query might return

```
:TRIGGER:A:PULSE:GLITCH:POLARITY POSITIVE
```

indicating that the polarity of the glitch must be positive for the trigger to occur.

## Example 2

```
TRIGger:A:PULse:GLItch:POLarity EITHer
```

This command specifies that the polarity of the glitch can be either positive or negative for the trigger to occur.

---

# TRIGger:A:PULse:GLItch:TRIGIF

## Description

This command sets or queries the acceptance/rejection of the glitch pulse trigger, based on width. This command is equivalent to selecting Glitch Setup from the Trig menu and choosing the desired Trig if Width setting.

## Group

Trigger

## Related Commands

TRIGger:A:PULse:GLItch:WIDth

## Syntax 1

```
TRIGger:A:PULse:GLItch:TRIGIF {ACcept|REJect}
```

## Syntax 2

```
TRIGger:A:PULse:GLItch:TRIGIF?
```

## Arguments

- ACcept

This specifies that the instrument will only trigger on pulses that are narrower than the specified width, when the trigger type is set to glitch. The width is specified using the TRIGger:A:PULse:GLItch:WIDth command.

- REJect

This specifies that the instrument will only trigger on pulses that are wider than the specified width, when the trigger type is set to glitch. The width is specified using the TRIGger:A:PULse:GLItch:WIDth command.

## Example 1

```
TRIGger:A:PULse:GLItch:TRIGIF?
```

This query might return

```
:TRIGGER:A:PULSE:GLITCH:TRIGIF ACCEPT
```

indicating that the instrument is set to trigger on pulses that are narrower than the specified width.

## Example 2

```
TRIGger:A:PULse:GLItch:TRIGIF REJect
```

This command specifies that the instrument triggers on pulses that are wider than the specified width.

---

# TRIGger:A:PULse:GLItch:WIDth

## Description

This command sets or queries the width for the A glitch trigger. This command is equivalent to selecting Glitch Setup from the Trig menu and then setting the desired Width.

For information about using the width value, refer to the TRIGger:A:PULse:GLItch:TRIGIF command.

## Group

Trigger

## Related Commands

TRIGger:A:LEVel, TRIGger:A:PULse:GLItch:TRIGIF

## Syntax 1

```
TRIGger:A:PULse:GLItch:WIDth <NR3>
```

## Syntax 2

```
TRIGger:A:PULse:GLItch:WIDth?
```

## Argument

- <NR3>

This argument specifies the width of the glitch in seconds.

## Example 1

```
TRIGger:A:PULse:GLItch:WIDth?
```

This query might return

```
:TRIGGER:A:PULSE:GLITCH:WIDTH 2.0000E-09
```

indicating that the width of the glitch is currently set at 2 ns.

## Example 2

```
TRIGger:A:PULse:GLItch:WIDth 15E-6
```

This command sets the width of the glitch to 15  $\mu$ s.

---

# TRIGger:A:PULse:GLItch?

## Description

This query-only command returns the current A glitch pulse trigger parameters. This command query is equivalent to selecting Glitch Setup from the Trig menu and viewing the current glitch trigger settings.

## Group

Trigger

## Related Commands

TRIGger:A:PULse:GLItch:POLarity,  
TRIGger:A:PULse:GLItch:TRIGIF,  
TRIGger:A:PULse:GLItch:WIDth

## Syntax

TRIGger:A:PULse:GLItch?

## Example

TRIGger:A:PULse:GLItch?

This query might return

```
:TRIGGER:A:PULSE:GLITCH:WIDTH 2.0000E-09;  
FILTER ACCEPT;POLARITY POSITIVE
```

---

# TRIGger:A:PULse:RUNT:LOGIc:INPUT:CH<x>

## Description

This command sets or queries the A runt logic input for the channel specified by <x>, which can be 1, 2, 3, or 4. This is equivalent to setting the runt logic inputs from the Trigger When section in the Runt Trigger menu. While all channels can be set or queried, only channels 3 and 4 can be used if the runt source is channel 1 or 2. Similarly, only channels 1 and 2 can be used if the runt source is channel 3 or 4.

**Note:** This command is not available in two-channel instruments.

## Group

Trigger

## Syntax 1

```
TRIGger:A:PULse:RUNT:LOGIc:INPUT:CH<x>  
{HIGH|LOW|X}
```

## Syntax 2

```
TRIGger:A:PULse:RUNT:LOGIc:INPUT:CH<x>?
```

## Arguments

- HIGH

This argument specifies logic high.

- LOW

This argument specifies logic low.

- X

This argument specifies a don't care state.

## Example 1

```
TRIGger:A:PULse:RUNT:LOGIc:INPUT:CH2?
```

This query might return

```
:TRIGGER:A:PULSE:RUNT:LOGIC:INPUT CH2 LOW
```

indicating that the runt logic trigger input is set to logic low for channel 2.

## Example 2

```
TRIGger:A:PULse:RUNT:LOGIc:INPUT:CH3 HIGH
```

This command sets the runt logic trigger input to logic high for channel 3.

---

# TRIGGER:A:PULSE:RUNT:LOGIC:INPUT?

## Description

This query-only command returns the current A runt pulse trigger logic input parameters. This query is equivalent to selecting Runt Setup from the Trig menu and then viewing the current settings.

**Note:** This command is not available in two-channel instruments.

## Group

Trigger

## Syntax

```
TRIGGER:A:PULSE:RUNT:LOGIC:INPUT?
```

## Example

```
TRIGGER:A:PULSE:RUNT:LOGIC:INPUT?
```

This query might return these parameters:

```
:TRIGGER:A:PULSE:RUNT:LOGIC:INPUT:CH1 HIGH;  
CH2 X; CH3 X; CH4 LOW.
```



---

# TRIGger:A:PULse:RUNT:LOGIc:THReshold:CH<x>

## Description

This command sets or queries the A runt logic threshold for the channel specified by <x>, which can be 1, 2, 3, or 4. This is equivalent to selecting Runt Setup from the Trig menu and setting the runt logic threshold. While all channels can be set or queried, only channels 3 and 4 can be used if the runt source is channel 1 or 2. Similarly only channels 1 and 2 can be used if the runt source is channel 3 or 4.

**Note:** This command is not available in two-channel instruments.

## Group

Trigger

## Syntax 1

```
TRIGger:A:PULse:RUNT:LOGIc:THReshold:CH<x>  
<NR3>
```

## Syntax 2

```
TRIGger:A:PULse:RUNT:LOGIc:THReshold:CH<x>?
```

## Argument

- <NR3>

This argument specifies the threshold in volts.

## Example 1

```
TRIGger:A:PULse:RUNT:LOGIc:THReshold:CH1?
```

This query might return

```
:TRIGGER:A:PULSE:RUNT:LOGIC:THRESHOLD:CH1  
0.5000
```

indicating that the runt logic trigger threshold is set to 0.5 V for channel 1.

## Example 2

```
TRIGger:A:PULse:RUNT:LOGIc:THReshold:CH3 0.5.
```

This command sets the runt logic trigger threshold to 0.5 V for channel 3.

---

# TRIGger:A:PULse:RUNT:LOGIc?

## Description

This query-only command returns the current A runt trigger logic parameters. This query is equivalent to selecting Runt Setup from the Trig menu and then viewing the current settings.

**Note:** This command is not available in two-channel instruments.

## Group

Trigger

## Syntax

```
TRIGger:A:PULse:RUNT:LOGIc?
```

## Example

```
TRIGger:A:PULse:RUNT:LOGIc?
```

This query might return

```
:TRIGGER:A:PULSE:RUNT:LOGIC:INPUT:CH1 HIGH;  
CH2 X; CH3 X; CH4 LOW;  
:TRIGGER:A:PULSE:RUNT:LOGIC:THRESHOLD:CH1  
1.2000;CH2 1.2000;CH3 1.2000;CH4 1.2000.
```

---

# TRIGger:A:PULse:RUNT:POLarity

## Description

This command sets or queries the polarity for the A pulse runt trigger. This command is equivalent to selecting Runt Setup from the Trig menu and then choosing the Polarity setting.

## Group

Trigger

## Related Commands

TRIGger:A:PULse:RUNT:THReshold?

## Syntax 1

```
TRIGger:A:PULse:RUNT:POLarity {POSITIVE|NEG-  
Ative}
```

## Syntax 2

```
TRIGger:A:PULse:RUNT:POLarity?
```

## Arguments

### ■ POSitive

This indicates that the rising edge crosses the low threshold and the falling edge re-crosses the low threshold without either edge ever crossing the high threshold.

### ■ NEGative

This indicates that the falling edge crosses the high threshold and the rising edge re-crosses the high threshold without either edge ever crossing the low threshold.

## Example 1

```
TRIGger:A:PULse:RUNT:POLarity?
```

This query might return

```
:TRIGGER:A:PULSE:RUNT:POLARITY POSITIVE
```

indicating that the polarity of the A pulse runt trigger is positive.

## Example 2

```
TRIGger:A:PULse:RUNT:POLarity NEGATIVE
```

This command specifies that the polarity of the A pulse runt trigger is negative.

---

# TRIGger:A:PULse:RUNT:THReshold:BOTh

## Description

This command (no query form) sets the upper and lower switching thresholds for the A pulse runt trigger. This command is equivalent to selecting Runt Setup from the Trig menu and then setting the Upper Level and Lower Level voltages.

## Group

Trigger

## Related Commands

TRIGger:A:PULse:RUNT:THReshold?

## Syntax

```
TRIGger:A:PULse:RUNT:THReshold:BOTh {TTL|ECL}
```

## Arguments

- TTL

This sets the upper threshold to 1.8 V and the lower threshold to 800 mV, which are the nominal TTL voltage levels.

- ECL

This sets the upper threshold to -1.1 V and the lower threshold to -1.5 V, which are the nominal ECL voltage levels.

## Example

```
TRIGger:A:PULse:RUNT:THReshold:BOTh TTL
```

This command sets the threshold of the pulse runt trigger to the nominal TTL voltage levels.

---

# TRIGger:A:PULse:RUNT:THReshold:HIGH

## Description

This command sets or queries the upper limit for the A pulse runt trigger. This command is equivalent to selecting Runt Setup from the Trig menu and setting the runt trigger Upper Level voltage.

## Group

Trigger

## Related Commands

TRIGger:A:PULse:RUNT:THReshold?

## Syntax 1

TRIGger:A:PULse:RUNT:THReshold:HIGH <NR3>

## Syntax 2

TRIGger:A:PULse:RUNT:THReshold:HIGH?

## Argument

- <NR3>

This specifies the threshold value, in volts.

## Example 1

TRIGger:A:PULse:RUNT:THReshold:HIGH?

This query might return

```
:TRIGGER:A:PULSE:RUNT:THRESHOLD:HIGH  
1.1000E+00
```

indicating that the upper limit of the pulse runt trigger is currently set to 1.1 V.

## Example 2

TRIGger:A:PULse:RUNT:THReshold:HIGH 120E-3

This command sets the upper limit of the pulse runt trigger to 120 mV.

---

# TRIGger:A:PULse:RUNT:THReshold:LOW

## Description

This command sets or queries the lower limit for the A pulse runt trigger. This command is equivalent to selecting Runt Setup from the Trig menu and then setting the Lower Level voltage.

## Group

Trigger

## Related Commands

TRIGger:A:PULse:RUNT:THReshold?

## Syntax 1

TRIGger:A:PULse:RUNT:THReshold:LOW <NR3>

## Syntax 2

TRIGger:A:PULse:RUNT:THReshold:LOW?

## Argument

- <NR3>

This specifies the threshold value in volts.

## Example 1

TRIGger:A:PULse:RUNT:THReshold:LOW?

This query might return

:TRIGGER:A:PULSE:RUNT:THRESHOLD:LOW 1.2000E-01

indicating that the lower limit of the pulse runt trigger is set to 120 mV.

## Example 2

TRIGger:A:PULse:RUNT:THReshold:LOW 50E-3

This command sets the lower limit of the pulse runt trigger to 50 mV.

---

# TRIGger:A:PULse:RUNT:THReshold?

## Description

This query-only command returns the upper and lower thresholds for the A pulse runt trigger. This command query is equivalent to selecting Runt Setup from the Trig menu and then viewing the Upper Level and Lower Level settings.

## Group

Trigger

## Related Commands

TRIGger:A:PULse:RUNT:POLarity,  
TRIGger:A:PULse:RUNT:THReshold:BOTH,  
TRIGger:A:PULse:RUNT:THReshold:HIGh,  
TRIGger:A:PULse:RUNT:THReshold:LOW

## Syntax

TRIGger:A:PULse:RUNT:THReshold?

## Example

TRIGger:A:PULse:RUNT:THReshold?

This query might return

```
:TRIGGER:A:PULSE:THRESHOLD:HIGh 2.0000E+00;  
LOW 8.0000E-01
```

indicating that the upper threshold is 2 V and that the lower threshold is 800 mV.

---

# TRIGger:A:PULse:RUNT:WHEn

## Description

This command sets or queries the type of pulse width the trigger checks for when it detects a runt. This is equivalent to selecting Runt Setup from the Trig menu and choosing the desired Trigger When setting from the drop-down list.

## Group

Trigger

## Related Commands

TRIGger:A:PULse:RUNT:WIDth

## Syntax 1 (Four-channel instruments)

```
TRIGger:A:PULse:RUNT:WHEn {LOGIC|OCCurs|WID-  
ERthan}
```

## Syntax 2 (Two-channel instruments)

```
TRIGger:A:PULse:RUNT:WHEn {OCCurs|WIDERthan}
```

## Syntax 3

```
TRIGger:A:PULse:RUNT:WHEn?
```

## Arguments

### ■ LOGIC

This argument specifies a trigger event when a runt occurs and the AND of the logic channels. This argument is not available for two-channel instruments.

### ■ OCCurs

This argument specifies a trigger event if a runt of any detectable width occurs.

### ■ WIDERthan

This specifies a trigger event if a runt greater than the specified width occurs.

## Example 1

```
TRIGger:A:PULse:RUNT:WHEn?
```

This query might return

```
:TRIGGER:A:PULSE:RUNT:WHEN OCCURS
```

indicating that a runt trigger will occur if the instrument detects a runt of any detectable width.



---

## Example 2

`TRIGger:A:PULse:RUNT:WHEn WIDERthan`

This command sets the runt trigger to occur when the instrument detects a runt in a pulse wider than the specified width.

---

# TRIGger:A:PULse:RUNT:WIDth

## Description

This command sets or queries the minimum width for an A Pulse Runt trigger. This command is equivalent to selecting Runt Setup from the Trig menu and then setting the Width.

## Group

Trigger

## Related Commands

TRIGger:A:PULse:RUNT:WHEn

## Syntax 1

TRIGger:A:PULse:RUNT:WIDth <NR3>

## Syntax 2

TRIGger:A:PULse:RUNT:WIDth?

## Argument

- <NR3>

This specifies the minimum width, in seconds.

## Example 1

TRIGger:A:PULse:RUNT:WIDth?

This query might return

:TRIGGER:A:PULSE:RUNT:WIDTH 2.0000E-09

indicating that the minimum width of a pulse runt trigger is 2 ns.

## Example 2

TRIGger:A:PULse:RUNT:WIDth 15E-6

This command sets the minimum width of the pulse runt trigger to 15  $\mu$ s.

---

# TRIGger:A:PULse:RUNT?

## Description

This query-only command returns the current A runt trigger parameters. This command query is equivalent to selecting Runt Setup from the Trig menu and then viewing the current settings.

## Group

Trigger

## Related Commands

TRIGger:A:PULse:GLItch?, TRIGger:A:PULse:TIMEOut?, TRIGger:A:PULse:TRANsition?, TRIGger:A:PULse:WIDth?

## Syntax

```
TRIGger:A:PULse:RUNT?
```

## Example

```
TRIGger:A:PULse:RUNT?
```

This query might return

```
:TRIGGER:A:PULSE:RUNT:LOGIC:INPUT:CH1 HIGH;  
CH2 X; CH3 X; CH4 LOW;:TRIG-  
GER:A:PULSE:RUNT:LOGIC:THRESHOLD:CH1 1.2000;  
CH2 1.2000;CH3 1.2000;CH4 1.2000;  
:TRIGGER:A:PULSE:RUNT:POLARITY POSITIVE  
:THRESHOLD:HIGH 1.2000;  
LOW 800.0000E-3;:TRIGGER:A:PULSE:RUNT:WHEN OC-  
CURS;WIDTH 2.0000E-9.
```

---

# TRIGger:A:PULse:SOUrce

## Description

This command sets or queries the source for the A pulse trigger. This source parameter applies to all classes of pulse triggers. This command is equivalent to selecting A Event (Main) Trigger Setup from the Trig menu, selecting the pulse type (Glitch, Width, Runt, Timeout, or Transition), and then choosing the desired channel from the Source pull-down list.

## Group

Trigger

## Related Commands

TRIGger:A:EDGE:SOUrce

## Syntax 1

```
TRIGger:A:PULse:SOUrce CH<x>
```

## Syntax 2

```
TRIGger:A:PULse:SOUrce?
```

## Argument

- CH<x>

This specifies one of the input channels, which range from 1 through 4 for four-channel instruments and 1 through 2 for two-channel instruments.

## Example 1

```
TRIGger:A:PULse:SOUrce?
```

This query might return

```
:TRIGGER:A:PULSE:SOURCE CH2
```

indicating that channel 2 is the source for the A pulse trigger.

## Example 2

```
TRIGger:A:PULse:SOUrce CH4
```

This command sets channel 4 as the source for the A pulse trigger.

---

# TRIGger:A:PULse:TIMEOut:POLarity

## Description

This command sets or queries the polarity for the A pulse timeout trigger. This command is equivalent to selecting Timeout Setup from the Trig menu and setting the desired polarity in the Trigger When box.

## Group

Trigger

## Related Commands

TRIGger:A:PULse:TIMEOut?

## Syntax 1

```
TRIGger:A:PULse:TIMEOut:POLarity {STAYS-  
High|STAYSLow|EITHer}
```

## Syntax 2

```
TRIGger:A:PULse:TIMEOut:POLarity?
```

## Arguments

- STAYSHigh

This indicates that a pulse edge must stay high (positive) for the required time period to permit timeout triggering to occur. This is the default polarity.

- STAYSLow

This indicates that a pulse edge must stay low (negative) for the required time period to permit timeout triggering to occur.

- EITHer

This indicates that the polarity of the timeout trigger can stay either high or low (positive or negative) for the required time period to permit time out triggering to occur.

## Example 1

```
TRIGger:A:PULse:TIMEOut:POLarity?
```

This query might return

```
:TRIGGER:A:PULSE:TIMEOUT:POLARITY EITHER
```

indicating that the polarity of the A timeout trigger can be either positive or negative.

## Example 2

```
TRIGger:A:PULse:TIMEOut:POLarity STAYSHigh
```

This command sets the polarity of the A timeout trigger to positive.

---

# TRIGger:A:PULse:TIMEOut:TIME

## Description

This command sets or queries the pulse timeout trigger time (measured in seconds). This command is equivalent to selecting Timeout Setup from the Trig menu and setting a value for Timer.

## Group

Trigger

## Related Commands

TRIGger:A:PULse:TIMEOut?

## Syntax 1

TRIGger:A:PULse:TIMEOut:TIME <NR3>

## Syntax 2

TRIGger:A:PULse:TIMEOut:TIME?

## Argument

- <NR3>

This argument specifies the timeout period in seconds.

## Example 1

TRIGger:A:PULse:TIMEOut:TIME?

This query might return

```
:TRIGGER:A:PULSE:TIMEOUT:TIME 2.0000E-9
```

indicating that the timeout is currently set to 2 ns.

## Example 2

TRIGger:A:PULse:TIMEOut:TIME 3.134E-6

This command sets the timeout to 3.134  $\mu$ s.

---

# TRIGger:A:PULse:TIMEOut?

## Description

This query-only command returns the polarity and time-out duration for the A pulse timeout trigger. This command is equivalent to selecting Timeout Setup from the Trig menu and viewing the polarity in the Trigger When box and the Timer setting.

## Group

Trigger

## Related Commands

TRIGger:A:PULse:TIMEOut:POLarity,  
TRIGger:A:PULse:TIMEOut:TIME

## Syntax

TRIGger:A:PULse:TIMEOut?

## Example

TRIGger:A:PULse:TIMEOut?

This query might return

```
:TRIGGER:A:PULSE:TIMEOUT:POLARITY STAYS-  
HIGH;TIME 2.0000E-9
```

indicating that the polarity setting for the A pulse trigger is STAYS-HIGH (positive) and the timeout duration is 2 ns.

---

# TRIGger:A:PULse:TRANSition:DELTATime

## Description

This command sets or queries the delta time used in calculating the transition value for the transition trigger. This is equivalent to selecting Transition Setup from the Trig menu and setting the Time.

## Group

Trigger

## Related Commands

TRIGger:A:PULse:TRANSition:POLarity, TRIGger:A:PULse:TRANSition:THReshold?

## Syntax 1

```
TRIGger:A:PULse:TRANSition:DELTATime <NR3>
```

## Syntax 2

```
TRIGger:A:PULse:TRANSition:DELTATime?
```

## Argument

- <NR3>

This specifies the delta time, in seconds.

## Example 1

```
TRIGger:A:PULse:TRANSition:DELTATime?
```

This query might return

```
:TRIGGER:A:PULSE:TRANSITION:DELTATIME  
2.0000E-09
```

indicating that the delta time of the transition trigger is set to 2 ns.

## Example 2

```
TRIGger:A:PULse:TRANSition:DELTATime 15E-6
```

This command sets the delta time of the transition trigger to 15  $\mu$ s.



---

# TRIGger:A:PULse:TRANSition:POLarity

## Description

This command sets or queries the polarity for the transition trigger. This command is equivalent to selecting Transition Setup from the Trig menu and choosing from the Polarity pull-down list.

## Group

Trigger

## Related Commands

TRIGger:A:PULse:TRANSition:DELTAtime

## Syntax 1

```
TRIGger:A:PULse:TRANSition:POLarity {POSITIVE|NEGative|EITHer}
```

## Syntax 2

```
TRIGger:A:PULse:TRANSition:POLarity?
```

## Arguments

### ■ POSITIVE

This indicates that a pulse edge must traverse from the lower (most negative) to higher (most positive) level for transition triggering to occur.

### ■ NEGative

This indicates that a pulse edge must traverse from the upper (most positive) to lower (most negative) level for transition triggering to occur.

### ■ EITHer

This indicates either positive or negative polarity.

## Example 1

```
TRIGger:A:PULse:TRANSition:POLarity?
```

This query might return

```
:TRIGGER:A:PULSE:TRANSITION:POLARITY EITHER
```

indicating that the polarity can be either positive or negative.

## Example 2

```
TRIGger:A:PULse:TRANSition:DeltaTime NEGative
```

This command sets the transition polarity to negative.

---

# TRIGger:A:PULse:TRANSition:THReshold:BOTh

## Description

This command (no query form) sets the upper and lower thresholds for the transition trigger. This command is equivalent to selecting Transition Setup from the Trig menu and setting the desired Upper Level and Lower Level voltages.

## Group

Trigger

## Related Commands

TRIGger:A:PULse:TRANSition:THReshold:HIGH,  
TRIGger:A:PULse:TRANSition:THReshold:LOW

## Syntax 1

```
TRIGger:A:PULse:TRANSition:THReshold:BOTh  
{TTL|ECL}
```

## Arguments

- TTL

This sets the upper threshold to 1.2 V and the lower threshold to 800 mV, which represent the nominal TTL voltage levels.

- ECL

This sets the upper threshold to -1.1 V and the lower threshold to -1.5 V, which represent the nominal ECL voltage levels.

## Example 1

```
TRIGger:A:PULse:TRANSition:THReshold:BOTh TTL
```

This command sets the thresholds of the transition trigger to the nominal TTL voltage levels.

---

# TRIGger:A:PULse:TRANSition:THReshold:HIGH

## Description

This command sets or queries the upper (most positive) transition trigger threshold. This command is equivalent to selecting Transition Setup from the Trig menu and then setting the desired Upper Level voltage.

## Group

Trigger

## Related Commands

TRIGger:A:PULse:TRANSition:THReshold:LOW

## Syntax 1

```
TRIGger:A:PULse:TRANSition:THReshold:HIGH  
<NR3>
```

## Syntax 2

```
TRIGger:A:PULse:TRANSition:THReshold:HIGH?
```

## Argument

- <NR3>

This specifies the upper threshold, in volts.

## Example 1

```
TRIGger:A:PULse:TRANSition:THReshold:HIGH?
```

This query might return

```
:TRIGGER:A:PULSE:TRANSITION:THRESHOLD:HIGH  
2.0000E+00
```

indicating that the upper limit of the pulse transition trigger is 2 V.

## Example 2

```
TRIGger:A:PULse:TRANSition:THReshold:HIGH  
120E-3
```

This command sets the upper limit of the pulse transition trigger to 120 mV.

---

# TRIGger:A:PULse:TRANSition:THReshold:LOW

## Description

This command sets or queries the lower (most negative) transition trigger threshold. This command is equivalent to selecting Transition Setup from the Trig menu and setting the desired Lower Level voltage.

## Group

Trigger

## Related Commands

TRIGger:A:PULse:TRANSition:THReshold:HIGH

## Syntax 1

TRIGger:A:PULse:TRANSition:THReshold:LOW <NR3>

## Syntax 2

TRIGger:A:PULse:TRANSition:THReshold:LOW?

## Argument

- <NR3>

This specifies the lower threshold, in volts.

## Example 1

TRIGger:A:PULse:TRANSition:THReshold:LOW?

This query might return

```
:TRIGGER:A:PULSE:TRANSITION:THRESHOLD:LOW  
50.0000E-03
```

indicating that the lower limit of the transition trigger is 50 mV.

## Example 2

TRIGger:A:PULse:TRANSition:THReshold:LOW 20E-3

This command sets the lower limit of the transition trigger to 20 mV.

---

# TRIGger:A:PULse:TRANSition:THReshold?

## Description

This query-only command returns the upper and lower threshold limits for the transition time trigger. This command is equivalent to selecting Transition Setup from the Trig menu and viewing the Upper Level and Lower Level voltage settings.

## Group

Trigger

## Related Commands

TRIGger:A:PULse:TRANSition:DELTATime,  
TRIGger:A:PULse:TRANSition:POLarity

## Syntax

```
TRIGger:A:PULse:TRANSition:THReshold?
```

## Example

```
TRIGger:A:PULse:TRANSition:THReshold?
```

This query might return

```
:TRIGGER:A:PULSE:TRANSITION:THRESHOLD:HIGH  
1.2000;LOW 800.0000E-3
```

indicating the upper and lower threshold limits for the transition time trigger.

---

# TRIGger:A:PULse:TRANSition:WHEn

## Description

This command sets or queries whether to check for a transitioning signal that is faster or slower than the specified delta time. This is equivalent to selecting Transition Setup from the Trig menu and choosing the Trigger When Transition Time setting.

## Group

Trigger

## Related Commands

TRIGger:A:PULse:TRANSition:DELTATime, TRIGger:A:PULse:TRANSition:POLarity, TRIGger:A:PULse:TRANSition:THReshold?

## Syntax 1

```
TRIGger:A:PULse:TRANSition:WHEn {FASTERthan|SLOWERthan}
```

## Syntax 2

```
TRIGger:A:PULse:TRANSition:WHEn?
```

## Arguments

- **FASTERthan**

This sets the trigger to occur when the transitioning signal is faster than the set volts/second rate.

- **SLOWERthan**

This sets the trigger to occur when the transitioning signal is slower than the set volts/second rate.

## Example 1

```
TRIGger:A:PULse:TRANSition:WHEn?
```

This query might return

```
:TRIGGER:A:PULSE:TRANSITION:WHEN FASTERTHAN
```

indicating that the transition triggers when the transitioning signal is faster than the set volts/second rate.

## Example 2

```
TRIGger:A:PULse:TRANSition:WHEn SLOWERthan
```

---

# TRIGger:A:PULse:TRANSition?

## Description

This query-only command returns delta time, polarity, and both upper and lower threshold limits for the transition time trigger. This command is equivalent to selecting Transition Setup from the Trig menu and then viewing the current transition settings.

## Group

Trigger

## Related Commands

TRIGger:A:TYPe, TRIGger:A:PULse:CLAss,  
TRIGger:A:PULse:GLItch, TRIGger:A:PULse:RUNT,  
TRIGger:A:PULse:TIMEOut, TRIGger:A:PULse:WIDth

## Syntax

```
TRIGger:A:PULse:TRANSition?
```

## Example

```
TRIGger:A:PULse:TRANSition?
```

This query might return

```
:TRIGGER:A:PULSE:TRANSITION:DELTATIME  
2.0000E-9;POLARITY POSITIVE;THRESHOLD:HIGH  
1.2000;LOW 800.0000E-3;:TRIGGER:A:PULSE:TRAN-  
SITION:WHEN SLOWERTHAN
```

indicating the current transition time trigger settings.

---

# TRIGger:A:PULse:WIDth:HIGHLimit

## Description

This command sets or queries the upper limit for the width trigger. This command is equivalent to selecting Width Setup from the Trig menu and setting the Upper Limit.

## Group

Trigger

## Related Commands

TRIGger:A:PULse:WIDth?

## Syntax 1

TRIGger:A:PULse:WIDth:HIGHLimit <NR3>

## Syntax 2

TRIGger:A:PULse:WIDth:HIGHLimit?

## Argument

- <NR3>

This specifies the width trigger upper limit in seconds.

## Example 1

TRIGger:A:PULse:WIDth:HIGHLimit?

This query might return

```
:TRIGGER:A:PULSE:WIDTH:HIGHLIMIT 2.0000E-9
```

indicating that the upper limit of the width trigger is set to 2 ns.

## Example 2

TRIGger:A:PULse:WIDth:HIGHLimit 5.0E-6

This command sets the upper limit of the width trigger to 5  $\mu$ s.



---

# TRIGger:A:PULse:WIDth:LOWLimit

## Description

This command sets or queries the lower limit for the width trigger. This command is equivalent to selecting Width Setup from the Trig menu and setting the pulse Lower Limit.

## Group

Trigger

## Related Commands

TRIGger:A:PULse:WIDth?

## Syntax 1

TRIGger:A:PULse:WIDth:LOWLimit <NR3>

## Syntax 2

TRIGger:A:PULse:WIDth:LOWLimit?

## Argument

- <NR3>

This specifies the A pulse width trigger lower limit, in seconds.

## Example 1

TRIGger:A:PULse:WIDth:LOWLimit?

This query might return

:TRIGGER:A:PULSE:WIDTH:LOWLIMIT 1.0000E-9

indicating that the lower limit of the width trigger is set to 1 ns.

## Example 2

TRIGger:A:PULse:WIDth:LOWLimit 2.0E-6

This command sets the lower limit of the width trigger to 2  $\mu$ s.

---

# TRIGger:A:PULse:WIDth:POLarity

## Description

This command sets or queries the polarity for the width trigger. This command is equivalent to selecting Width Setup from the Trig menu and selecting the Polarity.

## Group

Trigger

## Related Commands

TRIGger:A:PULse:WIDth?

## Syntax 1

```
TRIGger:A:PULse:WIDth:POLarity {NEGative|POSITIVE}
```

## Syntax 2

```
TRIGger:A:PULse:WIDth:POLarity?
```

## Arguments

- NEGative

This specifies a negative pulse.

- POSITIVE

This specifies a positive pulse.

## Example 1

```
TRIGger:A:PULse:WIDth:POLarity?
```

This query might return

```
:TRIGGER:A:PULSE:WIDTH:POLARITY POSITIVE
```

indicating a positive pulse.

## Example 2

```
TRIGger:A:PULse:WIDth:POLarity NEGative
```

This command sets the pulse polarity to negative.

---

# TRIGger:A:PULse:WIDth:WHEn

## Description

This command sets or queries whether to trigger on a pulse width that falls outside (or within) the specified range of limits. You can define or query trigger pulse width upper and lower limits using the TRIGger:A:PULse:WIDth:HIGHLimit and TRIGger:A:PULse:WIDth:LOWLimit commands.

This command is equivalent to selecting Width Setup from the Trig menu and then choosing from the Trig When drop-down list.

## Group

Trigger

## Related Commands

TRIGger:A:PULse:WIDth:HIGHLimit, TRIGger:A:PULse:WIDth:LOWLimit

## Syntax 1

```
TRIGger:A:PULse:WIDth:WHEn {OUTside|WIThin}
```

## Syntax 2

```
TRIGger:A:PULse:WIDth:WHEn?
```

## Arguments

### ■ OUTside

This argument causes a trigger event the duration of the pulse is greater than the high limit or less than the low limit specified. The high and low limits are specified with the TRIGger:A:PULse:WIDth:HIGHLimit and TRIGger:A:PULse:WIDth:LOWLimit commands respectively.

### ■ WIThin

This argument causes a trigger event when the duration of the pulse is within the high and low limits. The high and low limits are specified with the TRIGger:A:PULse:WIDth:HIGHLimit and TRIGger:A:PULse:WIDth:LOWLimit command respectively.

## Example 1

```
TRIGger:A:PULse:WIDth:WHEn?
```

This query might return

```
:TRIGGER:A:PULSE:WIDTH:WHEN OUTSIDE
```

indicating the conditions for generating a width trigger.

---

## Example 2

TRIGger:A:PULse:WIDth:WHEn WITHin

This command specifies that the duration of the A pulse will fall within defined high and low limits.

---

# TRIGger:A:PULse:WIDth?

## Description

This query-only command returns the width parameters for the A pulse width trigger. This command is equivalent to selecting Width Setup from the Trig menu and then viewing the current pulse width trigger Lower Limit, Upper Limit, Trig When and Polarity settings.

## Group

Trigger

## Related Commands

TRIGger:A:PULse:WIDth:HIGHLimit,  
TRIGger:A:PULse:WIDth:LOWLimit,  
TRIGger:A:PULse:WIDth:POLarity,  
TRIGger:A:PULse:WIDth:WHEn

## Syntax 1

```
TRIGger:A:PULse:WIDth
```

## Example

```
TRIGger:A:PULse:WIDth?
```

This query might return

```
:TRIGGER:A:PULSE:WIDTH:LOWLIMIT 2.0000E-9;  
HIGHLIMIT 2.0000E-9;WHEN WITHIN;  
POLARITY POSITIVE
```

as the current A width trigger parameters.

---

# TRIGger:A:PULse:WINDow:LOGIc:INPUT:CH<x>

## Description

This command sets or queries the A window logic input for the channel specified by <x>, which can be 1, 2, 3, or 4. This is equivalent to selecting Window Setup from the Trig menu, selecting Logic in the Trigger When box, and selecting a logic (H, L, or X) for the channel. While all channels can be set or queried, only channels 3 and 4 can be used if the window source is channel 1 or 2. Similarly, only channels 1 and 2 can be used if the window source is channel 3 or 4.

**Note:** This command is not available in two-channel instruments.

## Group

Trigger

## Syntax 1

```
TRIGger:A:PULse:WINDow:LOGIc:INPUT:CH<x>
{HIGH\LOW\X}
```

## Syntax 2

```
TRIGger:A:PULse:RUNT:LOGIc:INPUT:CH<x>?
```

## Arguments

- HIGH

This argument specifies logic high.

- LOW

This argument specifies logic low.

- X

This argument specifies a Don't Care state.

## Example 1

```
TRIGger:A:PULse:WINDow:LOGIc:INPUT:CH2?
```

This query might return

```
:TRIGGER:A:PULSE:WINDOW:LOGIC:INPUT CH2 LOW
```

indicating that the window logic trigger input is set to logic low for channel 2.

## Example 2

```
TRIGger:A:PULse:WINDow:LOGIc:INPUT:CH3 HIGH
```

This command sets the window logic trigger input to logic high for channel 3.

---

# TRIGger:A:PULse:WINDow:LOGIc:INPUT?

## Description

This query-only command returns the current window trigger logic input parameters.

**Note:** This command is not available in two-channel instruments.

## Group

Trigger

## Syntax

```
TRIGger:A:PULse:WINDow:LOGIcINPUT?
```

## Example

```
TRIGger:A:PULse:WINDow:LOGIc:INPUT?
```

This query might return

```
:TRIGGER:A:PULSE:WINDOW:LOGIC:INPUT:CH1 HIGH;  
CH2 X; CH3 X; CH4 LOW.
```

---

# TRIGGER:A:PULSE:WINDOW:LOGIC:THRESHOLD:CH<x>

## Description

This command sets or queries the window logic trigger threshold for the channel specified by <x>, which can be 1, 2, 3, or 4. This is equivalent to selecting Window Setup from the Trig menu and setting the window logic threshold in the Trigger When box. While all channels can be set or queried, only channels 3 and 4 can be used if the window source is channel 1 or 2. Similarly, only channels 1 and 2 can be used if the window source is channel 3 or 4.

**Note:** This command is not available in two-channel instruments.

## Group

Trigger

## Syntax 1

```
TRIGGER:A:PULSE:WINDOW:LOGIC:THRESHOLD:CH<x>  
<NR3>
```

## Syntax 2

```
TRIGGER:A:PULSE:WINDOW:LOGIC:THRESHOLD:CH<x>?
```

## Argument

- <NR3>

This argument specifies the window logic trigger threshold in volts.

## Example 1

```
TRIGGER:A:PULSE:WINDOW:LOGIC:THRESHOLD:CH1?
```

This query might return

```
:TRIGGER:A:PULSE:WINDOW:LOGIC:THRESHOLD:CH1  
0.5000
```

indicating that the window logic trigger threshold is set to 0.5 V for channel 1.

## Example 2

```
TRIGGER:A:PULSE:WINDOW:LOGIC:THRESHOLD:CH3 0.5
```

This command sets the window logic trigger threshold to 0.5 V for channel 3.



---

# TRIGger:A:PULse:WINDow:LOGIc?

## Description

This query-only command returns the current window trigger logic parameters.

**Note:** This command is not available in two-channel instruments.

## Group

Trigger

## Syntax

```
TRIGger:A:PULse:WINDow:LOGIc?
```

## Example

```
TRIGger:A:PULse:WINDow:LOGIc?
```

This query might return

```
:TRIGGER:A:PULSE:WINDOW:LOGIC:INPUT:CH1 HIGH;  
CH2 X; CH3 X; CH4 LOW;:TRIGGER:A:PULSE:WIN-  
DOW:LOGIC:THRESHOLD:CH1 1.2000;CH2 1.2000;CH3  
1.2000;CH4 1.2000.
```

---

# TRIGger:A:PULse:WINDow:THReshold:BOTh

## Description

This command (no query form) sets the upper and lower switching thresholds for the window trigger. This command is equivalent to selecting Window Setup from the Trig menu and then setting the Upper Level and Lower Level voltages.

## Group

Trigger

## Syntax

```
TRIGger:A:PULse:WINDow:THReshold:BOTh  
{TTL|ECL}
```

## Arguments

- TTL

This argument sets the upper threshold to 2.0 V and the lower threshold to 800 mV, which are the nominal TTL voltage levels.

- ECL

This argument sets the upper threshold to -1.1 V and the lower threshold to -1.5 V, which are the nominal ECL voltage levels.

## Example

```
TRIGger:A:PULse:WINDow:THReshold:BOTh TTL
```

This command sets the threshold of the window trigger to the nominal TTL voltage levels.

---

# TRIGger:A:PULse:WINDow:THReshold:HIGH

## Description

This command sets or queries the upper limit for the pulse window trigger. This command is equivalent to selecting Window Setup from the Trig menu and setting the window trigger Upper Level voltage.

## Group

Trigger

## Syntax 1

```
TRIGger:A:PULse:WINDow:THReshold:HIGH <NR3>
```

## Syntax 2

```
TRIGger:A:PULse:WINDow:THReshold:HIGH?
```

## Argument

- <NR3>

This specifies the threshold value in volts.

## Example 1

```
TRIGger:A:PULse:WINDow:THReshold:HIGH?
```

This query might return

```
:TRIGGER:A:PULSE:WINDOW:THRESHOLD:HIGH  
1.1000E+00
```

indicating that the upper limit of the window trigger is set to 1.1 V.

## Example 2

```
TRIGger:A:PULse:WINDow:THReshold:HIGH 2.0
```

This command sets the upper limit of the pulse window trigger to 2 V.

---

# TRIGger:A:PULse:WINDow:THReshold:LOW

## Description

This command sets or queries the lower limit for the window trigger. This command is equivalent to selecting Window Setup from the Trig menu and setting the Lower Level voltage.

## Group

Trigger

## Syntax 1

```
TRIGger:A:PULse:WINDow:THReshold:LOW <NR3>
```

## Syntax 2

```
TRIGger:A:PULse:WINDow:THReshold:LOW?
```

## Argument

- <NR3>

This specifies the threshold value in volts.

## Example 1

```
TRIGger:A:PULse:WINDow:THReshold:LOW?
```

This query might return

```
:TRIGGER:A:PULSE:WINDOW:THRESHOLD:LOW  
1.2000E-01
```

indicating that the lower limit of the window trigger is currently set to 120 mV.

## Example 2

```
TRIGger:A:PULse:WINDow:THReshold:LOW 0.8
```

This command sets the lower threshold of the window trigger to 0.8 V.

---

# TRIGger:A:PULse:WINDow:THReshold?

## Description

This query-only command returns the upper and lower thresholds for the window trigger. This command query is equivalent to selecting Window Setup from the Trig menu and viewing the window trigger Upper Level and Lower Level settings.

## Group

Trigger

## Syntax

```
TRIGger:A:PULse:WINDow:THReshold?
```

## Example

```
TRIGger:A:PULse:WINDow:THReshold?
```

This query might return

```
:TRIGGER:A:PULSE:WINDOW:THRESHOLD:HIGH  
1.2000;LOW 800.0000E-3
```

indicating that the upper threshold is 1.2 V and that the lower threshold is 800 mV.

---

# TRIGger:A:PULse:WINDow:TYPE

## Description

This command sets or queries the window trigger type. This command is equivalent to selecting Window Setup from the Trig menu and selecting Outside Limits or Inside Limits in the Trigger When section.

## Group

Trigger

## Syntax 1

```
TRIGger:A:PULse:WINDow:TYPE {INSide|OUTside}
```

## Syntax 2

```
TRIGger:A:PULse:WINDow:TYPE?
```

## Arguments

- **INSide**

This argument causes a trigger event to occur when a pulse enters the window defined by the upper and lower thresholds.

- **OUTside**

This argument causes a trigger event to occur when a pulse goes outside the window defined by the upper and lower thresholds.

## Example 1

```
TRIGger:A:PULse:WINDow:TYPE?
```

This query might return

```
:TRIGGER:A:PULSE:WINDOW:TYPE INSide
```

indicating that the window type is set to inside, which will cause a trigger event when a pulse enters the window defined by the upper and lower thresholds.

## Example 2

```
TRIGger:A:PULse:WINDow:TYPE OUTside
```

This command sets the window type to outside.

---

# TRIGger:A:PULse:WINDow:WHEn

## Description

This command sets or queries the window trigger violation qualification. This command is equivalent to selecting Window Setup from the Trig menu and selecting Logic, Occurs, or Wider than in the Trigger When box.

## Group

Trigger

## Related Commands

### Syntax 1

```
TRIGger:A:PULse:WINDow:WHEn {LOGic|OCCurs|WID-  
ERthan}
```

### Syntax 2

```
TRIGger:A:PULse:WINDow:WHEn?
```

## Arguments

#### ■ LOGic

This argument specifies a trigger event when a window violation occurs on the AND of the logic channels.

#### ■ OCCurs

This argument specifies a trigger event if any detectable window violation occurs.

#### ■ WIDERthan

This specifies a trigger event if a window violation greater than the specified width occurs.

## Example 1

```
TRIGger:A:PULse:WINDow:WHEn?
```

This query might return

```
:TRIGGER:A:PULSE:WINDOW:WHEN OCCURS
```

indicating that a trigger will occur if the instrument detects any window violation.

## Example 2

```
TRIGger:A:PULse:WINDow:WHEn OCCurs
```

This command sets the window violation qualifier to OCCurs, which causes a trigger to occur when the instrument detects any window violation.

---

# TRIGGER:A:PULSE:WINDOW:WIDTH

## Description

This command sets or queries the minimum width for a window violation. This command is equivalent to selecting Window Setup from the Trig menu, selecting Wider than in the Trigger When box, and setting the Width.

## Group

Trigger

## Syntax 1

```
TRIGGER:A:PULSE:WINDOW:WIDTH <NR3>
```

## Syntax 2

```
TRIGGER:A:PULSE:WINDOW:WIDTH?
```

## Argument

- <NR3>

This argument specifies the minimum width in seconds.

## Example 1

```
TRIGGER:A:PULSE:WINDOW:WIDTH?
```

This query might return

```
:TRIGGER:A:PULSE:WINDOW:WIDTH 2.0000E-09
```

indicating that the minimum width of a valid window violation is 2 ns.

## Example 2

```
TRIGGER:A:PULSE:WINDOW:WIDTH 15E-9
```

This command sets the minimum width of a valid window violation to 15 ns.



---

# TRIGger:A:PULse:WINDow?

## Description

This query-only command returns the current window trigger parameters.

## Group

Trigger

## Syntax

TRIGger:A:PULse:WINDow?

## Example

TRIGger:A:PULse:WINDow?

This query might return

```
:TRIGGER:A:PULSE:WINDOW:TYPE INSIDE;  
WHEN OCCURS;WIDTH 2.0000e-9;  
LOGIC:INPUT:CH1 HIGH;CH2 X; CH3 X; CH4 LOW;  
:TRIGGER:A:PULSE:WINDOW:LOGIC:THRESHOLD:CH1  
1.2000;CH2 1.2000;CH3 1.2000;CH4 1.2000;  
:TRIGGER:A:PULSE:WINDOW:THRESHOLD:HIGH  
1.2000;LOW 800.0000E-3.
```

---

# TRIGger:A:PULse?

## Description

This query-only command returns the A pulse trigger parameters.

## Group

Trigger

## Related Commands

TRIGger:A:EDGE?, TRIGger:A:LOGic?

## Syntax

TRIGger:A:PULse?

## Example

TRIGger:A:PULse?

This query might return

```
:TRIGGER:A:PULSE:CLASS GLITCH;SOURCE CH1;
GLITCH:WIDTH 2.0000E-9;
TRIGIF ACCEPT;POLARITY POSITIVE;
:TRIGGER:A:PULSE:RUNT:POLARITY POSITIVE;
THRESHOLD:HIGH 1.2000;LOW 800.0000E-3;
:TRIGGER:A:PULSE:RUNT:WHEN OCCURS;
WIDTH 2.0000E-9;
:TRIGGER:A:PULSE:TRANSITION:DELTATIME
2.0000E-9;POLARITY POSITIVE;
THRESHOLD:HIGH 1.2000;LOW 800.0000E-3;
:TRIGGER:A:PULSE:TRANSITION:WHEN SLOWERTHAN;
:TRIGGER:A:PULSE:WIDTH:LOWLIMIT 2.0000E-9;
HIGHLIMIT 2.0000E-9;WHEN WITHIN;
POLARITY POSITIVE;:TRIGGER:A:PULSE:TIMEOUT:PO-
LARITY STAYSHIGH;TIME 2.0000E-9
```

---

# TRIGger:A:TYPE

## Description

This command sets or queries the type of A trigger. The five types of triggers are of Edge, Logic, Pulse, Video, and Communication (Option SM). Logic and Pulse triggers contain classes. Logic triggers consist of State and Pattern classes; Pulse triggers consist of Glitch, Runt, Width, Transition, and Timeout classes. Once you have set the trigger type, you may also need to identify the associated trigger class. For details on selecting Logic and Pulse trigger classes, see TRIGger:A:LOGic:CLAss and TRIGger:A:PULse:CLAss respectively. This command is similar to selecting A Event (Main) Trigger Setup from the Trig menu and then selecting the desired Trigger Type.

## Group

Trigger

## Related Commands

TRIGger:A:EDGE, TRIGger:A:LOGic:CLAss, TRIGger:A:PULse:CLAss,

## Syntax 1

```
TRIGger:A:TYPE {EDGE | LOGic | PULse | VIDEo | COMMunication}
```

## Syntax 2

```
TRIGger:A:TYPE?
```

## Arguments

### ■ EDGE

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

### ■ LOGic

This specifies that a trigger occurs when specified conditions are met and is controlled by the TRIGger:A:LOGic commands.

### ■ PULse

This specifies that a trigger occurs when a specified pulse is found and is controlled by the TRIGger:A:PULse commands.

### ■ VIDEo

This specifies that the trigger occurs when a video signal is found.

### ■ COMMunication

This specifies that a trigger occurs when a communications signal is found. Supports AMI, HDB3, B3ZS, B6ZS, B8ZS, CMI, MLT3, Manchester, and NRZ encoded communications signals. COMMunication is available only if Option SM is installed.

---

### **Example 1**

`TRIGger:A:TYPe EDGE`

This command sets the A trigger type to EDGE.

### **Example 2**

`TRIGger:A:TYPe?`

This query might return

`:TRIGGER:A:TYPE PULSE`

indicating that the A trigger type is a pulse trigger.

---

# TRIGger:A:VIDeo:CUSTom:FORMat

## Description

This command sets or queries the video trigger format. Use this command only when the video format is set to custom.

## Group

Trigger

## Related Commands

### Syntax 1

```
TRIGger:A:VIDeo:CUSTom:FORMat {INTERLAcEd|PRO-  
Gressive}
```

### Syntax 2

```
TRIGger:A:VIDeo:CUSTom:FORMat?
```

## Arguments

#### ■ INTERLAcEd

This argument sets the format for interlaced video lines.

#### ■ PROGressive

This argument sets the format for progressive video lines.

## Example 1

```
TRIGger:A:VIDeo:CUSTom:FORMat?
```

This query might return

```
:TRIGger:A:VIDeo:CUSTom:FORMat INTERLACED
```

indicating that interlaced is selected as the custom format for the A video trigger.

## Example 2

```
TRIGger:A:VIDeo:CUSTom:FORMat PROGressive
```

This command sets the custom format for the A video trigger to progressive lines.

---

# TRIGger:A:VIDeo:CUSTom:SCAN

## Description

This command sets or queries the horizontal line scan rate of the A video trigger. Use this command only when the video format is set to custom. This is equivalent to selecting Video Setup from the Trig menu, setting Custom for Format, and selecting the Scan Rate from the drop-down menu.

## Group

Trigger

## Related Commands

### Syntax 1

```
TRIGger:A:VIDeo:CUSTom:SCAN  
{RATE1 | RATE2 | RATE3 | RATE4 | RATE5 }
```

### Syntax 2

```
TRIGger:A:VIDeo:CUSTom:SCAN?
```

## Arguments

#### ■ RATE1

This argument sets the range of the video line scan rate to 15 kHz through 20 kHz. This is the standard broadcast rate.

#### ■ RATE2

This argument sets the range of the video line scan rate to 20 kHz through 25 kHz.

#### ■ RATE3

This argument sets the range of the video line scan rate to 25 kHz through 35 kHz

#### ■ RATE4

This argument sets the range of the video line scan rate to 35 kHz through 50 kHz

#### ■ RATE5

This argument sets the range of the video line scan rate to 50 kHz through 65 kHz

## Example 1

```
TRIGger:A:VIDeo:CUSTom:SCAN?
```

This query might return

```
:TRIGger:A:VIDeo:CUSTom:SCAN RATE2
```

indicating that the video line rate for the A trigger custom video is set to Rate 2, which is 20 kHz to 23 kHz.

---

## Example 2

`TRIGger:A:VIDeo:CUSTom:SCAN RATE1`

This command sets the scan rate of the A trigger custom video to Rate 1, which is 15 kHz to 20 kHz (standard broadcast rate).

---

# TRIGger:A:VIDeo:CUSTom?

This query only command returns the custom video parameters for the A trigger.

## Group

Trigger

## Syntax

```
TRIGger:A:VIDeo:CUSTom?
```

## Example 1

```
TRIGger:A:VIDeo:CUSTom?
```

This query might return the parameters

```
:TRIGGER:A:VIDEO:CUSTOM:FORMAT INTERLACED;  
SCAN RATE1.
```



---

# TRIGger:A:VIDeo:FIELD

## Description

This command sets or queries the video field or line that the trigger detects.

## Group

Trigger

## Syntax 1

```
TRIGger:A:VIDeo:FIELD  
{ODD|EVEN|FIELD1|FIELD2|ALLFields|ALLLines|NUMERIC}
```

## Syntax 2

```
TRIGger:A:VIDeo:FIELD?
```

## Arguments

### ■ ODD

This argument sets the instrument to trigger on interlaced video odd fields.

### ■ EVEN

This argument sets the instrument to trigger on interlaced video even fields.

### ■ FIELD1

This argument sets the instrument to trigger on interlaced video odd fields (same as ODD).

### ■ FIELD2

This argument sets the instrument to trigger on interlaced video even fields (same as EVEN).

### ■ ALLFields

This argument sets the instrument to trigger on all fields.

### ■ ALLLines

This argument sets the instrument to trigger on all video lines.

### ■ NUMERIC

This argument sets the instrument to trigger on the video signal line specified by the TRIGger:A:VIDeo:LINE command.

---

### **Example 1**

TRIGger:A:VIDeo:FIELD?

This query might return

:TRIGGER:A:VIDEO:FIELD ALLFIELDS

indicating that the A video will trigger on all video fields.

### **Example 2**

TRIGger:A:VIDeo:FIELD EVEN

This command sets the A video trigger so that it will trigger on even fields.

---

# TRIGger:A:VIDeo:HOLdoff:FIELD

## Description

This command sets or queries the video trigger holdoff in terms of video fields.

## Group

Trigger

## Syntax 1

```
TRIGger:A:VIDeo:HOLdoff:FIELD <NR3>
```

## Syntax 2

```
TRIGger:A:VIDeo:HOLdoff:FIELD?
```

## Argument

- <NR3>

This argument is a real number from 0.0 to 8.5 in increments of 0.5. The argument sets the number of fields that the instrument waits before re-arming the video trigger.

## Example 1

```
TRIGger:A:VIDeo:HOLdoff:FIELD?
```

This query might return

```
:TRIGger:A:VIDeo:HOLdoff:FIELD 5
```

indicating that the instrument is set to wait 5 video fields before re-arming the trigger.

## Example 2

```
TRIGger:A:VIDeo:HOLdoff:FIELD 4.5
```

This command sets the instrument to wait 4.5 video fields before re-arming the trigger.

---

# TRIGger:A:VIDeo:LINE

## Description

This command sets or queries the video line number on which the instrument triggers. This command is equivalent to selecting Video Setup from the Trig menu, selecting Line # in the Trigger on box, and setting the line number. Use the TRIGger:A:VIDeo:FIELD command to actually trigger the instrument on the line that you specify with this command.

## Group

Trigger

## Related Commands

TRIGger:A:VIDeo:FIELD

## Syntax 1

TRIGger:A:VIDeo:LINE <NR1>

## Syntax 2

TRIGger:A:VIDeo:LINE?

## Argument

- <NR1>

This argument is an integer that sets the video line number on which the instrument triggers. The following table lists the valid choices, depending on the active video standard.

Video Line Numbering Ranges	
Video Standard	Line Number Range
CUSTOM	4–3000
NTSC	1–263 (odd) and 264–525 (even)
PAL	1–625
SECAM	1–625
HD480P60	1–520
HD720P60	1–750
HD1080I50	1–11235
HD1080I60	1–1125
HD1080P24	1–1125
HD1080P25	1–1125
HD1080SF24	1–1125

---

### **Example 1**

`TRIGger:A:VIDeo:LINE?`

This query might return

`:TRIGger:A:VIDeo:LINE 10`

indicating that the instrument is set to trigger on line 10.

### **Example 2**

`TRIGger:A:VIDeo:LINE 23`

This command sets the instrument to trigger on the line 23.

---

# TRIGger:A:VIDeo:POLarity

## Description

This command sets or queries the polarity of the A video trigger.

## Group

Trigger

## Syntax 1

```
TRIGger:A:VIDeo:POLarity {INVERTed|NORMAL}
```

## Syntax 2

```
TRIGger:A:VIDeo:POLarity?
```

## Argument

- INVERTed

This argument sets the instrument to trigger on a positive video sync pulse.

- NORMAL

This argument sets the instrument to trigger on a negative video sync pulse.

## Example 1

```
TRIGger:A:VIDeo:POLarity?
```

This query might return

```
:TRIGger:A:VIDeo:POLARITY INVERTED
```

indicating that the instrument is set to trigger on a positive video sync pulse.

## Example 2

```
TRIGger:A:VIDeo:POLarity NORMAL
```

This command sets the instrument to trigger on a negative video pulse.

---

# TRIGger:A:VIDeo:SCAN

## Description

This command sets or queries the video trigger horizontal line scan rate. This command is for compatibility with earlier instruments in the TDS series. This command is the same as the TRIGger:A:VIDeo:CUS-Tom:SCAN command.

## Group

Trigger

## Related Commands

### Syntax 1

```
TRIGger:A:VIDeo:SCAN  
{RATE1 | RATE2 | RATE3 | RATE4 | RATE5}
```

### Syntax 2

```
TRIGger:A:VIDeo:SCAN?
```

## Arguments

#### ■ RATE1

This argument sets the range of the video line scan rate to 15 kHz through 20 kHz. This is the standard broadcast rate.

#### ■ RATE2

This argument sets the range of the video line scan rate to 20 kHz through 25 kHz.

#### ■ RATE3

This argument sets the range of the video line scan rate to 25 kHz through 35 kHz.

#### ■ RATE4

This argument sets the range of the video line scan rate to 35 kHz through 50 kHz.

#### ■ RATE5

This argument sets the range of the video line scan rate to 50 kHz through 65 kHz.

## Example 1

```
TRIGger:A:VIDeo:SCAN?
```

This query might return

```
:TRIGger:A:VIDeo:SCAN RATE2
```

indicating that the video line rate for the A trigger is set to Rate 2, which is 20 kHz to 23 kHz.

---

## Example 2

`TRIGger:A:VIDeo:SCAN RATE1`

This command sets the scan rate of the A video trigger to Rate 1, which is 15 kHz to 20 kHz (standard broadcast rate).



---

# TRIGger:A:VIDeo:SOUrce

## Description

This command sets or queries the source for the A video trigger. This command is equivalent to selecting Video Setup from the Trig menu and selecting a channel from the Source drop-down menu.

## Group

Trigger

## Syntax 1

```
TRIGger:A:VIDeo:SOUrce {CH<x>}
```

## Syntax 2

```
TRIGger:A:VIDeo:SOUrce?
```

## Arguments

- CH<x>

This argument specifies one of the input channels of the instrument as the A video trigger. The value of x ranges from 1 through 4 for four-channel instruments or 1 through 2 for two-channel instruments.

## Example 1

```
TRIGger:A:VIDeo:SOUrce?
```

This query might return

```
:TRIGger:A:VIDeo:SOUrce CH2
```

indicating that the source for the A video trigger is set to Channel 2.

## Example 2

```
TRIGger:A:VIDeo:SOUrce CH1
```

This command sets the source for A video trigger to Channel 1.

---

# TRIGger:A:VIDeo:STANdard

## Description

This command sets or queries the video standard.

## Group

Trigger

## Syntax 1

```
TRIGger:A:VIDeo:STANdard {CUSTom|NTSc|PAL|  
SCAM|HD480P60|HD720P60|HD1080I50|HD1080I60|  
HD1080P24|HD1080P25|HD1080SF24}
```

## Syntax 2

```
TRIGger:A:VIDeo:SOURce?
```

## Arguments

### ■ CUSTom

This argument sets the instrument to use custom video horizontal scan rate parameters that you set with the TRIGger:A:VIDeo:SCAN command.

### ■ NTSc

This argument sets the instrument to trigger on video signals that meet the NTSC 525/60/2:1 standard (a line rate of 525 lines per frame and a field rate of 60 Hz).

### ■ PAL

This argument sets the instrument to trigger on video signals that meet the NTSC 625/50/2:1 standard (a line rate of 625 lines per frame and a field rate of 50 Hz).

### ■ SECAM

This argument sets the instrument to trigger on video signals that meet the SECAM standard.

### ■ HD480P60

This argument sets the instrument to trigger on the HDTV 480/60 progressive format.

### ■ HD720P60

This argument sets the instrument to trigger on the HDTV 720/60 progressive format.

### ■ HD1080I50

This argument sets the instrument to trigger on HDTV 1080/50 interlaced format.

### ■ HD1080I60

---

This argument sets the instrument to trigger on HDTV 1080/60 interlaced format.

- HD1080P24

This argument sets the instrument to trigger on HDTV 1080/24 progressive format.

- HD1080P25

This argument sets the instrument to trigger on HDTV 1080/25 progressive format.

- HD1080SF24

This argument sets the instrument to trigger on HDTV 1080/24 segmented frame format.

### Example 1

```
TRIGger:A:VIDeo:STANdard?
```

This query might return

```
:TRIGger:A:VIDeo:SOURCE CH2
```

indicating that the source for the A video trigger is set to Channel 2.

### Example 2

```
TRIGger:A:VIDeo:STANdard NTSC
```

This command sets the instrument to trigger on NTSC-standard video signals.

---

# TRIGger:A:VIDeo?

This query only command returns the video parameters for the A trigger.

## Group

Trigger

## Syntax

TRIGger:A:VIDeo?

## Example 1

TRIGger:A:VIDeo?

This query might return

```
:TRIGGER:A:VIDEO:CUSTOM:FORMAT INTERLACED;  
SCAN RATE1;  
:TRIGGER:A:VIDEO:FIELD ALLFIELDS;  
HOLDOFF:FIELD 1.0000;TIME 20.0000E-3;  
:TRIGGER:A:VIDEO:LINE 1;SCAN RATE1;  
SOURCE CH1;STANDARD NTSC.
```

---

# TRIGger:B

## Description

This command sets the B trigger level to 50% of minimum and maximum. The query form of this command returns the B trigger parameters. This command is similar to selecting B Event (Delayed) Trigger Setup from the Trig menu and then viewing the current setups.

## Group

Trigger

## Related Commands

TRIGger:A

## Syntax 1

TRIGger:B SETLevel

## Syntax 2

TRIGger:B?

## Argument

- SETLevel

This sets the B trigger level to 50% of MIN and MAX.

## Example 1

TRIGger:B?

This query might return the following B trigger parameters:

```
:TRIGGER:B:STATE 0;TYPE EDGE;  
LEVEL -220.0000E-3;BY TIME;EDGE:SOURCE  
CH1;SLOPE RISE;COUPLING DC;  
:TRIGGER:B:TIME 16.0000E-9;EVENTS:COUNT 2
```

## Example 2

TRIGger:B SETLevel

This command sets the B trigger level to 50% of MIN and MAX.

---

# TRIGger:B:BY

## Description

This command selects or returns whether the B trigger occurs after a specified number of events or a specified period of time after the A trigger. This is equivalent to selecting B Event (Delayed) Trigger Setup from the Trig menu, selecting the A→B Seq tab, and then choosing Trig After Time or Trig on nth event.

**Note:** The traditional Runs After functionality is now served by the Horizontal Delay function. For details, see the HORIZontal[:MAIN]:DELay:MODE and HORIZontal[:MAIN]:DELay:TIME commands.

## Group

Trigger

## Related Commands

TRIGger:B:EVENTS:COUNT, TRIGger:B:TIME, HORIZontal[:MAIN]:DELay:MODE, HORIZontal[:MAIN]:DELay:TIME

## Syntax 1

```
TRIGger:B:BY {EVENTS | TIME}
```

## Syntax 2

```
TRIGger:B:BY?
```

## Arguments

### ■ EVENTS

This sets the B trigger to take place following a set number of trigger events after the A trigger occurs. The number of events is specified by TRIGger:B:EVENTS:COUNT.

### ■ TIME

This sets the B trigger to occur a set time after the A trigger event. The time period is specified by TRIGger:B:TIME.

## Example 1

```
TRIGger:B:BY?
```

This query might return

```
:TRIGGER:B:BY EVENTS
```

indicating that the B trigger takes place following a set number of trigger events after the A trigger occurs.

---

## **Example 2**

**TRIGger:B:BY TIME**

This command sets the B trigger to occur at a set time after the A trigger event.

---

# TRIGger:B:EDGE:COUPLing

## Description

This command sets or queries the type of coupling for the B trigger. This command is equivalent to selecting B Event (Delayed) Trigger Setup from the Trig menu and choosing the setting from the Coupling drop-down list.

## Group

Trigger

## Related Commands

TRIGger:B:EDGE?

## Syntax 1

```
TRIGger:B:EDGE:COUPLing {DC|ATRIGger|NOISerej}
```

## Syntax 2

```
TRIGger:B:EDGE:COUPLing?
```

## Arguments

- ATRIGger

This sets the B trigger coupling to match the setting on the A trigger.

- DC

This selects DC trigger coupling.

- NOISerej

This selects DC low sensitivity.

## Example 1

```
TRIGger:B:EDGE:COUPLing?
```

This query might return

```
:TRIGGER:B:EDGE:COUPLING ATRIGGER
```

for the B trigger coupling.

## Example 2

```
TRIGger:B:EDGE:COUPLing DC
```

This command selects DC for the B trigger coupling.



---

# TRIGger:B:EDGE:SLOpe

## Description

This command sets or queries the slope for the B trigger. This command is equivalent to selecting B Event (Delayed) Trigger Setup from the Trig menu and choosing the Slope.

## Group

Trigger

## Related Commands

TRIGger:B:EDGE?

## Syntax 1

```
TRIGger:B:EDGE:SLOpe {RISe|FALL}
```

## Syntax 2

```
TRIGger:B:EDGE:SLOpe?
```

## Arguments

- RISe

This argument specifies the trigger on the rising or positive edge of a signal.

- FALL

This argument specifies the trigger on the falling or negative edge of a signal.

## Example 1

```
TRIGger:B:EDGE:SLOpe?
```

This query might return

```
:TRIGGER:B:EDGE:SLOPE RISE
```

indicating that the B edge trigger occurs on the rising slope.

## Example 2

```
TRIGger:B:EDGE:SLOpe FALL
```

This command sets the B edge trigger to occur on the falling slope.

---

# TRIGger:B:EDGE:SOURce

## Description

This command sets or queries the source for the B trigger. This command is equivalent to selecting B Event (Delayed) Trigger Setup from the Trig menu and choosing the desired setting from the Source drop-down list.

## Group

Trigger

## Related Commands

TRIGger:B:EDGE?

## Syntax 1

```
TRIGger:B:EDGE:SOURce {AUXiliary|CH<x>}
```

## Syntax 2

```
TRIGger:B:EDGE:SOURce?
```

## Arguments

### ■ AUXiliary

This specifies an external trigger (using the Auxiliary Trigger Input connector, located on the rear panel of the instrument) as the B trigger source.

### ■ CH<x>

This specifies one of the input channels as the B trigger source. Input channels are specified by x, which can range from 1 through 4 for four-channel instruments or 1 through 2 for two-channel instruments.

## Example 1

```
TRIGger:B:EDGE:SOURce?
```

This query might return

```
:TRIGGER:B:EDGE:SOURCE CH1
```

indicating that the current input source for the B trigger is channel 1.

## Example 2

```
TRIGger:B:EDGE:SOURce CH4
```

This command sets channel 4 as the input source for the B trigger.

---

# TRIGger:B:EDGE?

## Description

This query-only command returns the source, slope, and coupling for B trigger. This command is equivalent to selecting B Event (Delayed) Trigger Setup from the Trig menu and viewing the current Source, Slope, and Coupling settings.

## Group

Trigger

## Related Commands

TRIGger:B:EDGE:COUPling, TRIGger:B:EDGE:SLOpe,  
TRIGger:B:EDGE:SOUrce

## Syntax

TRIGger:B:EDGE?

## Example

TRIGger:B:EDGE?

This query might return

```
:TRIGGER:B:EDGE:SOURCE CH1;  
SLOPE RISE;COUPLING DC
```

---

# TRIGger:B:EVENTS:COUNT

## Description

This command sets or queries the number of events that must occur before the B trigger (when TRIG:DELay:BY is set to EVENTS). This command is equivalent to selecting B Event (Delayed) Trigger Setup from the Trig menu, selecting the A→B Seq tab, choosing Trig on nth event, and setting the desired Trig Event value.

## Group

Trigger

## Related Commands

TRIGger:B:EVENTS?

## Syntax 1

```
TRIGger:B:EVENTS:COUNT <NR1>
```

## Syntax 2

```
TRIGger:B:EVENTS:COUNT?
```

## Argument

- <NR1>

This is the number of B trigger events, which can range from 1 to 10,000,000.

## Example 1

```
TRIGger:B:EVENTS:COUNT?
```

This query might return

```
:TRIGGER:B:EVENTS:COUNT 2
```

indicating that two events must occur after the A trigger before the B trigger can occur.

## Example 2

```
TRIGger:B:EVENTS:COUNT 4
```

This command specifies that the B trigger will occur four trigger events after the A trigger.

---

# TRIGger:B:EVENTS?

## Description

This query-only command returns the current B trigger events parameter. This command is equivalent to selecting B Event (Delayed) Trigger Setup from the Trig menu, selecting the A→B Seq tab, choosing Trig on nth event, and viewing the Trig Event setting.

## Group

Trigger

## Related Commands

TRIGger:B:EVENTS:COUNT

## Syntax

TRIGger:B:EVENTS?

## Example

TRIGger:B:EVENTS?

This query might return

```
:TRIGGER:B:EVENTS:COUNT 2
```

indicating that 2 events must occur before the B trigger occurs.

---

# TRIGger:B:LEVel

## Description

This command sets or queries the level for the B trigger. This command is equivalent to selecting B Event (Delayed) Trigger Setup from the Trig menu, selecting the A→B Seq tab and setting the B Trig Level voltage.

## Group

Trigger

## Related Commands

TRIGger:A:LEVEL, TRIGger:B, TRIGger:B:EDGE:SOUrce

## Syntax 1

```
TRIGger:B:LEVel {ECL|TTL|<NR3>}
```

## Syntax 2

```
TRIGger:B:LEVel?
```

## Arguments

- ECL

This specifies a preset ECL level of -1.3 V.

- TTL

This specifies a preset TTL level of 1.4 V.

- <NR3>

This is the B trigger level, in volts.

## Example 1

```
TRIGger:B:LEVel?
```

This query might return

```
:TRIGGER:B:LEVEL 173.0000E-03
```

indicating that the B trigger level is currently set at 173 mV.

## Example 2

```
TRIGger:B:LEVel ECL
```

This command sets the B trigger level to -1.3 V.

---

# TRIGger:B:STATE

## Description

This command sets or queries the state of B trigger activity. If the B trigger state is on, the B trigger is part of the triggering sequence. If the B trigger state is off, then only the A trigger causes the trigger event.

## Group

Trigger

## Related Commands

TRIGger:A:MODE

## Syntax 1

```
TRIGger:B:STATE {ON|OFF|<NR1>}
```

## Syntax 2

```
TRIGger:B:STATE?
```

## Arguments

- ON

This argument indicates that the B trigger is active and in causes trigger events conjunction with the A trigger.

- OFF

This argument indicates that only the A trigger causes trigger events.

- <NR1>

A 0 turns off the B trigger; any other value activates the B trigger.

## Example 1

```
TRIGger:B:STATE?
```

This query might return

```
:TRIGGER:B:STATE 0
```

indicating that the B trigger is inactive and that only the A trigger causes trigger events.

## Example 2

```
TRIGger:B:STATE ON
```

This command sets the B trigger to active, making it capable of causing trigger events.

---

# TRIGger:B:TIME

## Description

This command sets or queries B trigger delay time. The B Trigger time applies only if TRIGger:B:BY is set to TIME. This command is equivalent to selecting B Event (Delayed) Trigger Setup from the Trig menu, choosing the A→B Seq tab, and setting Trig Delay.

## Group

Trigger

## Related Commands

TRIGger:B:BY, TRIGger:B:EVENTS:COUNT

## Syntax 1

```
TRIGger:B:TIME <NR3>
```

## Syntax 2

```
TRIGger:B:TIME?
```

## Argument

- <NR3>

This is the B trigger delay time in seconds.

## Example 1

```
TRIGger:B:TIME?
```

This query might return

```
:TRIGGER:B:TIME 16.0000E-9
```

indicating that the B trigger time is set to 16 ns.

## Example 2

```
TRIGger:B:TIME 4E-6
```

This command sets the B trigger delay time to 4  $\mu$ s.



---

# TRIGger:B:TYPe

## Description

This command sets or queries the type of B trigger. This command is equivalent to selecting B Event (Delayed) Trigger Setup from the Trig menu and choosing Edge.

## Group

Trigger

## Related Commands

TRIGger:A:TYPe

## Syntax 1

TRIGger:B:TYPe EDGE

## Syntax 2

TRIGger:B:TYPe?

## Argument

- EDGE

This sets the B trigger type to edge.

## Example 1

TRIGger:B:TYPe?

This query will return :TRIGGER:B:TYPe EDGE

## Example 2

TRIGger:B:TYPe EDGE

This command sets the B trigger type to edge.

---

# TRIGger:STATE?

## Description

This query-only command returns the current state of the triggering system. This command is equivalent to viewing the trigger status LEDs on the instrument front panel.

## Group

Trigger

## Related Commands

TRIGger:A:MODE

## Syntax

TRIGger:STATE?

## Outputs

- **ARMed**

This indicates that the instrument is acquiring pretrigger information. All triggers are ignored when TRIGger:STATE is arming.

- **AUTO**

This indicates that the instrument is in the automatic mode and acquires data even in the absence of a trigger.

- **DPO**

This indicates that the instrument is in DPO mode.

- **PARTial**

This indicates that the A trigger has occurred and the instrument is waiting for the B trigger to occur.

- **REAdy**

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

- **SAVe**

This indicates that the instrument is in save mode and is not acquiring data.

- **TRIGger**

This indicates that the instrument triggered and is acquiring the post trigger information.

---

### **Example**

TRIGger:STATE?

This query might return

:TRIGGER:STATE ARMED

indicating that the pretrigger data is being acquired.

---

# UNLock

## Description

This command (no query form) unlocks the front panel. The command is equivalent to LOCK NONE.

**Note:** If the instrument is in the Remote With Lockout State (RWLS), the UNLock command has no effect. For more information, see the ANSI-IEEE Std 488.1-1987 Standard Digital Interface for Programmable Instrumentation, section 2.8.3 on RL State Descriptions.

## Group

Miscellaneous

## Related Commands

LOCK

## Syntax

UNLock ALL

## Argument

- ALL

This specifies that all front-panel buttons and knobs are unlocked.

## Example

UNLock ALL

This command unlocks all front-panel buttons and knobs.

---

# VERBose

## Description

This command sets or queries the Verbose state that controls the length of keywords on query responses. Keywords can be both headers and arguments.

**Note:** This command does not affect IEEE Std 488.2-1987 Common Commands (those starting with an asterisk). However, this command does make a corresponding change in the Response Header Enable State of the opposite interface (physical or virtual GPIB interface). Refer to Introduction for more information.

## Group

Miscellaneous

## Related Commands

HEADer, \*LRN?, SET?

## Syntax

```
VERBose {OFF|ON|<NR1>}
```

## Arguments

- OFF

This sets the Verbose state to true, which returns full-length keywords for applicable setting queries.

- ON

This sets the Verbose state to false, which returns minimum-length keywords for applicable setting queries.

- <NR1>

A 0 returns minimum-length keywords for applicable setting queries; any other value returns full-length keywords.

## Example 1

```
VERBose ON
```

This command enables the Verbose state.

## Example 2

```
VERBOSE?
```

This query might return

```
:VERBOSE 0
```

indicating that the Verbose state is disabled.

---

# WFMinpre?

## Description

This query-only command returns the waveform formatting specification to be applied to the next incoming CURVe command data.

## Group

Waveform Transfer

## Related Commands

WFMOutpre?

## Syntax

WFMinpre?

## Example

WFMinpre?

This query might return the waveform formatting as

```
:WFMINPRE:BIT_NR 8;BN_FMT RI;BYT_NR 1;  
BYT_OR MSB;ENCDG BIN;NR_PT 500;PT_FMT Y;  
PT_OFF 0;XINCR 2.0000E-6;XZERO 1.7536E-6;  
XUNIT "s";YMULT 1.0000E-3;YOFF 0.0000;  
YZERO 0.0000;YUNIT "V"
```

---

# WFMInpre:BIT\_Nr

## Description

This command sets or returns the number of bits per binary waveform point for the waveform, as specified by the DATA:DESTINATION command. This specification is only meaningful when WFMInpre:ENCdg is set to BIN.

## Group

Waveform Transfer

## Related Commands

DATA:DESTINATION, WFMInpre:ENCdg, WFMInpre:BYT\_Nr, WFMOutpre:BIT\_Nr

## Syntax 1

```
WFMInpre:BIT_Nr <NR1>
```

## Syntax 2

```
WFMInpre:BIT_Nr?
```

## Argument

- <NR1>

This number of bits per data point can be 8, 16 (RI, RP) or 32 (FP).

## Example 1

```
WFMInpre:BIT_Nr?
```

This query might return

```
:WFMINPRE:BIT_NR 8
```

indicating that incoming RI or RP binary format data uses 8 bits per waveform point.

## Example 2

```
WFMInpre:BIT_Nr 16
```

This command sets the number of bits per waveform point to 16, for incoming RI and RP binary format data.

---

# WFMinpre:BN\_Fmt

## Description

This command sets or queries the format of binary data for incoming waveforms.

## Group

Waveform Transfer

## Related Commands

WFMOutpre:BN\_Fmt

## Syntax 1

```
WFMinpre:BN_Fmt {RI|RP|FP}
```

## Syntax 2

```
WFMinput:BN_Fmt?
```

## Arguments

- RI

This specifies signed integer data point representation.

- RP

This specifies positive integer data point representation.

- FP

This specifies single-precision binary floating point representation.

## Example 1

```
WFMinpre:BN_Fmt?
```

This query might return

```
:WFMINPRE:BN_FMT RI
```

indicating that the incoming data is currently interpreted as signed integers.

## Example 2

```
WFMinpre:BN_Fmt FP
```

This command specifies that incoming data will be interpreted as single-precision binary floating point numbers.



---

# WFMinpre:BYT\_Nr

## Description

This command sets or returns the binary field data width for the first ordered waveform, as specified by the DATA:DESTINATION command. This specification is only meaningful when WFMinpre:ENCdg is set to BIN and WFMinpre:BN\_Fmt is set to either RI or RP.

## Group

Waveform Transfer

## Related Commands

DATA:DESTINATION, WFMinpre:BN\_Fmt, WFMinpre:ENCdg, WFMinpre:BIT\_Nr, WFMinpre:BYT\_Nr

## Syntax 1

```
WFMinpre:BYT_Nr <NR1>
```

## Syntax 2

```
WFMinpre:BYT_Nr?
```

## Argument

- <NR1>

This is the number of bytes per data point and can be 1, 2 (RI, RP) or 4 (FP).

## Example 1

```
WFMinpre:BYT_Nr?
```

This query might return

```
:WFMINPRE:BYT_NR 2
```

indicating that there are 2 bytes per incoming waveform data point.

## Example 2

```
WFMinpre:BYT_Nr 1
```

This command sets the number of bytes per incoming waveform data point to 1, which is the default setting.

---

# WFMInpre:BYT\_Or

## Description

This command sets or queries which byte of binary waveform data is transmitted first for incoming waveform data when data points require more than one byte. This specification only has meaning when WFMInpre:ENCdg is set to BIN and WFMInpre:BYT\_Nr is greater than 1.

## Group

Waveform Transfer

## Related Commands

WFMInpre:ENCdg, WFMInpre:BYT\_Nr, WFMOutpre:BYT\_Or

## Syntax 1

```
WFMInpre:BYT_Or {LSB|MSB}
```

## Syntax 2

```
WFMInpre:BYT_Or?
```

## Arguments

- LSB

This specifies that the least significant byte will be transmitted first.

- MSB

This specifies that the most significant byte will be transmitted first.

## Example 1

```
WFMInpre:BYT_Or?
```

This query might return

```
:WFMINPRE:BYT_OR LSB
```

indicating that the least significant incoming CURVe data byte will be transmitted first.

## Example 2

```
WFMInpre:BYT_Or MSB
```

This command sets the most significant incoming byte of incoming waveform data to be transmitted first.

---

# WFMinpre:ENCdg

## Description

This command sets or queries the type of encoding for incoming waveform data.

## Group

Waveform Transfer

## Related Commands

WFMOutpre:ENCdg

## Syntax 1

```
WFMinpre:ENCdg {ASC|BIN}
```

## Syntax 2

```
WFMinpre:ENCdg?
```

## Arguments

- ASC

This specifies that the incoming data is in ASCII format.

- BIN

This specifies that the incoming data is in a binary format whose further interpretation requires knowledge of BYT\_NR, BIT\_NR, BN\_FMT, and BYT\_OR.

## Example 1

```
WFMinpre:ENCdg?
```

This query might return

```
:WFMINPRE:ENCDG BIN
```

indicating that the incoming waveform data is in binary format.

## Example 2

```
WFMinpre:ENCdg ASC
```

This command sets the format of incoming waveform data to ASCII format.

---

# WFMinpre:NR\_FR?

## Description

This query-only command returns one frame, indicating the number of frames in the transmitted waveform record.

## Group

Waveform Transfer

## Related Commands

CURVe, DATA, DATA:START, DATA:STOP, SAVE:WAVEform, SAVE:WAVEform:FILEFormat, WFMinpre:NR\_Pt, WFMinpre:NR\_FR?, WFMinpre:NR\_Pt?

## Syntax

```
WFMinpre:NR_FR?
```

## Example

```
WFMinpre:NR_FR?
```

This query will return

```
:WFMINPRE:NR_FR 1
```

indicating that there is 1 data frame in the transmitted waveform record.

---

# WFMinpre:NR\_Pt

## Description

This command sets or returns the number of data points that are in the transmitted waveform record.

## Group

Waveform Transfer

## Related Commands

CURVe , DATA, DATA:START, DATA:STOP, SAVE:WAVEform,  
SAVE:WAVEform:FILEFormat, WFMinpre:NR\_FR?,  
WFMinpre:NR\_FR?, WFMinpre:NR\_Pt?,

## Syntax 1

```
WFMinpre:NR_Pt <NR1>
```

## Syntax 2

```
WFMinpre:NR_Pt?
```

## Argument

- <NR1>

If WFMinpre:PT\_Fmt is set to Y, this is the number of data points; if WFMinpre:PT\_Fmt is set to ENV, this is the number of min-max pairs.

## Example 1

```
WFMinpre:NR_Pt?
```

This query might return

```
:WFMINPRE:NR_PT 8000
```

indicating that there are 8000 data points in the transmitted waveform record.

## Example 2

```
WFMinpre:NR_Pt 5000
```

This command specifies that 5000 data points will be transmitted.

---

# WFMinpre:PT\_Fmt

## Description

This command sets or queries the point format of the incoming waveform data. Regardless of the argument used, the scale, offset, and so on are interpreted similarly. When ENV is used, waveform data is interpreted over the min-max pair; when Y is used, it is interpreted over a single point.

## Group

Waveform Transfer

## Related Commands

WFMOutpre:PT\_Fmt

## Syntax 1

```
WFMinpre:PT_Fmt {ENV|Y}
```

## Syntax 2

```
WFMinpre:PT_Fmt?
```

## Arguments

### ■ ENV

This specifies that the waveform is transmitted in envelope mode as maximum and minimum point pairs. Only Y values are explicitly transmitted. Absolute coordinates are given by:

$$X_n = XZ_{Ero} + XIN_{cr} (n - PT_{Off})$$

$$Y_{nmax} = YZ_{Ero} + YMUlt (y_{nmax} - YOff)$$

$$Y_{nmin} = YZ_{Ero} + YMUlt (y_{nmin} - YOff)$$

### ■ Y

This specifies a normal waveform where one ASCII or binary data point is transmitted for each point in the waveform record. Only Y values are explicitly transmitted. Absolute coordinates are given by:

$$X_n = XZ_{Ero} + XIN_{cr} (N - PT_{Off})$$

$$Y_n = YZ_{Ero} + YMUlt (Y_n - YOff)$$

## Example 1

```
WFMinpre:PT_Fmt ENV
```

This command sets the incoming waveform data point format to enveloped.

---

## Example 2

WFMinpre:PT\_Fmt?

This query might return

:WFMINPRE:PT\_FMT ENV

indicating that the waveform is transmitted as maximum and minimum point pairs.

---

# WFMInpre:PT\_Off

## Description

This command specifies or returns the trigger point within the waveform record for the reference waveform specified by the DATA:DESTINATION command.

## Group

Waveform Transfer

## Related Commands

DATA:DESTINATION, DATA:START, WFMOutpre:PT\_Off

## Syntax 1

```
WFMInpre:PT_Off <NR1>
```

## Syntax 2

```
WFMInpre:PT_Off?
```

## Argument

- <NR1>

This is (record length - 1) to record length and is the position of the data point immediately following the actual trigger. <NR1> is expressed relative to DATA:START.

## Example 1

```
WFMInpre:PT_Off 0
```

This command specifies that the trigger point is the first point in the waveform record, which is the default.

## Example 2

```
WFMInpre:PT_Off?
```

This query might return

```
:WFMINPRE:PT_OFF 0
```

indicating that the incoming waveform trigger point is the first point in the waveform record.



---

# WFMInpre:WFId

## Description

This command (no query form) accepts but ignores the argument. This command is provided only to allow a waveform extracted from the instrument to be easily imported.

## Group

Waveform Transfer

## Related Commands

DATA:DESTination, WFMInpre:BN\_Fmt, WFMInpre:ENCdg,  
WFMOutpre:WFid

## Syntax

```
WFMInpre:WFId <String>
```

## Argument

- <String>

This must be a valid IEEE-488.2 string (but the contents are ignored).

## Example

```
WFMInpre:WFId "Ch1, DC coupling, 2.000V/div,  
400.0ns/div, 500 points, Sample mode"
```

This is a syntactically correct command.

---

# WFMinpre:XINcr

## Description

This command sets or queries the horizontal interval between incoming waveform points in units specified by WFMinpre:XUNit.

## Group

Waveform Transfer

## Related Commands

WFMinpre:XUNit, WFMinpre:XINcr

## Syntax 1

```
WFMinpre:XINcr <NR3>
```

## Syntax 2

```
WFMinpre:XINcr?
```

## Argument

- <NR3>

This is the horizontal interval representation.

## Example 1

```
WFMinpre:XINcr?
```

This query might return

```
:WFMINPRE:XINCR 1.0000E-3
```

indicating that if WFMinpre:XUNit is set to "s", there is a 1 ms interval between incoming waveform points.

## Example 2

```
WFMinpre:XINcr 3E-3
```

This command sets the interval between Incoming waveform points to 3 ms.

---

# WFMinpre:XUNit

## Description

This command sets or returns the horizontal units of the incoming waveform.

## Group

Waveform Transfer

## Related Commands

WFMOutpre:XUNit

## Syntax 1

```
WFMinpre:XUNit <String>
```

## Syntax 2

```
WFMinpre:XUNit?
```

## Argument

- <String>

This contains a maximum of three alpha characters that represent the horizontal unit of measure for the incoming waveform.

## Example 1

```
WFMinpre:XUNit?
```

This query might return

```
:WFMINPRE:XUNIT "s"
```

indicating that the horizontal units for the incoming waveform are seconds.

## Example 2

```
WFMinpre:XUNit "Hz"
```

This command specifies that the horizontal units for the incoming waveform are hertz.

---

# WFMinpre:XZEro

## Description

This command sets or queries the sub-sample time between the trigger sample (designated by PT\_OFF) and the occurrence of the actual trigger on the incoming waveform. This value is used to compute TTOFF for the incoming waveform, and is expressed in terms of WFMinpre:XUNit.

## Group

Waveform Transfer

## Related Commands

WFMinpre:PT\_Off, WFMinpre:XINcr, WFMinpre:XUnit, WFMinpre:XZEro

## Syntax 1

```
WFMinpre:XZEro <NR3>
```

## Syntax 2

```
WFMinpre:XZEro?
```

## Argument

- <NR3>

This argument is a floating point value that ranges from -WFMinpre:XINcr to 0.

## Example 1

```
WFMinpre:XZEro?
```

This query might return

```
:WFMINPRE:XZEro 7.5000E-6
```

indicating that the trigger occurs 7.5  $\mu$ s before the sample designated by WFMinpre:PT\_Off.

## Example 2

```
WFMinpre:XZEro 5.7E-6
```

This command specifies that the trigger actually occurred 5.7  $\mu$ s before the sample designated by WFMinpre:PT\_Off.

---

# WFMInpre:YMUlt

## Description

This command sets or queries the vertical scale factor (in units/digitizing level) for the reference waveform, specified by DATA:DESTINATION, upon a CURVe command.

## Group

Waveform Transfer

## Related Commands

DATA:DESTINATION, WFMInpre:BYT\_Nr, WFMInpre:YUNit

## Syntax 1

```
WFMInpre:YMUlt <NR3>
```

## Syntax 2

```
WFMInpre:UMUlt?
```

## Argument

- <NR3>

This is the vertical scale factor per digitizing level of the incoming waveform points.

## Example 1

```
WFMInpre:YMUlt?
```

This query might return

```
:WFMINPRE:YMULT 40.0000E-3
```

indicating that the vertical scale is 40 mV/digitizing level (1V/div).

## Example 2

```
WFMInpre:YMUlt 20E-3
```

This command specifies that (if WFMInpre:YUNit is "V" and WFMInpre:BYT\_Nr is 1), the vertical scale is 20 mV/digitizing level (500 mV/div).

---

# WFMinpre:YOFF

## Description

This command sets or queries the vertical position of the incoming waveform in digitizing levels. Variations in this number are analogous to changing the vertical position of the waveform. For those formats in which WFMinpre:BYT\_Nr is important (all nonfloating point formats), this command must take the location of the binary point implied by BYT\_NR into consideration.

## Group

Waveform Transfer

## Related Commands

WFMinpre:BYT\_Nr, WFMinpre:YMULT, WFMinpre:YOFF

## Syntax 1

```
WFMinpre:YOFF <NR3>
```

## Syntax 2

```
WFMinpre:YOFF?
```

## Argument

- <NR3>

This is the vertical offset in digitizing levels.

## Example 1

```
WFMinpre:YOFF?
```

This query might return

```
:WFMINPRE:YOFF 25
```

indicating the vertical position of the incoming waveform in digitizing levels.

## Example 2

```
WFMinpre:YOFF 50
```

This command specifies that the zero reference point for the incoming waveform is 50 digitizing levels (2 divisions) above the center of the data range.

---

# WFMInpre:YUNit

## Description

This command sets or queries the vertical units of the incoming waveform.

## Group

Waveform Transfer

## Related Commands

WFMOutpre:YUNit

## Syntax 1

```
WFMInpre:YUNit <String>
```

## Syntax 2

```
WFMInpre:YUNit?
```

## Argument

- <String>

This contains a maximum of three alpha characters that represent the vertical unit of measure for the incoming waveform.

## Example 1

```
WFMInpre:YUNit?
```

This query might return

```
:WFMINPRE:YUNIT "s"
```

indicating the vertical units for the incoming waveform are seconds.

## Example 2

```
WFMInpre:YUNit "Pa"
```

This command specifies that the vertical units for the incoming waveform are Pascal.

---

# WFMInpre:YZEro

## Description

This command sets or queries the offset of the incoming waveform in units specified by WFMInpre:YUNit. Variations in this number are analogous to changing the vertical offset of the waveform.

## Group

Waveform Transfer

## Related Commands

WFMInpre:YUNit, WFMOutpre:YZEro

## Syntax 1

```
WFMInpre:YZEro <NR3>
```

## Syntax 2

```
WFMInpre:YZEro?
```

## Argument

- <NR3>

This is the offset in YUNits.

## Example 1

```
WFMInpre:YZEro?
```

This query might return

```
:WFMINPRE:YZEro 7.5000E-6
```

indicating that the zero reference for the incoming waveform is 7.5  $\mu\text{V}$  below the center of the data range (given that WFMInpre:YUNit is set to V).

## Example 2

```
WFMInpre:YZEro 1.5E+0
```

This command specifies that the zero reference point for the incoming waveform is 1.5 V below the center of the data range (given that WFMInpre:YUNit is set to V).



---

# WFMOutpre?

## Description

This query-only command returns the waveform formatting data for the waveform specified by the DATA:SOURce command. The preamble components are considered to be of two types; formatting and interpretation. The formatting components are: ENCDg, BN\_Fmt, BYT\_Or, BYT\_Nr, BIT\_Nr. The interpretation components are derived from the DATA:SOURce specified waveform.

## Group

Waveform Transfer

## Syntax

WFMOutpre?

## Example

WFMOutpre:?

This query might return the waveform formatting data as:

```
:WFMOUTPRE:BIT_NR 8;BN_FMT RI;BYT_NR 1;
BYT_OR MSB;ENCDG BIN;NR_PT 500;
PT_FMT Y;PT_ORDER LINEAR;PT_OFF 0;
XINCR 8.0000E-9;XZERO 4.8794E-9;
XUNIT "s";YMULT -2000.0000E-3;
YOFF -4999.9995E-3;YZERO 0.0000;YUNIT "V";
WFID "Ch1, DC coupling, 2.000V/div, 400.0ns/
div,500 points, Sample mode"
```

---

# WFMOutpre:BIT\_Nr

## Description

This command sets and returns the number of bits per waveform point that outgoing waveforms contain, as specified by the DATA:SOURce command. Note that values will be constrained according to the underlying waveform data. This specification is only meaningful when WFMOutpre:ENCdg is set to BIN and WFMOutpre:BN\_Fmt is set to either RI or RP.

## Group

Waveform Transfer

## Related Commands

DATA:SOURce, WFMOutpre:BN\_Fmt, WFMOutpre:ENCdg

## Syntax 1

```
WFMOutpre:BIT_Nr <NR1>
```

## Syntax 2

```
WFMOutpre:BIT_Nr?
```

## Argument

- <NR1>

This number of bits per data point can be 8, 16, 32 or 64.

## Example 1

```
WFMOutpre:BIT_Nr?
```

This query might return

```
:WFMOUTPRE:BIT_NR 8
```

indicating that outgoing RI or RP binary format data uses 8 bits per waveform point.

## Example 2

```
WFMOutpre:BIT_Nr 16
```

This command sets the number of bits per waveform point to 16 for incoming RI and RP binary format data.

---

# WFMOutpre:BN\_Fmt

## Description

This command sets or queries the format of binary data for outgoing waveforms specified by the DATA:SOURce command.

## Group

Waveform Transfer

## Related Commands

DATA:SOURce

## Syntax 1

```
WFMOutpre:BN_Fmt {RI|RP|FP}
```

## Syntax 2

```
WFMOutput:BN_Fmt?
```

## Arguments

- RI

This specifies signed integer data point representation.

- RP

This specifies positive integer data point representation.

- FP

This specifies single-precision binary floating point data point representation.

## Example 1

```
WFMOutpre:BN_Fmt?
```

This query might return

```
:WFMOUTPRE:BN_FMT RI
```

indicating that the outgoing waveform data is currently in signed integer format.

## Example 2

```
WFMOutpre:BN_Fmt FP
```

This command specifies that outgoing waveform data will be in single-precision binary floating point format.

---

# WFMOutpre:BYT\_Nr

## Description

This command sets or returns the binary field data width for the waveform specified by the DATA:SOURce command. Note that values will be constrained according to the underlying waveform data. This specification is only meaningful when WFMOutpre:ENCdg is set to BIN, and WFMOutpre:BN\_Fmt is set to either RI or RP.

## Group

Waveform Transfer

## Related Commands

DATA:SOURce, WFMOutpre:BN\_Fmt, WFMOutpre:ENCdg

## Syntax 1

```
WFMOutpre:BYT_Nr <NR1>
```

## Syntax 2

```
WFMOutpre:BYT_Nr?
```

## Argument

- <NR1>

This is the number of bytes per data point and can be 1, 2, 4 or 8. A value of 1 or 2 bytes per waveform point indicates channel data; 4 bytes per waveform point indicate math data; 8 bytes per waveform point indicate pixel map (DPO) data.

## Example 1

```
WFMOutpre:BYT_Nr?
```

This query might return

```
:WFMOUTPRE:BYT_NR 2
```

indicating that there are 2 bytes per outgoing waveform data point.

## Example 2

```
WFMOutpre:BYT_Nr 1
```

This command sets the number of bytes per outgoing waveform data point to 1, which is the default setting.

---

# WFMOutpre:BYT\_Or

## Description

This command sets or queries which byte of binary waveform data is transmitted first, during a waveform data transfer, when data points require more than one byte. This specification only has meaning when WFMOutpre:ENCdg is set to BIN.

## Group

Waveform Transfer

## Related Commands

WFMOutpre:ENCdg

## Syntax 1

```
WFMOutpre:BYT_Or {LSB|MSB}
```

## Syntax 2

```
WFMOutpre:BYT_Or?
```

## Arguments

- LSB

This specifies that the least significant byte will be transmitted first.

- MSB

This specifies that the most significant byte will be transmitted first.

## Example 1

```
WFMOutpre:BYT_Or?
```

This query might return

```
:WFMOUTPRE:BYT_OR LSB
```

indicating that the least significant data byte will be transmitted first.

## Example 2

```
WFMOutpre:BYT_Or MSB
```

This command sets the most significant outgoing byte of waveform data to be transmitted first.

---

# WFMOutpre:ENCdg

## Description

This command sets and queries the type of encoding for outgoing waveforms.

## Group

Waveform Transfer

## Related Commands

DATA:ENCdg, WFMOutpre:BYT\_Nr, WFMOutpre:BYT\_Or, WFMOutpre:BIT\_Nr, WFMOutpre:BN\_Fmt

## Syntax 1

```
WFMOutpre:ENCdg {ASC|BIN}
```

## Syntax 2

```
WFMOutpre:ENCdg?
```

## Argument

### ■ ASC

This specifies that the outgoing data is to be in ASCII format. Waveforms internally stored as integers will be sent as <NR1> numbers, while those stored as floating point will be sent as <NR3> numbers.

### ■ BIN

This specifies that outgoing data is to be in a binary format whose further specification is determined by WFMOutpre:BYT\_Nr, WFMOutpre:BIT\_Nr, WFMOutpre:BN\_Fmt and WFMOutpre:BYT\_Or.

## Example 1

```
WFMOutpre:ENCdg?
```

This query might return

```
:WFMOUTPRE:ENCDG BIN
```

indicating that outgoing waveform data will be sent in binary format.

## Example 2

```
WFMOutpre:ENCdg ASC
```

This command specifies that the outgoing waveform data will be sent in ASCII format.

---

# WFMOutpre:NR\_FR?

## Description

This query-only command returns the number of frames for the DATA:SOURce waveform transmitted in response to a CURVe? query.

## Group

Waveform Transfer

## Related Commands

CURVe , DATA, DATA:START, DATA:STOP, SAVE:WAVEform, SAVE:WAVEform:FILEFormat, WFMIpre:NR\_FR?, WFMIpre:NR\_Pt, WFMOutpre:NR\_Pt?

## Syntax

```
WFMOutpre:NR_FR?
```

## Example

```
WFMOutpre:NR_FR?
```

This query might return

```
:WFMOUTPRE:NR_FR 500
```

indicating that there are 500 data frames to be sent.

---

# WFMOutpre:NR\_Pt?

## Description

This query-only command returns the number of points for the DATA:SOURce waveform that will be transmitted in response to a CURVe? query.

## Group

Waveform Transfer

## Related Commands

CURVe , DATA, DATA:START, DATA:STOP, SAVE:WAVEform, SAVE:WAVEform:FILEFormat, WFMinpre:NR\_FR?, WFMinpre:NR\_Pt, WFMOutpre:NR\_FR?

## Syntax

```
WFMOutpre:NR_Pt?
```

## Example

```
WFMOutpre:NR_Pt?
```

This query might return

```
:WFMOUTPRE:NR_PT 5000
```

indicating that there are 5000 data points to be sent.



---

# WFMOutpre:PT\_Fmt?

## Description

This query-only command returns the point format for the waveform specified by the DATA:SOURce command. The format specifies a set of equations describing how the scale factors in the preamble are used to give meaning to the CURVe data points.

An error is reported if the DATA:SOURce waveform does not exist.

## Group

Waveform Transfer

## Related Commands

CURVe?, DATA:SOURce

## Syntax

```
WFMOutpre:PT_Fmt?
```

## Example

```
WFMOutpre:PT_Fmt?
```

This query might return

```
:WFMOutpre:PT_Fmt ENV
```

indicating that the waveform data is a series of min-max pairs.

---

# WFMOutpre:PT\_Off?

## Description

This query-only command returns the trigger point relative to DATA:START for the waveform specified by the DATA:SOURce command.

**Note:** This returned value is the point immediately following the actual trigger.

## Group

Waveform Transfer

## Related Commands

DATA:SOURce, DATA:START, WFMOutpre:XZEro

## Syntax

```
WFMOutpre:PT_Off?
```

## Example

```
WFMOutpre:PT_Off?
```

This query might return

```
:WFMOUTPRE:PT_OFF 251
```

specifying that the trigger actually occurred between points 250 and 251.

---

# WFMOutpre:PT\_OR?

## Description

This query-only command specifies whether the source waveform is Fast Acquisition. A Fast Acquisition waveform is stored as a 200 (vertical) by 500 (horizontal) point bitmap. Each point represents display intensity for that screen location. Only CURVe? query functions are allowed on Fast Acquisition waveforms.

When the WFMOutpre:PT\_OR query returns Column, this indicates that the source is a Fast Acquisition waveform (and that each of 500 possible horizontal columns being transmitted contains 200 vertical points). When the WFMOutpre:PT\_OR? query returns Linear, this indicates that the source is not a Fast Acquisition waveform (and that each horizontal column being sent contains only one vertical point). Note that waveform points are transmitted in the following order: top to bottom, then left to right.

## Group

Waveform Transfer

## Related Commands

DATA:SOURce

## Syntax

```
WFMOutpre:PT_OR?
```

## Example 1

```
WFMOutpre:PT_OR?
```

This query might return

```
:WFMOUTPRE:PT_OR COL
```

specifying that the waveform designated by the DATA:SOURce waveform is a Fast Acquisition waveform.

## Example 2

```
WFMOutpre:PT_OR?
```

This query might return

```
:WFMOUTPRE:PT_OR LINEAR
```

specifying that the source waveform is a nonFast Acquisition waveform.

---

# WFMOutpre:WFId?

## Description

This query-only command returns a string describing several aspects of the acquisition parameters for the waveform specified by the DATA:SOURce command.

An error is reported if the DATA:SOURce waveform does not exist.

## Group

Waveform Transfer

## Related Commands

DATA:SOURce

## Syntax

WFMOutpre:WFId?

## Outputs

<string> comprises the following comma-separated fields documented in the tables below:

Waveform Suffixes		
Field	Description	Examples
Source	The source identification string as it appears in the front panel scale factor readouts.	"Ch1-4" "Math1-3" "Ref1-4"
Coupling	A string describing the vertical coupling of the waveform (the Source 1 waveform in the case of Dual Waveform Math).	"AC coupling" "DC coupling" "GND coupling"
Vert Scale	A string containing the vertical scale factor of the unzoomed waveform. The numeric portion will always be 4 digits. The examples cover all known internal units.	"100.0 mV/div" "20.00 dB/div" "45.00 deg/div" "785.4 mrad/div" "500.0 uVs/div" "10.00 kV/s/div" "200.0 mV/div" "50.00 unk/div"
Horiz Scale	A string containing the horizontal scale factor of the unzoomed waveform. The numeric portion will always be four digits. The examples cover all known internal units.	"100.0 ms/div" "10.00 kHz/div" "50.00 c/div"
Record Length	A string containing the number of waveform points available in the entire record. The numeric portion is given as an integer.	"500 points" "500000 points"
Acquisition Mode	A string describing the mode used to acquire the waveform.	"Sample mode" "Pk Detect mode" "Hi Res mode" "Envelope mode" "Average mode"
Primary Reference Offset	A string specifying the delta between the Primary Reference (typically, the A trigger) and the CURVe? zero reference location identified by a combination of PT_Off and XZEro in units of XUNits. For example, in Trigger After Delay, this number would be the actual time between the A and B trigger. In the event that this number is not meaningful, the string will be exactly "0".	"57.2345 ms" "87.3 Hz" "0"

### Example

WFMOutpre:WFID?

This query might return

```
:WFMOUTPRE:WFID "Ch1, DC coupling,100.0mVolts/div,500.0µs/div,500 points, Hi Res mode"
```

---

# WFMOutpre:XINcr?

## Description

This query-only command returns the horizontal point spacing in units of WFMOutpre:XUNit for the waveform specified by the DATA:SOURce command. This value corresponds to the sampling interval.

An error is reported if the DATA:SOURce waveform does not exist.

## Group

Waveform Transfer

## Related Commands

DATA:SOURce, WFMOutpre:XUNit<sub>x</sub>

## Syntax

WFMOutpre:XINcr?

## Example

WFMOutpre:XINcr?

This query might return

```
:WFMOUTPRE:XINCR 10.0000E-6
```

indicating that the horizontal sampling interval is 10  $\mu$ s/point (500  $\mu$ s/div).

---

# WFMOutpre:XUNIT?

## Description

This query-only command returns the horizontal units for the waveform specified by the DATA:SOURce command.

An error is reported if the DATA:SOURce waveform does not exist.

## Group

Waveform Transfer

## Related Commands

DATA:SOURce

## Syntax

```
WFMOutpre:XUNIT?
```

## Example

```
WFMOutpre:XUNIT?
```

This query might return

```
:WFMOUTPRE:XUNIT "Hz"
```

indicating that the horizontal units for the waveform are in hertz.

---

# WFMOutpre:XZErO?

## Description

This query-only command returns the sub-sample time between the trigger sample (designated by PT\_OFF) and the occurrence of the actual trigger for the waveform specified by the DATA:SOURce command. This value is in units of WFMOutpre:XUNit.

An error is reported if the DATA:SOURce waveform does not exist.

**Note:** During *stopped state operation* (i.e., ACQUIRE:STATE OFF), this is the only preamble that changes on each acquisition. If a query is run during steady state operation (that is, all control changes have settled and triggers are arriving on a regular basis), the XZErO value of the last stopped state is returned.

## Group

Waveform Transfer

## Related Commands

DATA:SOURce, WFMOutpre:XUNit

## Syntax

```
WFMOutpre:XZErO?
```

## Example

```
WFMOutpre:XZErO?
```

This query might return

```
:WFMOUTPRE:XZERO 5.6300E-9
```

indicating that the trigger actually occurred 5.63 ns before the trigger sample.



---

# WFMOutpre:YMUlt?

## Description

This query-only command returns the vertical scale factor per digitizing level in units specified by WFMOutpre:YUNit for the waveform specified by the DATA:SOURce command. For those formats in which WFMOutpre:BYT\_Nr is important (all nonfloating point formats), WFMOutpre:YMUlt? must take the location of the binary point implied by BYT\_NR into consideration.

An error is reported if the DATA:SOURce waveform does not exist.

## Group

Waveform Transfer

## Related Commands

DATA:SOURce

## Syntax

```
WFMOutpre:YMUlt?
```

## Example

```
WFMOutpre:YMUlt?
```

This query might return

```
:WFMOUTPRE:YMULT 4.0000E-3
```

indicating that the vertical scale for the corresponding waveform is 100 mV/div.

---

# WFMOutpre:YOFF?

## Description

This query-only command returns the vertical offset in digitized levels for the waveform specified by the DATA:SOURce command. For those formats in which BYT\_NR is important (all non-floating point formats), this command must take the location of the binary point implied by WFMOutpre:BYT\_Nr into consideration.

An error is reported if the DATA:SOURce waveform does not exist.

## Group

Waveform Transfer

## Related Commands

DATA:SOURce, WFMOutpre:BYT\_Nr

## Syntax

```
WFMOutpre:YOFF?
```

## Example

```
WFMOutpre:YOFF?
```

This query might return

```
:WFMOUTPRE:YOFF -50.0000E+0
```

indicating that the position indicator for the waveform was 50 digitizing levels (2 divisions) below center screen.

---

# WFMOutpre:YUNIT?

## Description

This query-only command returns the vertical units for the waveform specified by the DATA:SOURce command.

An error is reported if the DATA:SOURce waveform does not exist.

## Group

Waveform Transfer

## Related Commands

DATA:SOURce

## Syntax

```
WFMOutpre:YUNIT?
```

## Example

```
WFMOutpre:YUNIT?
```

This query might return

```
:WFMOUTPRE:YUNIT "dB"
```

indicating that the vertical units for the waveform are measured in decibels.

---

# WFMOutpre:YZero?

## Description

This query-only command returns the vertical offset in units specified by WFMOutpre:YUNit for the waveform specified by the DATA:SOURce command.

An error is reported if the DATA:SOURce waveform does not exist.

## Group

Waveform Transfer

## Related Commands

DATA:SOURce, WFMOutpre:YUNit

## Syntax

```
WFMOutpre:YZero?
```

## Example

```
WFMOutpre:YUNit?
```

This query might return

```
:WFMOUTPRE:YZERO -100.0000E-3
```

indicating that vertical offset is set to -100 mV.

---

# WFMPre:NR\_FR?

## Description

This query-only command returns the number of frames for the waveform transmitted in response to a CURVE? query.

## Group

Waveform Transfer

## Related Commands

CURVe , DATa, DATa:START, DATa:STOP, SAVe:WAVEform,  
SAVe:WAVEform:FILEFormat, WFMIpre:NR\_FR?,  
WFMIpre:NR\_Pt, WFMOupre:NR\_FR?, WFMOupre:NR\_Pt?,

## Syntax

```
WFMPre:NR_FR?
```

## Example

```
WFMPre:NR_FR?
```

This query might return

```
:WFMPRE:NR_FR:10
```

indicating that you have acquired 10 frames.

---

# WAVFrm?

## Description

This query-only command returns WFMOutpre? and CURVe? data for the waveform as specified by the DATA:SOURce command. This command is equivalent to sending both WFMOutpre? and CURVe?, with the additional provision that the response to WAVFrm? is guaranteed to provide a synchronized preamble and curve.

## Group

Waveform Transfer

## Related Commands

CURVe?, DATA:SOURce, WFMOutpre?

## Syntax

WAVFrm?

## Example

WAVFrm?

This query might return the waveform data as:

```
:WFMOOUTPRE:BIT_NR 8;BN_FMT RI;BYT_NR 1;
BYT_OR MSB;ENCDG ASC;NR_PT 500;PT_FMT Y;
PT_ORDER LINEAR;PT_OFF 0;XINCR 400.0000E-12;
XZERO 0.0000;XUNIT "s";YMULT 4.0000E-3;
YOFF 0.0000;YZERO 0.0000;YUNIT "V";
WFID "Ch1,DC coupling, 100.0mV/div, 200.0ns/
div, 5000 points,Sampl mode";
:CURVE51,50,51,48,51,48,50,49,51,49,51,48,51,4
8,51,49,50,49,50,48,49,49,52,49,49,50,50,48,50
,49,49,49,49,49,50,47,49,47,50,48,49,48,50,48,
49,47,49,48,51,48,49,47,50,48,50,47,51,47,49,4
8,48,48,50,46,50,46,48,45,48,47,49,47,49,48,49
,48,49,45,49,47,48,46,48,48,49,45,49,45,47,46,
46,46,49,47,48,46,48,46,48,47,47,46,47,47,48,4
6,48,45,48,46,47,46,47,47,46,45,46,45,47,47,47
,46,46,44,47,45,45,45,46,43,46,45,46,44,46,45,
48,45,45,44,46,45,46,45,45,45,45,43,45,44,46,4
4,47,44,44,44,45,43,44,44,45,44,46,43,46,43,45
,44,45,44,45,41,44,43,45,42,44,43,44,43,44,42,
43,42,44,41,44,41,44,42,43,42,43,42,45,42,43,4
1,42,41,43,42,44,42,43,42,43,40,42,40,42,40,42
,41,43,41,42,41,42,39,41,41,42,42,43,41,42,39,
41,39,41,40,41,40,41,39,41,38,41,39,41,38,42,3
9,41,37,40,38,40,38,39,38,39,38,41,39,39,38,40
,37,38,37,38,38,39,38,40,36,39,37,39,37,39,37,
39,36,40,38,38,35,38,36,38,37,39,37,38,37,38,3
7,37,36,38,35,38,36,37,36,38,36,38,35,39,34,36
,36,36,35,38,35,36,34,37,34,37,35,37,36,37,35,
37,35,36,34,36,33,36,35,36,34,36,33,34,33,34,3
```

---

4, 34, 34, 36, 33, 36, 34, 36, 33, 34, 33, 34, 32, 34, 32, 34  
, 33, 33, 31, 33, 32, 34, 32, 33, 32, 32, 31, 33, 31, 31, 32,  
33, 31, 32, 31, 32, 30, 32, 30, 32, 31, 31, 30, 31, 30, 31, 2  
9, 31, 29, 32, 31, 31, 30, 30, 29, 30, 29, 30, 30, 31, 29, 30  
, 28, 30, 29, 31, 28, 30, 28, 30, 29, 29, 27, 29, 28, 29, 27,  
30, 27, 30, 26, 28, 28, 29, 27, 28, 27, 28, 27, 28, 28, 28, 2  
7, 28, 27, 28, 26, 28, 26, 28, 26, 26, 25, 27, 25, 28, 27, 28  
, 26, 25, 25, 26, 25, 25, 25, 25, 25, 26, 25, 27, 25, 25, 24,  
25, 25, 26, 24, 26, 24, 24, 23, 25, 24, 24, 22, 25, 23, 24, 2  
3, 24, 22, 24, 22, 23, 22, 25, 24, 24, 21, 23, 22, 22, 21, 24  
, 22, 22, 22, 22, 21, 22, 20, 23, 21, 21, 20

---

# ZOOM

## Description

This command resets the zoom transforms to default values for all traces or live traces. The ZOOM query returns the current vertical and horizontal positioning and scaling of the display.

## Group

Zoom

## Syntax 1

ZOOM {RESET|RESETLive}

## Syntax 2

ZOOM?

## Arguments

- RESET

This resets the zoom transforms to default values for all traces.

- RESETLive

This resets the zoom transforms to default values for live traces.

## Example 1

ZOOM?

This query might return:

```
:ZOOM:MODE 0;GRATICULE:SIZE 80;SPLIT EIGHTYT-  
WENTY:ZOOM:SCROLL:DIRECTION STOP;  
LOCK 0;SPEED 1:ZOOM:ZOOM1:STATE 1;  
SCROLLLOCK 1;CH1:HORIZONTAL:POSITION 50.0000;  
SCALE 5:ZOOM:ZOOM1:CH1:VERTICAL:POSITION 0.0000;  
SCALE 1.0000:ZOOM:ZOOM1:CH2:HORIZONTAL:POSI-  
TION 50.0000;SCALE 5:ZOOM:ZOOM1:CH2:VER-  
TICAL:POSITION 0.0000;  
SCALE 1.0000:ZOOM:ZOOM1:CH3:HORIZONTAL:POSI-  
TION 50.0000;  
SCALE 5:ZOOM:ZOOM1:CH3:VERTICAL:POSITION  
0.0000;  
SCALE 1.0000:ZOOM:ZOOM1:CH4:HORIZONTAL:POSI-  
TION 50.0000;  
SCALE 5:ZOOM:ZOOM1:CH4:VERTICAL:POSITION  
0.0000;SCALE 1.0000:ZOOM:ZOOM1:MATH1:HORIZON-  
TAL:POSITION 50.0000;  
SCALE 5:ZOOM:ZOOM1:MATH1:VERTICAL:POSITION  
0.0000;SCALE 1.0000:ZOOM:ZOOM1:MATH2:HORIZON-  
TAL:POSITION 50.0000;  
SCALE 5:ZOOM:ZOOM1:MATH2:VERTICAL:POSITION  
0.0000;...
```



---

## **Example 2**

ZOOM RESET

This command resets the zoom transforms to default values for all traces.

---

# ZOOM:GRATICULE:SIZE

## Description

This command sets or queries the size, in percent, of the Zoom (lower) graticule.

## Group

Zoom

## Syntax 1

```
ZOOM:GRATICULE:SIZE {50|80|100}
```

## Syntax 2

```
ZOOM:GRATICULE:SIZE?
```

## Arguments

- 50

Set the size of the Zoom graticule to 50 %.

- 80

Set the size of the Zoom graticule to 80 %.

- 100

Set the size of the Zoom graticule to 100 %.

## Example 1

```
ZOOM:GRATICULE:SIZE?
```

This query might return

```
:ZOOM:GRATICULE:SIZE 100
```

Indicating that the zoom graticule size is set to 100%.

## Example 2

```
ZOOM:GRATICULE:SIZE 100
```

Setting the zoom graticule size to 100%.

---

# ZOOm:MODE

## Description

This command turns Zoom mode on or off. The Zoom query returns the current state of Zoom mode. This command is equivalent to pressing the **ZOOM** button located on the front panel.

## Group

Zoom

## Syntax 1

```
ZOOm:MODE {ON|OFF|<NR1>}
```

## Syntax 2

```
ZOOm:MODE
```

## Arguments

- ON

This turns on Zoom mode.

- OFF

This turns off Zoom mode.

- <NR1>

A 0 turns off Zoom mode; any other value turns on Zoom mode.

## Example 1

```
ZOOm:MODE?
```

This query might return

```
:ZOOM:MODE 1
```

indicating that Zoom mode is currently turned on.

## Example 2

```
ZOOm:MODE OFF
```

This command turns off Zoom mode.

---

# ZOOM:SCROLL:DIRECTION

## Description

This command sets or returns the direction for automatic scrolling of zoomed waveforms.

## Group

Zoom

## Syntax 1

```
ZOOM:SCROLL:DIRECTION {FWD|FFWD|REVERSE|  
FREVERSE|STOP}
```

## Syntax 2

```
ZOOM:SCROLL:DIRECTION?
```

## Arguments

- FWD

Starts AutoScroll. The Zoom Box moves from left to right.

- FFWD

Starts AutoScroll. Increases the scrolling speed in the forward direction.

- REVERSE

Starts AutoScroll. The Zoom Box moves from right to left.

- FREVERSE

Starts AutoScroll. Increases the scrolling speed in the reverse direction.

- STOP

Halts AutoScroll.

## Example 1

```
ZOOM:SCROLL:DIRECTION FWD
```

Starts AutoScroll. The Zoom Box moves from left to right.

## Example 2

```
ZOOM:SCROLL:DIRECTION?
```

Returns FWD, FFWD, REVERSE, FREVERSE, or STOP.

---

# ZOOm:SCROLL:LOCK

## Description

This command sets or returns the state of Scroll Lock, which "locks" zoomed waveforms under common control.

## Group

Zoom

## Syntax 1

```
ZOOm:SCROLL:LOCK {ON|OFF|NR1}
```

## Syntax 2

```
ZOOm:SCROLL:LOCK?
```

## Arguments

- ON

Enables Scroll Lock.

- OFF

Disables Scroll Lock.

- <NR1>

A 0 disables Scroll Lock for all zoom windows; any other value enables Scroll Lock for all zoom windows.

## Example 1

```
ZOOm:SCROLL:LOCK ON
```

Enables Scroll Lock for zoomed waveforms identified as locked.

## Example 2

```
ZOOm:SCROLL:LOCK?
```

This query might return

```
:ZOOM:SCROLL:LOCK OFF
```

indicating that the scroll lock function is disabled.

---

# ZOOm:SCROLL:SPEED

## Description

This command sets or returns the speed of automatic scrolling.

## Group

Zoom

## Syntax 1

```
ZOOm:SCROLL:SPEED <NR1>
```

## Syntax 2

```
ZOOm:SCROLL:SPEED?
```

## Arguments

- <NR1>

This is a value from 1 to 10.

## Example 1

```
ZOOm:SCROLL:SPEED 2
```

AutoScroll will move zoomed waveform(s) 2 divisions per update.

## Example 2

```
ZOOm:SCROLL:SPEED?
```

This query might return

```
:ZOOM:SCROLL:SPEED 1
```

indicating that the scroll speed is set to move zoomed waveform(s) 1 division per update.

---

# ZOOM:GRATICULE:SPLIT

## Description

This command sets or returns the sizes of the acquisition and zoom windows when Zoom is selected.

**Note:** For this command to have an effect on the display, you must have the zoom mode turned on. Use the ZOOM:MODE command to turn on the zoom mode. The ZOOM:GRATICULE:SPLIT command is equivalent to selecting Zoom Setup from the Horiz/Acq menu and selecting a value for the Graticule Split in the control window.

## Group

Zoom

## Related Commands

ZOOM:MODE

## Syntax 1

```
ZOOM:GRATICULE:SPLIT {FIFTYfifty|EIGHTytwen-  
ty|FuLl}
```

## Syntax 2

```
ZOOM:GRATICULE:SPLIT?
```

## Arguments

### ■ FIFTYfifty

This argument sets half of the available display to the zoomed graticule and half of the available display to the acquisition graticule; this argument is the default value.

### ■ EIGHTytwenty

This argument sets 80% of the available display to the zoomed graticule and 20% to the acquisition graticule.

### ■ FuLl

This argument sets the entire display to the zoomed graticule.

## Example 1

```
ZOOM:GRATICULE:SPLIT?
```

This query might return `:ZOOM:GRATICULE:SPLIT FIFTYFIFTY`, indicating that the display area is divided equally between the zoomed graticule and the acquisition graticule.

## Example 2

```
ZOOM:GRATICULE:SPLIT FuLl
```

This command sets the full display area to the zoomed graticule.

---

# ZOOM:ZOOM<x>

## Description

This command resets the zoom transforms to default values for all traces of the specified zoom, where x is an integer from 1 to 4 representing the desired zoom window. The ZOOM query returns the current vertical and horizontal positioning and scaling of the display.

## Group

Zoom

## Syntax 1

ZOOM:ZOOM<x> {RESET|RESETLive}

## Syntax 2

ZOOM:ZOOM<x>?

## Arguments

- RESET

Resets the zoom transforms to default values for all traces of the specified zoom.

- RESETLive

Resets the zoom transforms to default values for live traces of the specified zoom.

## Example 1

ZOOM:ZOOM1 RESET

This command resets the zoom transforms for all traces of Zoom1 to their default values

## Example 2

ZOOM:ZOOM1?

This query might return

```
:ZOOM:ZOOM1:STATE 1;SCROLLLOCK 1;  
CH1:HORIZONTAL POSITION 50.0000;SCALE 5;  
:ZOOM:ZOOM1:CH1:VERTICAL:POSITION 0.0000;  
SCALE 1.0000;  
:ZOOM:ZOOM1:CH2:HORIZONTAL:POSTION 50.0000;  
SCALE 5;  
:ZOOM:ZOOM1:CH2:VERTICAL:POSITION 0.0000;  
SCALE 1.0000;  
:ZOOM:ZOOM1:CH3:HORIZONTAL:POSITION50.0000;  
SCALE 5;;:ZOOM:ZOOM1:CH3:VERTICAL...
```

indicating the Zoom1 transforms for all traces.



---

# ZOOm:ZOOM<x>:<wfm>:HORizontal:POSition

## Description

This command sets or queries the horizontal position of the specified waveform for the specified zoom, where x is an integer from 1 to 4 representing the desired zoom window. The setting of the ZOOm:ZOOM<x>:SCROLLLOCK command determines the waveforms that are affected.

## Group

Zoom

## Syntax 1

```
ZOOm:ZOOM<x>:<wfm>:HORizontal:POSition <NR3>
```

## Syntax 2

```
ZOOm:ZOOM<x>:<wfm>:HORizontal:POSition?
```

## Argument

- <NR3>

This is a value from 0 to 100.00 and is the percent of the waveform that is to the left of screen center, when the zoom factor is 1x or greater.

## Example 1

```
ZOOm:ZOOM1:CH1:HORizontal:POSition 50
```

This command sets the Zoom1 reference pointer at 50% of acquired waveform.

## Example 2

```
ZOOm:ZOOM1:CH1:HORizontal:POSition?
```

This query might return

```
:ZOOM1:CH1:HORIZONTAL:POSITION 50.0000
```

indicating that the Zoom1 reference pointer for Channel 1 is currently set at 50% of acquired waveform.

---

# ZOOm:ZOOM<x>:<wfm>:HORizontal:SCAlE

## Description

This command sets or queries the zoom horizontal scale factor of the specified waveform for the specified zoom, where x is an integer from 1 to 4 representing the desired zoom window.

## Group

Zoom

## Syntax 1

```
ZOOm:ZOOM<x>:<wfm>:HORizontal:SCAlE <NR3>
```

## Syntax 2

```
ZOOm:ZOOM<x>:<wfm>:HORizontal:SCAlE?
```

## Argument

- <NR3>

This is the amount of expansion in the horizontal direction in 1-2 -5 increments.

## Example 1

```
ZOOm:ZOOM1:CH1:HORizontal:SCAlE 5
```

This command sets the horizontal scale factor of Channel 1 to 5.

## Example 2

```
ZOOm:ZOOM2:CH2:HORizontal:SCAlE?
```

This query might return :ZOOM2:CH2:HORIZONTAL:SCALE 1, indicating that Channel 2 has a horizontal scale factor of 1.

---

# ZOOm:ZOOM<x>:<wfm>:VERTical:POSition

## Description

This command sets or queries the vertical position of the specified waveform for the specified zoom, where x is an integer from 1 to 4 representing the desired zoom window. The setting of the ZOOm:ZOOM<x>:SCROLLLOCK command determines the waveforms that are affected.

## Group

Zoom

## Syntax 1

```
ZOOm:ZOOM<x>:<wfm>:VERTical:POSition <NR3>
```

## Syntax 2

```
ZOOm:ZOOM<x>:<wfm>:VERTical:POSition?
```

## Argument

- <NR3>

This is the vertical position, expressed in divisions.

## Example 1

```
ZOOm:ZOOM1:CH1:VERTical:POSition 2
```

This command sets the Zoom1 vertical position to 2, which centers the zoom trace at the second division of the acquired trace.

## Example 2

```
ZOOm:ZOOM1:CH1:VERTical:POSition?
```

This query might return :ZOOm1:CH1:VERTical:POSition 0.0000, indicating that the Zoom1 trace for Channel 1 is centered at division 0 of the acquired trace.

---

# ZOOM:ZOOM<x>:<wfm>:VERTical:SCALE

## Description

This command sets or queries the zoom vertical scale of the specified waveform for the specified zoom, where x is an integer from 1 to 4 representing the desired zoom window.

## Group

Zoom

## Syntax 1

```
ZOOM:ZOOM<x>:<wfm>:VERTical:SCALE <NR3>
```

## Syntax 2

```
ZOOM:ZOOM<x>:<wfm>:VERTical:SCALE?
```

## Argument

- <NR3>

This is the amount of vertical expansion or compression, which operates on a 1-2-5 sequence (for example, 1, 2, 5, 10, 20, 50, 100...). Based on the value entered, this command uses the nearest scale factor. Setting the vertical scale to 1 indicates unity (no zoom).

## Example 1

```
ZOOM:ZOOM2:CH2:VERTical:SCALE 5
```

This command sets the vertical scale of Channel 2 to 5x.

## Example 2

```
ZOOM:ZOOM2:CH2:VERTical:SCALE?
```

This query might return :ZOOM2:CH2:VERTICAL:SCALE 2.0000, indicating that the vertical scale is 2x.

---

# ZOOM:ZOOM<x>:SCROLLLock

## Description

This command sets or queries Scroll Lock for the specified zoom, where x is an integer from 1 to 4 representing the desired zoom window.

## Group

Zoom

## Syntax 1

```
ZOOM:ZOOM<x>:SCROLLLock {ON|OFF|1|0|NR1}
```

## Syntax 2

```
ZOOM:ZOOM<x>:SCROLLLock?
```

## Arguments

- ON {1}

Locks waveforms for the specified zoom window <x>.

- OFF {0}

Unlocks waveforms for the specified zoom window <x>.

- NR1

0 unlocks waveforms for the specified zoom window <x>; any other value locks waveforms for the specified zoom window <x>.

## Example 1

```
ZOOM:ZOOM1:SCROLLLock ON
```

This command turns Zoom1 scroll lock on.

## Example 2

```
ZOOM:ZOOM1:SCROLLLock?
```

This query returns either ON | 1 or OFF | 0 for Zoom1.

---

# ZOOM:ZOOM<x>:STATE

## Description

This command sets or queries the specified zoom on or off, where x is an integer from 1 to 4 representing the desired zoom window.

Default is Zoom1 on, Zoom2 - 4 off.

## Group

Zoom

## Syntax 1

```
ZOOM:ZOOM<x>:STATE {ON|OFF|<NR1>}
```

## Syntax 2

```
ZOOM:ZOOM<x>:STATE?
```

## Arguments

- ON

Turns Zoom 1-4 on.

- OFF

Turns Zoom 1-4 off.

- <NR1>

A 0 disables the specified zoom; any other value enables the specified zoom.

## Example 1

```
ZOOM:ZOOM1:STATE ON
```

This command turns Zoom1 on.

## Example 2

```
ZOOM:ZOOM1:STATE?
```

This query might return

```
:ZOOM:ZOOM2:STATE 1
```

indicating that Zoom2 is on.

---

# Status and Events

The instrument provides a status and event reporting system for the GPIB interfaces. This system informs you of certain significant events that occur within the instrument.

The instrument status handling system consists of five 8-bit registers and two queues for each interface. The remaining Status subtopics describe these registers and components. They also explain how the event handling system operates.

## Registers

### Overview

The registers in the event handling system fall into two functional groups:

- Status Registers contain information about the status of the instrument. They include the Standard Event Status Register (SESR)
- Enable Registers determine whether selected types of events are reported to the Status Registers and the Event Queue. They include the Device Event Status Enable Register (DESER), the Event Status Enable Register (ESER), and the Service Request Enable Register (SRER).

### Status

The Standard Event Status Register (SESR) and the Status Byte Register (SBR) record certain types of events that may occur while the instrument is in use. IEEE Std 488.2-1987 defines these registers.

Each bit in a Status Register records a particular type of event, such as an execution error or message available. When an event of a given type occurs, the instrument sets the bit that represents that type of event to a value of one. (You can disable bits so that they ignore events and remain at zero. See Enable Registers). Reading the status registers tells you what types of events have occurred.

**The Standard Event Status Register (SESR)** - The SESR records eight types of events that can occur within the instrument. Use the \*ESR? query to read the SESR register. Reading the register clears the bits of the register so that the register can accumulate information about new events.

**Note:** TekVISA applications use SESR bit 6 to respond to any of several events, including some front panel actions.

### Standard Event Status Register (SESR)

7	6	5	4	3	2	1	0
PON	URQ	CME	EXE	DDE	QYE	RQC	OPC

### SESR Bit Functions

Bit	Function
7 (MSB)	PON Power On. Shows that the oscilloscope was powered on. On completion, the diagnostic tests also set this bit.
6	URQ User Request. Indicates that an application event has occurred. *See note.
5	CME Command Error. Shows that an error occurred while the oscilloscope was parsing a command or query.
4	EXE Execution Error. Shows that an error occurred while the oscilloscope was executing a command or query.
3	DDE Device Error. Shows that a device error occurred.
2	QYE Query Error. Shows that either an attempt was made to read the Output Queue when no data was present or pending, or that data in the Output Queue was lost.
1	RQC Request Control. This is not used.
0 (LSB)	OPC Operation Complete. Shows that the operation is complete. This bit is set when all pending operations complete following an *OPC command.

**The Status Byte Register (SBR)** - Records whether output is available in the Output Queue, whether the instrument requests service, and whether the SESR has recorded any events.

Use a Serial Poll or the \*STB? query to read the contents of the SBR. The bits in the SBR are set and cleared depending on the contents of the SESR, the Event Status Enable Register (ESER), and the Output Queue. When you use a Serial Poll to obtain the SBR, bit 6 is the RQS bit. When you use the \*STB? query to obtain the SBR, bit 6 is the MSS bit. Reading the SBR does not clear the bits.

### Status Byte Register (SBR)

7	6	5	4	3	2	1	0
----	RQS	ESB	MAV	----	----	----	----
	6						
	MSS						



SBR Bit Functions	
Bit	Function
7 (MSB)	----- Not used.
6	RQS Request Service. Obtained from a serial poll. Shows that the oscilloscope requests service from the GPIB controller.
6	MSS Master Status Summary. Obtained from *STB? query. Summarizes the ESB and MAV bits in the SBR.
5	ESB Event Status Bit. Shows that status is enabled and present in the SESR.
4	MAV Message Available. Shows that output is available in the Output Queue.
3	----- Not used.
2	----- Not used.
1 – 0	----- Not used.

## Enable

DESER, ESER, and SRER allow you to select which events are reported to the Status Registers and the Event Queue. Each Enable Register acts as a filter to a Status Register (the DESER also acts as a filter to the Event Queue) and can prevent information from being recorded in the register or queue.

Each bit in an Enable Register corresponds to a bit in the Status Register it controls. In order for an event to be reported to a bit in the Status Register, the corresponding bit in the Enable Register must be set to one. If the bit in the Enable Register is set to zero, the event is not recorded.

Various commands set the bits in the Enable Registers. The Enable Registers and the commands used to set them are described below.

**The Device Event Status Enable Register (DESER)** - This register controls which types of events are reported to the SESR and the Event Queue. The bits in the DESER correspond to those in the SESR.

Use the DESE command to enable and disable the bits in the DESER. Use the DESE? query to read the DESER.

Device Event Status Enable Register (DESER)							
7 PON	6 URQ	5 CME	4 EXE	3 DDE	2 QYE	1 RQC	0 OPC

**The Event Status Enable Register (ESER)** - This register controls which types of events are summarized by the Event Status Bit (ESB) in the SBR. Use the \*ESE command to set the bits in the ESER. Use the \*ESE? query to read it.

---

### Event Status Enable Register (ESER)

7	6	5	4	3	2	1	0
PON	URQ	CME	EXE	DDE	QYE	RQC	OPC

**The Service Request Enable Register (SRER)** – This register controls which bits in the SBR generate a Service Request and are summarized by the Master Status Summary (MSS) bit.

Use the \*SRE command to set the SRER. Use the \*SRE? query to read the register. The RQS bit remains set to one until either the Status Byte Register is read with a Serial Poll or the MSS bit changes back to a zero.

### Service Request Enable Register (SRER)

7	6	5	4	3	2	1	0
-----	-----	ESB	MAV	-----	-----	-----	-----

### \*PSC Command

The \*PSC command controls the Enable Registers contents at power-on. Sending \*PSC 1 sets the Enable Registers at power on as follows:

- DESER 255 (equivalent to a DESe 255 command)
- ESER 0 (equivalent to an \*ESE 0 command)
- SRER 0 (equivalent to an \*SRE 0 command)

Sending \*PSC 0 lets the Enable Registers maintain their values in non-volatile memory through a power cycle.

Note: To enable the PON (Power On) event to generate a Service Request, send \*PSC 0, use the DESe and \*ESE commands to enable PON in the DESER and ESER, and use the \*SRE command to enable bit 5 in the SRER. Subsequent power-on cycles will generate a Service Request.

---

# Queues

The **\*PSC** command controls the Enable Registers contents at power-on. Sending **\*PSC 1** sets the Enable Registers at power on as follows:

## Output Queue

The instrument stores query responses in the Output Queue and empties this queue each time it receives a new command or query message after an **<EOM>**. The controller must read a query response before it sends the next command (or query) or it will lose responses to earlier queries.

**Caution:** When a controller sends a query, an **<EOM>**, and a second query, the instrument normally clears the first response and outputs the second while reporting a Query Error (QYE bit in the ESER) to indicate the lost response. A fast controller, however, may receive a part or all of the first response as well. To avoid this situation, the controller should always read the response immediately after sending any terminated query message or send a DCL (Device Clear) before sending the second query.

## Event Queue

The Event Queue stores detailed information on up to 33 events. If more than 32 events stack up in the Event Queue, the 32nd event is replaced by event code 350, "Queue Overflow."

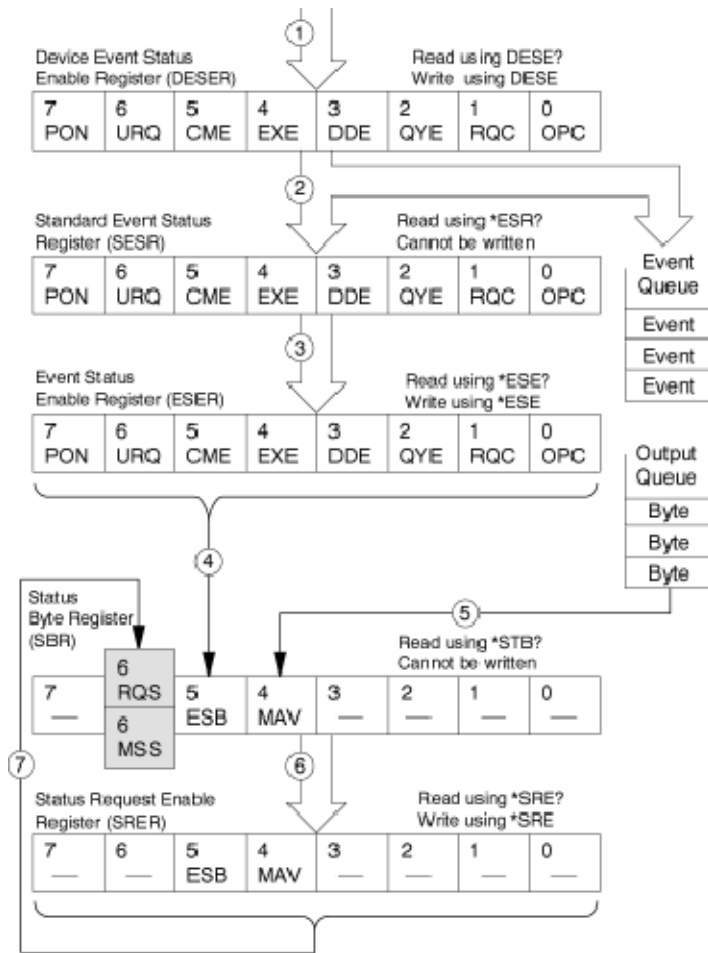
Read the Event Queue with the **EVENT?** query (which returns only the event number), with the **EVMSG?** query (which returns the event number and a text description of the event), or with the **ALLEV?** query (which returns all the event numbers along with a description of the event). Reading an event removes it from the queue.

Before reading an event from the Event Queue, you must use the **\*ESR?** query to read the summary of the event from the SESR. This makes the events summarized by the **\*ESR?** read available to the **EVENT?** and **EVMSG?** queries, and empties the SESR.

Reading the SESR erases any events that were summarized by previous **\*ESR?** reads but not read from the Event Queue. Events that follow an **\*ESR?** read are put in the Event Queue but are not available until **\*ESR?** is used again.

## Event Handling Sequence

The figure below shows how to use the status and event handling system. In the explanation that follows, numbers in parentheses refer to numbers in the figure.



When an event occurs, a signal is sent to the DESER (1). If that type of event is enabled in the DESER (that is, if the bit for that event type is set to 1), the appropriate bit in the SESR is set to one, and the event is recorded in the Event Queue (2). If the corresponding bit in the ESER is also enabled (3), then the ESB bit in the SBR is set to one (4).

When output is sent to the Output Queue, the MAV bit in the SBR is set to one (5).

When a bit in the SBR is set to one and the corresponding bit in the SRER is enabled (6), the MSS bit in the SBR is set to one and a service request is generated (7).

## Synchronization Methods

### Overview

Although most GPIB commands are completed almost immediately after being received by the instrument, some commands start a process that requires more time. For example, once a single sequence acquisition command is executed, depending upon the applied signals and trigger settings, it may be a few seconds before the acquisition is complete. Rather than remain idle while the operation is in process, the

---

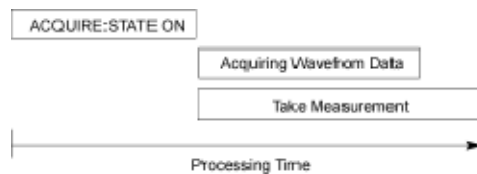
instrument will continue processing other commands. This means that some operations will not be completed in the order that they were sent.

Sometimes the result of an operation depends on the result of an earlier operation. A first operation must complete before the next one gets processed. The instrument status and event reporting system provides ways to do this.

For example, a typical application might involve acquiring a single-sequence waveform and then taking a measurement on the acquired waveform. You could use the following command sequence to do this:

```
/** Set up conditional acquisition **/  
ACQUIRE:STATE OFF  
SELECT:CH1 ON  
HORIZONTAL:RECORDLENGTH 500  
ACQUIRE:MODE SAMPLE  
ACQUIRE:STOPAFTER SEQUENCE  
  
/** Acquire waveform data **/  
ACQUIRE:STATE ON  
  
/** Set up the measurement parameters **/  
MEASUREMENT:IMMED:TYPE AMPLITUDE  
MEASUREMENT:IMMED:SOURCE CH1  
  
/** Take amplitude measurement **/  
MEASUREMENT:MEAS1:VALUE?
```

The acquisition of the waveform requires extended processing time. It may not finish before the instrument takes an amplitude measurement (see the following figure). This can result in an incorrect amplitude value.



To ensure the instrument completes waveform acquisition before taking the measurement on the acquired data, you can synchronize the program.



You can use four commands to synchronize the operation of the instrument with your application program: \*WAI, BUSY?, \*OPC, and \*OPC?

### **\*WAI**

The \*WAI command forces completion of previous commands that generate an OPC message. No commands after the \*WAI are processed before the OPC message(s) are generated

The same command sequence using the \*WAI command for synchronization looks like this:

```

/* Set up conditional acquisition */
ACQUIRE:STATE OFF

SELECT:CH1 ON

HORIZONTAL:RECORDLENGTH 500

ACQUIRE:MODE SAMPLE

ACQUIRE:STOPAFTER SEQUENCE

/* Acquire waveform data */
ACQUIRE:STATE ON

/* Set up the measurement parameters */
MEASUREMENT:IMMED:TYPE AMPLITUDE

MEASUREMENT:IMMED:SOURCE CH1

/* Wait until the acquisition is complete before taking the measurement*/

*/

*WAI

/* Take amplitude measurement */

MEASUREMENT:IMMED:VALUE?

```

The controller can continue to write commands to the input buffer of the instrument, but the commands will not be processed by the instrument until all OPC operations in process are complete. If the input buff-

---

er becomes full, the controller will be unable to write more commands to the buffer. This can cause a time-out.

## **BUSY?**

The BUSY? query allows you to find out whether the instrument is busy processing a command that has an extended processing time such as single-sequence acquisition.

The same command sequence, using the BUSY? query for synchronization, l

```
/* Set up conditional acquisition */  
  
ACQUIRE:STATE OFF  
  
SELECT:CH1 ON  
  
HORIZONTAL:RECORDLENGTH 500  
  
ACQUIRE:MODE SAMPLE  
  
ACQUIRE:STOPAFTER SEQUENCE  
  
  
/* Acquire waveform data */  
  
ACQUIRE:STATE ON  
  
  
/* Set up the measurement parameters */  
  
MEASUREMENT:IMMED:TYPE AMPLITUDE  
  
MEASUREMENT:IMMED:SOURCE CH1  
  
  
/* Wait until the acquisition is complete  
before taking the measurement */  
  
While BUSY? keep looping  
  
  
/* Take amplitude measurement */  
  
MEASUREMENT:IMMED:VALUE?
```

This sequence lets you create your own wait loop rather than using the \*WAI command. The BUSY? query helps you avoid time-outs caused by writing too many commands to the input buffer. The controller is still tied up though, and the repeated BUSY? query will result in more bus traffic.

## **\*OPC**

If the corresponding status registers are enabled, the \*OPC command sets the OPC bit in the Standard Event Status Register (SESR) when an

---

operation is complete. You achieve synchronization by using this command with either a serial poll or service request handler.

**Serial Poll Method:** Enable the OPC bit in the Device Event Status Enable Register (DESER) and the Event Status Enable Register (ESER) using the DESE and \*ESE commands.

When the operation is complete, the OPC bit in the Standard Event Status Register (SESR) will be enabled and the Event Status Bit (ESB) in the Status Byte Register will be enabled.

The same command sequence using the \*OPC command for synchronization with serial polling looks like this:

```
/* Set up conditional acquisition */
ACQUIRE:STATE OFF
SELECT:CH1 ON
HORIZONTAL:RECORDLENGTH 500
ACQUIRE:MODE SAMPLE
ACQUIRE:STOPAFTER SEQUENCE

/* Enable the status registers */
DESE 1
*ESE 1
*SRE 0

/* Acquire waveform data */
ACQUIRE:STATE ON

/* Set up the measurement parameters */
MEASUREMENT:IMMED:TYPE AMPLITUDE
MEASUREMENT:IMMED:SOURCE CH1

/* Wait until the acquisition is complete before taking the measurement.
*/
*OPC

While serial poll = 0, keep looping
/* Take amplitude measurement */
```



---

MEASUREMENT:IMMED:VALUE?

This technique requires less bus traffic than did looping on BUSY?.

**Service Request Method:** Enable the OPC bit in the Device Event Status Enable Register (DESER) and the Event Status Enable Register (ESER) using the DESE and \*ESE commands.

You can also enable service requests by setting the ESB bit in the Service Request Enable Register (SRER) using the \*SRE command. When the operation is complete, a Service Request will be generated.

The same command sequence using the \*OPC command for synchronization looks like this

```
/* Set up conditional acquisition */
ACQUIRE:STATE OFF
SELECT:CH1 ON
HORIZONTAL:RECORDLENGTH 500
ACQUIRE:MODE SAMPLE
ACQUIRE:STOPAFTER SEQUENCE

/* Enable the status registers */
DESE 1
*ESE 1
*SRE 32

/* Acquire waveform data */
ACQUIRE:STATE ON

/* Set up the measurement parameters */
MEASUREMENT:IMMED:TYPE AMPLITUDE
MEASUREMENT:IMMED:SOURCE CH1

/* Wait until the acquisition is complete before taking the measurement
*/
*OPC

The program can now do different tasks such as talk to other devices. The SRQ, when it
```

---

comes, interrupts those tasks and returns control to this task.

```
/* Take amplitude measurement */
```

```
MEASUREMENT:IMMED:VALUE?
```

This technique is more efficient but requires more sophisticated programming.

### **\*OPC?**

The \*OPC? query places a 1 in the Output Queue once an operation that generates an OPC message is complete. A time out could occur if you try to read the output queue before there is any data in it.

The same command sequence using the \*OPC? query for synchronization looks like this:

```
/* Set up conditional acquisition */
```

```
ACQUIRE:STATE OFF
```

```
SELECT:CH1 ON
```

```
HORIZONTAL:RECORDLENGTH 500
```

```
ACQUIRE:MODE SAMPLE
```

```
ACQUIRE:STOPAFTER SEQUENCE
```

```
/* Acquire waveform data */
```

```
ACQUIRE:STATE ON
```

```
/* Set up the measurement parameters */
```

```
MEASUREMENT:IMMED:TYPE AMPLITUDE
```

```
MEASUREMENT:IMMED:SOURCE CH1
```

```
/* Wait until the acquisition is complete before taking the measurement
```

```
*/
```

```
*OPC?
```

```
Wait for read from Output Queue.
```

```
/* Take amplitude measurement */
```

```
MEASUREMENT:IMMED:VALUE?
```

This is the simplest approach. It requires no status handling or loops. However, you must set the controller time-out for longer than the acquisition operation.

---

# Messages

## Overview

The information contained in the topic tabs above covers all the programming interface messages the instrument generates in response to commands and queries.

For most messages, a secondary message from the instrument gives more detail about the cause of the error or the meaning of the message. This message is part of the message string and is separated from the main message by a semicolon.

Each message is the result of an event. Each type of event sets a specific bit in the SESR and is controlled by the equivalent bit in the DESER. Thus, each message is associated with a specific SESR bit. In the message tables, the associated SESR bit is specified in the table title, with exceptions noted with the error message text.

## No Event

The following table shows the messages when the system has no events or status to report. These have no associated SESR bit.

No Event Messages	
Code	Message
0	No events to report; queue empty
1	No events to report; new events pending *ESR?

## Command Error

The following table shows the command error messages generated by improper syntax. Check that the command is properly formed and that it follows the rules in the section on command Syntax.

<b>Command Error Messages (CME Bit 5)</b>	
<b>Code</b>	<b>Message</b>
100	Command error
101	Invalid character
102	Syntax error
103	Invalid separator
104	Data type error
105	GET not allowed
108	Parameter not allowed
109	Missing parameter
110	Command header error
111	Header separator error
112	Program mnemonic too long
113	Undefined header
114	Header suffix out of range
120	Numeric data error
121	Invalid character in number
123	Exponent too large
124	Too many digits
128	Numeric data not allowed
130	Suffix error
131	Invalid suffix
134	Suffix too long
138	Suffix not allowed
140	Character data error
141	Invalid character data
144	Character data too long
148	Character data not allowed
150	String data error
151	Invalid string data
158	String data not allowed
160	Block data error
161	Invalid block data
168	Block data not allowed
170	Command expression error
171	Invalid expression
178	Expression data not allowed

## **Execution Error**

The following table lists the execution errors that are detected during execution of a command.

2209	Measurement error, No second Mid Ref crossing
2210	Measurement error, No Mid Ref crossing, second waveform
2211	Measurement error, No backwards Mid Ref crossing
2212	Measurement error, No negative crossing
2213	Measurement error, No positive crossing
2214	Measurement error, No crossing
2215	Measurement error, No crossing, second waveform
2216	Measurement error, No crossing, target waveform
2217	Measurement error, Constant waveform
2218	Measurement error, Unused
2219	Measurement error, No valid edge - No arm sample
2220	Measurement error, No valid edge - No arm cross
2221	Measurement error, No valid edge - No trigger cross
2222	Measurement error, No valid edge - No second cross
2223	Measurement error, Waveform mismatch
2225	Measurement error, No waveform to measure
2226	Measurement error, Null Waveform
2227	Measurement error, Positive and Negative Clipping
2228	Measurement error, Positive Clipping
2229	Measurement error, Negative Clipping
2230	Measurement error, High Ref < Low Ref
2231	Measurement error, no statistics available
2241	Waveform request is invalid
2243	This meas cannot be performed on this type of wfm
2244	Source wfm is not active
2248	This ref cannot be activated
2249	Reference deletion error, Waveform in use for math

2250	Reference error, Waveform reference file is invalid
2251	Reference error, Waveform preamble does not match actual
2252	Reference error, Source wfm is not valid
2253	Reference error, Too many points received
2254	Reference error, Too few points received
2259	File too big
2261	Calibration error, wait for the warmup interval to expire before invoking SPC
2400	Not enough memory available
2401	This channel cannot be activated
2402	Math/Meas/Histo have circular definition
2410	Empty math string
2411	Syntax error in math string
2412	Semantic error in math string
2413	Math expression is too complex
2420	Histogram cannot be performed on this type of wfm

## Device Error

The following table lists the device errors that can occur during instrument operation. These errors may indicate that the instrument needs repair.

Device Error Messages (DDE Bit 3)	
Code	Message
300	Device-specific error
310	System error
311	Memory error
312	PUD memory lost
313	Calibration memory lost
314	Save/recall memory lost
315	Configuration memory lost
350	Queue overflow (does not set DDE bit)

## System Event

The following table lists the system event messages. These messages are generated whenever certain system conditions occur.

System Event Messages	
Code	Message
400	Query event
401	Power on (PON bit 7 set)
402	Operation complete (OPC bit 0 set)
403	User request (URQ bit 6 set)
404	Power fail (DDE bit 3 set)
405	Request control
410	Query INTERRUPTED (QYE bit 2 set)
420	Query UNTERMINATED (QYE bit 2 set)
430	Query DEADLOCKED (QYE bit 2 set)
440	Query UNTERMINATED after indefinite response (QYE bit 2 set)

### Execution Warning

The following table lists warning messages that do not interrupt the flow of command execution. These notify you that you may get unexpected results.

Execution Warning Messages (EXE Bit 4)	
Code	Message
540	Measurement warning
541	Measurement warning, Low signal amplitude
542	Measurement warning, Unstable histogram
543	Measurement warning, Low resolution
544	Measurement warning, Uncertain edge
545	Measurement warning, Invalid in minmax
546	Measurement warning, Need 3 edges
547	Measurement warning, Clipping positive/negative
548	Measurement warning, Clipping positive
549	Measurement warning, Clipping negative
551	FASTAcq mode is active - deactivate to use math

### Internal Warning

The following table shows internal errors that indicate an internal fault in the instrument.

Internal Warning Messages	
Code	Message
600	Internal warning
630	Internal warning, 50 $\Omega$ overload

---

# Programming Examples

## Overview

Three example programs, which demonstrate methods that you can use to control the instrument through the General Purpose Interface Bus (GPIB), are included on your *TDS5000B Series Product Software CD*. These example programs are installed as part of the *GPIB Programmer* installation, which includes the TDS5000B Series Oscilloscope Programmer Online Guide.

To install the "GPIB Programmer", perform the following procedure:

1. Insert the *TDS5000B Series Product Software CD* in your CD drive.
2. Open the folder named *GPIB Programmer*.
3. Double-click *Setup.exe*.
4. Follow the Install Wizard directions.

An Examples directory will be created with the following path name:

*C:\Program Files\TekScope\Programmer\Examples*

Within the *Examples* directory are two subdirectories, *Source* and *Programs*.

*Source* contains the source files, written in Microsoft Visual C++ , Version 6.0, which is required to build executable files for each of the following example programs:

- *Meas.c* -- This example program demonstrates how to perform a periodic amplitude measurement on CH 1 of your instrument.
- *GetWfm.c* -- This example program demonstrates how to acquire and output the CH 1 waveform in a 500-point RIBANARY format. The program then queries the instrument to get the waveform preamble information, formats the binary waveform data as ASCII values, and, finally, writes out a report of the waveform preamble and ASCII data points to a file named *WFM\_DATA.PRN*.
- *Tl.c* -- This example program demonstrates how to use a general talker/listener program to allow the user to send commands and queries to the instrument and display the responses.

A *README* file in the *Source* directory has the latest documentation.

*Programs* contains compiled, executable files for each of the example programs in the *Source* directory (see above).

The programs run on a PC-compatible system equipped with a National Instruments GPIB board and associated drivers.



---

# Compiling GPIB Programs

The example GPIB programs make the following assumptions:

- The GPIB controller board is "GPIB0" (board 0). If you have more than one GPIB controller in your workstation, and you want to use a GPIB controller other than board 0, you must edit the source files accordingly.
- The instrument is connected to the GPIB controller (see above) and is set to address 1 (DEV1). If you want to use another address, then you must edit the source files accordingly.

Each program requires adding the following source files to the associated VC++ project (which you will create).

- *GPIBERR.C*
- *GPIBREAD.C*
- *GPIBWAIT.C*
- *GPIBWRIT.C*

In addition to the above source files, you will also need to add the following files provided by National Instruments:

- *decl-32.h*
- *gplib-32.obj*

## Visual C++

To make an executable for any of the example programs in the Source directory, perform the following steps:

1. Install Microsoft Visual C++, Version 6.0.
2. Install the National Instruments GPIB board and drivers.
3. Copy the following source files from your National Instruments GPIB drivers directory to this directory ("Examples\Source"):
  - *decl-32.h*
  - *gplib-32.obj*
4. Create a new "Win32 Console Application" project in Visual C++ located in this directory ("Examples\Source"). For example, if you want to build the Talker/Listener application, select this directory for the location, and enter a suitable name for the project, such as TL.
5. Add the following Visual C++ source files to the project:
  - *..\GPIBERR.C*
  - *..\GPIBREAD*

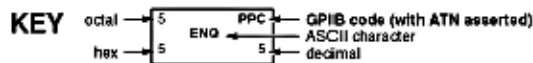
- 
- *..\GPIBWAIT.C*
  - *..\GPIBWRIT.C*
6. Add the following source files provided by National Instruments to the project:
    - *..\decl-32.h*
    - *..\gpib-32.obj*
  7. Add the appropriate program main source file to the project (see Overview for a list of the three source files).
  8. Build and test the project.
  9. To build another of the example projects, repeat steps 4 through 8.

# Miscellaneous

## Character Charts

Appendix A: ASCII & GPIB Code Chart

B7 B6 B5 BITS B4 B3 B2 B1	0 0 0		0 0 1		0 1 0		0 1 1		1 0 0		1 0 1		1 1 0		1 1 1	
	CONTROL				NUMBERS SYMBOLS				UPPER CASE				LOWER CASE			
0 0 0 0	0	NUL	0	DLE	40	LA0	60	LA16	100	TA0	120	TA16	140	SA0	160	SA16
0 0 0 1	1	GTL SOH	11	LLD DC1	41	LA1	61	LA17	101	TA1	121	TA17	141	SA1	161	SA17
0 0 1 0	2	STX	22	DC2	42	LA2	62	LA18	102	TA2	122	TA18	142	SA2	162	SA18
0 0 1 1	3	ETX	23	DC3	43	LA3	63	LA19	103	TA3	123	TA19	143	SA3	163	SA19
0 1 0 0	4	SDC EDT	44	LA4	64	LA20	104	TA4	124	TA20	144	SA4	164	SA20		
0 1 0 1	5	PPC ENQ	25	PPU NAK	45	LA5	65	LA21	105	TA5	125	TA21	145	SA5	165	SA21
0 1 1 0	6	ACK	26	SYN	46	LA6	66	LA22	106	TA6	126	TA22	146	SA6	166	SA22
0 1 1 1	7	BEL	27	ETB	47	LA7	67	LA23	107	TA7	127	TA23	147	SA7	167	SA23
1 0 0 0	8	GET BS	30	SPE CAN	50	LA8	70	LA24	110	TA8	130	TA24	150	SA8	170	SA24
1 0 0 1	9	TCT HT	31	SPD EM	51	LA9	71	LA25	111	TA9	131	TA25	151	SA9	171	SA25
1 0 1 0	10	LF A	32	BUB 2A	52	LA10	72	LA26	112	TA10	132	TA26	152	SA10	172	SA26
1 0 1 1	11	VT B	33	ESC 1B	53	LA11	73	LA27	113	TA11	133	TA27	153	SA11	173	SA27
1 1 0 0	12	FF C	34	FB 1C	54	LA12	74	LA28	114	TA12	134	TA28	154	SA12	174	SA28
1 1 0 1	13	CR D	35	GS 1D	55	LA13	75	LA29	115	TA13	135	TA29	155	SA13	175	SA29
1 1 1 0	14	SO E	36	RS 1E	56	LA14	76	LA30	116	TA14	136	TA30	156	SA14	176	SA30
1 1 1 1	15	SI F	37	US 1F	57	LA15	77	UNL ?	117	TA15 D	137	UNT	157	SA15 o	177	
		ADDRESSED COMMANDS		UNIVERSAL COMMANDS		LISTEN ADDRESSES		TALK ADDRESSES		SECONDARY ADDRESSES OR COMMANDS						



**Tektronix**  
 REF: ANSI STD X3.4-1977  
 IEEE STD 488.1-1987  
 ISO STD 646-2973

---

# Reserved Words

## Overview

This is a list of reserved words for the TDS5000B Series instruments. Capital letters identify the required minimum spelling. For the most robust code, use the full spelling, since spelling rules may change over time and among instrument models.

### \*CAL to DIAG

*CAL	*TRG	BETWeen	CONTROL
*CLS	*TST	BIT_Nr	COPy
*DDT	*WAI	BN_Fmt	COUNT
*ESE	A	BOTH	COUPLing
*ESR	ABSolute	Box	CURSor
*IDN	ACQuire	BOXPent	CURVe
*LRN	ACTUal	BUSY	DATA
*OPC	ALias	BY	DATE
*OPT	ALL	BYT_Nr	DEFIne
*PSC	ALLev	BYT_Or	DELay
*PUD	AREA	CALibrate	DELEte
*RCL	AUTOBright	CENTER	DELTA
*RST	AUTOSet	CH<x>	DELTATime
*SAV	AUXout	CLAss	DESE
*SDS	B	CLOCK	DESKew
*SRE	BANdwidth	CMDBatch	DESTination
*STB	BELI	COLOr	DIAg

---

## DIR to MAXimum

DIR	FASTAcq	HARDCopy	LABEL
DIREction	FILEName	HBArS	LAST
DISplay	FILESystem	HDELTA	LENGth
DIVisions	FILTer	HDR	LESSLimit
EDGE	FIRST	HEADer	LEVEL
EDGE2	FORMat	HIGH	LOCK
ENCdg	FRAME	HIGHLimit	LOGIc
EVENT	FREQuency	HIStoqram	LOOP
EVENTS	FUNCTion	HOLDoff	LOOPS
EVMsg	GAIN	HOLDTime	LOW
EVQty	GATEPOS	HORizontal	LOWLimit
EXECUTE	GATEWIDTH	HPOS<x>	MAG
EXTatten	GATING	ID	MAIn
EXTDBatten	GLItch	IMMed	MATH<x>
EXTUnits	GPIB	INTENSITy	MATHCOLOr
FACtory	GRATICule	INPut	MAXimum
FAILURES	HALT	ITEM	

---

## MEAN to Size

MEAN	NUMITEMS	PRObestate	RESULT
MEAS<x>	OFFSet	PT_Fmt	RESults
MEASUrement	PAIred	PT_Off	RMDir
METHod	PALETTE	PT_OR	RUNT
MID	PASSword	PULse	SAMPLERate
MID2	PATtern	PUMODE	SAVe
MINimum	PERCent	READFile	SCALE
MKDir	PERSistence	RECALL	SCREENSAVER
MODE	PHASE	RECOrdlength	SCREENSAVERDELAY
MORELimit	POLarity	REF	SELEct
NAME	PORT	REF<x>	SELECTED
NEWpass	POSition	REFCOLOr	SERnumber
NEXT	POSition<x>	REFLevel	SET
NR_Pt	PRInt	REName	SETHold
NUMACq	PRObe	REPEt	SETTime
NUMAVg	PROBECal	RESBw	SETUp
NUMEnv	PROBEFunc	RESistance	SIZE

---

## SLOpe to ZOOm

SLOpe	SUPPress	VALue	XINcr
SOURCE	TERminator	VARPersist	XPOS
SOURCE2	TEST	VBArs	XUNit
SPAN	THReshold	VDELTA	XZEro
SPC	TIMe	VERBose	YMUlt
SPECTral	TIMEOut	VOLTage	YOFF
SPLit	TIMEStamp	<wfm>	YPOS
START	TRACk	WAVEform	YUNit
STATE	TRANsition	WAVFrm	YZEro
STATIstics	TRIGBar	WEIghting	ZOOm
STDdev	TRIGger	WFId	
STOP	TRIGIF	WFMInpre	
STOPAfter	TRIGT	WFMOutpre	
STRing	TYPe	WHEn	
STYLE	UNIts	WIDth	
SUBITEMS	UNLock	WINdow	
SUBSYS	UNWRap	WRITEFile	

---

# Factory Default Setup Values

## Default Setup

The following tables list the default setup values by command group. The instrument sets these values when you press the front-panel DEFAULT SETUP button or sending the FACTORY command. Only those commands that have values set by the DEFAULT SETUP function are listed in these tables.

**Note:** The following commands do not affect the DATA parameters: TEKSecure, \*RST, RECALL:SETUP FACTORY. These are the initial factory default settings for DATA :DATA:DESTINATION REF1;ENCDG RIBINARY;SOURCE CH1;START 1;STOP 5000;FRAMESTART 1;FRAMESTOP 5000.

### Acquisition Default Values

The following table lists the Acquisition factory default setup values.

Command	Default Values
ACQUIRE:MODE	SAMPLE
ACQUIRE:NUMAVG	16
ACQUIRE:NUMENV	10
ACQUIRE:NUMSAMPLES	16000
ACQUIRE:SAMPLINGMODE	IT
ACQUIRE:STATE	1 (ON)
ACQUIRE:STOPAFTER	RUNSTOP
FASTACQ:STATE	0

### Alias Default Values

The following table lists the Alias factory default setup values.

Command	Default Values
ALIAS:STATE	0 (OFF)



---

## Cursor Default Values

The following table lists the Cursor factory default setup values.

<b>Command</b>	<b>Default Values</b>
CURSOR:FUNCTION	VBARS
CURSOR:HBARS:POSITION1	300.0000E-3
CURSOR:HBARS:POSITION2	-300.0000E-3
CURSOR:LINESTYLE	SOLID
CURSOR:MODE	INDEPENDENT
CURSOR:PAIRED:POSITION1	-1.6000E-6
CURSOR:PAIRED:POSITION2	1.6000E-6
CURSOR:SCREEN:STYLE	LINES
CURSOR:SOURCE1	CH1
CURSOR:SOURCE2	CH1
CURSOR:STATE	0
CURSOR:VBARS:POSITION1	-1.6000E-6
CURSOR:VBARS:POSITION2	1.6000E-6
CURSOR:VBARS:UNITS	SECONDS
CURSOR:WAVEFORM:POSITION1	-1.6000E-6
CURSOR:WAVEFORM:POSITION2	1.6000E-6
CURSOR:WAVEFORM:SOURCE2	CH1
CURSOR:WAVEFORM:STYLE	LINE_X
CURSOR:WAVEFORM:UNITS	BASE
CURSOR:XY:READOUT	RECTANGULAR
CURSOR:XY:RECTX1	0.0000
CURSOR:XY:RECTX2	0.0000
CURSOR:XY:RECTY1	0.0000
CURSOR:XY:RECTY2	0.0000

---

## Diagnostics Default Values

The following table lists the Diagnostics factory default setup values.

<b>Command</b>	<b>Default Values</b>
DIAG:LEVEL	SUBSYS

## Display Default Values

The following table lists the Display factory default setup values.

<b>Command</b>	<b>Default Values</b>
DISPLAY:CLOCK	1
DISPLAY:COLOR:MATHCOLOR	DEFAULT
DISPLAY:COLOR:PALETTE:IMAGEVIEW	TEMPERATURE
DISPLAY:COLOR:PALETTE:RECORDVIEW	NORMAL
DISPLAY:COLOR:PALETTE:USER:CARET	150,50,100
DISPLAY:COLOR:PALETTE:USER:CH1	180,50,100
DISPLAY:COLOR:PALETTE:USER:CH2	300,50,100
DISPLAY:COLOR:PALETTE:USER:CH3	60,50,100
DISPLAY:COLOR:PALETTE:USER:CH4	240,50,100
DISPLAY:COLOR:PALETTE:USER:GRATICULE	165,50,15
DISPLAY:COLOR:PALETTE:USER:HISTOGRAM	320,50,100
DISPLAY:COLOR:PALETTE:USER:MASK	0,25,75
DISPLAY:COLOR:PALETTE:USER:MASKHIGHLIGHT	140,50,100
DISPLAY:COLOR:PALETTE:USER:MATH1	160,50,100
DISPLAY:COLOR:PALETTE:USER:MATH2	40,60,100
DISPLAY:COLOR:PALETTE:USER:MATH3	120,60,100
DISPLAY:COLOR:PALETTE:USER:MATH4	195,50,100
DISPLAY:COLOR:PALETTE:USER:REF1	0,90,0
DISPLAY:COLOR:PALETTE:USER:REF2	0,90,100
DISPLAY:COLOR:PALETTE:USER:REF3	60,90,100
DISPLAY:COLOR:PALETTE:USER:REF4	240,90,100
DISPLAY:COLOR:REFCOLOR	DEFAULT
DISPLAY:CURSORTICK	SHORT
DISPLAY:FILTER	SINX

---

DISPLAY:FORMAT	YT
DISPLAY:GRATICULE	FULL
DISPLAY:INTENSITY:AUTOBRIGHT	1
DISPLAY:INTENSITY:SCREESAVER	1
DISPLAY:INTENSITY:SCREESAVERDELAY	28800
DISPLAY:INTENSITY:WAVEFORM:IMAGEVIEW	75.0000
DISPLAY:INTENSITY:WAVEFORM:RECORDVIEW	75.0000
DISPLAY:PERSISTENCE	OFF
DISPLAY:SCREENTEXT:LABEL1:NAME	""
DISPLAY:SCREENTEXT:LABEL1:XPOS	100
DISPLAY:SCREENTEXT:LABEL1:YPOS	5
DISPLAY:SCREENTEXT:LABEL2:NAME	""
DISPLAY:SCREENTEXT:LABEL2:XPOS	100
DISPLAY:SCREENTEXT:LABEL2:YPOS	20
DISPLAY:SCREENTEXT:LABEL3:NAME	""
DISPLAY:SCREENTEXT:LABEL3:XPOS	100
DISPLAY:SCREENTEXT:LABEL3:YPOS	35
DISPLAY:SCREENTEXT:LABEL4:NAME	""
DISPLAY:SCREENTEXT:LABEL4:XPOS	100
DISPLAY:SCREENTEXT:LABEL4:YPOS	5
DISPLAY:SCREENTEXT:LABEL5:NAME	""
DISPLAY:SCREENTEXT:LABEL5:XPOS	100
DISPLAY:SCREENTEXT:LABEL5:YPOS	65
DISPLAY:SCREENTEXT:LABEL6:NAME	""
DISPLAY:SCREENTEXT:LABEL6:XPOS	100
DISPLAY:SCREENTEXT:LABEL6:YPOS	80
DISPLAY:SCREENTEXT:LABEL7:NAME	""
DISPLAY:SCREENTEXT:LABEL7:XPOS	100
DISPLAY:SCREENTEXT:LABEL7:YPOS	95
DISPLAY:SCREENTEXT:LABEL8:NAME	""
DISPLAY:SCREENTEXT:LABEL8:XPOS	100

---

DISPLAY:SCREENTEXT:LABEL8:YPOS	110
DISPLAY:SCREENTEXT:STATE	0
DISPLAY:STYLE	VECTORS
DISPLAY:TRIGBAR	SHORT
DISPLAY:TRIGT	1
DISPLAY:VARPERSIST	500.0000E-3

## Email Default Values

The following table lists the Email setup values.

Command	Default Values
EMAIL:ATTEMPTS	1
EMAIL:AUTHLOGIN	""
EMAIL:FROM	"TDS5054B_<Instrument Serial Number>"
EMAIL:HOSTWANTED	""
EMAIL:IMAGE	0
EMAIL:LIMIT	0
EMAIL:MASK	0
EMAIL:MAXSIZE	1
EMAIL:MEASUREMENT	0
EMAIL:NUMEMAILS	1
EMAIL:SMTPPORT	25
EMAIL:SMTPSERVER	""
EMAIL:TIMEOUT	30
EMAIL:TO	""
EMAIL:TRIGGER	0
EMAIL:WAVEFORM	0

---

## Hardcopy Default Values

The following table lists the Hardcopy factory default setup values.

Command	Default Values
HARDCopy:FILENAME	""
HARDCopy:PORT	FILE

## Histogram Default Values

The following table lists the Histogram factory default setup values.

Command	Default Values
HISTOGRAM:BOXPCNT	30.0000,25.1000,70.0000,75.2000
HISTOGRAM:DISPLAY	LINEAR
HISTOGRAM:FUNCTION	HORIZONTAL
HISTOGRAM:SIZE	2.0000
HISTOGRAM:SOURCE	CH1
HISTOGRAM:STATE	0

## Horizontal Default Values

The following table lists the Horizontal factory default setup values.

Command	Default Values
HORIZONTAL:DELAY:MODE	0
HORIZONTAL:DELAY:POSITION	50.0000
HORIZONTAL:DELAY:TIME	0.000000000000
HORIZONTAL:FASTFRAME:COUNT	2
HORIZONTAL:FASTFRAME:LENGTH	5000
HORIZONTAL:FASTFRAME:MULTIPLEFRAMES :FRAMESTART:CH<1-4>	1
HORIZONTAL:FASTFRAME:MULTIPLEFRAMES :FRAMESTART:MATH<1-4>	1
HORIZONTAL:FASTFRAME:MULTIPLEFRAMES :FRAMESTART:REF<1-4>	1
HORIZONTAL:FASTFRAME:MULTIPLEFRAMES :MODE	OFF
HORIZONTAL:FASTFRAME:MULTIPLEFRAMES :NUMFRAMES:CH1-4>	2
HORIZONTAL:FASTFRAME:MULTIPLEFRAMES :NUMFRAMES:MATH<1-4>	2

---

HORIZONTAL:FASTFRAME:MULTIPLEFRAMES :NUMFRAMES:REF<1-4>	2
HORIZONTAL:FASTFRAME:READOUTS	0
HORIZONTAL:FASTFRAME:REF:FRAME	1
HORIZONTAL:FASTFRAME:REF:SOURCE	CH1
HORIZONTAL:FASTFRAME:SELECTED:CH1	2
HORIZONTAL:FASTFRAME:SELECTED:CH2	2
HORIZONTAL:FASTFRAME:SELECTED:CH3	2
HORIZONTAL:FASTFRAME:SELECTED:CH4	2
HORIZONTAL:FASTFRAME:SELECTED:MATH1	2
HORIZONTAL:FASTFRAME:SELECTED:MATH2	2
HORIZONTAL:FASTFRAME:SELECTED:MATH3	2
HORIZONTAL:FASTFRAME:SELECTED:MATH4	2
HORIZONTAL:FASTFRAME:SELECTED:REF1	2
HORIZONTAL:FASTFRAME:SELECTED:REF2	2
HORIZONTAL:FASTFRAME:SELECTED:REF3	2
HORIZONTAL:FASTFRAME:SELECTED:REF4	2
HORIZONTAL:FASTFRAME:STATE	0
HORIZONTAL:FASTFRAME:SUMFRAME	NONE
HORIZONTAL:FASTFRAME:TRACK	LIVE
HORIZONTAL:MAIN:POSITION	50.0000
HORIZONTAL:MAIN:SAMPLERATE	1.2500E+9
HORIZONTAL:MAIN:SCALE	400.0000E-9
HORIZONTAL:RECORDLENGTH	5000
HORIZONTAL:RESOLUTION	5000
HORIZONTAL:ROLL	AUTO

---

## Limit Test Default Values

The following table lists the Limit Test factory default setup values.

Command	Default Values
LIMIT:BEEP	0
LIMIT:COMPARE:<CH1-4>	NONE
LIMIT:COMPARE:MATH<1-4>	NONE
LIMIT:COMPARE:REF<1-4>	NONE
LIMIT:EMAIL	0
LIMIT:HARDCOPY	0
LIMIT:HIGHLIGHTHITS	1
LIMIT:LOCK	1
LIMIT:LOG	0
LIMIT:SAVEWFM	0
LIMIT:SRQ	0
LIMIT:STATE	0
LIMIT:STOPONVIOLATION	0
LIMIT:TEMPLATE:TOLERANCE:HORIZONTAL	40.0000E-3
LIMIT:TEMPLATE:TOLERANCE:VERTICAL	40.0000E-3

## Mask Default Values

The following table lists the Mask factory default setup values.

Command	Default Values
MASK:AUTOADJUST	10
MASK:AUTOADJUST:HDELTA	10
MASK:AUTOADJUST:VDELTA	10
MASK:AUTOSET:AUTOADJUST	1
MASK:AUTOSET:HPOS	1
MASK:AUTOSET:HSCALE	1
MASK:AUTOSET:MODE	MANUAL
MASK:AUTOSET:OFFSETADJ	1
MASK:AUTOSET:STANDARD	NONE
MASK:AUTOSET:TRIGGER	1

---

MASK:AUTOSET:VPOS	1
MASK:AUTOSET:VSCALE	1
MASK:COUNT:STATE	0
MASK:DISPLAY	1
MASK:FILTER	0
MASK:HIGHLIGHTHITS	1
MASK:INVERT	0
MASK:LOCK	1
MASK:MARGIN:PERCENT	5.0000
MASK:MARGIN:STATE	0
MASK:MASKPRE:AMPLITUDE	100.0000E-3
MASK:MASKPRE:HSCALE	200.0000E-9
MASK:MASKPRE:HTRIGPOS	500.0000E-3
MASK:MASKPRE:PATTERNBITS	1
MASK:MASKPRE:PRESAMPBITS	0
MASK:MASKPRE:RECORDLENGTH	5000
MASK:MASKPRE:TRIGTOSAMP	0.0000
MASK:MASKPRE:VOFFSET	0.0000
MASK:MASKPRE:VPOS	0.0000
MASK:MASKPRE:VSCALE	1.0000E-3
MASK:MASKPRE:WIDTH	0.0000
MASK:POLARITY	POSITIVE
MASK:SOURCE	CH1
MASK:STANDARD	NONE
MASK:STOPONVIOLATION	0
MASK:TEST:BEEP:COMPLETION	0
MASK:TEST:BEEP:FAILURE	0
MASK:TEST:DELAY	0.0000
MASK:TEST:HARDCOPY	0
MASK:TEST:LOG:FAILURE	0
MASK:TEST:REPEAT	0



---

MASK:TEST:SAMPLE:THRESHOLD	1
MASK:TEST:SAVEWFM	0
MASK:TEST:SAVEWFM:FILENAME	"C:/TekScope/Waveforms/"
MASK:TEST:SRQ:COMPLETION	0
MASK:TEST:SRQ:FAILURE	0
MASK:TEST:STATE	0
MASK:TEST:STOP:FAILURE	0
MASK:TEST:THRESHOLD	1
MASK:TEST:WAVEFORM	20
MASK:USER:AMPLITUDE	100.0000E-3
MASK:USER:BITRATE	1544000
MASK:USER:HSCALE	200.0000E-9
MASK:USER:HTRIGPOS	500.0000E-3
MASK:USER:LABEL	"User Mask"
MASK:USER:PATTERNBITS	1
MASK:USER:PRESAMPBITS	0
MASK:USER:RECORDLENGTH	5000
MASK:USER:TRIGTOSAMP	0.0000
MASK:USER:VOFFSET	0.0000
MASK:USER:VPOS	0.0000
MASK:USER:VSCALE	1.0000E-3
MASK:USER:WIDTH	0.0000

## Math Default Values

The following table lists the Math factory default setup values.

Command	Default Values
MATH<1-4>:DEFINE	" "
MATH<1-4>:LABEL:NAME	" "
MATH<1-4>:LABEL:XPOS	1.0000
MATH<1-4>:LABEL:YPOS	<1> 600.0000E-3 <2> 200.0000E-3 <3> -200.0000E-3

---

	<4>	-600.0000E-3
MATH<1-4>::NUMAVG		2
MATH<1-4>:SPECTRAL:CENTER		312.5000E+6
MATH<1-4>:SPECTRAL:GATEPOS		0.0000
MATH<1-4>:SPECTRAL:GATEWIDTH		2.0000E-6
MATH<1-4>:SPECTRAL:LOCK		0
MATH<1-4>:SPECTRAL:MAG		DB
MATH<1-4>:SPECTRAL:PHASE		DEGREES
MATH<1-4>:SPECTRAL:REFLEVEL		20.0000
MATH<1-4>:SPECTRAL:REFLEVELOFFSET		223.6000E-3
MATH<1-4>:SPECTRAL:RESBW		1000.0000E+3
MATH<1-4>:SPECTRAL:SPAN		625.0000E+6
MATH<1-4>:SPECTRAL:SUPPRESS		-200.0000
MATH<1-4>:SPECTRAL:UNWRAP		0
MATH<1-4>:SPECTRAL:WINDOW		GAUSSIAN
MATH<1-4>:UNITSTRING		"V"
MATH<1-4>:VERTICAL:POSITION		0.0000
MATH<1-4>:VERTICAL:SCALE		1.0000
MATHVAR:VAR<1-8>		0.0000

## Measurement Default Values

The following table lists the Measurement factory default setup values.

Command	Default Values
MEASUREMENT:GATING	OFF
MEASUREMENT:IMMED:DELAY:DIRECTION	FORWARDS
MEASUREMENT:IMMED:DELAY:EDGE1	RISE
MEASUREMENT:IMMED:DELAY:EDGE2	RISE
MEASUREMENT:IMMED:METHOD	HISTOGRAM
MEASUREMENT:IMMED:NOISE	HIGH
MEASUREMENT:IMMED:REFLEVEL:ABSOLUTE:HIGH	0.0000
MEASUREMENT:IMMED:REFLEVEL:ABSOLUTE:LOW	0.0000
MEASUREMENT:IMMED:REFLEVEL:ABSOLUTE:MID1	0.0000

---

MEASUREMENT:IMMED:REFLEVEL:ABSOLUTE:MID2	0.0000
MEASUREMENT:IMMED:REFLEVEL:METHOD	PERCENT
MEASUREMENT:IMMED:REFLEVEL:PERCENT:HIG	90.0000
MEASUREMENT:IMMED:REFLEVEL:PERCENT:LOW	10.0000
MEASUREMENT:IMMED:REFLEVEL:PERCENT:MID1	50.0000
MEASUREMENT:IMMED:REFLEVEL:PERCENT:MID2	50.0000
MEASUREMENT:IMMED:SOURCE1	CH1
MEASUREMENT:IMMED:SOURCE1:SIGTYPE	PULSE
MEASUREMENT:IMMED:SOURCE2	CH1
MEASUREMENT:IMMED:SOURCE2:SIGTYPE	PULSE
MEASUREMENT:IMMED:TYPE	UNDEFINED
MEASUREMENT:MEAS<1-8>:DELAY:DIRECTION	FORWARDS
MEASUREMENT:MEAS<1-8>:DELAY:EDGE1	RISE
MEASUREMENT:MEAS<1-8>:DELAY:EDGE2	RISE
MEASUREMENT:MEAS<1-8>:METHOD	HISTOGRAM
MEASUREMENT:MEAS<1-8>:NOISE	HIGH
MEASUREMENT:MEAS<1-8>:REFLEVEL:ABSOLUTE:HIG	0.0000
MEASUREMENT:MEAS<1-8>:REFLEVEL:ABSOLUTE:LOW	0.0000
MEASUREMENT:MEAS<1-8>:REFLEVEL:ABSOLUTE:MID1	0.0000
MEASUREMENT:MEAS<1-8>:REFLEVEL:ABSOLUTE:MID2	0.0000
MEASUREMENT:MEAS<1-8>:REFLEVEL:METHOD	PERCENT
MEASUREMENT:MEAS<1-8>:REFLEVEL:PERCENT:HIG9	0.0000
MEASUREMENT:MEAS<1-8>:REFLEVEL:PERCENT:LOW1	0.0000
MEASUREMENT:MEAS<1-8>:REFLEVEL:PERCENT:MID15	0.0000
MEASUREMENT:MEAS<1-8>:REFLEVEL:PERCENT:MID2	50.0000
MEASUREMENT:MEAS<1-8>:SOURCE1	CH1
MEASUREMENT:MEAS<1-8>:SOURCE1:SIGTYPE	PULSE
MEASUREMENT:MEAS<1-8>:SOURCE2	CH1
MEASUREMENT:MEAS<1-8>:SOURCE2:SIGTYPE	PULSE
MEASUREMENT:MEAS<1-8>:STATE	0
MEASUREMENT:MEAS<1-8>:TYPE	UNDEFINED

MEASUREMENT:STATISTICS:MODE	ALL
MEASUREMENT:STATISTICS:WEIGHTING	32

## Miscellaneous Default Values

The following table lists the Miscellaneous factory default setup values.

Command	Default Values
ALIAS:STATE	0 (OFF)
AUXOUT:EDGE	FALLING <cannot change>
AUXOUT:SOURCE	ATRIGGER
CMDBATCH	1 (ON)
HEADER	1 (ON)
LOCK	NONE
ROSC:SOURCE	INTERNAL
VERBOSE	1 (ON)

## Save and Recall Default Settings

The following table lists the Save and Recall factory default setup values.

Command	Default Values
SAVE:WAVEFORM:FILEFORMAT	INTERNAL

## Trigger Default Values

The following table lists the Trigger factory default setup values.

Command	Default Values
TRIGGER:A:COMMUNICATION:AMI:PULSEFORM	PLUSONE
TRIGGER:A:COMMUNICATION:AMI:THRESHOLD:HIGH	0.0000
TRIGGER:A:COMMUNICATION:AMI:THRESHOLD:LOW	0.0000
TRIGGER:A:COMMUNICATION:BITRATE	1544000
TRIGGER:A:COMMUNICATION:CLOCK:POLARITY	RISE
TRIGGER:A:COMMUNICATION:CMI:PULSEFORM	PLUSONE
TRIGGER:A:COMMUNICATION:CODE	NRZ
TRIGGER:A:COMMUNICATION:SOURCE	CH1
TRIGGER:A:COMMUNICATION:SOURCE:TYPE	DATA

---

TRIGGER:A:COMMUNICATION:STANDARD	CUSTOM
TRIGGER:A:EDGE:COUPLING	DC
TRIGGER:A:EDGE:SLOPE	RISE
TRIGGER:A:EDGE:SOURCE	CH1
TRIGGER:A:HOLDOFF:BY	DEFAULT
TRIGGER:A:HOLDOFF:TIME	1.5000E-6
TRIGGER:A:LEVEL	0.0000
TRIGGER:A:LOGIC:CLASS	PATTERN
TRIGGER:A:LOGIC:FUNCTION	AND
TRIGGER:A:LOGIC:INPUT:CH1	HIGH
TRIGGER:A:LOGIC:INPUT:CH2	X
TRIGGER:A:LOGIC:INPUT:CH3	X
TRIGGER:A:LOGIC:PATTERN:INPUT:CH4	X
TRIGGER:A:LOGIC:PATTERN:WHEN	TRUE
TRIGGER:A:LOGIC:PATTERN:WHEN:LESSLIMIT	5.0000E-9
TRIGGER:A:LOGIC:PATTERN:WHEN:MORELIMIT	5.0000E-9
TRIGGER:A:LOGIC:SETHOLD:CLOCK:EDGE	RISE
TRIGGER:A:LOGIC:SETHOLD:CLOCK:SOURCE	CH2
TRIGGER:A:LOGIC:SETHOLD:CLOCK:THRESHOLD	1.2000
TRIGGER:A:LOGIC:SETHOLD:DATA:SOURCE	CH1
TRIGGER:A:LOGIC:SETHOLD:DATA:THRESHOLD	1.2000
TRIGGER:A:LOGIC:SETHOLD:HOLDTIME	2.0000E-9
TRIGGER:A:LOGIC:SETHOLD:SETTIME	3.0000E-9
TRIGGER:A:LOGIC:STATE:INPUT:CH4	RISE
TRIGGER:A:LOGIC:STATE:WHEN	TRUE
TRIGGER:A:LOGIC:THRESHOLD:<CH1-4>	1.2000
TRIGGER:A:MODE	AUTO
TRIGGER:A:PULSE:CLASS	GLITCH
TRIGGER:A:PULSE:GLITCH:POLARITY	POSITIVE
TRIGGER:A:PULSE:GLITCH:TRIGIF	ACCEPT
TRIGGER:A:PULSE:GLITCH:WIDTH	2.0000E-9

---

TRIGGER:A:PULSE:RUNT:LOGIC:INPUT:CH<1-4>	HIGH
TRIGGER:A:PULSE:RUNT:LOGIC:THRESH- OLD:CH<1-4>	1.2000
TRIGGER:A:PULSE:RUNT:POLARITY	POSITIVE
TRIGGER:A:PULSE:RUNT:THRESHOLD:HIGH	1.2000
TRIGGER:A:PULSE:RUNT:THRESHOLD:LOW	800.0000E-3
TRIGGER:A:PULSE:RUNT:WHEN	OCCURS
TRIGGER:A:PULSE:RUNT:WIDTH	2.0000E-9
TRIGGER:A:PULSE:SOURCE	CH1
TRIGGER:A:PULSE:TIMEOUT:POLARITY	STAYSHIGH
TRIGGER:A:PULSE:TIMEOUT:TIME	2.0000E-9
TRIGGER:A:PULSE:TRANSITION:DELTATIME	2.0000E-9
TRIGGER:A:PULSE:TRANSITION:POLARITY	POSITIVE
TRIGGER:A:PULSE:TRANSITION:THRESHOLD:HIGH	1.2000
TRIGGER:A:PULSE:TRANSITION:THRESHOLD:LOW	800.0000E-3
TRIGGER:A:PULSE:TRANSITION:WHEN	FASTERTHAN
TRIGGER:A:PULSE:WIDTH:HIGHLIMIT	2.0000E-9
TRIGGER:A:PULSE:WIDTH:LOWLIMIT	2.0000E-9
TRIGGER:A:PULSE:WIDTH:POLARITY	POSITIVE
TRIGGER:A:PULSE:WIDTH:WHEN	WITHIN
TRIGGER:A:PULSE:WINDOW:LOGIC:INPUT:CH<1-4>	HIGH
TRIGGER:A:PULSE:WINDOW:LOGIC:THRESH- OLD:<CH1-4>	1.2000
TRIGGER:A:PULSE:WINDOW:THRESHOLD:HIGH	1.2000
TRIGGER:A:PULSE:WINDOW:THRESHOLD:LOW	800.0000E-3
TRIGGER:A:PULSE:WINDOW:TYPE	INSIDE
TRIGGER:A:PULSE:WINDOW:WHEN	OCCURS
TRIGGER:A:PULSE:WINDOW:WIDTH	2.0000E-9
TRIGGER:A:TYPE	EDGE
TRIGGER:A:VIDEO:CUSTOM:FORMAT	INTERLACED
TRIGGER:A:VIDEO:CUSTOM:SCAN	RATE1
TRIGGER:A:VIDEO:FIELD	ALLFIELDS

---

TRIGGER:A:VIDEO:HOLDOFF:FIELD	0.0000
TRIGGER:A:VIDEO:LINE	1
TRIGGER:A:VIDEO:POLARITY	NORMAL
TRIGGER:A:VIDEO:SCAN	RATE1
TRIGGER:A:VIDEO:SOURCE	CH1
TRIGGER:A:VIDEO:STANDARD	NTSC
TRIGGER:B:BY	EVENTS
TRIGGER:B:EDGE:COUPLING	DC
TRIGGER:B:EDGE:SLOPE	RISE
TRIGGER:B:EDGE:SOURCE	CH1
TRIGGER:B:EVENTS:COUNT	2
TRIGGER:B:LEVEL	0.0000
TRIGGER:B:STATE	0
TRIGGER:B:TIME	16.0000E-9
TRIGGER:B:TYPE	EDGE

### Vertical Default Values

The following table lists the Vertical factory default setup values.

Command	Default Values
CH<1-4>:BANDWIDTH	500.0000E+6
CH<1-4>:COUPLING	DC
CH<1-4>:DESKEW	0.0000
CH<1-4>:INVERT	0
CH<1-4>:LABEL:NAME	""
CH<1-4>:LABEL:XPOS	1.0000
CH<1-4>:LABEL:YPOS	<1> 2.2000 <2> 1.8000 <3> 1.4000 <4> 1.000
CH<1-4>:OFFSET	0.0000
CH<1-4>:POSITION	0.0000
CH<1-4>:PROBEFUNC:EXTATTEN	1.0000

---

CH<1-4>:PROBEFUNC:EXTUNITS	"V"
CH<1-4>:SCALE	100.0000E-3
CH<1-4>:TERMINATION	1.0000E+6
DATA:DESTINATION	REF1
DATA:ENCDG	RIBINARY
DATA:FRAMESTART	1
DATA:FRAMESTOP	5000
DATA:SOURCE	CH1
DATA:START	1
DATA:STOP	5000
REF1:LABEL:NAME	" "
REF1:LABEL:XPOS	1.0000
REF1:LABEL:YPOS	-1.0000
REF2:LABEL:NAME	" "
REF2:LABEL:XPOS	1.0000
REF2:LABEL:YPOS	-1.4000
REF3:LABEL:NAME	" "
REF3:LABEL:XPOS	1.0000
REF3:LABEL:YPOS	-1.8000
REF4:LABEL:NAME	" "
REF4:LABEL:XPOS	1.0000
REF4:LABEL:YPOS	-2.2000
SELECT:CH1	1
SELECT:CH2	0
SELECT:CH3	0
SELECT:CH4	0
SELECT:CONTROL	CH1
SELECT:MATH<1-4>	0
SELECT:REF<1-4>	0



---

## Waveform Default Values

The following table lists the Waveform factory default setup values.

<b>Command</b>	<b>Default Values</b>
DATA:DESTINATION	REF1
DATA:ENCDG	RIBINARY
DATA:FRAMESTART	1
DATA:FRAMESTOP	5000
DATA:SOURCE	CH1
DATA:START	1
DATA:STOP	5000

## Zoom Default Values

The following table lists the Zoom factory default setup values.

<b>Command</b>	<b>Default Values</b>
ZOOM:GRATICULE:SIZE	80
ZOOM:GRATICULE:SPLIT	EIGHTYTWENTY
ZOOM:MODE	0
ZOOM:SCROLL:DIRECTION	STOP
ZOOM:SCROLL:LOCK	0
ZOOM:SCROLL:SPEED	1
ZOOM:ZOOM1:CH<1-4>:HORIZONTAL:POSITION	50.0000
ZOOM:ZOOM1:CH<1-4>:HORIZONTAL:SCALE	5
ZOOM:ZOOM1:CH<1-4>:VERTICAL:POSITION	0.0000
ZOOM:ZOOM1:CH<1-4>:VERTICAL:SCALE	1.0000
ZOOM:ZOOM1:MATH<1-4>:HORIZONTAL:POSITION	50.0000
ZOOM:ZOOM1:MATH<1-4>:HORIZONTAL:SCALE	5
ZOOM:ZOOM1:MATH<1-4>:VERTICAL:POSITION	0.0000
ZOOM:ZOOM1:MATH<1-4>:VERTICAL:SCALE	1.0000
ZOOM:ZOOM1:REF<1-4>:HORIZONTAL:POSITION	50.0000
ZOOM:ZOOM1:REF<1-4>:HORIZONTAL:SCALE	5
ZOOM:ZOOM1:REF<1-4>:VERTICAL:POSITION	0.0000
ZOOM:ZOOM1:REF<1-4>:VERTICAL:SCALE	1.0000

---

ZOOM:ZOOM1:SCROLLLOCK	1
ZOOM:ZOOM1:STATE	1
ZOOM:ZOOM2:CH<1-4>:HORIZONTAL:POSITION	30.0000
ZOOM:ZOOM2:CH<1-4>:HORIZONTAL:SCALE	5
ZOOM:ZOOM2:CH<1-4>:VERTICAL:POSITION	0.0000
ZOOM:ZOOM2:CH<1-4>:VERTICAL:SCALE	1.0000
ZOOM:ZOOM2:MATH<1-4>:HORIZONTAL:POSITION	30.0000
ZOOM:ZOOM2:MATH<1-4>:HORIZONTAL:SCALE	5
ZOOM:ZOOM2:MATH<1-4>:VERTICAL:POSITION	0.0000
ZOOM:ZOOM2:MATH<1-4>:VERTICAL:SCALE	1.0000
ZOOM:ZOOM2:REF<1-4>:HORIZONTAL:POSITION	30.0000
ZOOM:ZOOM2:REF<1-4>:HORIZONTAL:SCALE	5
ZOOM:ZOOM2:REF<1-4>:VERTICAL:POSITION	0.0000
ZOOM:ZOOM2:REF<1-4>:VERTICAL:SCALE	1.0000
ZOOM:ZOOM2:SCROLLLOCK	1
ZOOM:ZOOM2:STATE	0
ZOOM:ZOOM3:CH<1-4>:HORIZONTAL:POSITION	70.0000
ZOOM:ZOOM3:CH<1-4>:HORIZONTAL:SCALE	5
ZOOM:ZOOM3:CH<1-4>:VERTICAL:POSITION	0.0000
ZOOM:ZOOM3:CH<1-4>:VERTICAL:SCALE	1.0000
ZOOM:ZOOM3:MATH<1-4>:HORIZONTAL:POSITION	70.0000
ZOOM:ZOOM3:MATH<1-4>:HORIZONTAL:SCALE	5
ZOOM:ZOOM3:MATH<1-4>:VERTICAL:POSITION	0.0000
ZOOM:ZOOM3:MATH<1-4>:VERTICAL:SCALE	1.0000
ZOOM:ZOOM3:REF<1-4>:HORIZONTAL:POSITION	70.0000
ZOOM:ZOOM3:REF<1-4>:HORIZONTAL:SCALE	5
ZOOM:ZOOM3:REF<1-4>:VERTICAL:POSITION	0.0000
ZOOM:ZOOM3:REF<1-4>:VERTICAL:SCALE	1.0000
ZOOM:ZOOM3:SCROLLLOCK	1
ZOOM:ZOOM3:STATE	0
ZOOM:ZOOM4:CH<1-4>:HORIZONTAL:POSITION	90.0000

---

ZOOM:ZOOM4:CH<1-4>:HORIZONTAL:SCALE	5
ZOOM:ZOOM4:CH<1-4>:VERTICAL:POSITION	0.0000
ZOOM:ZOOM4:CH<1-4>:VERTICAL:SCALE	1.0000
ZOOM:ZOOM4:MATH<1-4>:HORIZONTAL:POSITION	90.0000
ZOOM:ZOOM4:MATH<1-4>:HORIZONTAL:SCALE	5
ZOOM:ZOOM4:MATH<1-4>:VERTICAL:POSITION	0.0000
ZOOM:ZOOM4:MATH<1-4>:VERTICAL:SCALE	1.0000
ZOOM:ZOOM4:REF<1-4>:HORIZONTAL:POSITION	90.0000
ZOOM:ZOOM4:REF<1-4>:HORIZONTAL:SCALE	5
ZOOM:ZOOM4:REF<1-4>:VERTICAL:POSITION	0.0000
ZOOM:ZOOM4:REF<1-4>:VERTICAL:SCALE	1.0000
ZOOM:ZOOM4:SCROLLLOCK	1
ZOOM:ZOOM4:STATE	0

## GPIB Interface Specifications

### Overview

This topic describes details of the GPIB remote interface of the instrument. Normally, you will not need this information to use the instrument, but this information may be useful if you are connecting to controllers with unusual configurations.

### GPIB Functions

The following table lists the GPIB interface functions and electrical function subsets supported by this instrument and a brief description of each function.

<b>GPIB Functions</b>		
<b>Interface Function</b>	<b>Subset</b>	<b>Description</b>
Acceptor Handshake	AH1	The instrument can receive multiline messages across the GPIB from other devices.
Controller	C0	This instrument can be operated as a device or as a controller.
Device Clear	DC1	The instrument can respond to both the DCL (Device Clear) interface message and to the Selected Device Clear (SDC) interface message when the instrument is listen-addressed.
Device Trigger	DT1	Device Trigger capability; the instrument does respond to the GET (Group Execute Trigger) interface message.
Electrical	E2	The instrument uses tri-state buffers, which are optimal for high-speed data transfer.
Listener	L4	The instrument becomes a listener when it detects the listen address being sent over the bus with the ATN line asserted. The instrument ceases to be a listener and becomes a talker when it detects the talk address being sent over the bus with the ATN line asserted.
Parallel Poll	PP0	No Parallel Poll capability; the instrument does not respond to PPC (Parallel Poll Configure), PPD (Parallel Poll Disable), PPE (Parallel Poll Enable), or PPU (Parallel Poll Unconfigure) interface messages, nor does it send a status message when the ATN and EOI lines are asserted simultaneously.
Remote/Local	RL1	The instrument can respond to both the GTL (Go To Local) and LLO (Local Lock Out) interface messages.
Service Request	SR1	The instrument can assert the SRQ line to notify the controller in charge that it requires service.
Source Handshake	SH1	The instrument can initiate multiline messages to send across the GPIB to other devices.
Talker	T5	The instrument becomes a talker when it detects the talk address being sent over the bus with the ATN line asserted. The instrument ceases to be a talker and becomes a listener when it detects the listen address being sent over the bus with the ATN line asserted. The instrument also ceases to be a talker when it detects the talk address of another device being sent over the data lines with ATN asserted.

---

## Interface Messages

The following table shows the standard interface messages that the instrument supports.

Standard Interface Messages	
Message	GPIB
DCL	Yes
GET	Yes
GTL	Yes
LLO	Yes
PPC	No*
PPD	No*
PPE	No*
PPU	No*
SDC	Yes
SPD	Yes
SPE	Yes
TCT	No*
UNL	Yes
UNT	Yes
Listen Addresses	Yes
Talk Addresses	Yes

\*This instrument may be operated as a GPIB controller or as a device. When Operated in controller mode, your controller software should respond to these interface messages. The instrument will not respond to these messages when the GPIB port is used as a device.