Agilent InfiniiVision Series Oscilloscope Command Summary

Table 1. Common (*) Commands Summary

Command	Query	Options and Query Returns	Dependencies
*CLS	n/a	n/a	
*ESE <mask>	*ESE?	<mask> ::= 0 to 255; an integer in NR1 format: Bit Weight Name Enables --- ------ ---- ---------- 7 128 PON Power On 6 64 URQ User Request 5 32 CME Command Error 4 16 EXE Execution Error 3 8 DDE Dev. Dependent Error 2 4 QYE Query Error 1 2 RQL Request Control 0 1 OPC Operation Complete	

Command	Query	Options and Query Returns	Dependencies
n/a	*ESR?	<status> ::= 0 to 255; an integer in NR1 format	
n/a	*IDN?	AGILENT TECHNOLOGIES,<model>,<serial number>, X.XX.XX <model> ::= the model number of the instrument <serial number> ::= the serial number of the instrument <X.XX.XX> ::= the software revision of the instrument	
n/a	*LRN?	<learn_string> ::= current instrument setup as a block of data in IEEE 488.2 # format	
*OPC	*OPC?	ASCII "1" is placed in the output queue when all pending device operations have completed.	
n/a	*OPT?	<return_value> ::= 0,0,<license info> <license info> ::= <All field>, <reserved>, <Factory MSO>, <Upgraded MSO>, <Xilinx FPGA Probe>, <Memory>, <Low Speed Serial>, <Automotive Serial>, <reserved>, <Secure>, <Battery>, <Altera FPGA Probe>, <FlexRay Serial>, <Power Measurements>, <RS-232/UART Serial>, <reserved>, <Segmented Memory>, <Mask Test>, <reserved> <All field> ::= {0 | All} <reserved> ::= 0 <Factory MSO> ::= {0 | MSO} <Upgraded MSO> ::= {0 | MSO} <Xilinx FPGA Probe> ::= {0 | FPG} <Memory> ::= {0 | mem2M | mem8M} <Low Speed Serial> ::= {0 | LSS} <Automotive Serial> ::= {0 | AMS} <Secure> ::= {0 | SEC} <Battery> ::= {0 | BAT} (6000 Series only) <Altera FPGA Probe> ::= {0 | ALT} <FlexRay Serial> ::= {0 | FRS} <Power Measurements> ::= {0 | PWR} <RS-232/UART Serial> ::= {0 | 232} <Segmented Memory> ::= {0 | SGM} <Mask Test> ::= {0 | LMT}	6000/7000 Series[1] MSO[3] LSS[4] AMS[5] FRS[6] 232[7] SGM[8] LMT[9]
*RCL <value>	n/a	<value> ::= {0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9}	
*RST	n/a	See *RST (Reset)	
*SAV <value>	n/a	<value> ::= {0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9}	
*SRE <mask>	*SRE?	<mask> ::= sum of all bits that are set, 0 to 255; an integer in NR1 format. <mask> ::= following values: Bit Weight Name Enables --- ------ ---- ---------- 7 128 OPER Operation Status Reg 6 64 ---- (Not used.)	

Command	Query	Options and Query Returns	Dependencies
		5 32 ESB Event Status Bit 4 16 MAV Message Available 3 8 ---- (Not used.) 2 4 MSG Message 1 2 USR User 0 1 TRG Trigger	
n/a	*STB?	<value> ::= 0 to 255; an integer in NR1 format, as shown in the following: Bit Weight Name "1" Indicates --- ------ ---- --------------- 7 128 OPER Operation status condition occurred. 6 64 RQS/ MSS Instrument is requesting service. 5 32 ESB Enabled event status condition occurred. 4 16 MAV Message available. 3 8 ---- (Not used.) 2 4 MSG Message displayed. 1 2 USR User event condition occurred. 0 1 TRG A trigger occurred.	
*TRG	n/a	n/a	
n/a	*TST?	<result> ::= 0 or non-zero value; an integer in NR1 format	
*WAI	n/a	n/a	

Table 2. Root (:) Commands Summary

Command	Query	Options and Query Returns	Dependencies
:ACTivity	:ACTivity?	<return value> ::= <edges>,<levels> <edges> ::= presence of edges (32-bit integer in NR1 format) <levels> ::= logical highs or lows (32-bit integer in NR1 format)	
n/a	:AER?	{0 | 1}; an integer in NR1 format	
:AUToscale [<source>[,..,<source>]]	n/a	<source> ::= {CHANnel<n> | DIGital0,..,DIGital15 | POD{1 | 2}} <source> can be repeated up to 5 times <n> ::= 1-2 or 1-4 in NR1 format	MSO[3]
:AUToscale:AMODE <value>	:AUToscale:AMODE?	<value> ::= {NORMal | CURRent}	
:AUToscale:CHANnels <value>	:AUToscale:CHANnels?	<value> ::= {ALL | DISPlayed}	
:BLANk [<source>]	n/a	<source> ::= {CHANnel<n>} | FUNCtion | MATH | SBUS | DIGital0,..,DIGital15 | POD{1 | 2} | BUS{1 | 2}} <n> ::= 1-2 or 1-4 in NR1 format	4-Channel[2] MSO[3]
:CDISplay	n/a	n/a	
:DIGitize [<source>[,..,<source>]]	n/a	<source> ::= {CHANnel<n>} | FUNCtion | MATH | SBUS | DIGital0,..,DIGital15 | POD{1 | 2} | BUS{1 | 2}} <source> can be repeated up to 5 times	4-Channel[2] MSO[3]

Command	Query	Options and Query Returns	Dependencies
		<n> ::= 1-2 or 1-4 in NR1 format	
:HWEenable <n>	:HWEenable?	<n> ::= 16-bit integer in NR1 format	
n/a	:HWERregister:CONDition?	<n> ::= 16-bit integer in NR1 format	
n/a	:HWERegister[:EVENt]?	<n> ::= 16-bit integer in NR1 format	
:MERGe <pixel memory>	n/a	<pixel memory> ::= {PMEMory{0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9}}	
:MTEenable <n>	:MTEenable?	<n> ::= 16-bit integer in NR1 format	LMT[9]
n/a	:MTERegister[:EVENt]?	<n> ::= 16-bit integer in NR1 format	LMT[9]
:OPEE <n>	:OPEE?	<n> ::= 16-bit integer in NR1 format	
n/a	:OPERregister:CONDition?	<n> ::= 16-bit integer in NR1 format	
n/a	:OPERegister[:EVENt]?	<n> ::= 16-bit integer in NR1 format	
:OVLenable <mask>	:OVLenable?	<mask> ::= 16-bit integer in NR1 format as shown: Bit Weight Input --- ------ ---------- 10 1024 Ext Trigger Fault 9 512 Channel 4 Fault 8 256 Channel 3 Fault 7 128 Channel 2 Fault 6 64 Channel 1 Fault 4 16 Ext Trigger OVL 3 8 Channel 4 OVL 2 4 Channel 3 OVL 1 2 Channel 2 OVL 0 1 Channel 1 OVL	
n/a	:OVLRegister?	<value> ::= integer in NR1 format. See OVLenable for <value>	
:PRINt [<options>]	n/a	<options> ::= [<print option>][,..,<print option>] <print option> ::= {COLor | GRAYscale | PRINter0 | BMP8bit | BMP | PNG | NOFactors | FACTors} <print option> can be repeated up to 5 times.	
:RUN	n/a	n/a	
n/a	:SERial	<return value> ::= unquoted string containing serial number	
:SINGle	n/a	n/a	
n/a	:STATus? <display>	{0 | 1} <display> ::= {CHANnel<n> | FUNCtion | MATH | SBUS | DIGital0,..,DIGital15 | POD{1 | 2} | BUS{1 | 2}} <n> ::= 1-2 or 1-4 in NR1 format	4-Channel[2] MSO[3]
:STOP	n/a	n/a	
n/a	:TER?	{0 | 1}	
:VIEW <source>	n/a	<source> ::= {CHANnel<n> | PMEMory{0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9} | FUNCtion | MATH | SBUS | DIGital0,..,DIGital15 | POD{1 | 2} | BUS{1 | 2}}	4-Channel[2]

Command	Query	Options and Query Returns	Dependencies
		<n> ::= 1-2 or 1-4 in NR1 format	MSO[3]

Table 3. :ACQuire Commands Summary

Command	Query	Options and Query Returns	Dependencies
n/a	:ACQuire:AALias?	{1 | 0}	
:ACQuire:COMPlete <complete>	:ACQuire:COMPlete?	<complete> ::= 100; an integer in NR1 format	
:ACQuire:COUNt <count>	:ACQuire:COUNt?	<count> ::= an integer from 2 to 65536 in NR1 format	
:ACQuire:DAALias <mode>	:ACQuire:DAALias?	<mode> ::= {DISable | AUTO}	
:ACQuire:MODE <mode>	:ACQuire:MODE?	<mode> ::= {RTIMe | ETIMe | SEGMented}	SGM[8]
n/a	:ACQuire:POINts?	<# points> ::= an integer in NR1 format	
:ACQuire:RSIGnal <ref_signal_mode>	:ACQuire:RSIGnal?	<ref_signal_mode> ::= {OFF | OUT | IN}	
:ACQuire:SEGMented:ANALyze	n/a	n/a	SGM[8]
:ACQuire:SEGMented:COUNt <count>	:ACQuire:SEGMented:COUNt?	<count> ::= an integer from 2 to 250 (5000 Series) or 2 to 2000 (6000/7000 Series) in NR1 format	SGM[8]
:ACQuire:SEGMented:INDex <index>	:ACQuire:SEGMented:INDex?	<index> ::= an integer from 2 to 250 (5000 Series) or 2 to 2000 (6000/7000 Series) in NR1 format	SGM[8]
n/a	:ACQuire:SRATe?	<sample_rate> ::= sample rate (samples/s) in NR3 format	
:ACQuire:TYPE <type>	:ACQuire:TYPE?	<type> ::= {NORMal | AVERage | HRESolution | PEAK}	

Table 4. :BUS<n> Commands Summary

Command	Query	Options and Query Returns	Dependencies
:BUS<n>:BIT<m> {{0 | OFF} | {1 | ON}}	:BUS<n>:BIT<m>?	{0 | 1} <n> ::= 1 or 2; an integer in NR1 format <m> ::= 0-15; an integer in NR1 format	MSO[3]
:BUS<n>:BITS <channel_list>, {{0 | OFF} | {1 | ON}}	:BUS<n>:BITS?	<channel_list>, {0 | 1} <channel_list> ::= (@<m>,<m>:<m> ...) where "," is separator and ":" is range <n> ::= 1 or 2; an integer in NR1 format <m> ::= 0-15; an integer in NR1 format	MSO[3]
:BUS<n>:CLEar	n/a	<n> ::= 1 or 2; an integer in NR1 format	MSO[3]
:BUS<n>:DISPlay {{0 | OFF} | {1 | ON}}	:BUS<n>:DISPlay?	{0 | 1} <n> ::= 1 or 2; an integer in NR1 format	MSO[3]

Table 5. :CALibrate Commands Summary

Command	Query	Options and Query Returns	Dependencies
n/a	:CALibrate:DATE?	<return value> ::= <day>,<month>,<year>; all in NR1 format	
:CALibrate:LABel <string>	:CALibrate:LABel?	<string> ::= quoted ASCII string up to 32 characters	
:CALibrate:OUTPut <signal>	:CALibrate:OUTPut?	<signal> ::= {TRIGgers | SOURce | DSOurce | MASK}	
:CALibrate:STARt	n/a	n/a	
n/a	:CALibrate:STATus?	<return value> ::= ALL,<status_code>, <status_string> <status_code> ::= an integer status code <status_string> ::= an ASCII status string	
n/a	:CALibrate:SWITch?	{PROTected | UNPRotected}	
n/a	:CALibrate:TEMPerature?	<return value> ::= degrees C delta since last cal in NR3 format	
n/a	:CALibrate:TIME?	<return value> ::= <hours>,<minutes>,<seconds> all in NR1 format	

Table 6. :CHANnel<n> Commands Summary

Command	Query	Options and Query Returns	Dependencies
:CHANnel<n>:BWLimit {{0 | OFF} | {1 | ON}}	:CHANnel<n>:BWLimit?	{0 | 1} <n> ::= 1-2 or 1-4 in NR1 format	
:CHANnel<n>:COUPling <coupling>	:CHANnel<n>:COUPling?	<coupling> ::= {AC | DC} <n> ::= 1-2 or 1-4 in NR1 format	
:CHANnel<n>:DISPlay {{0 | OFF} | {1 | ON}}	:CHANnel<n>:DISPlay?	{0 | 1} <n> ::= 1-2 or 1-4 in NR1 format	
:CHANnel<n>:IMPedance <impedance>	:CHANnel<n>:IMPedance?	<impedance> ::= {ONEMeg | FIFTy} <n> ::= 1-2 or 1-4 in NR1 format	
:CHANnel<n>:INVert {{0 | OFF} | {1 | ON}}	:CHANnel<n>:INVert?	{0 | 1} <n> ::= 1-2 or 1-4 in NR1 format	
:CHANnel<n>:LABel <string>	:CHANnel<n>:LABel?	<string> ::= any series of 10 or less ASCII characters enclosed in quotation marks <n> ::= 1-2 or 1-4 in NR1 format	
:CHANnel<n>:OFFSet <offset>[suffix]	:CHANnel<n>:OFFSet?	<offset> ::= Vertical offset value in NR3 format [suffix] ::= {V | mV} <n> ::= 1-2 or 1-4; in NR1 format	
:CHANnel<n>:PROBe <attenuation>	:CHANnel<n>:PROBe?	<attenuation> ::= Probe attenuation ratio in NR3 format <n> ::= 1-2 or 1-4 in NR1 format	

Command	Query	Options and Query Returns	Dependencies
n/a	:CHANnel<n>:PROBe:ID?	<probe id> ::= unquoted ASCII string up to 11 characters <n> ::= 1-2 or 1-4 in NR1 format	
:CHANnel<n>:PROBe:SKEW <skew_value>	:CHANnel<n>:PROBe:SKEW?	<skew_value> ::= -100 ns to +100 ns in NR3 format <n> ::= 1-2 or 1-4 in NR1 format	
:CHANnel<n>:PROBe:STYPe <signal type>	:CHANnel<n>:PROBe:STYPe?	<signal type> ::= {DIFFerential | SINGle} <n> ::= 1-2 or 1-4 in NR1 format	
:CHANnel<n>:PROTection	:CHANnel<n>:PROTection?	{NORM | TRIP} <n> ::= 1-2 or 1-4 in NR1 format	
:CHANnel<n>:RANGe <range>[suffix]	:CHANnel<n>:RANGe?	<range> ::= Vertical full-scale range value in NR3 format [suffix] ::= {V | mV} <n> ::= 1-2 or 1-4 in NR1 format	
:CHANnel<n>:SCALe <scale>[suffix]	:CHANnel<n>:SCALe?	<scale> ::= Vertical units per division value in NR3 format [suffix] ::= {V | mV} <n> ::= 1-2 or 1-4 in NR1 format	
:CHANnel<n>:UNITs <units>	:CHANnel<n>:UNITs?	<units> ::= {VOLT | AMPere} <n> ::= 1-2 or 1-4 in NR1 format	
:CHANnel<n>:VERNier {{0 | OFF} | {1 | ON}}	:CHANnel<n>:VERNier?	{0 | 1} <n> ::= 1-2 or 1-4 in NR1 format	

Table 7. :DIGital<n> Commands Summary

Command	Query	Options and Query Returns	Dependencies
:DIGital<n>:DISPlay {{0 | OFF} | {1 | ON}}	:DIGital<n>:DISPlay?	{0 | 1} <n> ::= 0-15; an integer in NR1 format	MSO[3]
:DIGital<n>:LABel <string>	:DIGital<n>:LABel?	<string> ::= any series of 10 or less ASCII characters enclosed in quotation marks <n> ::= 0-15; an integer in NR1 format	MSO[3]
:DIGital<n>:POSition <position>	:DIGital<n>:POSition?	<n> ::= 0-15; an integer in NR1 format <position> ::= 0-7 if display size = large, 0-15 if size = medium, 0-31 if size = small	MSO[3]
:DIGital<n>:SIZE <value>	:DIGital<n>:SIZE?	<value> ::= {SMALl | MEDium | LARGe}	MSO[3]
:DIGital<n>:THReshold <value>[suffix]	:DIGital<n>:THReshold?	<n> ::= 0-15; an integer in NR1 format <value> ::= {CMOS | ECL | TTL | <user defined value>} <user defined value> ::= value in NR3 format from -8.00 to +8.00 [suffix] ::= {V | mV | uV}	MSO[3]

Table 8. :DISPlay Commands Summary

Command	Query	Options and Query Returns	Dependencies
:DISPlay:CLEar	n/a	n/a	
:DISPlay:DATA [<format>][,][<area>][,] [<palette>]<display data>	:DISPlay:DATA? [<format>][,][<area>][,] [<palette>]	<format> ::= {TIFF} (command) <area> ::= {GRATicule} (command) <palette> ::= {MONochrome} (command) <format> ::= {TIFF | BMP | BMP8bit | PNG} (query) <area> ::= {GRATicule | SCReen} (query) <palette> ::= {MONochrome | GRAYscale | COLor} (query) <display data> ::= data in IEEE 488.2 # format	
:DISPlay:LABel {{0 | OFF} | {1 | ON}}	:DISPlay:LABel?	{0 | 1}	
:DISPlay:LABList <binary block>	:DISPlay:LABList?	<binary block> ::= an ordered list of up to 75 labels, each 10 characters maximum, separated by newline characters	
:DISPlay:PERSistence <value>	:DISPlay:PERSistence?	<value> ::= {MINimum | INFinite}}	
:DISPlay:SOURce <value>	:DISPlay:SOURce?	<value> ::= {PMEMory{0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9}}	
:DISPlay:VECTors {{1 | ON} | {0 | OFF}}	:DISPlay:VECTors?	{1 | 0}	

Table 9. :EXTernal Trigger Commands Summary

Command	Query	Options and Query Returns	Dependencies
:EXTernal:BWLimit <bwlimit>	:EXTernal:BWLimit?	<bwlimit> ::= {0 | OFF}	
:EXTernal:IMPedance <value>	:EXTernal:IMPedance?	<impedance> ::= {ONEMeg | FIFTy}	
:EXTernal:PROBe <attenuation>	:EXTernal:PROBe?	<attenuation> ::= probe attenuation ratio in NR3 format	
n/a	:EXTernal:PROBe:ID?	<probe id> ::= unquoted ASCII string up to 11 characters	
:EXTernal:PROBe:STYPe <signal type>	:EXTernal:PROBe:STYPe?	<signal type> ::= {DIFFerential | SINGle}	
:EXTernal:PROTection[:CLEar]	:EXTernal:PROTection?	{NORM | TRIP}	
:EXTernal:RANGe <range>[<suffix>]	:EXTernal:RANGe?	<range> ::= vertical full-scale range value in NR3 format <suffix> ::= {V | mV}	
:EXTernal:UNITs <units>	:EXTernal:UNITs?	<units> ::= {VOLT | AMPere}	

Table 10. :FUNCtion Commands Summary

Command	Query	Options and Query Returns	Dependencies
:FUNCtion:CENTer <frequency>	:FUNCtion:CENTer?	<frequency> ::= the current center frequency in NR3 format. The range of legal values is from 0 Hz to 25 GHz.	
:FUNCtion:DISPlay {{0 | OFF} | {1 | ON}}	:FUNCtion:DISPlay?	{0 | 1}	
:FUNCtion:GOFT:OPERation <operation>	:FUNCtion:GOFT:OPERation?	<operation> ::= {ADD | SUBTract | MULTiply}	
:FUNCtion:GOFT:SOURce1 <source>	:FUNCtion:GOFT:SOURce1?	<source> ::= CHANnel<n> <n> ::= {1 | 2 | 3 | 4} for 4ch models <n> ::= {1 | 2} for 2ch models	
:FUNCtion:GOFT:SOURce2 <source>	:FUNCtion:GOFT:SOURce2?	<source> ::= CHANnel<n> <n> ::= {{1 | 2} | {3 | 4}} for 4ch models, depending on SOURce1 selection <n> ::= {1 | 2} for 2ch models	
:FUNCtion:OFFSet <offset>	:FUNCtion:OFFSet?	<offset> ::= the value at center screen in NR3 format. The range of legal values is +/-10 times the current sensitivity of the selected function.	
:FUNCtion:OPERation <operation>	:FUNCtion:OPERation?	<operation> ::= {ADD | SUBTract | MULTiply | INTegrate | DIFFerentiate | FFT | SQRT}	
:FUNCtion:RANGe <range>	:FUNCtion:RANGe?	<range> ::= the full-scale vertical axis value in NR3 format. The range for ADD, SUBT, MULT is 8E-6 to 800E+3. The range for the INTegrate function is 8E-9 to 400E+3. The range for the DIFFerentiate function is 80E-3 to 8.0E12 (depends on current sweep speed). The range for the FFT function is 8 to 800 dBV.	
:FUNCtion:REFerence <level>	:FUNCtion:REFerence?	<level> ::= the value at center screen in NR3 format. The range of legal values is +/-10 times the current sensitivity of the selected function.	
:FUNCtion:SCALe <scale value>[<suffix>]	:FUNCtion:SCALe?	<scale value> ::= integer in NR1 format <suffix> ::= {V | dB}	
:FUNCtion:SOURce1 <source>	:FUNCtion:SOURce1?	<source> ::= {CHANnel<n> | GOFT} <n> ::= {1 | 2 | 3 | 4} for 4ch models <n> ::= {1 | 2} for 2ch models GOFT is only for FFT, INTegrate, DIFFerentiate, and SQRT operations.	
:FUNCtion:SOURce2 <source>	:FUNCtion:SOURce2?	<source> ::= {CHANnel<n> | NONE} <n> ::= {{1 | 2} | {3 | 4}} for 4ch models, depending on SOURce1 selection <n> ::= {1 | 2} for 2ch models	
:FUNCtion:SPAN	:FUNCtion:SPAN?	::= the current frequency span in NR3 format. Legal values are 1 Hz to 100 GHz.	

Table 11. :HARDcopy Commands Summary

Command	Query	Options and Query Returns	Dependencies
:HARDcopy:AREA <area>	:HARDcopy:AREA?	<area> ::= SCReen	
:HARDcopy:APRinter <active_printer>	:HARDcopy:APRinter?	<active_printer> ::= {<index> | <name>} <index> ::= integer index of printer in list <name> ::= name of printer in list	
:HARDcopy:FACTors {{0 | OFF} | {1 | ON}}	:HARDcopy:FACTors?	{0 | 1}	
:HARDcopy:FFEed {{0 | OFF} | {1 | ON}}	:HARDcopy:FFEed?	{0 | 1}	
:HARDcopy:INKSaver {{0 | OFF} | {1 | ON}}	:HARDcopy:INKSaver?	{0 | 1}	
:HARDcopy:LAYout <layout>	:HARDcopy:LAYout?	<layout> ::= {LANDscape | PORTrait}	
:HARDcopy:PALette <palette>	:HARDcopy:PALette?	<palette> ::= {COLor | GRAYscale | NONE}	
n/a	:HARDcopy:PRINter:LIST?	<list> ::= [<printer_spec>] ... [printer_spec] <printer_spec> ::= "<index>,<active>,<name>;" <index> ::= integer index of printer <active> ::= {Y | N} <name> ::= name of printer	
:HARDcopy:STARt	n/a	n/a	

Table 12. :MARKer Commands Summary

Command	Query	Options and Query Returns	Dependencies
:MARKer:MODE <mode>	:MARKer:MODE?	<mode> ::= {OFF | MEASurement | MANual | WAVeform}	
:MARKer:X1Position <position>[suffix]	:MARKer:X1Position?	<position> ::= X1 cursor position value in NR3 format [suffix] ::= {s | ms | us | ns | ps | Hz | kHz | MHz} <return_value> ::= X1 cursor position value in NR3 format	
:MARKer:X1Y1source <source>	:MARKer:X1Y1source?	<source> ::= {CHANnel<n> | FUNCtion | MATH} <n> ::= 1-2 or 1-4 in NR1 format <return_value> ::= <source>	
:MARKer:X2Position <position>[suffix]	:MARKer:X2Position?	<position> ::= X2 cursor position value in NR3 format [suffix] ::= {s | ms | us | ns | ps | Hz | kHz | MHz} <return_value> ::= X2 cursor position value in NR3 format	
:MARKer:X2Y2source <source>	:MARKer:X2Y2source?	<source> ::= {CHANnel<n> | FUNCtion | MATH} <n> ::= 1-2 or 1-4 in NR1 format	

Command	Query	Options and Query Returns	Dependencies
		<return_value> ::= <source>	
n/a	:MARKer:XDELta?	<return_value> ::= X cursors delta value in NR3 format	
:MARKer:Y1Position <position>[suffix]	:MARKer:Y1Position?	<position> ::= Y1 cursor position value in NR3 format [suffix] ::= {V | mV | dB} <return_value> ::= Y1 cursor position value in NR3 format	
:MARKer:Y2Position <position>[suffix]	:MARKer:Y2Position?	<position> ::= Y2 cursor position value in NR3 format [suffix] ::= {V | mV | dB} <return_value> ::= Y2 cursor position value in NR3 format	
n/a	:MARKer:YDELta?	<return_value> ::= Y cursors delta value in NR3 format	

Table 13. :MEASure Commands Summary

Command	Query	Options and Query Returns	Dependenci
:MEASure:CLEar	n/a	n/a	
:MEASure:COUNter [<source>]	:MEASure:COUNter? [<source>]	<source> ::= {CHANnel<n> | DIGital0,..,DIGital15 | EXTernal} <n> ::= 1-2 or 1-4 in NR1 format <return_value> ::= counter frequency in Hertz in NR3 format	MSO[3]
:MEASure:DEFine DELay, <delay spec>	:MEASure:DEFine? DELay	<delay spec> ::= <edge_spec1>,<edge_spec2> edge_spec1 ::= [<slope>]<occurrence> edge_spec2 ::= [<slope>]<occurrence> <slope> ::= {+ | -} <occurrence> ::= integer	
:MEASure:DEFine THResholds, <threshold spec>	:MEASure:DEFine? THResholds	<threshold spec> ::= {STANdard} | {<threshold mode>,<upper>, <middle>,<lower>} <threshold mode> ::= {PERCent | ABSolute}	
:MEASure:DELay [<source1>] [,<source2>]	:MEASure:DELay? [<source1>] [,<source2>]	<source1,2> ::= {CHANnel<n> | FUNCtion | MATH} <n> ::= 1-2 or 1-4 in NR1 format <return_value> ::= floating-point number delay time in seconds in NR3 format	
:MEASure:DUTYcycle [<source>]	:MEASure:DUTYcycle? [<source>]	<source> ::= {CHANnel<n> | FUNCtion | MATH | DIGital0,..,DIGital15} <n> ::= 1-2 or 1-4 in NR1 format <return_value> ::= ratio of positive pulse width to period in NR3 format	MSO[3]

Command	Query	Options and Query Returns	Dependenci
:MEASure:FALLtime [<source>]	:MEASure:FALLtime? [<source>]	<source> ::= {CHANnel<n> | FUNCtion | MATH | DIGital0,..,DIGital15} <n> ::= 1-2 or 1-4 in NR1 format <return_value> ::= time in seconds between the lower and upper thresholds in NR3 format	MSO[3]
:MEASure:FREQuency [<source>]	:MEASure:FREQuency? [<source>]	<source> ::= {CHANnel<n> | FUNCtion | MATH | DIGital0,..,DIGital15} <n> ::= 1-2 or 1-4 in NR1 format <return_value> ::= frequency in Hertz in NR3 format	MSO[3]
:MEASure:NWIDth [<source>]	:MEASure:NWIDth? [<source>]	<source> ::= {CHANnel<n> | FUNCtion | MATH | DIGital0,..,DIGital15} <n> ::= 1-2 or 1-4 in NR1 format <return_value> ::= negative pulse width in seconds in NR3 format	MSO[3]
:MEASure:OVERshoot [<source>]	:MEASure:OVERshoot? [<source>]	<source> ::= {CHANnel<n> | FUNCtion | MATH} <n> ::= 1-2 or 1-4 in NR1 format <return_value> ::= the percent of the overshoot of the selected waveform in NR3 format	
:MEASure:PERiod [<source>]	:MEASure:PERiod? [<source>]	<source> ::= {CHANnel<n> | FUNCtion | MATH | DIGital0,..,DIGital15} <n> ::= 1-2 or 1-4 in NR1 format <return_value> ::= waveform period in seconds in NR3 format	MSO[3]
:MEASure:PHASe [<source1>] [,<source2>]	:MEASure:PHASe? [<source1>] [,<source2>]	<source1,2> ::= {CHANnel<n> | FUNCtion | MATH} <n> ::= 1-2 or 1-4 in NR1 format <return_value> ::= the phase angle value in degrees in NR3 format	
:MEASure:PREShoot [<source>]	:MEASure:PREShoot? [<source>]	<source> ::= {CHANnel<n> | FUNCtion | MATH} <n> ::= 1-2 or 1-4 in NR1 format <return_value> ::= the percent of preshoot of the selected waveform in NR3 format	
:MEASure:PWIDth [<source>]	:MEASure:PWIDth? [<source>]	<source> ::= {CHANnel<n> | FUNCtion | MATH | DIGital0,..,DIGital15} <n> ::= 1-2 or 1-4 in NR1 format <return_value> ::= width of positive pulse in seconds in NR3 format	MSO[3]
n/a	:MEASure:RESults? <results_list>	<results_list> ::= comma-separated list of measurement results	
:MEASure:RISEtime [<source>]	:MEASure:RISEtime? [<source>]	<source> ::= {CHANnel<n> | FUNCtion | MATH} <n> ::= 1-2 or 1-4 in NR1 format <return_value> ::= rise time in seconds in NR3 format	
:MEASure:SDEViation [<source>]	:MEASure:SDEViation? [<source>]	<source> ::= {CHANnel<n> | FUNCtion | MATH} <n> ::= 1-2 or 1-4 in NR1 format	

Command	Query	Options and Query Returns	Dependenci
		<return_value> ::= calculated std deviation in NR3 format	
:MEASure:SHOW {1 | ON}	:MEASure:SHOW?	{1}	
:MEASure:SOURce <source1> [,<source2>]	:MEASure:SOURce?	<source1,2> ::= {CHANnel<n> | FUNCtion | MATH | DIGital0,..,DIGital15 | EXTernal} <n> ::= 1-2 or 1-4 in NR1 format <return_value> ::= {<source> | EXT | NONE} EXTernal is only a valid source for the counter measurement (and <source1>).	MSO^3
:MEASure:STATistics <type>	:MEASure:STATistics?	<type> ::= {{ON | 1} | CURRent | MEAN | MINimum | MAXimum | STDDev | COUNt} ON ::= all statistics returned	
:MEASure:STATistics:INCRement	n/a	n/a	
:MEASure:STATistics:RESet	n/a	n/a	
n/a	:MEASure:TEDGe? <slope><occurrence>[,<source>]	<slope> ::= direction of the waveform <occurrence> ::= the transition to be reported <source> ::= {CHANnel<n> | FUNCtion | MATH | DIGital0,..,DIGital15} <n> ::= 1-2 or 1-4 in NR1 format <return_value> ::= time in seconds of the specified transition	MSO^3
n/a	:MEASure:TVALue? <value>, [<slope>]<occurrence> [,<source>]	<value> ::= voltage level that the waveform must cross. <slope> ::= direction of the waveform when <value> is crossed. <occurrence> ::= transitions reported. <return_value> ::= time in seconds of specified voltage crossing in NR3 format <source> ::= {CHANnel<n> | FUNCtion | MATH | DIGital0,..,DIGital15} <n> ::= 1-2 or 1-4 in NR1 format	MSO^3
:MEASure:VAMPlitude [<source>]	:MEASure:VAMPlitude? [<source>]	<source> ::= {CHANnel<n> | FUNCtion | MATH} <n> ::= 1-2 or 1-4 in NR1 format <return_value> ::= the amplitude of the selected waveform in volts in NR3 format	
:MEASure:VAVerage [<source>]	:MEASure:VAVerage? [<source>]	<source> ::= {CHANnel<n> | FUNCtion | MATH} <n> ::= 1-2 or 1-4 in NR1 format <return_value> ::= calculated average voltage in NR3 format	
:MEASure:VBASe [<source>]	:MEASure:VBASe? [<source>]	<source> ::= {CHANnel<n> | FUNCtion | MATH} <n> ::= 1-2 or 1-4 in NR1 format <base_voltage> ::= voltage at the base of the selected waveform in NR3 format	

Command	Query	Options and Query Returns	Dependenci
:MEASure:VMAX [<source>]	:MEASure:VMAX? [<source>]	<source> ::= {CHANnel<n> | FUNCtion | MATH} <n> ::= 1-2 or 1-4 in NR1 format <return_value> ::= maximum voltage of the selected waveform in NR3 format	
:MEASure:VMIN [<source>]	:MEASure:VMIN? [<source>]	<source> ::= {CHANnel<n> | FUNCtion | MATH} <n> ::= 1-2 or 1-4 in NR1 format <return_value> ::= minimum voltage of the selected waveform in NR3 format	
:MEASure:VPP [<source>]	:MEASure:VPP? [<source>]	<source> ::= {CHANnel<n> | FUNCtion | MATH} <n> ::= 1-2 or 1-4 in NR1 format <return_value> ::= voltage peak-to-peak of the selected waveform in NR3 format	
:MEASure:VRATio [<source1>] [,<source2>]	:MEASure:VRATio? [<source1>] [,<source2>]	<source1,2> ::= {CHANnel<n> | FUNCtion | MATH} <n> ::= 1-2 or 1-4 in NR1 format <return_value> ::= the ratio value in dB in NR3 format	
:MEASure:VRMS [<source>]	:MEASure:VRMS? [<source>]	<source> ::= {CHANnel<n> | FUNCtion | MATH} <n> ::= 1-2 or 1-4 in NR1 format <return_value> ::= calculated dc RMS voltage in NR3 format	
n/a	:MEASure:VTIMe? <vtime>[,<source>]	<vtime> ::= displayed time from trigger in seconds in NR3 format <return_value> ::= voltage at the specified time in NR3 format <source> ::= {CHANnel<n> | FUNCtion | MATH | DIGital0,..,DIGital15} <n> ::= 1-2 or 1-4 in NR1 format	MSO[3]
:MEASure:VTOP [<source>]	:MEASure:VTOP? [<source>]	<source> ::= {CHANnel<n> | FUNCtion | MATH} <n> ::= 1-2 or 1-4 in NR1 format <return_value> ::= voltage at the top of the waveform in NR3 format	
:MEASure:XMAX [<source>]	:MEASure:XMAX? [<source>]	<source> ::= {CHANnel<n> | FUNCtion | MATH} <n> ::= 1-2 or 1-4 in NR1 format <return_value> ::= horizontal value of the maximum in NR3 format	
:MEASure:XMIN [<source>]	:MEASure:XMIN? [<source>]	<source> ::= {CHANnel<n> | FUNCtion | MATH} <n> ::= 1-2 or 1-4 in NR1 format <return_value> ::= horizontal value of the maximum in NR3 format	

Table 14. :MTESt Commands Summary

Command	Query	Options and Query Returns	Dependencies
:MTESt:AMASk:CREate	n/a	n/a	LMT[9]

Table 15. :POD<n> Commands Summary

Command	Query	Options and Query Returns	Dependencies
:POD<n>:DISPlay {{0 | OFF} | {1 | ON}}	:POD<n>:DISPlay?	{0 | 1} <n> ::= 1-2 in NR1 format	MSO[3]
:POD<n>:SIZE <value>	:POD<n>:SIZE?	<value> ::= {SMALl | MEDium | LARGe}	MSO[3]
:POD<n>:THReshold <type>[suffix]	:POD<n>:THReshold?	<n> ::= 1-2 in NR1 format <type> ::= {CMOS | ECL | TTL | <user defined value>} <user defined value> ::= value in NR3 format [suffix] ::= {V | mV | uV}	MSO[3]

Table 16. :RECall Commands Summary

Command	Query	Options and Query Returns	Dependencies
:RECall:FILename <base_name>	:RECall:FILename?	<base_name> ::= quoted ASCII string	
:RECall:IMAGe[:STARt] [<file_spec>]	n/a	<file_spec> ::= {<internal_loc> | <file_name>} <internal_loc> ::= 0-9; an integer in NR1 format <file_name> ::= quoted ASCII string	
:RECall:MASK[:STARt] [<file_spec>]	n/a	<file_spec> ::= {<internal_loc> | <file_name>} <internal_loc> ::= 0-3; an integer in NR1 format <file_name> ::= quoted ASCII string	LMT[9]
:RECall:PWD <path_name>	:RECall:PWD?	<path_name> ::= quoted ASCII string	
:RECall:SETup[:STARt] [<file_spec>]	n/a	<file_spec> ::= {<internal_loc> | <file_name>} <internal_loc> ::= 0-9; an integer in NR1 format <file_name> ::= quoted ASCII string	

Table 17. :SAVE Commands Summary

Command	Query	Options and Query Returns	Dependencies
:SAVE:FILename <base_name>	:SAVE:FILename?	<base_name> ::= quoted ASCII string	
:SAVE:IMAGe[:STARt] [<file_spec>]	n/a	<file_spec> ::= {<internal_loc> | <file_name>} <internal_loc> ::= 0-9; an integer in NR1 format <file_name> ::= quoted ASCII string	
n/a	:SAVE:IMAGe:AREA?	<area> ::= {GRAT | SCR}	
:SAVE:IMAGe:FACTors {{0 | OFF} | {1 | ON}}	:SAVE:IMAGe:FACTors?	{0 | 1}	
:SAVE:IMAGe:FORMat <format>	:SAVE:IMAGe:FORMat?	<format> ::= {TIFF | {BMP | BMP24bit} | BMP8bit | PNG | NONE}	

Command	Query	Options and Query Returns	Dependencies
:SAVE:IMAGe:IGColors {{0 | OFF} | {1 | ON}}	:SAVE:IMAGe:IGColors?	{0 | 1}	
:SAVE:IMAGe:PALette <palette>	:SAVE:IMAGe:PALette?	<palette> ::= {COLor | GRAYscale | MONochrome}	
:SAVE:MASK[:STARt] [<file_spec>]	n/a	<file_spec> ::= {<internal_loc> | <file_name>} <internal_loc> ::= 0-3; an integer in NR1 format <file_name> ::= quoted ASCII string	LMT[9]
:SAVE:PWD <path_name>	:SAVE:PWD?	<path_name> ::= quoted ASCII string	
:SAVE:SETup[:STARt] [<file_spec>]	n/a	<file_spec> ::= {<internal_loc> | <file_name>} <internal_loc> ::= 0-9; an integer in NR1 format <file_name> ::= quoted ASCII string	
:SAVE:WAVeform[:STARt] [<file_name>]	n/a	<file_name> ::= quoted ASCII string	
:SAVE:WAVeform:FORMat <format>	:SAVE:WAVeform:FORMat?	<format> ::= {ALB | ASCiixy | CSV | BINary | NONE}	
:SAVE:WAVeform:LENGth <length>	:SAVE:WAVeform:LENGth?	<length> ::= 100 to max. length; an integer in NR1 format	
:SAVE:WAVeform:SEGMented <option>	:SAVE:WAVeform:SEGMented?	<option> ::= {ALL | CURRent}	SGM[8]

Table 18. :SBUS Commands Summary

Command	Query	Options and Query Returns	Dependencies
:SBUS:BUSDoctor:ADDRess <value>	:SBUS:BUSDoctor:ADDRess?	<value> ::= <field value>, <field value>, <field value>, <field value> <field value> ::= integer from 0-255 in NR1 format	FRS[6]
:SBUS:BUSDoctor:BAUDrate <baudrate>	:SBUS:BUSDoctor:BAUDrate?	<baudrate> ::= {2500000 | 5000000 | 10000000}	FRS[6]
:SBUS:BUSDoctor:CHANnel <channel>	:SBUS:BUSDoctor:CHANnel?	<channel> ::= {A | B}	FRS[6]
:SBUS:BUSDoctor:MODE <mode>	:SBUS:BUSDoctor:MODE?	<mode> ::= {ASYNchronous | SYNChronous | PC}	FRS[6]
n/a	:SBUS:CAN:COUNt:ERRor?	<frame_count> ::= integer in NR1 format	AMS[5]
n/a	:SBUS:CAN:COUNt:OVERload?	<frame_count> ::= integer in NR1 format	AMS[5]
:SBUS:CAN:COUNt:RESet	n/a	n/a	AMS[5]
n/a	:SBUS:CAN:COUNt:TOTal?	<frame_count> ::= integer in NR1 format	AMS[5]
n/a	:SBUS:CAN:COUNt:UTILization?	<percent> ::= floating-point in NR3 format	AMS[5]

Command	Query	Options and Query Returns	Dependencies
:SBUS:DISPlay {{0 | OFF} | (1 | ON}}	:SBUS:DISPlay?	{0 | 1}	4-Channel[2]
n/a	:SBUS:FLEXray:COUNt:NULL?	<frame_count> ::= integer in NR1 format	FRS[6]
:SBUS:FLEXray:COUNt:RESet	n/a	n/a	FRS[6]
n/a	:SBUS:FLEXray:COUNt:SYNC?	<frame_count> ::= integer in NR1 format	FRS[6]
n/a	:SBUS:FLEXray:COUNt:TOTal?	<frame_count> ::= integer in NR1 format	FRS[6]
:SBUS:IIC:ASIZe <size>	:SBUS:IIC:ASIZe?	<size> ::= {BIT7 | BIT8}	LSS[4]
:SBUS:LIN:PARity {{0 | OFF} | {1 | ON}}	:SBUS:LIN:PARity?	{0 | 1}	AMS[5]
:SBUS:MODE <mode>	:SBUS:MODE?	<mode> ::= {IIC | SPI | CAN | LIN | FLEXray}	4-Channel[2] LSS[4] AMS[5] FRS[6]
:SBUS:SPI:WIDTh <word_width>	:SBUS:SPI:WIDTh?	<word_width> ::= integer 4-16 in NR1 format	LSS[4]
:SBUS:UART:BASE <base>	:SBUS:UART:BASE?	<base> ::= {ASCii | BINary | HEX}	232[7]
n/a	:SBUS:UART:COUNt:ERRor?	<frame_count> ::= integer in NR1 format	232[7]
:SBUS:UART:COUNt:RESet	n/a	n/a	232[7]
n/a	:SBUS:UART:COUNt:RXFRames?	<frame_count> ::= integer in NR1 format	232[7]
n/a	:SBUS:UART:COUNt:TXFRames?	<frame_count> ::= integer in NR1 format	232[7]
:SBUS:UART:FRAMing <value>	:SBUS:UART:FRAMing?	<value> ::= {OFF | <decimal> | <nondecimal>} <decimal> ::= 8-bit integer in decimal from 0-255 (0x00-0xff) <nondecimal> ::= #Hnn where n ::= {0,..,9 | A,..,F} for hexadecimal <nondecimal> ::= #Bnn...n where n ::= {0 | 1} for binary	232[7]

Table 19. :SYSTem Commands Summary

Command	Query	Options and Query Returns	Dependencies

Command	Query	Options and Query Returns	Dependencies
:SYSTem:DATE <date>	:SYSTem:DATE?	<date> ::= <year>,<month>,<day> <year> ::= 4-digit year in NR1 format <month> ::= {1,..,12 | JANuary | FEBruary | MARch | APRil | MAY | JUNe | JULy | AUGust | SEPtember | OCTober | NOVember | DECember} <day> ::= {1,..31}	
:SYSTem:DSP <string>	n/a	<string> ::= up to 254 characters as a quoted ASCII string	
n/a	:SYSTem:ERRor?	<error> ::= an integer error code <error string> ::= quoted ASCII string. See "Error Messages" in *Programmer's Reference*.	
:SYSTem:LOCK	:SYSTem:LOCK?	<value> ::= {{1 | ON} | {0 | OFF}}	
:SYSTem:PROTection:LOCK	:SYSTem:PROTection:LOCK?	<value> ::= {{1 | ON} | {0 | OFF}}	
:SYSTem:SETup <setup_data>	:SYSTem:SETup?	<setup_data> ::= data in IEEE 488.2 # format.	
:SYSTem:TIME <time>	:SYSTem:TIME?	<time> ::= hours,minutes,seconds in NR1 format	

Table 20. :TIMebase Commands Summary

Command	Query	Options and Query Returns	Dependencies
:TIMebase:MODE <value>	:TIMebase:MODE?	<value> ::= {MAIN | WINDow | XY | ROLL}	
:TIMebase:POSition <pos>	:TIMebase:POSition?	<pos> ::= time from the trigger event to the display reference point in NR3 format	
:TIMebase:RANGe <range_value>	:TIMebase:RANGe?	<range_value> ::= 5 ns through 500 s in NR3 format	
:TIMebase:REFClock {{0 | OFF} | {1 | ON}}	:TIMebase:REFClock?	{0 | 1}	
:TIMebase:REFerence {LEFT | CENTer | RIGHt}	:TIMebase:REFerence?	<return_value> ::= {LEFT | CENTer | RIGHt}	
:TIMebase:SCALe <scale_value>	:TIMebase:SCALe?	<scale_value> ::= scale value in seconds in NR3 format	
:TIMebase:VERNier {{0 | OFF} | {1 | ON}}	:TIMebase:VERNier?	{0 | 1}	
:TIMebase:WINDow:POSition <pos>	:TIMebase:WINDow:POSition?	<pos> ::= time from the trigger event to the delayed view reference point in NR3 format	
:TIMebase:WINDow:RANGe <range_value>	:TIMebase:WINDow:RANGe?	<range value> ::= range value in seconds in NR3 format for the delayed window	
:TIMebase:WINDow:SCALe <scale_value>	:TIMebase:WINDow:SCALe?	<scale_value> ::= scale value in seconds in NR3 format for the delayed window	

Table 21. General :TRIGger Commands Summary

Command	Query	Options and Query Returns	Dependencies

Command	Query	Options and Query Returns	Dependencies
:TRIGger:HFReject {{0 | OFF} | {1 | ON}}	:TRIGger:HFReject?	{0 | 1}	
:TRIGger:HOLDoff <holdoff_time>	:TRIGger:HOLDoff?	<holdoff_time> ::= 60 ns to 10 s in NR3 format	
:TRIGger:MODE <mode>	:TRIGger:MODE?	<mode> ::= {EDGE | GLITch | PATTern | CAN | DURation | IIC | EBURst | LIN | SEQuence | SPI | TV | USB | FLEXray | UART} <return_value> ::= {<mode> | <none>} <none> ::= query returns "NONE" if the :TIMebase:MODE is ROLL or XY	
:TRIGger:NREJect {{0 | OFF} | {1 | ON}}	:TRIGger:NREJect?	{0 | 1}	
:TRIGger:PATTern <value>, <mask> [,<edge source>,<edge>]	:TRIGger:PATTern?	<value> ::= integer in NR1 format or <string> <mask> ::= integer in NR1 format or <string> <string> ::= "0xnnnnn"; n ::= {0,..,9 | A,..,F} (# bits = # channels) <edge source> ::= {CHANnel<n> | EXTernal | NONE} for DSO models <edge source> ::= {CHANnel<n> | DIGital0,..,DIGital15 | NONE} for MSO models <edge> ::= {POSitive | NEGative} <n> ::= 1-2 or 1-4 in NR1 format	MSO[3]
:TRIGger:SWEep <sweep>	:TRIGger:SWEep?	<sweep> ::= {AUTO | NORMal}	

Table 22. :TRIGger:CAN Commands Summary

Command	Query	Options and Query Returns	Depen
:TRIGger:CAN:PATTern:DATA <value>, <mask>	:TRIGger:CAN:PATTern:DATA?	<value> ::= 64-bit integer in decimal, <nondecimal>, or <string> (with Option AMS) <mask> ::= 64-bit integer in decimal, <nondecimal>, or <string> <nondecimal> ::= #Hnn...n where n ::= {0,..,9 | A,..,F} for hexadecimal <nondecimal> ::= #Bnn...n where n ::= {0 | 1} for binary <string> ::= "0xnn...n" where n ::= {0,..,9 | A,..,F} for hexadecimal	AMS[5]
:TRIGger:CAN:PATTern:DATA:LENGth <length>	:TRIGger:CAN:PATTern:DATA:LENGth?	<length> ::= integer from 1 to 8 in NR1 format (with Option AMS)	AMS[5]
:TRIGger:CAN:PATTern:ID <value>, <mask>	:TRIGger:CAN:PATTern:ID?	<value> ::= 32-bit integer in decimal, <nondecimal>, or <string> (with Option AMS) <mask> ::= 32-bit integer in decimal, <nondecimal>, or <string> <nondecimal> ::= #Hnn...n where n ::= {0,..,9 | A,..,F} for hexadecimal <nondecimal> ::= #Bnn...n where n ::= {0 | 1} for binary	AMS[5]

Command	Query	Options and Query Returns	Depen
		<string> ::= "0xnn...n" where n ::= {0,..,9 | A,..,F} for hexadecimal	
:TRIGger:CAN:PATTern:ID:MODE <value>	:TRIGger:CAN:PATTern:ID:MODE?	<value> ::= {STANdard | EXTended} (with Option AMS)	AMS[5]
:TRIGger:CAN:SAMPlepoint <value>	:TRIGger:CAN:SAMPlepoint?	<value> ::= {60 | 62.5 | 68 | 70 | 75 | 80 | 87.5} in NR3 format	
:TRIGger:CAN:SIGNal:BAUDrate <baudrate>	:TRIGger:CAN:SIGNal:BAUDrate?	<baudrate> ::= integer from 10000 to 1000000 in 100 b/s increments	
:TRIGger:CAN:SOURce <source>	:TRIGger:CAN:SOURce?	<source> ::= {CHANnel<n> | EXTernal} for DSO models <source> ::= {CHANnel<n> | DIGital0,..,DIGital15} for MSO models <n> ::= 1-2 or 1-4 in NR1 format	MSO[3]
:TRIGger:CAN:TRIGger <condition>	:TRIGger:CAN:TRIGger?	<condition> ::= {SOF | DATA | ERRor | IDData | IDEither | IDRemote | ALLerrors | OVERload | ACKerror}	AMS[5]

Table 23. :TRIGger:DURation Commands Summary

Command	Query	Options and Query Returns	Dependencie
:TRIGger:DURation:GREaterthan <greater_than_time>[suffix]	:TRIGger:DURation:GREaterthan?	<greater_than_time> ::= floating-point number in NR3 format [suffix] ::= {s | ms | us | ns | ps}	
:TRIGger:DURation:LESSthan <less_than_time>[suffix]	:TRIGger:DURation:LESSthan?	<less_than_time> ::= floating-point number in NR3 format [suffix] ::= {s | ms | us | ns | ps}	
:TRIGger:DURation:PATTern <value>, <mask>	:TRIGger:DURation:PATTern?	<value> ::= integer or <string> <mask> ::= integer or <string> <string> ::= ""0xnnnnnn"" n ::= {0,..,9 | A,..,F}	
:TRIGger:DURation:QUALifier <qualifier>	:TRIGger:DURation:QUALifier?	<qualifier> ::= {GREaterthan | LESSthan | INRange | OUTRange | TIMeout}	
:TRIGger:DURation:RANGe <less_than_time>[suffix], <greater_than_time>[suffix]	:TRIGger:DURation:RANGe?	<less_than_time> ::= 15 ns to 10 seconds in NR3 format <greater_than_time> ::= 10 ns to 9.99 seconds in NR3 format [suffix] ::= {s | ms | us | ns | ps}	

Table 24. :TRIGger:EBURst Commands Summary

Command	Query	Options and Query Returns	Dependencies
:TRIGger:EBURst:COUNt <count>	:TRIGger:EBURst:COUNt?	<count> ::= integer in NR1 format	6000/7000 Series[1]
:TRIGger:EBURst:IDLE <time_value>	:TRIGger:EBURst:IDLE?	<time_value> ::= time in seconds in NR3 format	6000/7000 Series[1]

Command	Query	Options and Query Returns	Dependencies
:TRIGger:EBURst:SLOPe <slope>	:TRIGger:EBURst:SLOPe?	<slope> ::= {NEGative | POSitive}	6000/7000 Series[1]

Table 25. :TRIGger[:EDGE] Commands Summary

Command	Query	Options and Query Returns	Dependencies
:TRIGger[:EDGE]:COUPling {AC | DC | LF}	:TRIGger[:EDGE]:COUPling?	{AC | DC | LF}	
:TRIGger[:EDGE]:LEVel <level> [,<source>]	:TRIGger[:EDGE]:LEVel? [<source>]	For internal triggers, <level> ::= .75 x full-scale voltage from center screen in NR3 format. For external triggers, <level> ::= ±(external range setting) in NR3 format. For digital channels (MSO models), <level> ::= ±8 V. <source> ::= {CHANnel<n> | EXTernal} for DSO models <source> ::= {CHANnel<n> | DIGital0,..,DIGital15 | EXTernal } for MSO models <n> ::= 1-2 or 1-4 in NR1 format	MSO[3]
:TRIGger[:EDGE]:REJect {OFF | LF | HF}	:TRIGger[:EDGE]:REJect?	{OFF | LF | HF}	
:TRIGger[:EDGE]:SLOPe <polarity>	:TRIGger[:EDGE]:SLOPe?	<polarity> ::= {POSitive | NEGative | EITHer | ALTernate}	
:TRIGger[:EDGE]:SOURce <source>	:TRIGger[:EDGE]:SOURce?	<source> ::= {CHANnel<n> | EXTernal} for DSO models <source> ::= {CHANnel<n> | DIGital0,..,DIGital15 | EXTernal} for MSO models <n> ::= 1-2 or 1-4 in NR1 format	MSO[3]

Table 26. :TRIGger:FLEXray Commands Summary

Command	Query	Options and Query Returns	De
:TRIGger:FLEXray:ERRor:TYPE <error_type>	:TRIGger:FLEXray:ERRor:TYPE?	<error_type> ::= {ALL | CODE | TSS | HCRC | FCRC | END | BOUNdary | IDLE | SYMbol | SLOT | NULL | SOS | FID | CCOunt | PLENgth}	FF
:TRIGger:FLEXray:FRAMe:CCBase <cycle_count_base>	:TRIGger:FLEXray:FRAMe:CCBase?	<cycle_count_base> ::= integer from 0-63	FF
:TRIGger:FLEXray:FRAMe:CCRepetition <cycle_count_repetition>	:TRIGger:FLEXray:FRAMe:CCRepetition?	<cycle_count_repetition> ::= {ALL | <rep #>} <rep #> ::= integer from 2-64	FF
:TRIGger:FLEXray:FRAMe:ID <frame_id>	:TRIGger:FLEXray:FRAMe:ID?	<frame_id> ::= {ALL | <frame #>} <frame #> ::= integer from 1-2047	FF
:TRIGger:FLEXray:FRAMe:TYPE <frame_type>	:TRIGger:FLEXray:FRAMe:TYPE?	<frame_type> ::= {NORMal | STARtup | NULL | SYNC | NSTArtup | NNULl | NSYNc}	FF

Command	Query	Options and Query Returns	De
:TRIGger:FLEXray:TIME:CBASe <cycle_base>	:TRIGger:FLEXray:TIME:CBASe?	<cycle_base> ::= integer from 0-63	FF
:TRIGger:FLEXray:TIME:CREPetition <cycle_repetition>	:TRIGger:FLEXray:TIME:CREPetition?	<cycle_repetition> ::= {ALL | <rep #>} <rep #> ::= integer from 2-64	FF
:TRIGger:FLEXray:TIME:SEGMent <segment_type>	:TRIGger:FLEXray:TIME:SEGMent?	<segment_type> ::= {STATic | DYNamic | SYMbol | IDLE}	FF
:TRIGger:FLEXray:TIME:SLOT <slot_type>, <slot_id>	::TRIGger:FLEXray:TIME:SLOT?	<slot_type> ::= {ALL | EMPTY} <slot_id> ::= {ALL | <slot #>} <slot #> ::= integer from 1-2047	FF
:TRIGger:FLEXray:TRIGger <condition>	:TRIGger:FLEXray:TRIGger?	<condition> ::= {FRAMe | TIME | ERRor}	FF

Table 27. :TRIGger:GLITch Commands Summary

Command	Query	Options and Query Returns	Dependencies
:TRIGger:GLITch:GREaterthan <greater_than_time>[suffix]	:TRIGger:GLITch:GREaterthan?	<greater_than_time> ::= floating-point number in NR3 format [suffix] ::= {s | ms | us | ns | ps}	
:TRIGger:GLITch:LESSthan <less_than_time>[suffix]	:TRIGger:GLITch:LESSthan?	<less_than_time> ::= floating-point number in NR3 format [suffix] ::= {s | ms | us | ns | ps}	
:TRIGger:GLITch:LEVel <level> [<source>]	:TRIGger:GLITch:LEVel?	For internal triggers, <level> ::= .75 x full-scale voltage from center screen in NR3 format. For external triggers (DSO models), <level> ::= ±(external range setting) in NR3 format. For digital channels (MSO models), <level> ::= ±8 V. <source> ::= {CHANnel<n> | EXTernal} for DSO models <source> ::= {CHANnel<n> | DIGital0,..,DIGital15} for MSO models <n> ::= 1-2 or 1-4 in NR1 format	MSO[3]
:TRIGger:GLITch:POLarity <polarity>	:TRIGger:GLITch:POLarity?	<polarity> ::= {POSitive | NEGative}	
:TRIGger:GLITch:QUALifier <qualifier>	:TRIGger:GLITch:QUALifier?	<qualifier> ::= {GREaterthan | LESSthan | RANGe}	
:TRIGger:GLITch:RANGe <less_than_time>[suffix], <greater_than_time>[suffix]	:TRIGger:GLITch:RANGe?	<less_than_time> ::= 15 ns to 10 seconds in NR3 format <greater_than_time> ::= 10 ns to 9.99 seconds in NR3 format [suffix] ::= {s | ms | us | ns | ps}	
:TRIGger:GLITch:SOURce <source>	:TRIGger:GLITch:SOURce?	<source> ::= {CHANnel<n> | EXTernal} for DSO models <source> ::= {CHANnel<n> | DIGital0,..,DIGital15}	MSO[3]

Command	Query	Options and Query Returns	Dependencies
		for MSO models <n> ::= 1-2 or 1-4 in NR1 format	

Table 28. :TRIGger:IIC Commands Summary

Command	Query	Options and Query Returns	Dependenc
:TRIGger:IIC:PATTern:ADDRess <value>	:TRIGger:IIC:PATTern:ADDRess?	<value> ::= integer or <string> <string> ::= "0xnn" n ::= {0,..,9 | A,..,F}	
:TRIGger:IIC:PATTern:DATA <value>	:TRIGger:IIC:PATTern:DATA?	<value> ::= integer or <string> <string> ::= "0xnn" n ::= {0,..,9 | A,..,F}	
:TRIGger:IIC:PATTern:DATa2 <value>	:TRIGger:IIC:PATTern:DATa2?	<value> ::= integer or <string> <string> ::= "0xnn" n ::= {0,..,9 | A,..,F}	
:TRIGger:IIC[:SOURce]:CLOCk <source>	:TRIGger:IIC[:SOURce]:CLOCk?	<source> ::= {CHANnel<n> | EXTernal} for DSO models <source> ::= {CHANnel<n> | DIGital0,..,DIGital15} for MSO models <n> ::= 1-2 or 1-4 in NR1 format	MSO[3]
:TRIGger:IIC[:SOURce]:DATA <source>	:TRIGger:IIC[:SOURce]:DATA?	<source> ::= {CHANnel<n> | EXTernal} for DSO models <source> ::= {CHANnel<n> | DIGital0,..,DIGital15} for MSO models <n> ::= 1-2 or 1-4 in NR1 format	MSO[3]
:TRIGger:IIC:TRIGger:QUALifier <value>	:TRIGger:IIC:TRIGger:QUALifier?	<value> ::= {EQUal | NOTequal | LESSthan | GREaterthan}	
:TRIGger:IIC:TRIGger[:TYPE] <type>	:TRIGger:IIC:TRIGger[:TYPE]?	<type> ::= {STARt | STOP | READ7 | READEprom | WRITe7 | WRITe10 | NACKnowledge | ANACknowledge | R7Data2 | W7Data2 | RESTart}	

Table 29. :TRIGger:LIN Commands Summary

Command	Query	Options and Query Returns	Dependenc
:TRIGger:LIN:ID <value>	:TRIGger:LIN:ID?	<value> ::= 7-bit integer in decimal, <nondecimal>, or <string> from 0-63 or 0x00-0x3f (with Option AMS) <nondecimal> ::= #Hnn where n ::= {0,..,9 | A,..,F} for hexadecimal <nondecimal> ::= #Bnn...n where n ::= {0 | 1} for binary <string> ::= "0xnn" where n ::= {0,..,9 | A,..,F} for hexadecimal	AMS[5]
:TRIGger:LIN:SAMPlepoint <value>	:TRIGger:LIN:SAMPlepoint?	<value> ::= {60 | 62.5 | 68 | 70 | 75 | 80 | 87.5} in NR3 format	
:TRIGger:LIN:SIGNal:BAUDrate <baudrate>	:TRIGger:LIN:SIGNal:BAUDrate?	<baudrate> ::= integer from 2400 to 625000 in 100 b/s increments	

Command	Query	Options and Query Returns	Dependenc
:TRIGger:LIN:SOURce <source>	:TRIGger:LIN:SOURce?	<source> ::= {CHANnel<n> | EXTernal} for DSO models <source> ::= {CHANnel<n> | DIGital0,..,DIGital15} for MSO models <n> ::= 1-2 or 1-4 in NR1 format	MSO[3]
:TRIGger:LIN:STANdard <std>	:TRIGger:LIN:STANdard?	<std> ::= {LIN13 | LIN20}	
:TRIGger:LIN:SYNCbreak <value>	:TRIGger:LIN:SYNCbreak?	<value> ::= integer = {11 | 12 | 13}	
:TRIGger:LIN:TRIGger <condition>	:TRIGger:LIN:TRIGger?	<condition> ::= {SYNCbreak | ID}	AMS[5]

Table 30. :TRIGger:SEQuence Commands Summary

Command	Query	Options and Query Returns	Dependenc
:TRIGger:SEQuence:COUNt <count>	:TRIGger:SEQuence:COUNt?	<count> ::= integer in NR1 format	6000/7000 Series[1]
:TRIGger:SEQuence:EDGE{1|2} <source>, <slope>	:TRIGger:SEQuence:EDGE{1|2}?	<source> ::= {CHANnel<n> | EXTernal} for the DSO models <source> ::= {CHANnel<n> | DIGital0,..,DIGital15} for the MSO models <slope> ::= {POSitive | NEGative} <n> ::= 1-2 or 1-4 in NR1 format <return_value> ::= query returns "NONE" if edge source is disabled	6000/7000 Series[1] MSO[3]
:TRIGger:SEQuence:FIND <value>	:TRIGger:SEQuence:FIND?	<value> ::= {PATTern1,ENTered | PATTern1,EXITed | EDGE1 | PATTern1,AND,EDGE1}	6000/7000 Series[1]
:TRIGger:SEQuence:PATTern{1|2} <value>, <mask>	:TRIGger:SEQuence:PATTern{1|2}?	<value> ::= integer or <string> <mask> ::= integer or <string> <string> ::= "0xnnnnnn" n ::= {0,..,9 | A,..,F}	6000/7000 Series[1]
:TRIGger:SEQuence:RESet <value>	:TRIGger:SEQuence:RESet?	<value> ::= {NONE | PATTern1,ENTered | PATTern1,EXITed | EDGE1 | PATTern1,AND,EDGE1 | PATTern2,ENTered | PATTern2,EXITed | EDGE2 | TIMer} Values used in find and trigger stages not available. EDGE2 not available if EDGE2,COUNt used in trigger stage.	6000/7000 Series[1]
:TRIGger:SEQuence:TIMer <time_value>	:TRIGger:SEQuence:TIMer?	<time_value> ::= time from 10 ns to 10 seconds in NR3 format	6000/7000 Series[1]
:TRIGger:SEQuence:TRIGger <value>	:TRIGger:SEQuence:TRIGger?	<value> ::= {PATTern2,ENTered | PATTern2,EXITed | EDGE2 | PATTern2,AND,EDGE2 | EDGE2,COUNt | EDGE2,COUNt,NREFind}	6000/7000 Series[1]

Table 31. :TRIGger:SPI Commands Summary

Command	Query	Options and Query Returns	Dependencies
:TRIGger:SPI:CLOCk:SLOPe <slope>	:TRIGger:SPI:CLOCk:SLOPe?	<slope> ::= {NEGative | POSitive}	
:TRIGger:SPI:CLOCk:TIMeout <time_value>	:TRIGger:SPI:CLOCk:TIMeout?	<time_value> ::= time in seconds in NR1 format	
:TRIGger:SPI:FRAMing <value>	:TRIGger:SPI:FRAMing?	<value> ::= {CHIPselect | NOTChipselect | TIMeout}	
:TRIGger:SPI:PATTern:DATA <value>, <mask>	:TRIGger:SPI:PATTern:DATA?	<value> ::= integer or <string> <mask> ::= integer or <string> <string> ::= "0xnnnnnn" where n ::= {0,..,9 | A,..,F}	
:TRIGger:SPI:PATTern:WIDTh <width>	:TRIGger:SPI:PATTern:WIDTh?	<width> ::= integer from 4 to 32 in NR1 format	
:TRIGger:SPI:SOURce:CLOCk <source>	:TRIGger:SPI:SOURce:CLOCk?	<value> ::= {CHANnel<n> | EXTernal} for the DSO models <value> ::= {CHANnel<n> | DIGital0,..,DIGital15} for the MSO models <n> ::= 1-2 or 1-4 in NR1 format	MSO[3]
:TRIGger:SPI:SOURce:DATA <source>	:TRIGger:SPI:SOURce:DATA?	<value> ::= {CHANnel<n> | EXTernal} for the DSO models <value> ::= {CHANnel<n> | DIGital0,..,DIGital15} for the MSO models <n> ::= 1-2 or 1-4 in NR1 format	MSO[3]
:TRIGger:SPI:SOURce:FRAMe <source>	:TRIGger:SPI:SOURce:FRAMe?	<value> ::= {CHANnel<n> | EXTernal} for the DSO models <value> ::= {CHANnel<n> | DIGital0,..,DIGital15} for the MSO models <n> ::= 1-2 or 1-4 in NR1 format	MSO[3]

Table 32. :TRIGger:TV Commands Summary

Command	Query	Options and Query Returns	Dependencies
:TRIGger:TV:LINE <line number>	:TRIGger:TV:LINE?	<line number> ::= integer in NR1 format	
:TRIGger:TV:MODE <tv mode>	:TRIGger:TV:MODE?	<tv mode> ::= {FIEld1 | FIEld2 | AFIelds | ALINes | LINE | VERTical | LFIeld1 | LFIeld2 | LALTernate | LVERtical}	
:TRIGger:TV:POLarity <polarity>	:TRIGger:TV:POLarity?	<polarity> ::= {POSitive | NEGative}	
:TRIGger:TV:SOURce <source>	:TRIGger:TV:SOURce?	<source> ::= {CHANnel<n>} <n> ::= 1-2 or 1-4 integer in NR1 format	
:TRIGger:TV:STANdard <standard>	:TRIGger:TV:STANdard?	<standard> ::= {GENeric | NTSC | PALM | PAL | SECam | {P480L60HZ | P480} | {P720L60HZ | P720} | {P1080L24HZ | P1080} | P1080L25HZ | {I1080L50HZ | I1080} | I1080L60HZ}	

Table 33. :TRIGger:UART Commands Summary

Command	Query	Options and Query Returns	Dependencies
:TRIGger:UART:BASE <base>	:TRIGger:UART:BASE?	<base> ::= {ASCii | HEX}	232[7]
:TRIGger:UART:BAUDrate <baudrate>	:TRIGger:UART:BAUDrate?	<baudrate> ::= integer from 1200 to 3000000 in 100 b/s increments	232[7]
:TRIGger:UART:BITorder <bitorder>	:TRIGger:UART:BITorder?	<bitorder> ::= {LSBFirst | MSBFirst}	232[7]
:TRIGger:UART:BURSt <value>	:TRIGger:UART:BURSt?	<value> ::= {OFF | 1 to 4096 in NR1 format}	232[7]
:TRIGger:UART:DATA <value>	:TRIGger:UART:DATA?	<value> ::= 8-bit integer from 0-255 (0x00-0xff) in decimal, <hexadecimal>, <binary>, or <quoted_string> format <hexadecimal> ::= #Hnn where n ::= {0,..,9 | A,..,F} for hexadecimal <binary> ::= #Bnn...n where n ::= {0 | 1} for binary <quoted_string> ::= any of the 128 valid 7-bit ASCII characters (or standard abbreviations)	232[7]
:TRIGger:UART:IDLE <time_value>	:TRIGger:UART:IDLE?	<time_value> ::= time from 1 us to 10 s in NR3 format	232[7]
:TRIGger:UART:PARity <parity>	:TRIGger:UART:PARity?	<parity> ::= {EVEN | ODD | NONE}	232[7]
:TRIGger:UART:POLarity <polarity>	:TRIGger:UART:POLarity?	<polarity> ::= {HIGH | LOW}	232[7]
:TRIGger:UART:QUALifier <value>	:TRIGger:UART:QUALifier?	<value> ::= {EQUal | NOTequal | GREaterthan | LESSthan}	232[7]
:TRIGger:UART:SOURce:RX <source>	:TRIGger:UART:SOURce:RX?	<source> ::= {CHANnel<n> | EXTernal} for DSO models <source> ::= {CHANnel<n> | DIGital0,..,DIGital15} for MSO models <n> ::= 1-2 or 1-4 in NR1 format	232[7] MSO[3]
:TRIGger:UART:SOURce:TX <source>	:TRIGger:UART:SOURce:TX?	<source> ::= {CHANnel<n> | EXTernal} for DSO models <source> ::= {CHANnel<n> | DIGital0,..,DIGital15} for MSO models <n> ::= 1-2 or 1-4 in NR1 format	232[7] MSO[3]
:TRIGger:UART:TYPE <value>	:TRIGger:UART:TYPE?	<value> ::= {RSTArt | RSTOp | RDATa | RD1 | RD0 | RDX | PARityerror | TSTArt | TSTOp | TDATa | TD1 | TD0 | TDX}	232[7]
:TRIGger:UART:WIDTh <width>	:TRIGger:UART:WIDTh?	<width> ::= {5 | 6 | 7 | 8 | 9}	232[7]

Table 34. :TRIGger:USB Commands Summary

Command	Query	Options and Query Returns	Dependencies

Command	Query	Options and Query Returns	Dependencies
:TRIGger:USB:SOURce:DMINus <source>	:TRIGger:USB:SOURce:DMINus?	<source> ::= {CHANnel<n> | EXTernal} for the DSO models <source> ::= {CHANnel<n> | DIGital0,..,DIGital15} for the MSO models <n> ::= 1-2 or 1-4 in NR1 format	6000/7000 Series[1] MSO[3]
:TRIGger:USB:SOURce:DPLus <source>	:TRIGger:USB:SOURce:DPLus?	<source> ::= {CHANnel<n> | EXTernal} for the DSO models <source> ::= {CHANnel<n> | DIGital0,..,DIGital15} for the MSO models <n> ::= 1-2 or 1-4 in NR1 format	6000/7000 Series[1] MSO[3]
:TRIGger:USB:SPEed <value>	:TRIGger:USB:SPEed?	<value> ::= {LOW | FULL}	6000/7000 Series[1]
:TRIGger:USB:TRIGger <value>	:TRIGger:USB:TRIGger?	<value> ::= {SOP | EOP | ENTersuspend | EXITsuspend | RESet}	6000/7000 Series[1]

Table 35. :WAVeform Commands Summary

Command	Query	Options and Query Returns	Dependencies
:WAVeform:BYTeorder <value>	:WAVeform:BYTeorder?	<value> ::= {LSBFirst | MSBFirst}	
n/a	:WAVeform:COUNt?	<count> ::= an integer from 1 to 65536 in NR1 format	
n/a	:WAVeform:DATA?	<binary block length bytes>,<binary data> For example, to transmit 1000 bytes of data, the syntax would be: #800001000<1000 bytes of data><NL> 8 is the number of digits that follow 00001000 is the number of bytes to be transmitted <1000 bytes of data> is the actual data	
:WAVeform:FORMat <value>	:WAVeform:FORMat?	<value> ::= {WORD | BYTE | ASCII}	
:WAVeform:POINts <# points>	:WAVeform:POINts?	<# points> ::= {100 | 250 | 500 | 1000 | <points_mode>} if waveform points mode is NORMal <# points> ::= {100 | 250 | 500 | 1000 | 2000 ... 8000000 in 1-2-5 sequence | <points_mode>} if waveform points mode is MAXimum or RAW <points_mode> ::= {NORMal | MAXimum | RAW}	
:WAVeform:POINts:MODE <points_mode>	:WAVeform:POINts:MODE?	<points_mode> ::= {NORMal | MAXimum | RAW}	
n/a	:WAVeform:PREamble?	<preamble_block> ::= <format NR1>,<type NR1>, <points NR1>,<count NR1>, <xincrement NR3>, <xorigin NR3>, <xreference NR1>, <yincrement NR3>, <yorigin NR3>, <yreference NR1> <format> ::= an integer in NR1 format:	

Command	Query	Options and Query Returns	Dependencies
		• 0 for BYTE format • 1 for WORD format • 2 for ASCii format <type> ::= an integer in NR1 format: • 0 for NORMal type • 1 for PEAK detect type • 2 for AVERage type • 3 for HRESolution type <count> ::= Average count, or 1 if PEAK detect type or NORMal; an integer in NR1 format	
n/a	:WAVeform:SEGMented:COUNt?	<count> ::= an integer from 2 to 250 (5000 Series) or 2 to 2000 (6000/7000 Series) in NR1 format	SGM[8]
n/a	:WAVeform:SEGMented:TTAG?	<time_tag> ::= in NR3 format	SGM[8]
:WAVeform:SOURce <source>	:WAVeform:SOURce?	<source> ::= {CHANnel<n> | FUNCtion | MATH | SBUS | POD{1 | 2} | BUS{1 | 2}} <n> ::= 1-2 or 1-4 in NR1 format	4-Channel[2] MSO[3]
:WAVeform:SOURce:SUBSource <subsource>	:WAVeform:SOURce:SUBSource?	<subsource> ::= {{NONE | RX} | TX}	232[7]
n/a	:WAVeform:TYPE?	<return_mode> ::= {NORM | PEAK | AVER | HRES}	
:WAVeform:UNSigned {{0 | OFF} | {1 | ON}}	:WAVeform:UNSigned?	{0 | 1}	
:WAVeform:VIEW <view>	:WAVeform:VIEW?	<view> ::= {MAIN}	
n/a	:WAVeform:XINCrement?	<return_value> ::= x-increment in the current preamble in NR3 format	
n/a	:WAVeform:XORigin?	<return_value> ::= x-origin value in the current preamble in NR3 format	
n/a	:WAVeform:XREFerence?	<return_value> ::= 0 (x-reference value in the current preamble in NR1 format)	
n/a	:WAVeform:YINCrement?	<return_value> ::= y-increment value in the current preamble in NR3 format	
n/a	:WAVeform:YORigin?	<return_value> ::= y-origin in the current preamble in NR3 format	
n/a	:WAVeform:YREFerence?	<return_value> ::= y-reference value in the current preamble in NR1 format	

Table 36. Command Dependencies Descriptions

[1]6000/7000 Series — requires a 6000 Series or 7000 Series oscilloscope. MSO, memory, FPGA probe, and serial options can be added. The 6000 Series oscilloscope battery option is an order-time only option.

[2]4-Channel — requires a 4-channel oscilloscope. Highlighted syntax is only valid on 4-channel oscilloscope models when a serial decode option has been licensed.

[3]MSO — requires a mixed-signal oscilloscope.

[4]LSS — requires the low-speed serial option for a 4-channel oscilloscope.

[5]AMS — requires the automotive serial option for a 4-channel oscilloscope.

[6]FRS — requires the FlexRay serial option for a 4-channel, mixed-signal oscilloscope.

[7]232 — requires the RS-232/UART serial option for a 4-channel, mixed-signal oscilloscope.

[8]SGM — requires the segmented memory option.

[9]LMT — requires the mask test option.