



---

# 12-Bit, 200 MS/s VXIbus Digital Oscilloscope

---

Command Interface Manual

Model ZT432VXI

Copyright © ZTEC Instruments Inc. 2003. All rights reserved under copyright laws of the United States and other countries. All technical data and computer software documentation contained herein is proprietary and confidential to ZTEC Instruments Inc. or its licensor. The reproduction and/or transmission this publication, in whole or in part, by any means, electronics or mechanical, is prohibited without the prior written consent of ZTEC Instruments Inc.

The material in this manual is for informational purposes only and is subject to change without notice. ZTEC Instruments Inc. assumes no responsibility for any error or for consequential damages that may result from the use or misinterpretation of any of the procedures in this publication.

Product names listed are trademarks of their respective manufacturers. Company names listed are trademarks or trade names of their respective companies.

LabVIEW™	National Instruments Corporation
Windows™	Microsoft Corporation



## **HANDLING PRECAUTIONS FOR ELECTRONIC DEVICES SUBJECT TO DAMAGE BY STATIC ELECTRICITY**

The **ZT432VXI** is susceptible to ESD damage. Place instrument or module in a conductive (anti-static) envelope or carrier, when transported. Open only at an ESD approved work surface. An ESD safe work surface is defined as follows:

- 1) The work surface must be conductive and reliably connected to earth ground through a safety resistance of approximately 250 kilohms.
- 2) The surface must NOT be metal. (A resistivity of 30 to 300 kilohms per square inch is suggested.)

Ground the frame of any line-powered equipment, chassis, test instruments, lamps, soldering irons, etc., directly to earth ground. To avoid shorting out the safety resistance, be sure that grounded equipment has rubber feet or other means of insulation from the work surface.

Avoid placing tools or electrical parts on insulators. Do not use any hand tool (such as non-conductive plunger-type solder suckers) that can generate a static charge. Ground yourself reliably, through a resistance, to the work surface using, for example, a conductive strap or cable with a wrist cuff. The cuff must make electrical contact directly with your skin; do NOT wear it over clothing. (Resistance between skin contact and work surface through a commercially available personnel grounding device is typically 250 kilohms to 1 megohm.)

Avoid circumstances that are likely to produce static charges, such as wearing clothes of synthetic material, sitting on a plastic-covered stool (particularly while wearing wool), combing your hair, or making extensive erasures. These circumstances are most significant when the air is dry.

When testing static sensitive devices, be sure DC power is on before, during, and after application of test signals. Be sure all pertinent voltages have been switched off while boards or components are removed or inserted.

### 1.1.1 REVISION HISTORY

Rev	Date	Section	Description
-	9 July 2003		Original Release
1	30 August 2003		Rev_1
2	14 September 2003		Rev_2
3	15 January 2004		Major rewrite of the entire manual

## Table of Contents

<b>HANDLING PRECAUTIONS FOR ELECTRONIC DEVICES SUBJECT TO DAMAGE BY STATIC ELECTRICITY</b>	<b>ii</b>
1.1.1 REVISION HISTORY	iii
<b>1 COMMAND INTERFACE</b>	<b>1</b>
<b>1.1 Command List</b>	<b>3</b>
1.1.1 IEEE Common Commands	3
1.1.2 SCPI Instrument Specific Commands	4
<b>1.2 IEEE 488.2 Common Commands</b>	<b>10</b>
1.2.1 Calibration (*CAL?)	10
1.2.2 Clear Status (*CLS)	11
1.2.3 Event Status Enable (*ESE)	12
1.2.4 Event Status Register (*ESR?)	13
1.2.5 Identification (*IDN?)	14
1.2.6 Operation Complete (*OPC)	15
1.2.7 Options (*OPT?)	16
1.2.8 Recall (*RCL)	17
1.2.9 Reset (*RST)	18
1.2.10 Save (*SAV)	19
1.2.11 Service Request Enable (*SRE)	20
1.2.12 Status Byte (*STB?)	21
1.2.13 Trigger (*TRG)	22
1.2.14 Test (*TST?)	23
1.2.15 Wait (*WAI)	24
<b>1.3 High Level Word Serial Commands</b>	<b>25</b>
1.3.1 Abort (ABORT)	25
1.3.2 Arm (ARM?)	26
1.3.3 Arm Slope (ARM:SLOPe)	27
1.3.4 Arm Source (ARM:SOURce)	28
1.3.4 Calculate Waveform Addition (CALCulate:ADD)	29
1.3.6 Calculate Waveform Copy (CALCulate:COPI)	30
1.3.7 Calculate Waveform Derivative (CALCulate:DERivative)	31
1.3.8 Calculate Waveform Function (CALCulate:FUNCTion?)	32
1.3.9 Calculate Waveform IMMEDIATE (CALCulate:IMMEDIATE)	33
1.3.10 Calculate Waveform Integral (CALCulate:INTegral)	34
1.3.11 Calculate Waveform Invert (CALCulate:INVert)	35
1.3.12 Calculate Waveform MULTiply (CALCulate:MULTiply)	36
1.3.13 Calculate Waveform Offset (CALCulate:OFFSet)	37
1.3.14 Calculate Waveform Range (CALCulate:RANGe)	38
1.3.15 Calculate State (CALCulate:STATe)	39
1.3.16 Calculate Waveform Subtraction (CALCulate:SUBTract)	40
1.3.17 Calibration Report (CALibration:REPort?)	41
1.3.18 Format Byte Order (FORMat:BORDER)	42
1.3.19 Initiate (INITiate?)	43

1.3.20	Initiate Immediate (INITiate:IMMediate)	44
1.3.21	Measure Edge (MEASure:EDGE)	45
1.3.22	Measure Reference Levels (MEASure:REFerence)	46
1.3.23	Measure Voltage, AC (MEASure:VOLTag:e:AC?)	47
1.3.24	Measure Voltage, Amplitude (MEASure:VOLTag:e:AMPLitude?)	48
1.3.25	Measure Voltage, Average (MEASure:VOLTag:e:AVErAge?)	49
1.3.26	Measure Voltage, DC (MEASure:VOLTag:e:DC?)	50
1.3.27	Measure Voltage, Delay (MEASure:VOLTag:e:DELay?)	51
1.3.28	Measure Voltage, Frequency (MEASure:VOLTag:e:FREQuency?)	52
1.3.29	Measure Voltage, Fall Time (MEASure:VOLTag:e:FTIME?)	53
1.3.30	Measure Voltage, High Level (MEASure:VOLTag:e:HIGH?)	54
1.3.31	Measure Voltage, Low Level (MEASure:VOLTag:e:LOW?)	55
1.3.32	Measure Voltage, Maximum Level (MEASure:VOLTag:e:MAXimum?)	56
1.3.33	Measure Voltage, Minimum Level (MEASure:VOLTag:e:MINimum?)	57
1.3.34	Measure Voltage, Negative Duty Cycle (MEASure:VOLTag:e:NDUTY?)	58
1.3.35	Measure Voltage, Negative Width (MEASure:VOLTag:e:NWIDth?)	59
1.3.36	Measure Voltage, Overshoot (MEASure:VOLTag:e:OVERshoot?)	60
1.3.37	Measure Voltage, Positive Duty Cycle (MEASure:VOLTag:e:PDUTY?)	61
1.3.38	Measure Voltage, Period (MEASure:VOLTag:e:PERiod?)	62
1.3.39	Measure Voltage, Preshoot (MEASure:VOLTag:e:PREShoot?)	63
1.3.40	Measure Voltage, Peak-to-Peak (MEASure:VOLTag:e:PTPeak?)	64
1.3.41	Measure Voltage, Positive Width (MEASure:VOLTag:e:PWIDth?)	65
1.3.42	Measure Voltage, Rise Time (MEASure:VOLTag:e:RTIME?)	66
1.3.43	Measure Voltage, Crossing Time (MEASure:VOLTag:e:TCross?)	67
1.3.44	Measure Voltage, Time at Maximum (MEASure:VOLTag:e:TMAXimum?)	68
1.3.45	Measure Voltage, Time at Minimum (MEASure:VOLTag:e:TMINimum?)	69
1.3.46	Measure Voltage, Time at Voltage (MEASure:VOLTag:e:TVOLTage?)	70
1.3.47	Measure Voltage, Voltage at Time (MEASure:VOLTag:e:VTIME?)	71
1.3.48	Output ECL Trigger Polarity (OUTPut:ECLTrg<n>:POLarity)	72
1.3.49	Output ECL Trigger Source (OUTPut:ECLTrg<n>:SOURce)	73
1.3.50	Output ECL Trigger State (OUTPut:ECLTrg<n>:STATe)	74
1.3.51	Output Reference Source (OUTPut:REFerence:SOURce)	75
1.3.52	Output Reference State (OUTPut:REFerence:STATe)	76
1.3.53	Output TTL Trigger Polarity (OUTPut:TTLTrg<n>:POLarity)	77
1.3.54	Output TTL Trigger Source (OUTPut:TTLTrg<n>:SOURce)	78
1.3.55	Output TTL Trigger State (OUTPut:TTLTrg<n>:STATe)	79
1.3.56	Sense Autoscale (SENSE:AUToscale)	80
1.3.57	Sense Average Count (:SENSE:AVErAge:COUNt)	81
1.3.58	Sense Average State (SENSE:AVErAge:STATe)	82
1.3.59	Sense Average Type (SENSE:AVErAge:TYPE)	83
1.3.60	Sense Channel Count (SENSE:CHANnel:COUNt?)	84
1.3.61	Sense Channel Mode (SENSE:CHANnel:MODE)	85
1.3.62	Sense Input Attenuation (SENSE:INPut:ATTenuation)	86
1.3.63	Sense Input Coupling (SENSE:INPut:COUPling)	87
1.3.64	Sense Input Impedance (SENSE:INPut:IMPedance)	88
1.3.65	Sense Input Preset (SENSE:INPut:PRESet)	89
1.3.66	Sense Input Smoothing (SENSE:INPut:SMOothing:POINts)	90
1.3.67	Sense Input Smoothing State (SENSE:INPut:SMOothing:STATe)	91
1.3.68	Sense Input State (SENSE:INPut:STATe)	92
1.3.69	Sense Reference Oscillator Source (SENSE:ROSCillator:SOURce)	93

1.3.70	Sense Sweep Mode (SENSE:SWEEP:MODE)	94
1.3.71	Sense Sweep Offset Time (SENSE:SWEEP:OFFSET:TIME)	95
1.3.72	Sense Sweep Offset Reference Location (SENSE:SWEEP:REFERENCE:LOCATION)	96
1.3.73	Sense Sweep Points (SENSE:SWEEP:POINTS)	97
1.3.74	Sense Sweep Sample Rate (SENSE:SWEEP:SRATE)	99
1.3.75	Sense Sweep Time (SENSE:SWEEP:TIME)	100
1.3.76	Sense Sweep Time Interval (SENSE:SWEEP:TINTERVAL?)	101
1.3.77	Sense Voltage Range Offset (SENSE:VOLTAGE:RANGE:OFFSET?)	102
1.3.78	Sense Voltage Range Peak-to-Peak (SENSE:VOLTAGE:RANGE:PTPEAK)	103
1.3.79	Status Operation Event (STATUS:OPERATION:EVENT?)	105
1.3.80	Status Operation Condition (STATUS:OPERATION:CONDITION?)	106
1.3.81	Status Operation Enable (STATUS:OPERATION:ENABLE)	107
1.3.82	Status Preset (STATUS:PRESET)	108
1.3.83	Status Questionable Event (STATUS:QUESTIONABLE?)	109
1.3.84	Status Questionable Condition (STATUS:QUESTIONABLE:CONDITION?)	110
1.3.85	Status Questionable Enable (STATUS:QUESTIONABLE:ENABLE)	111
1.3.86	System Error All (SYSTEM:ERROR:ALL?)	112
1.3.87	System Error Count (SYSTEM:ERROR:COUNT?)	113
1.3.88	System Error Next (SYSTEM:ERROR:NEXT?)	114
1.3.89	System Memory (SYSTEM:MEMORY?)	115
1.3.90	System Version (SYSTEM:VERSION?)	116
1.3.91	Test Report (TEST:REPORT?)	117
1.3.92	Trace Address (TRACE:ADDRESS?)	118
1.3.93	Trace Data (TRACE:DATA)	119
1.3.94	Trace Download Immediate (TRACE:DOWNLOAD:IMMEDIATE)	121
1.3.95	Trace Download State (TRACE:DOWNLOAD:STATE)	122
1.3.96	Trace Flags State (TRACE:FLAGS:STATE)	123
1.3.97	Trace Preamble (TRACE:PREAMBLE)	124
1.3.98	Trace Reference Copy (TRACE:REFERENCE:COPY)	127
1.3.99	Trace Time Stamp (TRACE:TIMESTAMP?)	128
1.3.100	Trigger A Event Count (TRIGGER:A:ECOUNT)	129
1.3.101	Trigger A Pattern Mask (TRIGGER:A:PATTERN:MASK)	130
1.3.102	Trigger A Pattern Truth (TRIGGER:A:PATTERN:TRUTH)	131
1.3.103	Trigger A Pulse High Limit (TRIGGER:A:PULSE:HLIMIT)	132
1.3.104	Trigger A Pulse Low Limit (TRIGGER:A:PULSE:LLIMIT)	133
1.3.105	Trigger A Slope (TRIGGER:A:SLOPE)	134
1.3.106	Trigger A Source (TRIGGER:A:SOURCE)	135
1.3.107	Trigger A Type (TRIGGER:A:TYPE)	136
1.3.108	Trigger B Slope (TRIGGER:B:SLOPE)	138
1.3.109	Trigger B Source (TRIGGER:B:SOURCE)	139
1.3.110	Trigger B State (TRIGGER:B:SOURCE)	140
1.3.111	Trigger Complete (TRIGGER:COMPLETE)	141
1.3.112	Trigger Count (TRIGGER:COUNT)	142
1.3.113	Trigger External Level (TRIGGER:EXTERNAL:LEVEL)	143
1.3.114	Trigger Holdoff (TRIGGER:HOLDOFF)	144
1.3.115	Trigger Input Level (TRIGGER:INPUT:LEVEL)	145
1.3.116	Trigger Maximum Count (TRIGGER:MCCOUNT)	146

## 1 Command Interface

The ZT432VXI is an IEEE 488.2 compliant VXIbus instrument. This indicates that the ZT432VXI supports the data transfer and recognition of ASCII strings using the Byte Available and Byte Request low-level VXIbus commands. These ASCII strings correspond to commands and inquiries directed at the ZT432VXI. In accordance with the IEEE 488.2 specification, the ZT432VXI supports a several IEEE-488 common commands as listed in Section 5.1 and defined in Section 5.2. More information on the IEEE 488.2 common commands can be found in the IEEE Standard 488.2-1992 specification.

The ZTZ432VXI also supports a number of SCPI related commands to control instrument features as listed in Section 5.1 and described in Section 5.3. More information on the SCPI commands can be found In the SCPI 1997 specification.

The following definitions are used in the command descriptions:

<b>n/a</b>	not applicable
<b>None</b>	the command does not support this value
<b>[name:]</b>	optional command mnemonic, example for the command prototype [SENSe:]INPut:COUPling? SENSe:INPut:COUPling? and INPut:COUPling? are equivalent
<b>COMMand</b>	command mnemonic showing the long and short versions of the command, for the example shown at left, either COMM or COMMAND represent the same mnemonic
<b>&lt;name&gt;</b>	command or query parameter or channel number suffix

Where optional command mnemonics are allowed, command descriptions are shown in two forms: full (with optional mnemonics) and minimum (without optional mnemonics). The two command forms are equivalent. An example of this is shown below:

<b>Command Syntax</b>	INITiate[:IMMediate] INITiate	full minimum
<b>Query Syntax</b>	n/a	

The scale modifiers shown in Table 5-1 are allowed when entering numeric values. Default values are shown in **BOLD** face.

<b>Multiplier</b>	<b>Frequency</b>	<b>Ohms</b>	<b>Time</b>	<b>Voltage</b>
1.0e+9	GHZ			
1.0e+6	MHZ	MOHM		
1.0e+3	KHZ	KOHM		KV
1.0e+0	HZ	OHM	S	V
1.0e-3			MS	MV
1.0e-6			US	UV
1.0e-9			NS	

**Table 5-1: Numeric Value Scale Modifiers**

Text sections shown in *italics* are planned features, supported by the command interface but not implemented.

Three command parameter types are used: numeric, character, and string.

Numeric values may be entered in one of formats:

- decimal            number without a decimal or exponent, ex. 255
- float             number with a decimal, exponent, or both, ex. 1.0e1
- binary            binary number beginning with #B, ex. #B101101
- octal             octal number beginning with #O, ex. #O7654
- hexadecimal     hex number beginning with #H, ex. #HABCD

Character values are words containing up to 12 characters followed by a decimal suffix, ex. INP1 or MAXimum. In some cases, the character value may be abbreviated. Numeric suffixes are retained without modification by the abbreviation. The use of abbreviations is allowed when specifically indicated by the CAPS/non-caps parameter prototype. Examples are:

- INP1              character value INP, suffix value 1
- MAXimum        character value that may be represented by MAXIMUM or MAX
- CALCulate1     character value that may be represented by CALCulate1or CALC1

String values are designated by single or double quotes. The contents of the paired quotes are considered a single value. The pair quotes may be either single or double quotes, and may not be mixed within a single string value. Examples are:

- 'TEST'            string value TEST
- "TEST"            string value TEST
- 'TEST"            illegal string value, single and double quotes may not be mixed

# 1

---

## 1.1 Command List

---

### 1.1.1 IEEE Common Commands

\*CAL?  
\*CLS  
\*ESE <mask>  
\*ESE?  
\*ESR?  
\*IDN?  
\*OPC  
\*OPC?  
\*OPT?  
\*RCL <state>  
\*RST  
\*SAV <state>  
\*SRE <mask>  
\*SRE?  
\*STB?  
\*TRG  
\*TST?  
\*WAI

## 1.1.2 SCPI Instrument Specific Commands

### Arm, Abort and Initiate Subsystems

ABORt  
ARM?  
ARM:SLOPe <POS | NEG>  
ARM:SLOPe?  
ARM:SOURce <ARM | TTLTn | ECLTn | IMM>  
ARM:SOURce?  
INITiate?  
INITiate:IMMediate

### Calculate Subsystem

CALCulate<n>:ADD <INPn | REFn | CALCn>,<INPn | REFn | CALCn>,<num>,<num>  
CALCulate<n>:COPY <INPn | REFn | CALCn>,<num>  
CALCulate<n>:DERivative <INPn | REFn | CALCn>,<num>  
CALCulate<n>:FUNction?  
CALCulate<n>:IMMediate  
CALCulate<n>:INTEgral <INPn | REFn | CALCn>,<num>  
CALCulate<n>:INVert <INPn | REFn | CALCn>,<num>  
CALCulate<n>:MULTiply <INPn | REFn | CALCn>,<INPn | REFn | CALCn>,<num>,<num>  
CALCulate<n>:OFFSet <volts>  
CALCulate<n>:OFFSet?  
CALCulate<n>:RANGe <volts>  
CALCulate<n>:RANGe?  
CALCulate<n>:STATe <ON | OFF>  
CALCulate<n>:STATe?  
CALCulate<n>:SUBTract <INPn | REFn | CALCn>,<INPn | REFn | CALCn>,<num>,<num>

### Calibration Subsystem

CALibration:REPort?

### Format Subsystem

FORMat:BORDer <SWAP | NORM>  
FORMat:BORDER?

## Measure Subsystem

MEASure:EDGE <num>,<num>  
MEASure:EDGE?  
MEASure:REFerence <num>,<num>,<num>  
MEASure:REFerence?  
MEASure:VOLTage:AC? <INPn | REFn | CALCn>,<num>  
MEASure:VOLTage:AMPLitude? <INPn | REFn | CALCn>,<num>  
MEASure:VOLTage:AVERage? <INPn | REFn | CALCn>,<num>  
MEASure:VOLTage:DC? <INPn | REFn | CALCn>,<num>  
MEASure:VOLTage:DELay? <INPn | REFn | CALCn>,<INPn | REFn | CALCn>,<num>  
MEASure:VOLTage:FREQuency? <INPn | REFn | CALCn>,<num>  
MEASure:VOLTage:FTIME? <INPn | REFn | CALCn>,<num>  
MEASure:VOLTage:HIGH? <INPn | REFn | CALCn>,<num>  
MEASure:VOLTage:LOW? <INPn | REFn | CALCn>,<num>  
MEASure:VOLTage:MAXimum? <INPn | REFn | CALCn>,<num>  
MEASure:VOLTage:MINimum? <INPn | REFn | CALCn>,<num>  
MEASure:VOLTage:NDUTYcycle? <INPn | REFn | CALCn>,<num>  
MEASure:VOLTage:NWIDTH? <INPn | REFn | CALCn>,<num>  
MEASure:VOLTage:OVERshoot? <INPn | REFn | CALCn>,<num>  
MEASure:VOLTage:PDUTYcycle? <INPn | REFn | CALCn>,<num>  
MEASure:VOLTage:PERiod? <INPn | REFn | CALCn>,<num>  
MEASure:VOLTage:PREShoot? <INPn | REFn | CALCn>,<num>  
MEASure:VOLTage:PTPeak? <INPn | REFn | CALCn>,<num>  
MEASure:VOLTage:PWIDth? <INPn | REFn | CALCn>,<num>  
MEASure:VOLTage:RTIME? <INPn | REFn | CALCn>,<num>  
MEASure:VOLTage:TCrossing? <INPn | REFn | CALCn>,<num>  
MEASure:VOLTage:TMAXimum? <INPn | REFn | CALCn>,<num>  
MEASure:VOLTage:TMINimum? <INPn | REFn | CALCn>,<num>  
MEASure:VOLTage:TVOLT? <INPn | REFn | CALCn>,<num>,<volts>  
MEASure:VOLTage:VTIME? <INPn | REFn | CALCn>,<num>,<time>

## Output Subsystem

OUTPut:ECLTrg:POLarity <POS | NEG>  
OUTPut:ECLTrg:POLarity?  
OUTPut:ECLTrg:SOURce <TRG | OPC | ARM | NONE>  
OUTPut:ECLTrg:SOURce?  
OUTPut:ECLTrg:STATe <ON | OFF>  
OUTPut:ECLTrg:STATe?  
OUTPut:REFeRence:SOURce <COMP | PULS | REF | VOLT | TRGA | TRGB>  
OUTPut:REFeRence:SOURce?  
OUTPut:REFeRence:STATe <ON | OFF>  
OUTPut:REFeRence:STATe?  
OUTPut:TTLTrg:POLarity <POS | NEG>  
OUTPut:TTLTrg:POLarity?  
OUTPut:TTLTrg:SOURce <TRG | OPC | ARM | NONE>  
OUTPut:TTLTrg:SOURce?  
OUTPut:TTLTrg:STATe <ON | OFF>  
OUTPut:TTLTrg:STATe?

## Sense Subsystem

SENSe:AUToscale  
SENSe:AVERage:COUNT <count>  
SENSe:AVERage:COUNT?  
SENSe:AVERage:STATe <ON | OFF>  
SENSe:AVERage:STATe?  
SENSe:AVERage:TYPE <NORM | ENV>  
SENSe:AVERage:TYPE?  
SENSe:CHANnel:COUNT?  
SENSe:CHANnel:MODE <ALL4 | CHAN12 | CHAN34>  
SENSe:CHANnel:MODE?  
SENSe:INPut<n>:ATTenuation <atten>  
SENSe:INPut<n>:ATTenuation?  
SENSe:INPut<n>:COUPling <AC | DC>  
SENSe:INPut<n>:COUPling?

**Sense Subsystem (cont.)**

SENSe:INPut<n>:IMPedance <50 | 1e6 | MIN | MAX>

SENSe:INPut<n>:IMPedance?

SENSe:INPut<n>:SMOothing:POINts <10 | 50>

SENSe:INPut<n>:SMOothing:POINts?

SENSe:INPut<n>:SMOothing:STATe <ON | OFF>

SENSe:INPut<n>:SMOothing:STATe?

SENSe:INPut<n>:PRESet <TTL | ECL>

SENSe:INPut<n>:STATe <ON | OFF>

SENSe:INPut<n>:STATe?

SENSe:ROSCillator:SOURce <INT | VXI>

SENSe:ROSCillator:SOURce?

SENSe:SWEEp:ESCLock:STATe <ON | OFF>

SENSe:SWEEp:ESCLock:STATe?

SENSe:SWEEp:ESCLock:TINTerval <time>

SENSe:SWEEp:ESCLock:TINTerval?

SENSe:SWEEp:MODE <AUTO | NORM>

SENSe:SWEEp:MODE?

SENSe:SWEEp:OFFSet:TIME <time>

SENSe:SWEEp:OFFSet:TIME?

SENSe:SWEEp:OREFerance:LOCation <position>

SENSe:SWEEp:OREFerance:LOCation?

SENSe:SWEEp:POINts <points>

SENSe:SWEEp:POINts?

SENSe:SWEEp:SRATE?

SENSe:SWEEp:TIME <time>

SENSe:SWEEp:TIME?

SENSe:SWEEp:TINTerval?

SENSe:VOLTage<n>:RANGe:OFFSet?

SENSe:VOLTage<n>:RANGe:PTPeak <volts>

SENSe:VOLTage<n>:RANGe:PTPeak?

**Status Subsystem**

STATus:OPERation:CONDition?  
STATus:OPERation:ENABle <mask>  
STATus:OPERation:ENABle?  
STATus:OPERation:EVENT?  
STATus:PRESet  
STATus:QUEStionable:CONDition?  
STATus:QUEStionable:ENABle <mask>  
STATus:QUEStionable:ENABle?  
STATus:QUEStionable:EVENT?

**System Subsystem**

SYSTem:ERRor:ALL?  
SYSTem:ERRor:COUNt?  
SYSTem:ERRor:NEXt?  
SYSTem:MEMory?  
SYSTem:VERSion?

**Test Subsystem**

TEST:REPort?

**Trace Subsystem**

TRACe:ADDReSS? <INPn | REFn | CALCn | ALL>,<num>  
TRACe:DATA <REFn>  
TRACe:DATA? <INPn | REFn | CALCn>,<num>  
TRACe:DOWNload:IMMediate  
TRACe:DOWNload:STATe <ON | OFF>  
TRACe:DOWNload:STATe?  
TRACe:FLAGs:STATe <ON | OFF>  
TRACe:FLAGs:STATe?  
TRACe:PREamble <REFn>  
TRACe:PREamble? <INPn | REFn | CALCn>,<num>  
TRACe:REFerence<n>:COPY <INPn | REFn | CALCn>,<num>  
TRACe:REFerence<n>:STATe <ON | OFF>  
TRACe:REFerence<n>:STATe?  
TRACe:TIMestamp? <num>

## Trigger Subsystem

TRIGger:A:ECOunt <events>  
TRIGger:A:ECOunt?  
TRIGger:A:PATTern:MASK <mask>  
TRIGger:A:PATTern:MASK?  
TRIGger:A:PATTern:TRUTH <truth>  
TRIGger:A:PATTern:TRUTH?  
TRIGger:A:PULSe:HLIMit <time>  
TRIGger:A:PULSe:HLIMit?  
TRIGger:A:PULSe:LLIMit <time>  
TRIGger:A:PULSe:LLIMit?  
TRIGger:A:SLOPe <POS | NEG>  
TRIGger:A:SLOPe?  
TRIGger:A:SOURce <INPn | EXT | TTLTn | ECLTn | ARM | PATT | NONE>  
TRIGger:A:SOURce?  
TRIGger:A:TYPE <EDGE | PIN | POUT | PLTH | PGTH>  
TRIGger:A:TYPE?  
TRIGger:B:STATe <ON | OFF>  
TRIGger:B:STATe?  
TRIGger:B:SLOPe <POS | NEG>  
TRIGger:B:SLOPe?  
TRIGger:B:SOURce <INPn | EXT | TTLTn | ECLTn | ARM | PATT | NONE>  
TRIGger:B:SOURce?  
TRIGger:COMPLete?  
TRIGger:COUNt <count>  
TRIGger:COUNt?  
TRIGger:EXTernal:LEVel <volts>  
TRIGger:EXTernal:LEVel?  
TRIGger:HOLDoff?  
TRIGger:HOLDoff <time>  
TRIGger:INPut<n>:LEVel <volts>  
TRIGger:INPut<n>:LEVel?  
TRIGger:MCOunt?

## 1.2 IEEE 488.2 Common Commands

In accordance with the IEEE 488.2 specification, the ZT432VXI supports the common commands listed in this section. More information on the IEEE 488.2 common commands can be found in the IEEE Standard 488.2-1992 specification.

### 1.2.1 Calibration (\*CAL?)

**COMMAND SYNTAX** n/a

**QUERY SYNTAX** \*CAL?

This command Initiates the unit self-calibration process and returns a pass/fail code upon completion of the process. Refer to The Installation, Operation, Programming, & Diagnostics Manual, Appendix B for descriptions of the return codes. The following considerations apply when using the \*CAL? query:

- If more than one failure occurs, the response includes only the first failure code. The return code is saved and can be queried at any time using the :CALibration:REPort? Query
- Results of the calibration are saved in volatile memory. Factory defaults are restored after the next power up or \*RST command
- All front panel inputs must be removed before sending this command.
- No other process should be initiated while the self-calibration process is running.
- The STATus:OPERation:CONDition? query can be used to determine the state of the calibration process.

**COMMAND PARAMETERS** None

<b>RESPONSE PARAMETERS</b>	<code>	self-calibration results code
	0	calibration successful
	other	auto advance enabled

**DEFAULT VALUE** n/a

**EXAMPLES** \*CAL? initiate unit self-calibration, example response: 0 (or self-calibration passed)

**STATUS EVENTS** None

**AFFECTIVITY** None

**RELATED COMMANDS** CALibration:RESults?

## 1.2.2 Clear Status (\*CLS)

**COMMAND SYNTAX** \*CLS

**QUERY SYNTAX** n/a

Clears all SCPI and IEEE 488.2 defined event registers and the Status Queue. Event reporting enables are not affected, use the STATus:PRESet command to clear the event reporting enable registers.

**COMMAND PARAMETERS** None

**RESPONSE PARAMETERS** None

**DEFAULT VALUE** n/a

**EXAMPLES** \*CLS clear event status and the Status Queue

**STATUS EVENTS** None

**AFFECTIVITY** The Operation Status Event Register, Questionable Status Event Register, and Standard Event Status Register are cleared.

**RELATED COMMANDS** STATus:PRESet

### 1.2.3 Event Status Enable (\*ESE)

**COMMAND SYNTAX** \*ESE <mask>

**QUERY SYNTAX** \*ESE?

Sets or queries the Standard Event Status reporting mask. The following considerations apply when using the \*ESE command:

- Each bit in the mask corresponds to the associated bit in the Standard Event Status Register, refer to The Installation, Operation, Programming, & Diagnostics Manual for more information on the Standard Event Status Register.
- Enabled event flags in the Standard Event Register are summarized and reported in bit 5 of the Status Byte.
- The Standard Event Status Register can be read using the \*ESR? query.

<b>COMMAND PARAMETERS</b>	<mask>	event reporting mask	0 to 255	bit-mapped mask
<b>RESPONSE PARAMETERS</b>	<mask>	event reporting mask		
<b>DEFAULT VALUE</b>	0			
<b>EXAMPLES</b>	*ESE 0	disable event reporting from the Standard Event Register		
	*ESE?	query current event reporting mask, example response: 129 (or enable power-on and OPC event reporting)		
<b>STATUS EVENTS</b>	-222	attempt to set the mask to an illegal value, command was ignored		
<b>AFFECTIVITY</b>	None			
<b>RELATED COMMANDS</b>	*ESR?			

## 1.2.4 Event Status Register (\*ESR?)

**COMMAND SYNTAX** n/a

**QUERY SYNTAX** \*ESR?

Queries the current Standard Event Status Register contents. The Standard Event Status Register is a bit-mapped record of certain status events, refer to The Installation, Operation, Programming, & Diagnostics Manual for more information on the Standard Event Status Register. The following considerations apply when using the \*ESR? query:

- the Standard Event Status Register records the events occurring since the last \*ESR? query.
- the \*ESR? clears the Standard Event Status Register after reading it
- selected flags in the Standard Event Status Register may be enabled for reporting to the Status Byte, refer to the \*ESE command form more information.

**COMMAND PARAMETERS** n/a

**RESPONSE PARAMETERS** <flags> record of events occurring since the last \*ESR? query  
0 to 255 bit-mapped status flags

**DEFAULT VALUE** n/a

**EXAMPLES** \*ESR? query the Standard Event Status Register, example response: 129 (or power-on and OPC events have occurred since the last \*ESR? query)

**STATUS EVENTS** None

**AFFECTIVITY** Reading this register clears it.

**RELATED COMMANDS** \*ESE

### 1.2.5 Identification (\*IDN?)

**COMMAND SYNTAX**    n/a

**QUERY SYNTAX**        \*IDN?

Returns the unit identification message. The following, comma-separated, information is contained in the response:

- first field, manufacturer name
- second field, unit model name
- third field, unit serial number
- fourth field, unit firmware version number

**COMMAND PARAMETERS**    n/a

**RESPONSE PARAMETERS**    <id>                    unit identification string

**DEFAULT VALUE**            n/a

**EXAMPLES**                \*IDN?                    query id string, example response: ZTEC Inc.,ZT432VXI,S/N 35,Version 00.15

**STATUS EVENTS**            None

**AFFECTIVITY**             None

**RELATED COMMANDS** None

## 1.2.6 Operation Complete (\*OPC)

**COMMAND SYNTAX** \*OPC

**QUERY SYNTAX** \*OPC?

Generates a response when all currently running processes are complete. The \*OPC and \*OPC? result in different unit actions as described below. The following considerations apply when using the \*OPC command:

- this command is intended to synchronize command execution with the chassis controller.
- \*OPC causes an Operation Complete event to be generated in the Standard Event Status Register when all currently running processes are complete, refer to the \*ESR? query for more information on reading the Standard Event Status Register.
- \*OPC? causes an ASCII '1' to be placed in the response queue when all currently running processes are complete.
- If there are multiple commands in a command string, the \*OPC or \*OPC? commands must be the last entry.
- The \*OPC? query does not complete until all currently running processes are complete, this has the effect of temporarily stopping the command interface. The command interface can be restarted if necessary using the low-level word-serial CLEAR command.
- \*OPC monitors process activity in the unit background execution and does not stop the command interface
- the \*WAI command performs an action similar to the \*OPC? query but without generating a response when complete.

**COMMAND PARAMETERS** None

**RESPONSE PARAMETERS** None

**DEFAULT VALUE** n/a

**EXAMPLES** \*OPC

generate an Operation Complete event in the Standard Event Status Register when all currently running processes are complete

\*OPC?

generate a response queue entry when all currently running processes are complete, example response: 1

**STATUS EVENTS** None

**AFFECTIVITY**

The \*OPC? query temporarily stops the command interface until all currently running processes are complete.

**RELATED COMMANDS** \*ESR?

\*WAI

## 1.2.7 Options (\*OPT?)

**COMMAND SYNTAX** n/a

**QUERY SYNTAX** \*OPT?

Queries the unit installed options as a comma-separated list. Contact ZTEC for information on the options string interpretation.

**COMMAND PARAMETERS** n/a

**RESPONSE PARAMETERS** <opt> unit installed options list

**DEFAULT VALUE** n/a

**EXAMPLES** \*OPT? query unit installed options, example response: V128

**STATUS EVENTS** None

**AFFECTIVITY** None

**RELATED COMMANDS** None

# 1

## 1.2.8 Recall (\*RCL)

**COMMAND SYNTAX** \*RCL <num>

**QUERY SYNTAX** n/a

Recalls a previously saved instrument configuration. The following considerations apply when using the \*RCL command:

- up to 48 user-defined instrument settings may be saved in non-volatile memory using the \*SAV command.
- Factory defaults may be restored using the \*RST command.
- Saved instrument configurations include waveform capture and processing configurations only, unit status, status reporting, and reference waveforms are not affected
- \*CLS and STATus:PRESet commands may be used to clear unit status and status reporting structures, respectively

<b>COMMAND PARAMETERS</b>	<num>	instrument configuration setting to recall 1 to 48 setting number
<b>RESPONSE PARAMETERS</b>	n/a	
<b>DEFAULT VALUE</b>	n/a	
<b>EXAMPLES</b>	*RCL 1	recall instrument configuration number 1
<b>STATUS EVENTS</b>	-101	Attempted to recall an uninitialized or corrupted instrument setting, command was ignored
	-222	Attempted to recall an invalid setting number, command was ignored
<b>AFFECTIVITY</b>		All instrument settings except status structures and reference waveforms are updated.
<b>RELATED COMMANDS</b>	*CLS *SAV *RS STATus:PRESet	

**1.2.9 Reset (\*RST)**

**COMMAND SYNTAX** \*RST

**QUERY SYNTAX** n/a

Resets the unit to factory defaults instrument settings. The following considerations apply when using the \*RST command:

- the \*RST command will stop all currently running processes and restore factory default instrument settings and calibration values.
- unit instrument reset updates waveform capture and processing configurations only, unit status, status reporting, and reference waveforms are not affected
- \*CLS and STATus:PRESet commands may be used to clear unit status and status reporting structures, respectively.

**COMMAND PARAMETERS** None

**RESPONSE PARAMETERS** n/a

**DEFAULT VALUE** n/a

**EXAMPLES** \*RST restore factory default settings

**STATUS EVENTS** None

**AFFECTIVITY** All instrument settings except status structures and reference waveforms are updated.

**RELATED COMMANDS** \*CLS  
\*RCL  
\*SAV  
STATus:PRESet

## 1.2.10 Save (\*SAV)

**COMMAND SYNTAX** \*SAV <num>

**QUERY SYNTAX** n/a

Saves the current instrument settings in nonvolatile memory. The following considerations apply when using the \*SAV command:

- up to 48 user-defined instrument settings may be saved in non-volatile memory.
- user-defined instrument settings may be restored using the \*RCL command.
- Factory defaults may be restored using the \*RST command.
- Saved instrument configurations include waveform capture and processing configurations only, unit status, status reporting, and reference waveforms are not affected

<b>COMMAND PARAMETERS</b>	<num>	instrument configuration setting to save 1 to 48 setting number
<b>RESPONSE PARAMETERS</b>	n/a	
<b>DEFAULT VALUE</b>	n/a	
<b>EXAMPLES</b>	*SAV 4	save the current instrument configuration to storage location 4
<b>STATUS EVENTS</b>	-222	Attempted to save to an invalid setting number, command was ignored
<b>AFFECTIVITY</b>	None	
<b>RELATED COMMANDS</b>	*CLS *RCL *RST STATus:PRESet	

### 1.2.11 Service Request Enable (\*SRE)

**COMMAND SYNTAX**    \*SRE <mask>

**QUERY SYNTAX**        n/a

Sets or queries the service request enable mask. The following considerations apply when using the \*SRE command:

- Each bit in the mask corresponds to the associated bit in the Status Byte, refer to The Installation, Operation, Programming, & Diagnostics Manual for more information on the Status Byte.
- The enabled Status Byte flags are summarized in the Master Status Summary bit, bit 6.
- When enabled, the summary bit will also generate an interrupt to the system controller, refer to the low-level VXIbus ASSIGN INTERRUPTER LINE and CONTROL EVENT commands
- The Status Byte can be read using the \*STB? query.

<b>COMMAND PARAMETERS</b>	<mask>	event reporting mask 0 to 255	bit-mapped mask
<b>RESPONSE PARAMETERS</b>	<mask>	event reporting mask	
<b>DEFAULT VALUE</b>	0		
<b>EXAMPLES</b>	*SRE 0	disable summary bit and service request generation	
	*SRE?	query current event reportingmask, example response: 32	
<b>STATUS EVENTS</b>	-222	attempt to set the mask to an illegal value, command was ignored	
<b>AFFECTIVITY</b>	None		
<b>RELATED COMMANDS</b>	*STB?		

# 1

## 1.2.12 Status Byte (\*STB?)

**COMMAND SYNTAX** n/a

**QUERY SYNTAX** \*STB?

Queries the Status Byte contents. The Status Byte is bit-mapped register reflecting the state of the unit status, refer to The Installation, Operation, Programming, & Diagnostics Manual for more information on the Standard Byte. The following considerations apply when using the \*STB? query:

- status reporting to the Status Byte from the Operation Status Event, Questionable Event, and Standard Event Status Registers is enabled using the STATus:OPERation:ENABle, STATus:QUEStionable:ENABle, and \*ESR commands, respectively.
- The Status Byte is mas ked with the Service Request Enable Register to form the Master Status Summary bit, bit 6, refer to the \*SRE command for information on setting the summary bit generation mask.
- When enabled, the summary bit will also generate an interrupt to the system controller, refer to the low-level VXIbus ASSIGN INTERRUPTER LINE and CONTROL EVENT commands
- the Status Byte is also accessible using the VXIbus low-level READ STB command. This command avoids the command interface protocols providing a very low overhead method to access the Status Byte. The READ STB command is available even when the command interface is busy.
- Reading the Status Byte does not clear it.

**COMMAND** n/a

**PARAMETERS**

**RESPONSE** <flag>  
**PARAMETERS**

current unit status  
0 to 255 bit-mapped status flags

**DEFAULT VALUE** n/a

**EXAMPLES** \*STB? query the Status Byte, example response: 96 (or the Master Status Summary and Standard Event reporting bits are set)

**STATUS EVENTS** None

**AFFECTIVITY** None

**RELATED** \*ESE  
**COMMANDS** \*SRE

STATus:OPERation:ENABle  
STATus:QUEStionable:ENABle

**1.2.13 Trigger (\*TRG)**

**COMMAND SYNTAX** \*TRG

**QUERY SYNTAX** n/a

Forces a trigger event in the waveform capture controls. The following considerations apply when using the \*TRG command:

- a waveform capture cycle must be in progress for this command to have an effect
- the \*TRG command overrides all other trigger controls and forces Trigger A and B events. This command overrides the trigger count and causes an immediate trigger
- The \*TRG command can be made the only trigger source using the TRIGger:A:SOURce command with NONE as the trigger source.
- A trigger event can also be forced using the VXIbus low-level TRIGGER command. This command avoids the command interface protocols providing a very low overhead method of forcing a trigger event. The TRIGGER command is available even when the command interface is busy.

**COMMAND PARAMETERS** None

**RESPONSE PARAMETERS** n/a

**DEFAULT VALUE** n/a

**EXAMPLES** \*TRG force a trigger event

**STATUS EVENTS** None

**AFFECTIVITY** The \*TRG command over-rides all other trigger selections and controls.

**RELATED COMMANDS** TRIGger:A:SOURce  
TRIGger:B:SOURce

# 1

---

## 1.2.14 Test (\*TST?)

**COMMAND SYNTAX** n/a

**QUERY SYNTAX** \*TST?

Initiates the unit self-test process and returns a pass/fail code upon completion of the process. Refer to The Installation, Operation, Programming, & Diagnostics Manual, Appendix B of the The Installation, Operation, Programming, & Diagnostics Manual for descriptions of the return codes. The following considerations apply when using the \*TST? query:

- If more than one failure occurs, the response includes only the first failure code.
- The return code is saved and can be queried at any time using the :TEST:REPort? query
- No other process should be initiated while the self-calibration process is running.
- The self-test process state can be determined using the :STATus:OPERation:CONDition? query.

**COMMAND PARAMETERS** n/a

**RESPONSE PARAMETERS** <code> self-test results code

**DEFAULT VALUE** n/a

**EXAMPLES** \*TST? initiate unit self-test, example response: 0

**STATUS EVENTS** None

**AFFECTIVITY** None

**RELATED COMMANDS** TEST:RESults?

**1.2.15 Wait (\*WAI)**

**COMMAND SYNTAX** \*WAI

**QUERY SYNTAX** n/a

Temporarily stops the command interface until all currently running processes are complete. The following considerations apply when using the \*WAI command:

- this command is intended to synchronize command execution with the chassis controller.
- If there are multiple commands in the command string, the \*WAI command must be the last entry.
- The \*WAI command does not complete until all currently running processes are complete, this has the effect of temporarily stopping the command interface. The command interface can be restarted if necessary using the low-level word-serial CLEAR command.
- the \*OPC? query performs an action similar to the \*WAI command but generates a response queue entry when complete.

**COMMAND PARAMETERS** None

**RESPONSE PARAMETERS** n/a

**DEFAULT VALUE** n/a

**EXAMPLES** \*WAI stop the command interface until all currently running processes are complete

**STATUS EVENTS** None

**AFFECTIVITY** The \*WAI command temporarily stops the command interface until all currently running processes are complete.

**RELATED COMMANDS** \*OPC?

---

## 1.3 High Level Word Serial Commands

---

The ZT432VXI supports various high-level commands that are specific to its operation. These commands and queries are sent to the ZT432VXI as ASCII strings using the Byte Available and Byte Request low-level VXIbus commands.

### 1.3.1 Abort (ABORt)

**COMMAND SYNTAX** ABORt

**QUERY SYNTAX** n/a

Terminates all ongoing processes and returns the unit to the idle state. Data resulting from the on-going processes may be corrupt.

**COMMAND PARAMETERS** n/a

**RESPONSE PARAMETERS** n/a

**DEFAULT VALUE** n/a

**EXAMPLES** ABOR aborts all processes

**STATUS EVENTS** None

**AFFECTIVITY** All on-going processes are aborted, including waveform capture, self-calibration, and self-test.

**RELATED COMMANDS** None

### 1.3.2 Arm (ARM?)

**COMMAND SYNTAX** n/a

**QUERY SYNTAX** ARM?

Queries the waveform acquisition Arm state. The unit arm state may also be queried using the :STATus:OPERation:CONDition? query.

**COMMAND PARAMETERS** n/a

<b>RESPONSE PARAMETERS</b>	<state>	current acquisition state	
		0	not initiated
		or initiated and waiting for arm	
		1	armed and ready for triggering

**DEFAULT VALUE** n/a

**EXAMPLES** ARM? query current acquisition state, example response: 0 (or not armed)

**STATUS EVENTS** None

**AFFECTIVITY** None

**RELATED COMMANDS** INITiate  
:STATus:OPERation:CONDition?



### 1.3.3 Arm Slope (ARM:SLOPe)

**COMMAND SYNTAX** ARM:SLOPe <slope>

**QUERY SYNTAX** ARM:SLOPe?

Sets or queries the arm signal active state. The following considerations apply when setting the arm slope:

- The active arm slope is detected after the arm source is selected from the available sources
- The front panel ARM IN signal is inverted before the source selector
- POSITIVE slope defines the active state as the selected source in or transitioning to its high state
- NEGATIVE slope defines the active state as the selected source in or transitioning to its low state

<b>COMMAND PARAMETERS</b>	<slope>	desired arm signal active state POSitive active high arm signal NEGative active low arm signal
<b>RESPONSE PARAMETERS</b>	<slope>	current active arm state POS active high arm signal NEG active low arm signal
<b>DEFAULT VALUE</b>	POS	
<b>EXAMPLES</b>	ARM:SLOP NEG	selects arm signal active state
	ARM:SLOP?	query arm signal active state, example response: POS (or active high)
<b>STATUS EVENTS</b>	-141	invalid slope name selected, command was ignored
<b>AFFECTIVITY</b>	None	
<b>RELATED COMMANDS</b>	ARM:SOURe	

### 1.3.4 Arm Source (ARM:SOURce)

**COMMAND SYNTAX**    ARM:SOURce <source>

**QUERY SYNTAX**        ARM:SOURce?

Sets or queries the arm signal source. The following considerations apply when setting the arm source:

- When a waveform acquisition has been initiated, the arm signal allows the unit to begin trigger detection.
- The front panel ARM IN signal can be selected using the ARM source setting, note that the ARM IN signal is inverted before input to the source selector.
- The VXIbus ECL or TTL signals can be selected using the ECLTrg or TTLTrg source settings, respectively.
- The waveform acquisition system ARM state may be bypassed and trigger detection begun immediately using the IMMEDIATE source setting.
- The arm signal active state can be set using the :ARM:SLOPe command.

<b>COMMAND PARAMETERS</b>	<source>	desired arm signal source  ARM        front panel ARM IN signal ECLTrg<n>    VXIbus ECL trigger line, where <n> may be 0 or 1 IMMEDIATE    bypass arm detection TTLTrg<n>    VXIbus TTL trigger line, where <n> may be 0, 1, 2, ... or 7
<b>RESPONSE PARAMETERS</b>	<source>	current arm signal source  ARM        front panel ARM IN signal ECLT<n>    VXIbus ECL trigger line, where <n> may be 0 or 1 IMM        arm detection bypassed TTLT<n>    VXIbus TTL trigger line, where <n> may be 0, 1, 2, ... or 7
<b>DEFAULT VALUE</b>	IMM	
<b>EXAMPLES</b>	ARM:SOUR IMM ARM:SOUR?	bypass arm signal detection query current arm signal source, example response: ARM (or ARM IN)
<b>STATUS EVENTS</b>	-141	invalid arm signal source selected, command was ignored
<b>AFFECTIVITY</b>	None	
<b>RELATED COMMANDS</b>	ARM:SLOPe	

# 1

## 1.3.5 Calculate Waveform Addition (CALCulate:ADD)

**COMMAND SYNTAX** CALCulate<n>:ADD <src1>,<src2>,<num1>,<num2>

**QUERY SYNTAX** n/a

Sets Calculate Channel <n> to add two waveforms and place the result in its output. The following considerations apply when using the ADD calculate function:

- <num1> and <num2> select the waveform numbers for <src1> and <src2>. This number is used when multiple waveforms are available for an input signal channel, refer to the TRIGger:COUNT command. These values must be set to 1 for other waveform sources.
- The source and waveform number selections are position sensitive.
- Refer to the CALCulate:FUNCTion? query to read the current function for Calculate Channel <n>.

<b>COMMAND PARAMETERS</b>	<n>	calculate channel number, where <n> may be 1 or 2
	<src1> and <src2>	desired waveform sources INPut<n> front panel input signals, where <n> may be 1, 2, 3, or 4 CALCulate<n> waveform math channel, where <n> may be 1 or 2 REFerence<n> reference waveform, where <n> may be 1, 2, 3,... or 6 defaults to INPut1 if omitted
	<num1> and <num2>	desired waveform sources INPut<n> front panel input signals, where <n> may be 1, 2, 3, or 4
<b>RESPONSE PARAMETERS</b>	n/a	
<b>DEFAULT VALUE</b>	n/a	
<b>EXAMPLES</b>	CALC:ADD INP1,REF1,3	add input signal 1, waveform 3, to reference channel 1
<b>STATUS EVENTS</b>	-141	invalid waveform source selected, command was ignored
	-222	out-of-range waveform number selected, the waveform number was coerced in range
	-241	invalid calculate channel selected, command was ignored
<b>AFFECTIVITY</b>		This command will over-ride any previous waveform math function and waveform sources
<b>RELATED COMMANDS</b>	CALCulate Subsystem TRIGger:COUNT	



### 1.3.7 Calculate Waveform Derivative (CALCulate:DERivative)

**COMMAND SYNTAX** CALCulate<n>:DERivative <src>,<num>

**QUERY SYNTAX** n/a

Sets Calculate Channel <n> to calculate the derivative of a source waveform and place the result into its output. The following considerations apply when using the DERivative calculate function:

- <num> selects the waveform number for <src>. This number is used when multiple waveforms are available for an input signal channel, refer to the TRIGger:COUNT command. <num> must be set to 1 for other waveform sources.
- The source and waveform number selections are position sensitive.
- Refer to the CALCulate:FUNCTION? query to read the current function for Calculate Channel <n>.

<b>COMMAND</b>	<n>	calculate channel number, where <n> may be 1 or 2
<b>PARAMETERS</b>	<src>	desired waveform source INPut<n> front panel input signals, where <n> may be 1, 2, 3, or 4 CALCulate<n> waveform math channel, where <n> may be 1 or 2 REFerence<n> reference waveform, where <n> may be 1, 2, 3,... or 6 defaults to INPut1 if omitted
	<num>	desired waveform numbers, defaults to 1 if omitted
<b>RESPONSE</b>	n/a	
<b>PARAMETERS</b>	n/a	
<b>DEFAULT VALUE</b>	n/a	
<b>EXAMPLES</b>	CALC:DER INP2 TRIGger:COUNT	calculate the derivative of input signal 2
<b>STATUS EVENTS</b>	-141 -222 -241	invalid waveform source selected, command was ignored out-of-range waveform number selected, the waveform number was coerced in range invalid calculate channel selected, command was ignored
<b>AFFECTIVITY</b>		This command will over-ride any previous waveform math function and waveform sources
<b>RELATED COMMANDS</b>	CALCulate Subsystem TRIGger:COUNT	

### 1.3.8 Calculate Waveform Function (CALCulate:FUNCtion?)

**COMMAND SYNTAX** n/a

**QUERY SYNTAX** CALCulate<n>:FUNCtion?

Queries the current Calculate Channel <n> function.

<b>COMMAND PARAMETERS</b>	<n>	calculate channel number, where <n> may be 1 or 2
<b>RESPONSE PARAMETERS</b>	<func>,<src1>,<src2>,<num1>,<num2>	function configuration
	<func>	abbreviated math function name, may be ADD, COPY, DER, INT, INV, MULT, or SUBT
	<src1>,<src2>	abbreviated waveform source names, may be INPn, CALCn, or REFn
	<num1>,<num2>	waveform numbers, may be 1 to 1024 for INPn channels or 1 for other sources
<b>DEFAULT VALUE</b>	n/a	
<b>EXAMPLES</b>	CALC1:FUNC?	query the current CALC1 function, example response: ADD,INP1,INP2,1,1
<b>STATUS EVENTS</b>	None	
<b>AFFECTIVITY</b>	None	
<b>RELATED COMMANDS</b>	CALCulate Subsystem	

# 1

---

## 1.3.9 Calculate Waveform IMMEDIATE (CALCulate:IMMEDIATE)

**COMMAND SYNTAX** CALCulate<n>:IMMEDIATE

**QUERY SYNTAX** n/a

Forces an immediate re-evaluation of Calculate Channel <n>.

**COMMAND PARAMETERS** <n> calculate channel number, where <n> may be 1 or 2

**RESPONSE PARAMETERS** n/a

**DEFAULT VALUE** n/a

**EXAMPLES** CALC1:IMM re-evaluate the current CALC1 function

**STATUS EVENTS** None

**AFFECTIVITY** None

**RELATED COMMANDS** CALCulate Subsystem

### 1.3.10 Calculate Waveform Integral (CALCulate:INTEgral)

**COMMAND SYNTAX** CALCulate<n>:INTEgral <src>,<num>

**QUERY SYNTAX** n/a

Sets Calculate Channel <n> to calculate the integral of a source waveform and place the result into its output. The following considerations apply when using the INTEgral calculate function:

- <num> selects the waveform number for <src>. This number is used when multiple waveforms are available for an input signal channel, refer to the TRIGger:COUNT command. <num> must be set to 1 for other waveform sources.
- The source and waveform number selections are position sensitive.
- Refer to the “CALCulate:FUNCtion?” query to read the current function for Calculate Channel <n>.

<b>COMMAND PARAMETERS</b>	<n>	calculate channel number, where <n> may be 1 or 2
	<src>	desired waveform source INPut<n> front panel input signals, where <n> may be 1, 2, 3, or 4 CALCulate<n> waveform math channel, where <n> may be 1 or 2 REFerence<n> reference waveform, where <n> may be 1, 2, 3,... or 6 defaults to INPut1 if omitted
	<num>	desired waveform numbers, defaults to 1 if omitted

**RESPONSE PARAMETERS** n/a

**DEFAULT VALUE** n/a

**EXAMPLES** CALC:INT REF5 calculate the integral of reference waveform 5

<b>STATUS EVENTS</b>	-141	invalid waveform source selected, command was ignored
	-222	out-of-range waveform number selected, the waveform number was coerced in range
	-241	invalid calculate channel selected, command was ignored

**AFFECTIVITY** This command will over-ride any previous waveform math function and waveform sources

**RELATED COMMANDS** CALCulate Subsystem  
TRIGger:COUNT

# 1

## 1.3.11 Calculate Waveform Invert (CALCulate:INVert)

**COMMAND SYNTAX** CALCulate<n>:INVert <src>,<num>

**QUERY SYNTAX** n/a

Sets Calculate Channel <n> to invert the sign of a source waveform and place the result into its output. The following considerations apply when using the INVert calculate function:

- <num> selects the waveform number for <src>. This number is used when multiple waveforms are available for an input signal channel, refer to the TRIGger:COUNT command. <num> must be set to 1 for other waveform sources.
- The source and waveform number selections are position sensitive, thus if <num> is present, then <src> must also be present.
- Refer to the CALCulate:FUNCTION? query to read the current function for Calculate Channel <n>.

<b>COMMAND PARAMETERS</b>	<n>	calculate channel number, where <n> may be 1 or 2
	<src>	desired waveform source INPut<n> front panel input signals, where <n> may be 1, 2, 3, or 4 CALCulate<n> waveform math channel, where <n> may be 1 or 2 REFerence<n> reference waveform, where <n> may be 1, 2, 3,... or 6 defaults to INPut1 if omitted
	<num>	desired waveform numbers, defaults to 1 if omitted
<b>RESPONSE PARAMETERS</b>	n/a	
<b>DEFAULT VALUE</b>	n/a	
<b>EXAMPLES</b>	CALC:INV INP2,7	calculate the inverse of input signal 2, waveform number 7
<b>STATUS EVENTS</b>	-141	invalid waveform source selected, command was ignored
	-222	out-of-range waveform number selected, the waveform number was coerced in range
	-241	invalid calculate channel selected, command was ignored
<b>AFFECTIVITY</b>	This command will over-ride any previous waveform math function and waveform sources	
<b>RELATED COMMANDS</b>	CALCulate Subsystem TRIGger:COUNT	



# 1

## 1.3.13 Calculate Waveform Offset (CALCulate:OFFSet)

**COMMAND SYNTAX** CALCulate<n>:OFFSet <offset>

**QUERY SYNTAX** CALCulate<n>:OFFSet?

Sets or queries the user defined offset for Calculate Channel <n>. When a Calculate channel is configured, the unit derives a range and offset for the channel output. The derived offset value can be overridden using this command. The derived range may be overridden using the CALCulate:RANGe command.

**COMMAND PARAMETERS** <n> calculate channel number, where <n> may be 1 or 2  
<offset> desired waveform source

**RESPONSE PARAMETERS** <offset> current calculate channel offset

**DEFAULT VALUE** 0.0

**EXAMPLES** CALC3:OFFS 0.1 V calculate the inverse of input signal 2, waveform number 7  
CALC:OFFS? query the Calculate Channel 1 offset, example response 1.0E-1

**STATUS EVENTS** None

**AFFECTIVITY** None

**RELATED COMMANDS** :CALCulate<n>:RANGe

### 1.3.14 Calculate Waveform Range (CALCulate:RANGe)

**COMMAND SYNTAX** CALCulate<n>:RANGe <range>

**QUERY SYNTAX** CALCulate<n>:RANGe?

Sets or queries the user defined peak-to-peak range for Calculate Channel <n>. When a Calculate channel is configured, the unit derives a range and offset for the channel output. The derived range value can be overridden using this command. The offset may be overridden using the CALCulate:OFFSet command.

**COMMAND PARAMETERS**

<n>	calculate channel number, where <n> may be 1 or 2
<range>	new range value

**RESPONSE PARAMETERS**

<range>	current calculate channel peak-to-peak range
---------	----------------------------------------------

**DEFAULT VALUE** 5.0

**EXAMPLES**

CALC:RANG 100 mV	calculate the inverse of input signal 2, waveform number 7
CALC:RANG?	query the Calculate Channel 1 range, example response 1.0E-1

**STATUS EVENTS** None

**AFFECTIVITY** None

**RELATED COMMANDS** None

# 1

## 1.3.15 Calculate State (CALCulate:STATe)

**COMMAND SYNTAX** CALCulate<n>:STATe <sw>

**QUERY SYNTAX** CALCulate<n>:STATe?

Sets or queries the Calculate Channel <n> processing enable. When this switch is set to OFF, the associated calculate channel is not evaluated.

<b>COMMAND PARAMETERS</b>	<n> <sw>	calculate channel number, where <n> may be 1 or 2 desired calculate channel processing enable state ON or 1 enable processing OFF or 0 disable processing
<b>RESPONSE PARAMETERS</b>	<sw>	current calculate channel processing enable state 1 processing enabled 0 processing disabled
<b>DEFAULT VALUE</b>	0	
<b>EXAMPLES</b>	CALC1:STAT OFF CALC:STAT?	disable Calculate Channel 1 processing query the Calculate Channel 1 state, example response 0 (or processing disabled)
<b>STATUS EVENTS</b>	-141	invalid state name selected, command was ignored
<b>AFFECTIVITY</b>	None	
<b>RELATED COMMANDS</b>	None	

### 1.3.16 Calculate Waveform Subtraction (CALCulate:SUBTract)

**COMMAND SYNTAX** CALCulate<n>:SUBTract <src1>,<src2>,<num1>,<num2>

**QUERY SYNTAX** n/a

Sets Calculate Channel <n> to subtract two waveforms and place the result in its output. The following considerations apply when using the subtract calculate function:

- the function evaluated is <src1> - <src2>.
- <num1> and <num2> select the waveform numbers for <src1> and <src2>. This number is used when multiple waveforms are available for an input signal channel, refer to the TRIGger:COUNT command. These values must be set to 1 for other waveform sources.
- The source and waveform number selections are position sensitive.
- Refer to the CALCulate:FUNCTION? query to read the current function for Calculate Channel <n>.

<b>COMMAND PARAMETERS</b>	<n>	calculate channel number, where <n> may be 1 or 2
	<src1> and <src2>	desired waveform sources INPut<n> front panel input signals, where <n> may be 1, 2, 3, or 4 CALCulate<n> waveform math channel, where <n> may be 1 or 2 REFerence<n> reference waveform, where <n> may be 1, 2, 3,... or 6 defaults to INPut1 if omitted
	<num1> and <num2>	desired waveform sources INPut<n> front panel input signals, where <n> may be 1, 2, 3, or 4

**RESPONSE PARAMETERS** n/a

**DEFAULT VALUE** n/a

**EXAMPLES** CALC:SUBT INP1,REF1,3 subtract reference channel 1 from input signal 1, waveform 3

**STATUS EVENTS**

-141	invalid waveform source selected, command was ignored
-222	out-of-range waveform number selected, the waveform number was coerced in range
-241	invalid calculate channel selected, command was ignored

**AFFECTIVITY** This command will over-ride any previous waveform math function and waveform sources.

**RELATED COMMANDS** CALCulate Subsystem  
TRIGger:COUNT

### 1.3.17 Calibration Report (CALibration:REPort?)

**COMMAND SYNTAX** n/a

**QUERY SYNTAX** CALibration:REPort?

Queries the results of the most recent calibration cycle. Refer to The Installation, Operation, Programming, & Diagnostics Manual, Appendix B for information on calibration report codes. A self-calibration cycle may be initiated using the IEEE-488 \*CAL? command.

**COMMAND** n/a

**PARAMETERS**

**RESPONSE** <code> calibration results report code  
**PARAMETERS** > 0 information codes  
0 successful calibration  
< 0 calibration failure

**DEFAULT VALUE** n/a

**EXAMPLES** CAL:REP? query the calibration results code, example response 0 (or successful calibration)

**STATUS EVENTS** None

**AFFECTIVITY** None

**RELATED** \*CAL?  
**COMMANDS**

### 1.3.18 Format Byte Order (FORMat:BORDER)

**COMMAND SYNTAX** FORMat:BORDER <order>

**QUERY SYNTAX** FORMat:BORDER?

Sets or queries the byte order used to transfer binary data through all external interfaces. This format is used by block data transfers such as waveform load operations. The following orders are defined:

- NORMAl, high byte then low order byte transfer order, also know as “Motorola Order”
- SWAPped, low byte then high order byte transfer order, also know as “Intel Order”

<b>COMMAND PARAMETERS</b>	<order>	desired byte order NORMAl high byte then low byte SWAPped low byte then high byte
<b>RESPONSE PARAMETERS</b>	<order>	current byte order NORM high byte then low byte SWAP low byte then high byte
<b>DEFAULT VALUE</b>	NORM	
<b>EXAMPLES</b>	FORM:BORD SWAP FORM:BORD?	transfer low byte then high byte query current byte order, example response: NORM
<b>STATUS EVENTS</b>	-141	invalid byte order name selected, command was ignored
<b>AFFECTIVITY</b>	None	
<b>RELATED COMMANDS</b>	None	

# 1

## 1.3.19 Initiate (INITiate?)

**COMMAND SYNTAX** n/a  
**QUERY SYNTAX** INITiate?

Queries the initiated state of the waveform acquisition system. The following considerations apply when reading the acquisition system state:

- The acquisition system state may also be read using the STATus:OPERation:CONDition command.
- The low level VXIbus Control/Status Register bit 4, (0010<sub>16</sub> or 256<sub>10</sub>), will be set during waveform acquisition and cleared upon acquisition completion.

**COMMAND PARAMETERS** n/a

**RESPONSE PARAMETERS** <state> waveform acquisition system state  
1 acquiring  
0 idle

**DEFAULT VALUE** n/a

**EXAMPLES** INIT? query waveform acquisition system state, example response  
1 (or acquiring waveforms)

**STATUS EVENTS** n/a

**AFFECTIVITY** None

**RELATED COMMANDS** STATus:OPERation:CONDition

### 1.3.20 Initiate Immediate (INITiate:IMMediate)

<b>COMMAND</b>	INITiate[:IMMediate]	full
<b>SYNTAX</b>	INITiate	minimum
<b>QUERY SYNTAX</b>	n/a	

Starts a waveform acquisition cycle. The following considerations apply when using the INITiate command:

- This command will cause the acquisition controller to leave its idle state and begin polling the selected arm signal for its active state. When an active arm signal is detected the unit will begin waveform capture sequencing, refer to the ARM subsystem commands for information on configuring arm detection.
- The unit will remain initiated until all requested trigger conditions are met, including input signal digitization, CALCulate channels execution, and automatic waveform download and all post-capture signal processing is complete.
- Upon completion of all acquisition activities, the controller will return to its idle state.
- Waveform acquisition may be aborted using the ABORt command.
- The state of the acquisition controller may be determined using the :STATus:OPERation:CONDition command which reports the overall acquisition system state, the arm detection status, and the trigger detection status.
- The state of the acquisition controller may also be determined using the INITiate? query which reports the overall acquisition system state only.
- The \*OPC or \*OPC? commands may be used to cause the unit to generate additional acquisition complete indicators in the \*ESR OPC bit or the command response buffer, respectively.
- Acquired waveforms may be automatically copied to VXIbus A32 memory after a capture cycle by using the TRACe:DOWNload:STATe command.
- The low level VXIbus Control/Status Register bit 4, (0010<sub>16</sub> or 256<sub>10</sub>), will be set during waveform acquisition and cleared upon acquisition completion.
- The low level VXIbus Control/Status Register bit 5, (0020<sub>16</sub> or 512<sub>10</sub>), will be cleared during waveform acquisition and set when enabled waveforms are available in VXIbus A32 memory.

<b>COMMAND PARAMETERS</b>	n/a	
<b>RESPONSE PARAMETERS</b>	n/a	
<b>DEFAULT VALUE</b>	n/a	
<b>EXAMPLES</b>	INIT	begin waveform acquisition
<b>STATUS EVENTS</b>	-213	INITiate command received before previous acquisition has completed, command was ignored
<b>AFFECTIVITY</b>	None	
<b>RELATED COMMANDS</b>	ARM Subsystem ABORt INITiate? *OPC, *OPC? :STATus:OPERation:CONDition TRACe:DOWNload	

# 1

## 1.3.21 Measure Edge (MEASure:EDGE)

**COMMAND SYNTAX** MEASure:EDGE <edge1>,<edge2>

**QUERY SYNTAX** MEASure:EDGE?

Sets or queries the waveform edge numbers used in edge related measurements, ex. delay. The following considerations apply when using the measurement edge command:

- <edge1> selects the first edge to be analyzed. <edge1> may be positive or negative, where positive numbers select rising edges and negative numbers select falling edges. <edge1> establishes the zero time for time measurements.
- <edge2> selects the second edge to be analyzed. <edge2> may be positive or negative, where positive numbers select rising edges and negative numbers select falling edges. Not all measurements use <edge2>.
- In some cases the edge polarity is ignored.
- If edges are specified in reverse order, negative times will be generated, ex. :MEASure:EDGE 2,1 will result in (-1 \* the result produced by :MEASure:EDGE 1,2).
- The source and waveform number selections are position sensitive.

<b>COMMAND PARAMETERS</b>	<edge1> and <edge2>	first and second edges to be analyzed, <edge1> defaults to 1, <edge2> defaults to 2 -N select the N <sup>th</sup> falling edge from the waveform record start 0 select the first edge of either polarity from the waveform record start +N select the N <sup>th</sup> rising edge from the waveform record start
<b>RESPONSE PARAMETERS</b>	<edge1>,<edge2>	selected analysis edges
<b>DEFAULT VALUE</b>	1, 2	
<b>EXAMPLES</b>	MEAS:EDGE 4,2 MEAS:EDGE?	analyze the 4 <sup>th</sup> rising edge then the 2 <sup>nd</sup> rising edge query the analysis edges, example response 4,2 (or analyze the 4 <sup>th</sup> rising edge and then the 2 <sup>nd</sup> rising edge)
<b>STATUS EVENTS</b>	None	
<b>AFFECTIVITY</b>	None	
<b>RELATED COMMANDS</b>	MEASure System	

### 1.3.22 Measure Reference Levels (MEASure:REFerence)

**COMMAND SYNTAX** MEASure:REFerence <lref>,<mref>,<href>

**QUERY SYNTAX** MEASure:REFerence?

Sets or queries the reference levels used in waveform voltage analysis as a % of full scale. The following considerations apply when using the MEASure:REFerence command:

- <lref> selects the threshold for detection of the input signal low state.
- <mref> selects the threshold for detection of the input signal middle level.
- <href> selects the threshold for detection of the input signal high state.
- The allowed reference values range from 0.0 to 100.0 and are measured in percent.
- The reference level selections are position sensitive and must be specified in ascending values.

<b>COMMAND PARAMETERS</b>	<lref>	signal analysis low level detection threshold, if not specified defaults to 10.0
	<mref>	signal analysis middle level detection threshold, if not specified defaults to 50.0
	<href>	signal analysis high level detection threshold, if not specified defaults to 90.0
<b>RESPONSE PARAMETERS</b>	<lref>,<mref>,<href>	selected signal analysis reference levels
<b>DEFAULT VALUE</b>	10.0, 50.0, 90.0	
<b>EXAMPLES</b>	MEAS:REF 25,50	select the reference levels as lref = 25, mref = 50, and href = 90 (where href is defaulted)
	MEAS:REF?	query the reference levels, example response 2.5E+1,5.0E+1,9.0E+1
<b>STATUS EVENTS</b>	-222	out-of-range reference level selected, reference level coerced in range
<b>AFFECTIVITY</b>	None	
<b>RELATED COMMANDS</b>	MEASure System	

### 1.3.23 Measure Voltage, AC (MEASure:VOLTage:AC?)

**COMMAND SYNTAX** n/a

**QUERY SYNTAX** MEASure:VOLTage:AC? <src>,<num>

Queries the RMS level of one cycle of the selected waveform source. The following considerations apply when using the measure AC query:

- The AC measurement is the “AC RMS” value of the selected waveform, defined as the RMS value of (waveform values – waveform average value) for one waveform cycle. Use the :MEASure:VOLTage:DC? query to determine the overall signal RMS level.
- <num> selects the waveform number from the waveform source to be analyzed. <num> is used to select a particular waveform in the case of multiple waveform capture, refer to the TRIGger:COUNT command. <num> must be 1 for all waveform sources except input signal channels.
- The query parameters are position sensitive.

<b>COMMAND PARAMETERS</b>	<src>	waveform source, if not specified defaults to INPut1 CALCulate<n> waveform math channel, where <n> may be 1 or 2 INPut<n> front panel input signals, where <n> may be 1, 2, 3, or 4 REFerence<n> reference waveform, where <n> may be 1, 2, 3,... or 6
	<num>	desired waveform number, if not specified defaults to 1
<b>RESPONSE PARAMETERS</b>	<ac>	the AC RMS value of the selected waveform
<b>DEFAULT VALUE</b>	n/a	
<b>EXAMPLES</b>	MEAS:VOLT:AC?	query the AC RMS waveform value, example response 2.5E+0
<b>STATUS EVENTS</b>	-141	invalid waveform source name specified, command was ignored
	-222	out-of-range waveform number selected, value was coerced in range
<b>AFFECTIVITY</b>	None	
<b>RELATED COMMANDS</b>	MEASure:VOLTage:DC? TRIGger:COUNT	

### 1.3.24 Measure Voltage, Amplitude (MEASure:VOLTage:AMPLitude?)

**COMMAND SYNTAX** n/a

**QUERY SYNTAX** MEASure:VOLTage:AMPLitude? <src>,<num>

Queries the amplitude of the selected waveform source. The following considerations apply when using the measure amplitude query:

- The amplitude measurement assumes a bi-level signal with distinct high and low levels and is defined as (waveform high level – waveform low level). Use the :MEASure:VOLTage:PTPeak? query to detect signal amplitude without assuming a bi-level signal.
- <num> selects the waveform number from the waveform source to be analyzed. <num> is used to select a particular waveform in the case of multiple waveform capture, see the TRIGger:COUNT command. <num> must be 1 for all waveform sources except input signal channels.
- The query parameters are position sensitive.

**COMMAND PARAMETERS**

<src>	waveform source, if not specified defaults to INPut1 CALCulate<n> waveform math channel, where <n> may be 1 or 2 INPut<n> front panel input signals, where <n> may be 1, 2, 3, or 4 REFerence<n> reference waveform, where <n> may be 1, 2, 3,... or 6
<num>	desired waveform number, if not specified defaults to 1

**RESPONSE PARAMETERS** <amplitude> the selected waveform amplitude

**DEFAULT VALUE** n/a

**EXAMPLES** MEAS:VOLT:AMPL? query the waveform amplitude, example response 5.0E-1

**STATUS EVENTS**

-141	invalid waveform source name specified, command was ignored
-222	out-of-range waveform number selected, value was coerced in range

**AFFECTIVITY** None

**RELATED COMMANDS** MEASure:VOLTage:PTPeak  
TRIGger:COUNT

### 1.3.25 Measure Voltage, Average (MEASure:VOLTage:AVERage?)

**COMMAND SYNTAX** n/a

**QUERY SYNTAX** MEASure:VOLTage:AVERage? <src>,<num>

Queries the average level of the selected waveform source. The following considerations apply when using the measure average query:

- The average is defined as the sum of all the sample values in the waveform record divided by the number of samples.
- <num> selects the waveform number from the waveform source to be analyzed. <num> is used to select a particular waveform in the case of multiple waveform capture, see the TRIGger:COUNT command. <num> must be 1 for all waveform sources except input signal channels.
- The query parameters are position sensitive.

<b>COMMAND PARAMETERS</b>	<src>	waveform source, if not specified defaults to INPut1 CALCulate<n> waveform math channel, where <n> may be 1 or 2 INPut<n> front panel input signals, where <n> may be 1, 2, 3, or 4 REFerence<n> reference waveform, where <n> may be 1, 2, 3,... or 6
	<num>	desired waveform number, if not specified defaults to 1
<b>RESPONSE PARAMETERS</b>	<avg>	the selected waveform average
<b>DEFAULT VALUE</b>	n/a	
<b>EXAMPLES</b>	MEAS:VOLT:AVER?	query the waveform average value, example response 5.0E-1
<b>STATUS EVENTS</b>	-141	invalid waveform source name specified, command was ignored
	-222	out-of-range waveform number selected, value was coerced in range
<b>AFFECTIVITY</b>	None	
<b>RELATED COMMANDS</b>	TRIGger:COUNT	

### 1.3.26 Measure Voltage, DC (MEASure:VOLTage:DC?)

**COMMAND SYNTAX** n/a

**QUERY SYNTAX** MEASure:VOLTage:DC? <src>,<num>

Queries the RMS level of one cycle of the selected waveform source. The following considerations apply when using the measure DC query:

- The DC measurement is the “DC RMS” value of the selected waveform and is defined as the overall RMS level of one waveform cycle. Use the :MEASure:VOLTage:AC? query to determine the RMS level of the selected waveform’s AC content only.
- <num> selects the waveform number from the waveform source to be analyzed. <num> is used to select a particular waveform in the case of multiple waveform capture, see the TRIGger:COUNT command. <num> must be 1 for all waveform sources except input signal channels.
- The query parameters are position sensitive.

<b>COMMAND PARAMETERS</b>	<src>	waveform source, if not specified defaults to INPut1 CALCulate<n> waveform math channel, where <n> may be 1 or 2 INPut<n> front panel input signals, where <n> may be 1, 2, 3, or 4 REFerence<n> reference waveform, where <n> may be 1, 2, 3,... or 6
	<num>	desired waveform number, if not specified defaults to 1
<b>RESPONSE PARAMETERS</b>	<dc>	the DC RMS value of the selected waveform
<b>DEFAULT VALUE</b>	n/a	
<b>EXAMPLES</b>	MEAS:VOLT:DC?	query the DC RMS waveform value, example response 2.5E+0
<b>STATUS EVENTS</b>	-141 -222	invalid waveform source name specified, command was ignored out-of-range waveform number selected, value was coerced in range
<b>AFFECTIVITY</b>	None	
<b>RELATED COMMANDS</b>	MEASure:VOLTage:AC? TRIGger:COUNT	

### 1.3.27 Measure Voltage, Delay (MEASure:VOLTage:DELay?)

**COMMAND SYNTAX** n/a

**QUERY SYNTAX** MEASure:VOLTage:DELay? <src1>,<src2>,<num1>,<num2>

Queries the time between selected edges in two waveforms. The following considerations apply when using the measure delay query:

- The delay measurement is defined as the time at <src1>,<num1>,<edge1> minus the time at <src2>,<num2>,<edge2>. See the :MEASure:EDGE command for information on setting <edge1> and <edge2>.
- <num1> and <num2> select the waveform number from the waveform source to be analyzed. <num1> and <num2> are used to select a particular waveform in the case of multiple waveform capture, see the TRIGger:COUNT command. <num1> and <num2> must be 1 for all waveform sources except input signal channels.
- <src1> and <src2> may refer to the same source.
- The query parameters are position sensitive.

<b>COMMAND PARAMETERS</b>	<src1> and <src2>	waveform sources, if not specified default to INPut1 CALCulate<n> waveform math channel, where n> may be 1 or 2 INPut<n> front panel input signals, where <n> may be 1, 2, 3, or 4 REFerence<n> reference waveform, where <n> may be 1, 2, 3,... or 6
	<num1> and <num2>	desired waveform numbers, if not specified defaults to 1
<b>RESPONSE PARAMETERS</b>	<delay>	the time between waveform <src1>,<num1>,<edge1> and waveform <src2>,<num2>,<edge2>
<b>DEFAULT VALUE</b>	n/a	
<b>EXAMPLES</b>	MEAS:VOLT:DEL? INP1,INP2,2	query the time between waveform INPut1, number 2, rising edge 1 and waveform INPut2, number 1, falling edge 2, (where INPut2 number 1 is defaulted). This assumes that the :MEASure:EDGE command was used to set <edge1> = 1 and <edge2> = -2, example response 2.5000E+0
<b>STATUS EVENTS</b>	-141	invalid waveform source name specified, command was ignored
	-222	out-of-range waveform number selected, value was coerced in range
<b>AFFECTIVITY</b>	None	
<b>RELATED COMMANDS</b>	MEASure:EDGE TRIGger:COUNT	

### 1.3.28 Measure Voltage, Frequency (MEASure:VOLTage:FREQuency?)

**COMMAND SYNTAX** n/a

**QUERY SYNTAX** MEASure:VOLTage:FREQuency? <src>,<num>

Queries the frequency of the selected waveform source. The following considerations apply when using the measure frequency query:

- In order to ensure reliable frequency measurements, the waveform record must contain two full cycles of the input signal and the number of points per cycle must be as large as practical. Frequency measurement accuracy is directly proportional to the number of samples per cycle.
- <num> selects the waveform number from the waveform source to be analyzed. <num> is used to select a particular waveform in the case of multiple waveform capture, see the TRIGger:COUNT command. <num> must be 1 for all waveform sources except input signal channels.
- The query parameters are position sensitive.

<b>COMMAND PARAMETERS</b>	<src>	waveform source, if not specified defaults to INPut1 CALCulate<n> waveform math channel, where <n> may be 1 or 2 INPut<n> front panel input signals, where <n> may be 1, 2, 3, or 4 REFerence<n> reference waveform, where <n> may be 1, 2, 3,... or 6
	<num>	desired waveform number, if not specified defaults to 1
<b>RESPONSE PARAMETERS</b>	<freq>	the selected waveform frequency
<b>DEFAULT VALUE</b>	n/a	
<b>EXAMPLES</b>	MEAS:VOLT:FREQ?	query the frequency of waveform INPut1, number 1, where the waveform source and waveform number are defaulted, example response 5.0000E+4
<b>STATUS EVENTS</b>	-141 -222	invalid waveform source name specified, command was ignored out-of-range waveform number selected, value was coerced in range
<b>AFFECTIVITY</b>	None	
<b>RELATED COMMANDS</b>	TRIGger:COUNT	

### 1.3.29 Measure Voltage, Fall Time (MEASure:VOLTage:FTIME?)

**COMMAND SYNTAX** n/a

**QUERY SYNTAX** MEASure:VOLTage:FTIME? <src>,<num>

Queries the fall time of a selected waveform edge. The following considerations apply when using the measure fall time query:

- The fall time measurement is defined as the time between the waveform signal level crossing <href> and then crossing <lref> on <edge1>. See the :MEASure:EDGE command for information on setting the edge to analyze, <edge1>. See the :MEASure:REFerence command for setting the signal reference levels, <lref> and <href>.
- Falling edges are assumed and the polarity of <edge1> is ignored.
- <num> selects the waveform number from the waveform source to be analyzed. <num> is used to select a particular waveform in the case of multiple waveform capture, see the TRIGger:COUNT command. <num> must be 1 for all waveform sources except input signal channels.
- The query parameters are position sensitive.

<b>COMMAND PARAMETERS</b>	<src>	waveform source, if not specified defaults to INPut1 CALCulate<n> waveform math channel, where <n> may be 1 or 2 INPut<n> front panel input signals, where <n> may be 1, 2, 3, or 4 REFerence<n> reference waveform, where <n> may be 1, 2, 3,... or 6
	<num>	desired waveform numbers, if not specified defaults to 1
<b>RESPONSE PARAMETERS</b>	<time>	the time between the waveform level crossing <href> and then <lref> on <edge1>
<b>DEFAULT VALUE</b>	n/a	
<b>EXAMPLES</b>	MEAS:VOLT:FTIM? INP4,3	query fall time of waveform INPut4, waveform number 3, example response 1.3286E-6
<b>STATUS EVENTS</b>	-141	invalid waveform source name specified, command was ignored
	-222	out-of-range waveform number selected, value was coerced in range
<b>AFFECTIVITY</b>	None	
<b>RELATED COMMANDS</b>	MEASure:EDGE MEASure:REFerence TRIGger:COUNT	

### 1.3.30 Measure Voltage, High Level (MEASure:VOLTage:HIGH?)

**COMMAND SYNTAX** n/a

**QUERY SYNTAX** MEASure:VOLTage:HIGH? <src>,<num>

Queries the high level of the selected waveform source. The following considerations apply when using the measure high query:

- The high level measurement assumes a bi-level signal with distinct high and low levels and is defined as the waveform upper level. Use the :MEASure:VOLTage:MAXimum? query to detect the most positive waveform voltage level without assuming a bi-level signal.
- <num> selects the waveform number from the waveform source to be analyzed. <num> is used to select a particular waveform in the case of multiple waveform capture, see the TRIGger:COUNT command. <num> must be 1 for all waveform sources except input signal channels.
- The query parameters are position sensitive.

<b>COMMAND PARAMETERS</b>	<src>	waveform source, if not specified defaults to INPut1 CALCulate<n> waveform math channel, where <n> may be 1 or 2 INPut<n> front panel input signals, where <n> may be 1, 2, 3, or 4 REFerence<n> reference waveform, where <n> may be 1, 2, 3,... or 6
	<num>	desired waveform number, if not specified defaults to 1
<b>RESPONSE PARAMETERS</b>	<high>	the selected waveform high level
<b>DEFAULT VALUE</b>	n/a	
<b>EXAMPLES</b>	MEAS:VOLT:HIGH? CALC2	query the Calculate2 waveform high level, example response 5.1876E-1
<b>STATUS EVENTS</b>	-141 -222	invalid waveform source name specified out-of-range waveform number selected
<b>AFFECTIVITY</b>	None	
<b>RELATED COMMANDS</b>	MEASure:VOLTage:MAXimum TRIGger:COUNT	

### 1.3.31 Measure Voltage, Low Level (MEASure:VOLTage:LOW?)

**COMMAND SYNTAX** n/a

**QUERY SYNTAX** MEASure:VOLTage:LOW? <src>,<num>

Queries the low level of the selected waveform source. The following considerations apply when using the measure low query:

- The low level measurement assumes a bi-level signal with distinct high and low levels and is defined as the waveform lower level. Use the :MEASure:VOLTage:MINimum? query to detect the most negative waveform voltage level without assuming a bi-level signal.
- <num> selects the waveform number from the waveform source to be analyzed. <num> is used to select a particular waveform in the case of multiple waveform capture, see the TRIGger:COUNT command. <num> must be 1 for all waveform sources except input signal channels.
- The query parameters are position sensitive.

<b>COMMAND PARAMETERS</b>	<src>	waveform source, if not specified defaults to INPut1 CALCulate<n> waveform math channel, where <n> may be 1 or 2 INPut<n> front panel input signals, where <n> may be 1, 2, 3, or 4 REFerence<n> reference waveform, where <n> may be 1, 2, 3,... or 6
	<num>	desired waveform number, if not specified defaults to 1
<b>RESPONSE PARAMETERS</b>	<low>	the selected waveform low level
<b>DEFAULT VALUE</b>	n/a	
<b>EXAMPLES</b>	MEAS:VOLT:LOW? INP2	query the waveform INPut2, waveform number 1, high level, where the waveform number was defaulted, example response -5.1876E-1
<b>STATUS EVENTS</b>	-141 -222	invalid waveform source name specified, command was ignored out-of-range waveform number selected, value was coerced in range
<b>AFFECTIVITY</b>	None	
<b>RELATED COMMANDS</b>	MEASure:VOLTage:MINimum TRIGger:COUNT	

### 1.3.32 Measure Voltage, Maximum Level (MEASure:VOLTage:MAXimum?)

**COMMAND SYNTAX** n/a

**QUERY SYNTAX** MEASure:VOLTage:MAXimum? <src>,<num>

Queries the most positive voltage of the selected waveform source. The following considerations apply when using the measure maximum query:

- The maximum level measurement is defined as the waveform most positive voltage level. Use the :MEASure:VOLTage:HIGH? query to detect the upper level in a bi-level signal such as a square wave.
- <num> selects the waveform number from the waveform source to be analyzed. <num> is used to select a particular waveform in the case of multiple waveform capture, see the TRIGger:COUNT command. <num> must be 1 for all waveform sources except input signal channels.
- The query parameters are position sensitive.

<b>COMMAND PARAMETERS</b>	<src>	waveform source, if not specified defaults to INPut1 CALCulate<n> waveform math channel, where <n> may be 1 or 2 INPut<n> front panel input signals, where <n> may be 1, 2, 3, or 4 REFerence<n> reference waveform, where <n> may be 1, 2, 3,... or 6
	<num>	desired waveform number, if not specified defaults to 1
<b>RESPONSE PARAMETERS</b>	<max>	the selected waveform most positive voltage
<b>DEFAULT VALUE</b>	n/a	
<b>EXAMPLES</b>	MEAS:VOLT:MAX? CALC1	query the Calculate1 waveform maximum voltage level, example response 5.1876E-1
<b>STATUS EVENTS</b>	-141 -222	invalid waveform source name specified, command was ignored out-of-range waveform number selected, value was coerced in range
<b>AFFECTIVITY</b>	None	
<b>RELATED COMMANDS</b>	MEASure:VOLTage:HIGH TRIGger:COUNT	

# 1

## 1.3.33 Measure Voltage, Minimum Level (MEASure:VOLTage:MINimum?)

**COMMAND SYNTAX** n/a

**QUERY SYNTAX** MEASure:VOLTage:MINimum? <src>,<num>

Queries the most negative voltage of the selected waveform source. The following considerations apply when using the measure minimum query:

- The minimum level measurement is defined as the waveform most negative voltage level. Use the :MEASure:VOLTage:LOW? query to detect the lower level in a bi-level signal such as a square wave.
- <num> selects the waveform number from the waveform source to be analyzed. <num> is used to select a particular waveform in the case of multiple waveform capture, see the TRIGger:COUNT command. <num> must be 1 for all waveform sources except input signal channels.
- The query parameters are position sensitive.

<b>COMMAND PARAMETERS</b>	<src>	waveform source, if not specified defaults to INPut1 CALCulate<n> waveform math channel, where <n> may be 1 or 2 INPut<n> front panel input signals, where <n> may be 1, 2, 3, or 4 REFerence<n> reference waveform, where <n> may be 1, 2, 3,... or 6
	<num>	desired waveform number, if not specified defaults to 1
	<b>RESPONSE PARAMETERS</b>	<min> the selected waveform most negative voltage
	<b>DEFAULT VALUE</b>	n/a
<b>EXAMPLES</b>	MEAS:VOLT:MIN? INP2,5	query the waveform INPut2, waveform number 5, minimum level, example response -5.1876E-1
<b>STATUS EVENTS</b>	-141	invalid waveform source name specified, command was ignored
	-222	out-of-range waveform number selected, value was coerced in range
<b>AFFECTIVITY</b>	None	
<b>RELATED COMMANDS</b>	MEASure:VOLTage:LOW	
	TRIGger:COUNT	

### 1.3.34 Measure Voltage, Negative Duty Cycle (MEASure:VOLTage:NDUTy?)

**COMMAND SYNTAX** n/a

**QUERY SYNTAX** MEASure:VOLTage:NDUTy? <src>,<num>

Queries the percent of a cycle the selected waveform is below the middle reference. The following considerations apply when using the measure negative duty query:

- The negative duty cycle is defined as the percent of a cycle that the selected waveform is below the signal <mref>. Use the :MEASure:REFerence command to set the <mref> value.
- <num> selects the waveform number from the waveform source to be analyzed. <num> is used to select a particular waveform in the case of multiple waveform capture, see the TRIGger:COUNt command. <num> must be 1 for all waveform sources except input signal channels.
- The query parameters are position sensitive.

<b>COMMAND PARAMETERS</b>	<src>	waveform source, if not specified defaults to INPut1 CALCulate<n> waveform math channel, where <n> may be 1 or 2 INPut<n> front panel input signals, where <n> may be 1, 2, 3, or 4 REFerence<n> reference waveform, where <n> may be 1, 2, 3,... or 6
	<num>	desired waveform number, if not specified defaults to 1
<b>RESPONSE PARAMETERS</b>	<nduty>	the selected waveform negative duty cycle
<b>DEFAULT VALUE</b>	n/a	
<b>EXAMPLES</b>	MEAS:VOLT:NDUT? INP1	query the negative duty cycle of waveform INPut1, waveform number 1, where the waveform number is defaulted, example response 4.9000E+1
<b>STATUS EVENTS</b>	-141 -222	invalid waveform source name specified, command was ignored out-of-range waveform number selected, value was coerced in range
<b>AFFECTIVITY</b>	None	
<b>RELATED COMMANDS</b>	MEASure:REFerence TRIGger:COUNt	

### 1.3.35 Measure Voltage, Negative Width (MEASure:VOLTage:NWIDth?)

**COMMAND SYNTAX** n/a

**QUERY SYNTAX** MEASure:VOLTage:NWIDth? <src>,<num>

Queries the time that the selected waveform is below the middle reference. The following considerations apply when using the measure negative width query:

- The negative width is defined as the time that the selected waveform is below the signal <mref>. Use the :MEASure:REFerence command to set the <mref> value.
- <num> selects the waveform number from the waveform source to be analyzed. <num> is used to select a particular waveform in the case of multiple waveform capture, see the TRIGger:COUNT command. <num> must be 1 for all waveform sources except input signal channels.
- The query parameters are position sensitive.

<b>COMMAND PARAMETERS</b>	<src>	waveform source, if not specified defaults to INPut1 CALCulate<n> waveform math channel, where <n> may be 1 or 2 INPut<n> front panel input signals, where <n> may be 1, 2, 3, or 4 REFerence<n> reference waveform, where <n> may be 1, 2, 3,... or 6
---------------------------	-------	-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

<num>	desired waveform number, if not specified defaults to 1
-------	---------------------------------------------------------

<b>RESPONSE PARAMETERS</b>	<nwid>	the selected waveform negative width
----------------------------	--------	--------------------------------------

<b>DEFAULT VALUE</b>	n/a
----------------------	-----

<b>EXAMPLES</b>	MEAS:VOLT:NWID? INP3	query the negative width of waveform INPut3, waveform number 1, where the waveform number is defaulted, example response 1.2500E-5
-----------------	----------------------	------------------------------------------------------------------------------------------------------------------------------------

<b>STATUS EVENTS</b>	-141	invalid waveform source name specified, command was ignored
	-222	out-of-range waveform number selected, value was coerced in range

<b>AFFECTIVITY</b>	None
--------------------	------

<b>RELATED COMMANDS</b>	MEASure:REFerence
	TRIGger:COUNT

### 1.3.36 Measure Voltage, Overshoot (MEASure:VOLTage:OVERshoot?)

**COMMAND SYNTAX** n/a

**QUERY SYNTAX** MEASure:VOLTage:OVERshoot? <src>,<num>

Queries the voltage overshoot of the selected waveform. The following considerations apply when using the measure overshoot query:

- Overshoot is defined as the amount of voltage past the high level of a bi-level signal that a signal travels as it transitions from its low state to its high state.
- Overshoot is calculated as the signal (maximum voltage – high level voltage).
- <num> selects the waveform number from the waveform source to be analyzed. <num> is used to select a particular waveform in the case of multiple waveform capture, see the TRIGger:COUNT command. <num> must be 1 for all waveform sources except input signal channels.
- The query parameters are position sensitive.

<b>COMMAND PARAMETERS</b>	<src>	waveform source, if not specified defaults to INPut1 CALCulate<n> waveform math channel, where <n> may be 1 or 2 INPut<n> front panel input signals, where <n> may be 1, 2, 3, or 4 REFerence<n> reference waveform, where <n> may be 1, 2, 3,... or 6
	<num>	desired waveform number, if not specified defaults to 1
<b>RESPONSE PARAMETERS</b>	<over>	the selected waveform overshoot
<b>DEFAULT VALUE</b>	n/a	
<b>EXAMPLES</b>	MEAS:VOLT:OVER? CALC2	query the overshoot of the Calculate2 output waveform, example response 3.6712E-3
<b>STATUS EVENTS</b>	-141	invalid waveform source name specified, command was ignored
	-222	out-of-range waveform number selected, value was coerced in range
<b>AFFECTIVITY</b>	None	
<b>RELATED COMMANDS</b>	TRIGger:COUNT	

### 1.3.37 Measure Voltage, Positive Duty Cycle (MEASure:VOLTage:PDUTy?)

**COMMAND SYNTAX** n/a

**QUERY SYNTAX** MEASure:VOLTage:PDUTy? <src>,<num>

Queries the percent of a cycle the selected waveform is above the middle reference. The following considerations apply when using the measure positive duty query:

- The positive duty cycle is defined as the percent of a cycle the selected waveform is above the signal <mref>. Use the :MEASure:REFerence command to set the <mref> value.
- <num> selects the waveform number from the waveform source to be analyzed. <num> is used to select a particular waveform in the case of multiple waveform capture, see the TRIGger:COUNT command. <num> must be 1 for all waveform sources except input signal channels.
- The query parameters are position sensitive.

<b>COMMAND PARAMETERS</b>	<src>	waveform source, if not specified defaults to INPut1 CALCulate<n> waveform math channel, where <n> may be 1 or 2 INPut<n> front panel input signals, where <n> may be 1, 2, 3, or 4 REFerence<n> reference waveform, where <n> may be 1, 2, 3,... or 6
	<num>	desired waveform number, if not specified defaults to 1
<b>RESPONSE PARAMETERS</b>	<pduTy>	the selected waveform positive duty cycle
<b>DEFAULT VALUE</b>	n/a	
<b>EXAMPLES</b>	MEAS:VOLT:PDUT? INP1	query the negative duty cycle of waveform INPut1, waveform number 1, where the waveform number is defaulted, example response 5.1000E+1
<b>STATUS EVENTS</b>	-141	invalid waveform source name specified, command was ignored
	-222	out-of-range waveform number selected, value was coerced in range
<b>AFFECTIVITY</b>	None	
<b>RELATED COMMANDS</b>	MEASure:REFerence TRIGger:COUNT	

### 1.3.38 Measure Voltage, Period (MEASure:VOLTage:PERiod?)

**COMMAND SYNTAX** n/a

**QUERY SYNTAX** MEASure:VOLTage:PERiod? <src>,<num>

Queries the period of the selected waveform source. The following considerations apply when using the measure period query:

- In order to ensure reliable period measurements, the waveform record must contain two full cycles of the input signal and the number of points per cycle must be as large as practical. Period measurement resolution is directly proportional to the number of samples per cycle.
- <num> selects the waveform number from the waveform source to be analyzed. <num> is used to select a particular waveform in the case of multiple waveform capture, see the TRIGger:COUNT command. <num> must be 1 for all waveform sources except input signal channels.
- The query parameters are position sensitive.

<b>COMMAND PARAMETERS</b>	<src>	waveform source, if not specified defaults to INPut1 CALCulate<n> waveform math channel, where <n> may be 1 or 2 INPut<n> front panel input signals, where <n> may be 1, 2, 3, or 4 REFerence<n> reference waveform, where <n> may be 1, 2, 3,... or 6
	<num>	desired waveform number, if not specified defaults to 1
<b>RESPONSE PARAMETERS</b>	<period>	the selected waveform period
<b>DEFAULT VALUE</b>	n/a	
<b>EXAMPLES</b>	MEAS:VOLT:PER?	query the period of waveform INPut1, number 1, where the waveform source and waveform number are defaulted, example response 2.4010E-5
<b>STATUS EVENTS</b>	-141 -222	invalid waveform source name specified out-of-range waveform number selected
<b>AFFECTIVITY</b>	None	
<b>RELATED COMMANDS</b>	Related Commands TRIGger:COUNT	

### 1.3.39 Measure Voltage, Preshoot (MEASure:VOLTage:PREShoot?)

**COMMAND SYNTAX** n/a

**QUERY SYNTAX** MEASure:VOLTage:PREShoot? <src>,<num>

Queries the voltage preshoot of the selected waveform. The following considerations apply when using the measure preshoot query:

- Preshoot is defined as the amount of voltage past the low level of a bi-level signal that a signal travels as it transitions from its high state to its low state.
- Overshoot is calculated as the signal (minimum voltage – low level voltage).
- <num> selects the waveform number from the waveform source to be analyzed. <num> is used to select a particular waveform in the case of multiple waveform capture, see the TRIGger:COUNT command. <num> must be 1 for all waveform sources except input signal channels.
- The query parameters are position sensitive.

<b>COMMAND PARAMETERS</b>	<src>	waveform source, if not specified defaults to INPut1 CALCulate<n> waveform math channel, where <n> may be 1 or 2 INPut<n> front panel input signals, where <n> may be 1, 2, 3, or 4 REFerence<n> reference waveform, where <n> may be 1, 2, 3,... or 6
	<num>	desired waveform number, if not specified defaults to 1
	<b>RESPONSE PARAMETERS</b>	<pres> the selected waveform preshoot
	<b>DEFAULT VALUE</b>	n/a
<b>EXAMPLES</b>	MEAS:VOLT:PRES? INP4	query the preshoot of the INPut4 output waveform, example response 3.6712E-3
<b>STATUS EVENTS</b>	-141	invalid waveform source name specified, command was ignored
	-222	out-of-range waveform number selected, value was coerced in range
<b>AFFECTIVITY</b>	None	
<b>RELATED COMMANDS</b>	TRIGger:COUNT	

### 1.3.40 Measure Voltage, Peak-to-Peak (MEASure:VOLTage:PTPeak?)

**COMMAND SYNTAX** n/a

**QUERY SYNTAX** MEASure:VOLTage:PTPeak? <src>,<num>

Queries the peak-to-peak amplitude of the selected waveform source. The following considerations apply when using the measure peak-to-peak query:

- The peak-to-peak measurement is defined as (waveform maximum level – waveform minimum level). Use the :MEASure:VOLTage:AMPLitude? query for less noise sensitive amplitude detection of bi-level signals, such as square waves.
- <num> selects the waveform number from the waveform source to be analyzed. <num> is used to select a particular waveform in the case of multiple waveform capture, see the TRIGger:COUNT command. <num> must be 1 for all waveform sources except input signal channels.
- The query parameters are position sensitive

**COMMAND PARAMETERS**

<src>	waveform source, if not specified defaults to INPut1
CALCulate<n>	waveform math channel, where <n> may be 1 or 2
INPut<n>	front panel input signals, where <n> may be 1, 2, 3, or 4
REFerence<n>	reference waveform, where <n> may be 1, 2, 3,... or 6
<num>	desired waveform number, if not specified defaults to 1

**RESPONSE PARAMETERS** <ptp> the selected waveform peak-to-peak amplitude

**DEFAULT VALUE** n/a

**EXAMPLES** MEAS:VOLT:PTP? INP2 query the waveform peak-to-peak amplitude of waveform INPut2, number 1, where the waveform number is defaulted, example response 5.0000E-1

**STATUS EVENTS**

-141	invalid waveform source name specified, command was ignored
-222	out-of-range waveform number selected, value was coerced in range

**AFFECTIVITY** None

**RELATED COMMANDS** MEASure:VOLTage:AMPLitude  
TRIGger:COUNT

# 1

## 1.3.41 Measure Voltage, Positive Width (MEASure:VOLTage:PWIDth?)

**COMMAND SYNTAX** n/a

**QUERY SYNTAX** MEASure:VOLTage:PWIDth? <src>,<num>

Queries the time that the selected waveform is above the middle reference. The following considerations apply when using the measure positive width query:

- The positive width is defined as the time that the selected waveform is above the signal <mref>. Use the :MEASure:REFerence command to set the <mref> value.
- <num> selects the waveform number from the waveform source to be analyzed. <num> is used to select a particular waveform in the case of multiple waveform capture, see the TRIGger:COUNT command. <num> must be 1 for all waveform sources except input signal channels.
- The query parameters are position sensitive.

<b>COMMAND PARAMETERS</b>	<src>	waveform source, if not specified defaults to INPut1 CALCulate<n> waveform math channel, where <n> may be 1 or 2 INPut<n> front panel input signals, where <n> may be 1, 2, 3, or 4 REFerence<n> reference waveform, where <n> may be 1, 2, 3,... or 6
	<num>	desired waveform number, if not specified defaults to 1
<b>RESPONSE PARAMETERS</b>	<pwid>	the selected waveform positive width
<b>DEFAULT VALUE</b>	n/a	
<b>EXAMPLES</b>	MEAS:VOLT:PWID? INP3	query the negative width of waveform INPut3, waveform number 1, where the waveform number is defaulted, example response 3.6500E-5
<b>STATUS EVENTS</b>	-141	invalid waveform source name specified, command was ignored
	-222	out-of-range waveform number selected, value was coerced in range
<b>AFFECTIVITY</b>	None	
<b>RELATED COMMANDS</b>	MEASure:REFerence TRIGger:COUNT	

### 1.3.42 Measure Voltage, Rise Time (MEASure:VOLTage:RTIME?)

**COMMAND SYNTAX** n/a

**QUERY SYNTAX** MEASure:VOLTage:RTIME? <src>,<num>

Queries the rise time of a selected waveform edge. The following considerations apply when using the measure rise time query:

- The rise time measurement is defined as the time between the waveform signal level crossing <lref> and then crossing <href> on <edge1>. See the :MEASure:EDGE command for information on setting the edge to analyze, <edge1>. See the :MEASure:REFerence command for setting the signal reference levels, <lref> and <href>.
- Rising edges are assumed and the polarity of <edge1> is ignored.
- <num> selects the waveform number from the waveform source to be analyzed. <num> is used to select a particular waveform in the case of multiple waveform capture, see the TRIGger:COUNT command. <num> must be 1 for all waveform sources except input signal channels.
- The query parameters are position sensitive.

<b>COMMAND PARAMETERS</b>	<src>	waveform source, if not specified defaults to INPut1 CALCulate<n> waveform math channel, where <n> may be 1 or 2 INPut<n> front panel input signals, where <n> may be 1, 2, 3, or 4 REFerence<n> reference waveform, where <n> may be 1, 2, 3,... or 6
	<num>	desired waveform numbers, if not specified defaults to 1
<b>RESPONSE PARAMETERS</b>	<time>	the time between the waveform level crossing <lref> and then <href> on <edge>
<b>DEFAULT VALUE</b>	n/a	
<b>EXAMPLES</b>	MEAS:VOLT:RTIM? INP1,2	query rise time of waveform INPut1, waveform number 2, example response 1.3286E-6
<b>STATUS EVENTS</b>	-141 -222	invalid waveform source name specified, command was ignored out-of-range waveform number selected, value was coerced in range
<b>AFFECTIVITY</b>	None	
<b>RELATED COMMANDS</b>	MEASure:EDGE MEASure:REFerence TRIGger:COUNT	

### 1.3.43 Measure Voltage, Crossing Time (MEASure:VOLTage:TCross?)

**COMMAND SYNTAX** n/a

**QUERY SYNTAX** MEASure:VOLTage:TCross? <src>,<num>

Queries the crossing rise time of a selected waveform edge. The following considerations apply when using the measure time crossing query:

- The crossing time measurement is defined as the time that the waveform signal <edge1> crosses <mref>. See the :MEASure:EDGE command for information on setting the edge to analyze, <edge1>. See the :MEASure:REFerence command for setting the signal reference level, <mref>.
- Rising edges are assumed and the polarity of <edge1> is ignored.
- <num> selects the waveform number from the waveform source to be analyzed. <num> is used to select a particular waveform in the case of multiple waveform capture, see the TRIGger:COUNT command. <num> must be 1 for all waveform sources except input signal channels.
- The query parameters are position sensitive.

<b>COMMAND PARAMETERS</b>	<src>	waveform source, if not specified defaults to INPut1 CALCulate<n> waveform math channel, where <n> may be 1 or 2 INPut<n> front panel input signals, where <n> may be 1, 2, 3, or 4 REFerence<n> reference waveform, where <n> may be 1, 2, 3,... or 6
	<num>	desired waveform numbers, if not specified defaults to 1

<b>RESPONSE PARAMETERS</b>	<time>	the <mref >crossing time of waveform <edge1>
----------------------------	--------	----------------------------------------------

**DEFAULT VALUE** n/a

<b>EXAMPLES</b>	MEAS:VOLT:TCR? INP1,2	query time that INPut1, waveform number 2, crosses <mref> for the second time, this assumes that MEASure:EDGE was used to set <edge1> to a 2, example response 1.3286E-6
-----------------	-----------------------	--------------------------------------------------------------------------------------------------------------------------------------------------------------------------

<b>STATUS EVENTS</b>	-141	invalid waveform source name specified, command was ignored
	-222	out-of-range waveform number selected, value was coerced in range

**AFFECTIVITY** None

**RELATED COMMANDS**

MEASure:EDGE  
MEASure:REFerence  
TRIGger:COUNT

### 1.3.44 Measure Voltage, Time at Maximum (MEASure:VOLTage:T MAXimum?)

**COMMAND SYNTAX** n/a

**QUERY SYNTAX** MEASure:VOLTage:TMAXimum? <src>,<num>

Queries the time at which the selected waveform reaches its maximum value. The following considerations apply when using the measure time at maximum query:

- The time at maximum measurement is defined as the time that the waveform signal reaches its maximum value.
- <num> selects the waveform number from the waveform source to be analyzed. <num> is used to select a particular waveform in the case of multiple waveform capture, see the TRIGger:COUNT command. <num> must be 1 for all waveform sources except input signal channels.
- The query parameters are position sensitive.

<b>COMMAND PARAMETERS</b>	<src>	waveform source, if not specified defaults to INPut1 CALCulate<n> waveform math channel, where <n> may be 1 or 2 INPut<n> front panel input signals, where <n> may be 1, 2, 3, or 4 REFerence<n> reference waveform, where <n> may be 1, 2, 3,... or 6
	<num>	desired waveform number, if not specified defaults to 1
<b>RESPONSE PARAMETERS</b>	<time>	the time when the waveform reaches its maximum value
<b>DEFAULT VALUE</b>	n/a	
<b>EXAMPLES</b>	MEAS:VOLT:TMAX? INP1,2	query time that INPut1, waveform number 2, reaches its maximum value, example response 1.3286E-6
<b>STATUS EVENTS</b>	-141	invalid waveform source name specified, command was ignored
	-222	out-of-range waveform number selected, value was coerced in range
<b>AFFECTIVITY</b>	None	
<b>RELATED COMMANDS</b>	TRIGger:COUNT	

# 1

## 1.3.45 Measure Voltage, Time at Minimum (MEASure:VOLTage:TMINimum?)

**COMMAND SYNTAX** n/a

**QUERY SYNTAX** MEASure:VOLTage:TMINimum? <src>,<num>

Queries the time at which the selected waveform reaches its minimum value. The following considerations apply when using the measure time at minimum query:

- The time at minimum measurement is defined as the time that the waveform signal reaches its minimum value.
- <num> selects the waveform number from the waveform source to be analyzed. <num> is used to select a particular waveform in the case of multiple waveform capture, see the TRIGger:COUNT command. <num> must be 1 for all waveform sources except input signal channels.
- The query parameters are position sensitive.

<b>COMMAND PARAMETERS</b>	<src>	waveform source, if not specified defaults to INPut1 CALCulate<n> waveform math channel, where <n> may be 1 or 2 INPut<n> front panel input signals, where <n> may be 1, 2, 3, or 4 REFerence<n> reference waveform, where <n> may be 1, 2, 3,... or 6
	<num>	desired waveform number, if not specified defaults to 1
<b>RESPONSE PARAMETERS</b>	<time>	the time when the waveform reaches its minimum value
<b>DEFAULT VALUE</b>	n/a	
<b>EXAMPLES</b>	MEAS:VOLT:TMIN? INP1,2	query time that INPut1, waveform number 2, reaches its minimum value, example response 1.3286E-6
<b>STATUS EVENTS</b>	-141 -222	invalid waveform source name specified, command was ignored out-of-range waveform number selected, value was coerced in range
<b>AFFECTIVITY</b>	None	
<b>RELATED COMMANDS</b>	TRIGger:COUNT	

### 1.3.46 Measure Voltage, Time at Voltage (MEASure:VOLTage:TVOLTage?)

**COMMAND SYNTAX** n/a

**QUERY SYNTAX** MEASure:VOLTage:TVOLTage? <src>,<num>,<volt>

Queries the time at which the selected waveform first reaches a selected voltage. The following considerations apply when using the measure time at voltage query:

- <num> selects the waveform number from the waveform source to be analyzed. <num> is used to select a particular waveform in the case of multiple waveform capture, see the TRIGger:COUNT command. <num> must be 1 for all waveform sources except input signal channels.
- The query parameters are position sensitive.

<b>COMMAND PARAMETERS</b>	<src>	waveform source, if not specified defaults to INPut1 CALCulate<n> waveform math channel, where <n> may be 1 or 2 INPut<n> front panel input signals, where <n> may be 1, 2, 3, or 4 REFerence<n> reference waveform, where <n> may be 1, 2, 3,... or 6
	<num>	desired waveform number, if not specified defaults to 1
	<volt>	desired voltage to detect, if not specified defaults to 0.0
<b>RESPONSE PARAMETERS</b>	<time>	the time when the waveform reaches its minimum value
<b>DEFAULT VALUE</b>	n/a	
<b>EXAMPLES</b>	MEAS:VOLT:TVOL? INP1,2,0.1V	query time that INPut1, waveform number 2, first reaches 0.1 V, example response 1.3286E-6
<b>STATUS EVENTS</b>	-141	invalid waveform source name specified, command was ignored
	-222	out-of-range waveform number selected, value was coerced in range
<b>AFFECTIVITY</b>	None	
<b>RELATED COMMANDS</b>	TRIGger:COUNT	

# 1

## 1.3.47 Measure Voltage, Voltage at Time (MEASure:VOLTage:VTIME?)

**COMMAND SYNTAX** n/a

**QUERY SYNTAX** MEASure:VOLTage:VTIME? <src>,<num>,<time>

Queries the voltage at a selected waveform time. The following considerations apply when using the measure voltage at time query:

- <num> selects the waveform number from the waveform source to be analyzed. <num> is used to select a particular waveform in the case of multiple waveform capture, see the TRIGger:COUNT command. <num> must be 1 for all waveform sources except input signal channels.
- The query parameters are position sensitive.

<b>COMMAND PARAMETERS</b>	<src>	waveform source, if not specified defaults to INPut1 CALCulate<n> waveform math channel, where <n> may be 1 or 2 INPut<n> front panel input signals, where <n> may be 1, 2, 3, or 4 REFerence<n> reference waveform, where <n> may be 1, 2, 3,... or 6
	<num>	desired waveform number, if not specified defaults to 1
	<time>	desired time to read, if not specified defaults to 0.0
<b>RESPONSE PARAMETERS</b>	<volt>	the voltage at the selected waveform time
<b>DEFAULT VALUE</b>	n/a	
<b>EXAMPLES</b>	MEAS:VOLT:VTIM? INP1,2,2 US	query voltage at INPut1, waveform number 2, time 2 usec, example response 1.5E-1
<b>STATUS EVENTS</b>	-141	invalid waveform source name specified, command was ignored
	-222	out-of-range waveform number selected, value was coerced in range
<b>AFFECTIVITY</b>	None	
<b>RELATED COMMANDS</b>	TRIGger:COUNT	

### 1.3.48 Output ECL Trigger Polarity (OUTPut:ECLTrg<n>:POLarity)

**COMMAND SYNTAX**        OUTPut:ECLTrg<n>:POLarity <pol>

**QUERY SYNTAX**        OUTPut:ECLTrg<n>:POLarity?

Sets or queries the unit VXIbus ECLTRGn output driver polarity. The following considerations apply when using the ECL output polarity command:

- When positive output polarity is selected, the VXIbus output driver will force a logic 1 onto the ECL output when the signal source is active, i.e. for a POSitive polarity setting and an ARM source signal, the ECL trigger line will be a logic 1 when the unit is armed.
- Output driver polarity does not affect VXIbus ECLTRG line sensing used by other unit functions.
- Each output line polarity is selected individually
- ECLTRG lines can be sourced and sensed simultaneously.

<b>COMMAND PARAMETERS</b>	<n> <pol>	VXIbus ECLTRG line to update, may be 0 or 1 desired output driver polarity POSitive    output driver does not invert the selected source signal NEGative    output driver inverts the inverted source signal
<b>RESPONSE PARAMETERS</b>	<pol>	current output driver polarity POS        output driver does not invert the selected source signal NEG        output driver inverts the inverted source signal
<b>DEFAULT VALUE</b>	POS	
<b>EXAMPLES</b>	OUTP:ECLT0:POL NEG OUTP:ECLT0?	inverted output driver query current output driver setting, example response: NEG
<b>STATUS EVENTS</b>	-141 -241	invalid driver polarity name, command was ignored invalid output line, command was ignored
<b>AFFECTIVITY</b>	None	
<b>RELATED COMMANDS</b>	OUTPut:ECLTrg<n>:SOURce OUTPut:ECLTrg<n>:STATE	

### 1.3.49 Output ECL Trigger Source (OUTPut:ECLTrg<n>:SOURce)

**COMMAND SYNTAX** OUTPut:ECLTrg<n>:SOURce <src>

**QUERY SYNTAX** OUTPut:ECLTrg<n>:SOURce?

Sets or queries the unit VXIbus ECLTRGn output driver source. The following considerations apply when using the ECL output source command:

- The output driver may be enabled or disabled and the polarity selected, refer to the OUTPut:ECLTrg<n>:POLarity and OUTPut:ECLTrg<n>:SOURce commands.
- Each output line source is selected individually
- ECLTRG lines can be sourced and sensed simultaneously.

<b>COMMAND PARAMETERS</b>	<n>	VXIbus ECLTRG line to update, may be 0 or 1
	<src>	desired ECLTRG signal source
	ARM	the unit "arm detected" signal
	TRGA	the unit "primary trigger detected" event
	TRGB	the unit "secondary trigger detected" event
	OPC	the unit "operation complete" indicator
<b>RESPONSE PARAMETERS</b>	<src>	current ECLTRG source
<b>DEFAULT VALUE</b>	TRGA	(VXIbus ECLTRG0)
	TRGB	(VXIbus ECLTRG1)
<b>EXAMPLES</b>	OUTP:ECLT1:SOUR ARM	when ECL output 1 is enabled, drive the unit armed signal onto the bus
	OUTP:ECLT1:SOUR?	query current output driver source, example response: ARM (or select the unit armed status as the output source)
<b>STATUS EVENTS</b>	-141	invalid driver source name, command was ignored
	-241	invalid output line, command was ignored
<b>AFFECTIVITY</b>	None	
<b>RELATED COMMANDS</b>	OUTPut:ECLTrg<n>:POLarity	
	OUTPut:ECLTrg<n>:STATe	

### 1.3.50 Output ECL Trigger State (OUTPut:ECLTrg<n>:STATe)

<b>COMMAND SYNTAX</b>	OUTPut:ECLTrg<n>[:STATe] <sw>	full
	OUTPut:ECLTrg<n> <sw>	minimum
<b>QUERY SYNTAX</b>	OUTPut:ECLTrg<n>[:STATe]?	full
	OUTPut:ECLTrg<n>?	minimum

Sets or queries the unit VXIbus ECLTRGn output driver enable. The following considerations apply when using the ECL output state command:

- The output source and polarity are selectable, refer to the OUTPut:ECLTrg<n>:POLarity and OUTPut:ECLTrg<n>:SOURce commands.
- Each output line driver state is selected individually
- ECLTRG lines can be sourced and sensed simultaneously.

<b>COMMAND PARAMETERS</b>	<n>	VXIbus ECLTRG line to update, may be 0 or 1
	<sw>	desired output driver state 0 or OFF    disable output driver 1 or ON    enable output driver
<b>RESPONSE PARAMETERS</b>	<sw>	current output driver state 0    output driver disabled 1    output driver enabled
	<b>DEFAULT VALUE</b>	0
<b>EXAMPLES</b>	OUTP:ECLT0 ON	enable output driver
	OUTP:ECLT0?	query current output driver setting, example response: 1 (or output enabled)
<b>STATUS EVENTS</b>	-141	invalid driver state name, command was ignored
	-241	invalid output line, command was ignored
<b>AFFECTIVITY</b>	None	
<b>RELATED COMMANDS</b>	OUTPut:ECLTrg<n>:POLarity	
	OUTPut:ECLTrg<n>:SOURce	

### 1.3.51 Output Reference Source (OUTPut:REfERENCE:SOURce)

**COMMAND SYNTAX** OUTPut:REfERENCE:SOURce <func>

**QUERY SYNTAX** OUTPut:REfERENCE: SOURce?

Sets or queries the function output on the unit front panel REF OUT connector when the output is enabled. The following considerations apply when using the reference source command:

- The REF OUT connector is only driven when enabled, refer to the OUTPut:REfERENCE:STATe command.
- The TRGA and TRGB sources are active high
- The VOLTage source is only capable of sourcing 1 mA while remaining within tolerance, all other sources are capable of driving a 2 Vpp signal into a 50  $\Omega$  load.

<b>COMMAND PARAMETERS</b>	<func>	desired output function
		COMPensation a 500 Hz square wave, probe compensation signal
		PULSe a 1 kHz repetition rate, 10 nsec wide pulse signal
		REFerence the selected 10 MHz reference source
		TRGA the unit "primary trigger detected" event
		TRGB the unit "secondary trigger detected" event
		VOLTage an 8 V reference voltage
<b>RESPONSE PARAMETERS</b>	<func>	current output function
		COMP a 500 Hz square wave, probe compensation signal
		PULS a 1 kHz repetition rate, 10 nsec wide pulse signal
		REF the selected 10 MHz reference source
		TRGA the unit "primary trigger detected" event
		TRGB the unit "secondary trigger detected" event
		VOLT an 8 V reference voltage
<b>DEFAULT VALUE</b>	COMP	
<b>EXAMPLES</b>	OUTP:REF:SOUR COMP	select the 500 Hz probe compensation signal for output
	OUTP:REF:SOUR?	query current output signal source, example response: COMP (or output the probe compensation signal)
<b>STATUS EVENTS</b>	-141	invalid output signal source name, command was ignored
<b>AFFECTIVITY</b>	None	
<b>RELATED COMMANDS</b>	OUTPut:REfERENCE:STATe	

### 1.3.52 Output Reference State (OUTPut:REFeRence:STATe)

**COMMAND SYNTAX**      OUTPut:REFeRence[:STATe] <sw>  
**QUERY SYNTAX**      OUTPut:REFeRence[:STATe]?

Sets or queries the front panel REF OUT driver enable. The signal driven onto the REF OUT connector is selectable among several sources, refer to the OUTPut:REFeRence:SOURce command:

- The signal produced on the REF OUT connector is selectable among several sources, refer to the OUTPut:REFeRence:FUNcTion command.

**COMMAND PARAMETERS**      <sw>      desired output driver enable  
                                          ON or 1      enable the output driver  
                                          OFF or 0      disable the output driver

**RESPONSE PARAMETERS**      <sw>      current output driver enable state  
                                          1      output driver enabled  
                                          0      output driver disabled

**DEFAULT VALUE**      0

**EXAMPLES**      OUTP:REF 0      disable the output driver  
                          OUTP:REF?      query current output driver state, example response: 0

**STATUS EVENTS**      -141      invalid output driver control name, command was ignored

**AFFECTIVITY**      None

**RELATED COMMANDS**      OUTPut:REFeRence:FUNcTion

### 1.3.53 Output TTL Trigger Polarity (OUTPut:TTLTrg<n>:POLarity)

**COMMAND SYNTAX** OUTPut:TTLTrg<n>:POLarity <pol>

**QUERY SYNTAX** OUTPut:TTLTrg<n>:POLarity?

Sets or queries the unit VXIbus TTLTRGn output driver polarity. The following considerations apply when using the TTL output polarity command:

- When positive output polarity is selected, the output driver will force a logic 0 onto the TTL output when the signal source is active, i.e. for a POSitive polarity setting and an ARM source signal, the TTL trigger line will be a logic 0 when the unit is armed.
- Output driver polarity does not affect VXIbus TTLTRG line sensing used by other unit functions.
- Each output line polarity is selected individually
- TTLTRG lines can be sourced and sensed simultaneously.

<b>COMMAND PARAMETERS</b>	<n>	VXIbus TTLTRG line to update, may be 0, 1, 2, ..., or 7
	<pol>	desired output driver polarity POSitive output driver follows the inverted selected source signal NEGative output driver follows the source signal
<b>RESPONSE PARAMETERS</b>	<pol>	current output driver polarity
	POS	output driver follows the inverted selected source signal
	NEG	output driver follows the source signal
<b>DEFAULT VALUE</b>	POS	
<b>EXAMPLES</b>	OUTP:TTLT0:POL NEG	inverted output driver
	OUTP:TTLT0?	query current output driver setting, example response: NEG
<b>STATUS EVENTS</b>	-141	invalid driver polarity name, command was ignored
	-241	invalid output line, command was ignored
<b>AFFECTIVITY</b>	None	
<b>RELATED COMMANDS</b>	OUTPut:TTLTrg<n>:SOURce	
	OUTPut:TTLTrg<n>:STATe	

### 1.3.54 Output TTL Trigger Source (OUTPut:TTLTrg<n>:SOURce)

**COMMAND SYNTAX**    OUTPut:TTLTrg<n>:SOURce <src>

**QUERY SYNTAX**        OUTPut:TTLTrg<n>:SOURce?

Sets or queries the unit VXIbus TTLTRGn output driver source. The following considerations apply when using the TTL output source command:

- The output driver may be enabled or disabled and the polarity selected, refer to the OUTPut:TTLTrg<n>:POLarity and OUTPut:TTLTrg<n>:SOURce commands.
- Each output line source is selected individually
- TTLTRG lines can be sourced and sensed simultaneously.

<b>COMMAND PARAMETERS</b>	<n>  <src>	VXIbus TTLTRG line to update, may be 0, 1, 2, ..., or 7  desired TTLTRG signal source ARM    the unit "arm detected" signal TRGA   the unit "primary trigger detected" event TRGB   the unit "secondary trigger detected" event OPC    the unit "operation complete" indicator
<b>RESPONSE PARAMETERS</b>	<src>	current TTLTRG source
<b>DEFAULT VALUE</b>	ARM TRGA TRGB OPC ARM TRGA TRGB OPC	(VXIbus TTLTRG0) (VXIbus TTLTRG1) (VXIbus TTLTRG2) (VXIbus TTLTRG3) (VXIbus TTLTRG4) (VXIbus TTLTRG5) (VXIbus TTLTRG6) (VXIbus TTLTRG7)
<b>EXAMPLES</b>	OUTP:TTL1:SOUR ARM  OUTP:TTL1:SOUR?	when TTL output 1 is enabled, drive the unit armed signal onto the bus  query current output driver source, example response: ARM (or select the unit armed status as the output source)
<b>STATUS EVENTS</b>	-141 -241	invalid driver source name, command was ignored invalid output line, command was ignored
<b>AFFECTIVITY</b>	None	
<b>RELATED COMMANDS</b>	OUTPut:TTLTrg<n>:POLarity OUTPut:TTLTrg<n>:STATe	

### 1.3.55 Output TTL Trigger State (OUTPut:TTLTrg<n>:STATe)

<b>COMMAND SYNTAX</b>	OUTPut:TTLTrg<n>[:STATe] <sw>	full
	OUTPut:TTLTrg<n> <sw>	minimum
<b>QUERY SYNTAX</b>	OUTPut:TTLTrg<n>[:STATe]?	full
	OUTPut:TTLTrg<n>?	minimum

Sets or queries the unit VXIbus TTLTRGn output driver enable. The following considerations apply when using the TTL output state command:

- The output source and polarity are selectable, refer to the OUTPut:TTLTrg<n>:POLarity and OUTPut:TTLTrg<n>:SOURce commands.
- Each output line driver state is selected individually
- TTLTRG lines can be sourced and sensed simultaneously.

<b>COMMAND PARAMETERS</b>	<n>	VXIbus TTLTRG line to update, may be 0 or 1
	<sw>	desired output driver state
		0 or OFF    disable output driver 1 or ON     enable output driver
<b>RESPONSE PARAMETERS</b>	<sw>	current output driver state
		0    output driver disabled 1    output driver enabled
<b>DEFAULT VALUE</b>	0	
<b>EXAMPLES</b>	OUTP:TTLT0 ON	enable output driver
	OUTP:TTLT0?	query current output driver setting, example response: 1 (or output enabled)
<b>STATUS EVENTS</b>	-141	invalid driver state name, command was ignored
	-241	invalid output line, command was ignored
<b>AFFECTIVITY</b>	None	
<b>RELATED COMMANDS</b>	OUTPut:TTLTrg<n>:POLarity	
	OUTPut:TTLTrg<n>:SOURce	

### 1.3.56 Sense Autoscale (SENSE:AUToscale)

**COMMAND SYNTAX** [:SENSe]:AUToscale full  
 AUToscale minimum

**QUERY SYNTAX** n/a

Commands the unit to automatically detect the input signal amplitude and frequency and to configure the input channels and sweep rate to capture these signals. The following considerations apply when using the autoscale command:

- Peak-to-peak range is independently adjusted for each channel to force the peak signal levels to approximately 2/3 of full scale.
- The sweep time is adjusted capture approximately 2 cycles of the signal on INPut1.
- The trigger is set to edge mode and channel 1 source.
- The input impedance, input coupling, and sweep points are not changed

**COMMAND PARAMETERS** n/a

**RESPONSE PARAMETERS** n/a

**DEFAULT VALUE** n/a

**EXAMPLES** AUT autoscale the waveform acquisition parameters

**STATUS EVENTS** None

**AFFECTIVITY** None

**RELATED COMMANDS** None

### 1.3.57 Sense Average Count (:SENSe:AVERage:COUNT)

<b>COMMAND SYNTAX</b>	[SENSe]:AVERage:COUNT <cnt>	full
	AVERage:COUNT <cnt>	minimum
<b>QUERY SYNTAX</b>	[SENSe]:AVERage:COUNT?	full
	AVERage:COUNT?	minimum

Sets or queries the number input waveforms to average in order to produce an output waveform. The following considerations apply when using the :SENSe:AVERage:COUNT command:

- The averager resides between the input signal digitizers and the rest of the unit functionality. When the averager is enabled, only the final output waveform is retained, the component waveforms are lost.
- The averager may be disabled and its operation mode selected, refer to the SENSe:AVERager:STATe and SENSe:AVERager:TYPE commands respectively.
- The averager count setting affects all active input channels, other waveform sources are not affected.
- The average and multiple waveform capture modes can not be invoked at the same time. When the average count is set to a value other than 1, the :TRIGger:COUNT is reset to 1.

<b>COMMAND PARAMETERS</b>	<cnt>	desired number of waveforms to average 1 to 2048
<b>RESPONSE PARAMETERS</b>	<cnt>	current number of waveforms to average
<b>DEFAULT VALUE</b>	2	
<b>EXAMPLES</b>	AVER:COUN 4	average four waveforms before generating an output waveform
	AVER:COUN?	query current waveform count, example response: 4 (or average 4 waveforms)
<b>STATUS EVENTS</b>	-222	the average count is out-of-range, the count has been forced in range
<b>AFFECTIVITY</b>		Setting the average count to a value > 1 forces the trigger count to 1.
<b>RELATED COMMANDS</b>	SENSe:AVERage:STATe SENSe:AVERage:TYPE TRIGger:COUNT	

### 1.3.58 Sense Average State (SENSe:AVERage:STATe)

<b>COMMAND SYNTAX</b>	[SENSe:]AVERage[:STATe] <sw>	full
	AVERage <sw>	minimum
<b>QUERY SYNTAX</b>	[SENSe:]AVERage[:STATe]?	full
	AVERage?	minimum

Sets or queries the waveform averager enable. The following considerations apply when using the average state command:

- The averager resides between the input signal digitizers and the rest of the unit functionality. When the averager is enabled, only the final output waveform is retained, the raw, un-averaged data is not available.
- When the averager is disabled, (NORMAL acquisition mode), waveforms from the digitizers are passed through without modification.
- The number of waveforms averaged and the averager operation mode can be selected using the SENSe:AVERager:COUNT and SENSe:AVERager:TYPE commands respectively.
- The averager enable setting affects all active input channels, other waveform sources are not affected.
- The average and multiple waveform capture modes can not be invoked at the same time. When the average count is set to a value other than 1, the :TRIGger:COUNT is reset to 1.

<b>COMMAND PARAMETERS</b>	<sw>	desired averager state 0 or OFF    disable averaging 1 or ON    enable averaging
<b>RESPONSE PARAMETERS</b>	<sw>	current averager state 0    averager disabled 1    averager enabled
<b>DEFAULT VALUE</b>	0	
<b>EXAMPLES</b>	AVER ON	enable averager
	AVER?	query current averager enable setting, example response: 1 (or averager enabled)
<b>STATUS EVENTS</b>	-141	invalid averager enable name, command was ignored
<b>AFFECTIVITY</b>		Enabling the averager forces the trigger count to 1.
<b>RELATED COMMANDS</b>	SENSe:AVERage:COUNT SENSe:AVERage:TYPE TRIGger:COUNT	

### 1.3.59 Sense Average Type (SENSe:AVERage:TYPE)

<b>COMMAND SYNTAX</b>	[SENSe]:AVERage:TYPE <type>	full
	AVERage:TYPE <type>	minimum
<b>QUERY SYNTAX</b>	[SENSe]:AVERage:TYPE?	full
	AVERage:TYPE?	minimum

Sets or queries the averager operation mode. The following considerations apply when using the SENSe:AVERage:TYPE command:

- The averager resides between the input signal digitizers and the rest of the unit functionality. When the averager is enabled, only the final output waveform is retained, the component waveforms are lost.
- Two forms of averaging are provided: an “boxcar” averager and an envelope detector.
- The boxcar averager sums each point in the acquired waveforms with associated points in the other captured waveforms and then divides the result by the number of waveforms summed.
- The boxcar averager reduces random signal noise by approximately 3 dB for each doubling of the COUNT, ex. averaging 1 waveform (no averaging) results in a 0 dB noise improvement, 2 waveforms results in a 3 dB improvement, 4 waveforms results in 6 dB improvement, ... Averaging does not work well when the noise is related to the desired signal.
- The averager output record length is equal to the input record size.
- The envelope detector forms two output waveforms, one containing the maximum value of all the input waveforms for each sample in the output record, the second containing the minimum values.
- The enveloper output record contains two waveforms, each equal in length to the input record size.
- The number of waveforms averaged and averager enable may be selected, refer to the SENSe:AVERager:TYPE and SENSe:AVERager:STATe commands respectively.
- The averager type setting affects all active input channels, other waveform sources are not affected.
- The average and multiple waveform capture modes can not be invoked at the same time. When the average count is set to a value other than 1, the :TRIGger:COUNT is reset to 1.

<b>COMMAND PARAMETERS</b>	<type>	desired averager mode NORMal boxcar averager mode ENVelope envelope mode
<b>RESPONSE PARAMETERS</b>	<type>	current averager mode NORM boxcar averager mode ENV envelope mode
<b>DEFAULT VALUE</b>	NORM	
<b>EXAMPLES</b>	AVER:TYPE NORM	select boxcar averager mode
	AVER:TYPE?	query current averager mode, example res ponse: NORM
<b>STATUS EVENTS</b>	-141	invalid averager mode name selected
<b>AFFECTIVITY</b>		Enabling the averager forces the trigger count to 1.
<b>RELATED COMMANDS</b>	SENSe:AVERage:COUNT SENSe:AVERage:STATe TRIGger:COUNT	

### 1.3.60 Sense Channel Count (SENSE:CHANnel:COUNT?)

**COMMAND SYNTAX** n/a

**QUERY SYNTAX** [SENSe]:CHANnel:COUNT? full  
 CHANnel:COUNT? minimum

Queries the number of sense channels enabled. The ZT432VXI provides both 2 and 4 channel operation, refer to the SENSE:CHANnel:MODE command for information on setting the number of active channels.

**COMMAND PARAMETERS** n/a

**RESPONSE PARAMETERS** <cnt> current number of channels enabled  
 2 channel modes CHAN12 or CHAN34 invoked  
 4 channel mode ALL4 invoked

**DEFAULT VALUE** n/a

**EXAMPLES** CHAN:COUN? query number of channels enabled, example response: 4

**STATUS EVENTS** None

**AFFECTIVITY** None

**RELATED COMMANDS** SENSe:CHANnel:MODE

# 1

## 1.3.61 Sense Channel Mode (SENSe:CHANnel:MODE)

<b>COMMAND SYNTAX</b>	[SENSe]:CHANnel:MODE <mode>	full
	CHANnel:MODE <mode>	minimum
<b>QUERY SYNTAX</b>	[SENSe]:CHANnel:MODE?	full
	CHANnel:MODE?	minimum

Sets or queries the input channel mode. The following considerations apply when using the SENSe:CHANnel:MODE command:

- The unit can operate in 2 channel or 4 channel mode.
- The following channel modes are supported

active	ALL4	4 channel mode, all channels are
are active	CHAN12	2 channel mode, channels 1 and 2
are active	CHAN34	2 channel mode, channels 3 and 4

- When in 2 channel mode each channel may be sampled at up to the full 200 MS/s ADC rate. The final sample rate may be extended to 2 GS/s using the unit's interpolation function. Refer to The Installation, Operation, Programming, & Diagnostics Manual for a list of available sample rates.
- When in 2 channel mode each channel may use up to the full digitizer memory for waveform record storage. Refer to the SENSe:SWEep:POINts command for information on selecting the waveform record size.
- When in 4 channel mode each channel may be sampled at up to ½ the 2 channel rates and may use up to ½ of the full digitizer memory for waveform record storage.
- Changing the channel mode will result in the sweep time and points being re-evaluated and possibly changed. If the points or time are changed, a warning will be issued.
- The number of active channels may be determined using the SENSe:CHANnel:COUNT? query.

<b>COMMAND PARAMETERS</b>	<mode>	desired input channel mode
	ALL4	4 channel mode, all channels active
	CHAN12	2 channel mode, channels 1 and 2 active
	CHAN34	2 channel mode, channels 3 and 4 active

**RESPONSE PARAMETERS** <mode> current input channel mode

**DEFAULT VALUE** ALL4

**EXAMPLES** CHAN:MODE CHAN12 select 2 channel mode, channels 1 and 2 active  
 CHAN:MODE? query the channel mode, example response: CHAN12 (or channels 1 and 2 are active)

**STATUS EVENTS** -141 invalid channel mode name selected, command was ignored

**AFFECTIVITY** None

**RELATED COMMANDS** SENSe:CHANnel:COUNT  
 SENSe:SWEep:POINts

### 1.3.62 Sense Input Attenuation (SENSe:INPut:ATTenuation)

<b>COMMAND SYNTAX</b>	[SENSe]:INPut<n>:ATTenuation <atten>	full
	INPut<n>:ATTenuation <atten>	minimum
<b>QUERY SYNTAX</b>	[SENSe]:INPut<n>:ATTenuation?	full
	INPut<n>:ATTenuation?	minimum

Sets or queries the external attenuation for an input signal. The attenuation feature allows the user to set voltage levels using unattenuated values. The following considerations apply when using the input attenuation command:

- The attenuation command would be used for example when a 10x probe is attached to an input channel. If the user desires to capture a 3 Vpp signal then the following settings could be used:  
 Unit input attenuation = 1.0 and input signal range = 0.3 Vpp (3 Vpp / 10.0)  
 Unit input attenuation = 10.0 and input signal range = 3 Vpp
- As seen above the attenuation command allows the user to set voltages in the source signal levels. Refer to the VOLTage:RANGe:PTPeak command for information on input signal range setting.
- When the input attenuation is changed, the current input range will be scaled to match the attenuation setting according to the following equation:  

$$\text{New Input Range} = \text{new attenuation} * (\text{old range} / \text{old attenuation})$$
- An attenuation setting of 1.0 indicates that no external attenuation is used, i.e. the attenuation feature is disabled.
- Each input channel's attenuation is set independently.

<b>COMMAND PARAMETERS</b>	<n>	signal input number, can be 1, 2, 3, or 4, defaults to 1 if omitted
	<atten>	external signal attenuation, can range from 0.9 to 1000.0
<b>RESPONSE PARAMETERS</b>	<atten>	current input signal attenuation setting
<b>DEFAULT VALUE</b>	1.0	
<b>EXAMPLES</b>	INP3:ATT 10.0	set input attenuation to account for an external 10x probe
	INP:ATT?	query the current attenuation setting for input signal channel 1, example response: 1.0 (or attenuation function disabled)
<b>STATUS EVENTS</b>	-222	attenuation setting out-of-range, attenuation coerced in range
	-241	invalid input channel number, command was ignored
<b>AFFECTIVITY</b>		Attenuation setting affects all voltage setting commands and queries on the affected channel, ex. voltage range, voltage measurements, and trigger levels.
<b>RELATED COMMANDS</b>	VOLTage:RANGe:PTPeak	

### 1.3.63 Sense Input Coupling (SENSe:INPut:COUPling)

**COMMAND SYNTAX** [SENSe]:INPut<n>:COUPling <couple> full  
 INPut<n>:COUPling <couple> minimum

**QUERY SYNTAX** [SENSe]:INPut<n>:COUPling? full  
 INPut<n>:COUPling? minimum

Sets or queries the input signal coupling. The signal coupling feature allows the user to selectively pass or block the DC component of an input signal. The following considerations apply when using the input coupling command:

- DC coupling passes all frequencies equally
- AC coupling blocks low frequencies.
- AC coupling and input impedance setting interact to set the low frequency cutoff frequency. AC and high input impedance attenuates frequencies below 50 Hz. AC coupling and low input impedance attenuates frequencies below 1 MHz. Refer to the SENSe:INPut:IMPedance command for information on impedance setting.

**COMMAND PARAMETERS** <n> signal input number, can be 1, 2, 3, or 4, defaults to 1 if omitted  
 <couple> desired input signal coupling  
 DC pass all frequencies  
 AC block low frequencies

**RESPONSE PARAMETERS** <couple> current input signal coupling

**DEFAULT VALUE** DC

**EXAMPLES** INP3:COUP AC block low frequencies on input signal channel 3  
 INP:COUP? query the current coupling on input signal channel 1, example response: DC (or DC component passed)

**STATUS EVENTS** -141 invalid input coupling name selected, command was ignored  
 -241 invalid input channel number, command was ignored

**AFFECTIVITY** None

**RELATED COMMANDS** SENSe:INPut:IMPedance

### 1.3.64 Sense Input Impedance (SENSe:INPut:IMPedance)

**COMMAND SYNTAX** [SENSe:]INPut<n>:IMPedance <ohms> full  
 INPut<n>:IMPedance <ohms> minimum

**QUERY SYNTAX** [SENSe:]INPut<n>:IMPedance? full  
 INPut<n>:IMPedance? minimum

- Sets or queries the input impedance of the specified input channel. The impedances provided by the ZT432VXI are 50 Ω and 1 MΩ.

**COMMAND PARAMETERS**

<n>	signal input number, can be 1, 2, 3, or 4, defaults to 1 if omitted
<ohms>	desired input impedance
	50.050 Ω, low impedance input
	1.0e6 1 MΩ, high impedance input
	MINimum 50 Ω, low impedance input
	MAXimum 1 MΩ, high impedance input

**RESPONSE PARAMETERS**

<ohms>	current input impedance setting
	5.0e+1 low impedance input
	1.0e+6 high impedance input

**DEFAULT VALUE** 1.0e+6

**EXAMPLES**

INP:IMP MIN	select low impedance input for signal input channel 1
INP:IMP?	query current input impedance, example response: 5.0e+1 (or low impedance)

**STATUS EVENTS**

-141	invalid impedance setting name, command was ignored
-222	impedance value out-of-range, value coerced in range
-241	invalid input channel number, command was ignored

**AFFECTIVITY** Input impedance and input coupling settings interact, refer to the SENSe:INPut:COUpling command

**RELATED COMMANDS** SENSe:INPut:COUpling

# 1

## 1.3.65 Sense Input Preset (SENSe:INPut:PRESet)

**COMMAND SYNTAX** [SENSe]:INPut<n>:PRESet <mode> full  
 INPut<n>:PRESet <mode> minimum

**QUERY SYNTAX** n/a

Configures a signal input channel for capturing a particular signal type. The following considerations apply when using the SENSe:INPut:PRESet command:

- Presets are provided for TTL and ECL logic signals.
- Sweep parameters such as sweep points and sweep time are not affected.
- Presets affect only the selected input channel.

**COMMAND PARAMETERS** <n> signal input number, can be 1, 2, 3, or 4, defaults to 1 if omitted  
 <mode> desired input preset type  
 ECLconfigure for -5 V ECL logic  
 TTL configure for +5 V TTL logic

**RESPONSE PARAMETERS** n/a

**DEFAULT VALUE** n/a

**EXAMPLES** INP3:PRESet ECL preset input channel 3 for ECL signal detection

**STATUS EVENTS** -141 invalid preset type name selected, command was ignored  
 -241 invalid input channel number, command was ignored

**AFFECTIVITY** The PRESet command modifies the SENSe:VOLTage:RANGe:PTPeak setting.

**RELATED COMMANDS** SENSe:VOLTage:RANGe:PTPeak

### 1.3.66 Sense Input Smoothing (SENSe:INPut:SMOothing:POINts)

<b>COMMAND SYNTAX</b>	[SENSe]:INPut<n>:SMOothing:POINts <pts>	full
	INPut<n>:SMOothing:POINts <pts>	minimum
<b>QUERY SYNTAX</b>	[SENSe]:INPut<n>:SMOothing:POINts?	full
	INPut<n>:SMOothing:POINts?	minimum

Sets or queries the number of points used to smooth the captured input signal. The following considerations apply when using the SENSe:INPut:SMOothing:POINts command:

- Both 10 point and 50 point, linear phase, IIR smoothing filters are provided
- Smoothing must be enabled to affect the captured signals, refer to the SENSe:INPut:SMOothing:STATe command
- Smoothing is applied during waveform readout to samples stored in digitizer memory. The raw, unsmoothed data may be accessed by disabling smoothing.
- Smoothing is configured independently for each input channel.

<b>COMMAND PARAMETERS</b>	<n>	signal input number, can be 1, 2, 3, or 4, defaults to 1 if omitted
	<pts>	desired smoothing filter length
	10	short smoothing filter
	50	long smoothing filter
	MINimum	short smoothing filter
	MAXimum	long smoothing filter

<b>RESPONSE PARAMETERS</b>	<pts>	current smoothing filter length
	10	short smoothing filter
	50	long smoothing filter

**DEFAULT VALUE** 10

<b>EXAMPLES</b>	INP4:SMO:POIN MIN	select the short smoothing filter for input channel 4
	INP4:SMO:POIN?	query current filter length, example response: 10

<b>STATUS EVENTS</b>	-141	invalid smoothing filter length name selected, command was ignored
	-222	smoothing points value out-of-range, value was coerced in range
	-241	invalid input channel number, command was ignored

**AFFECTIVITY** None

**RELATED COMMANDS** SENSe:INPut:SMOothing:STATe

### 1.3.67 Sense Input Smoothing State (SENSe:INPut:SMOothing:STATe)

<b>COMMAND SYNTAX</b>	[SENSe]:INPut<n>:SMOothing[:STATe] <sw>	full
	INPut<n>:SMOothing <sw>	minimum
<b>QUERY SYNTAX</b>	[SENSe]:INPut<n>:SMOothing[:STATe]?	full
	INPut<n>:SMOothing?	minimum

Sets or queries the smoothing filter enable. Smoothing replaces each sample in a waveform record with a filtered value determined from a specified number of preceding points in the waveform record, refer to the SENSe:INPut:SMOothing:POINts command for information on setting the filter length

<b>COMMAND PARAMETERS</b>	<n>	signal input number, can be 1, 2, 3, or 4, defaults to 1 if omitted
	<sw>	desired smoothing filter state 0 or OFF    disable the smoothing filter 1 or ON     enable the smoothing filter
<b>RESPONSE PARAMETERS</b>	<sw>	current smoothing filter state 0            smoothing filter disabled 1            smoothing filter enabled
<b>DEFAULT VALUE</b>	0	
<b>EXAMPLES</b>	INP:SMO 0	disable smoothing for input channel 1
	INP:SMO?	query current auto-advance mode setting, example response: 0 (or smoothing disabled)
<b>STATUS EVENTS</b>	-141	invalid smoothing filter state name selected, command was ignored
<b>AFFECTIVITY</b>	None	
<b>RELATED COMMANDS</b>	SENSe:INPut:SMOothing:POINts	

### 1.3.68 Sense Input State (SENSE:INPut:STATe)

<b>COMMAND SYNTAX</b>	[SENSe]:INPut<n>[:STATe] <sw>	full
	INPut<n> <sw>	minimum
<b>QUERY SYNTAX</b>	[SENSe]:INPut<n>[:STATe]?	full
	INPut<n>?	minimum

Sets or queries the input channel automatic download enable. When an input channel's download is disabled then that channel will not be copied when post-waveform capture automatic downloads are performed, refer to the TRACe:DOWNload:STATe command. A waveform is captured for every active channel, this command speeds acquisition completion when automatic downloading is enabled by not copying disabled channels.

<b>COMMAND PARAMETERS</b>	<n>	signal input number, can be 1, 2, 3, or 4, defaults to 1 if omitted
	<sw>	desired enable state 0 or OFF     disable the input channel copying 1 or ON     enable the input channel copying
<b>RESPONSE PARAMETERS</b>	<sw>	current enable state 0     input channel copying disabled 1     input channel copying enabled
<b>DEFAULT VALUE</b>	0	
<b>EXAMPLES</b>	INP1 ON	enable copying for input channel 1
	INP1?	query current download enable setting, example response: 0 (or automatic download disabled)
<b>STATUS EVENTS</b>	-141	invalid smoothing filter state name selected, command was ignored
<b>AFFECTIVITY</b>	None	
<b>RELATED COMMANDS</b>	TRACe:DOWNload:STATe	

### 1.3.69 Sense Reference Oscillator Source (SENSe:ROSCillator:SOURce)

<b>COMMAND SYNTAX</b>	SENSe:]ROSCillator:SOURce <src>	full
	ROSCillator:SOURce <src>	minimum
<b>QUERY SYNTAX</b>	SENSe:]ROSCillator:SOURce?	full
	ROSCillator:SOURce?	minimum

Sets or queries the unit 10 MHz clock reference clock source. The selected 10 MHz reference source is used to create the sample clock and other acquisition related time references. The following considerations apply when using the SENSe:ROSCillator:SOURce command:

- The unit default reference clock is the on-board 2.5 ppm temperature compensated oscillator, (TCXO).
- The VXIbus CLK10, 10 MHz reference signal may be selected to improve timing consistency across multiple VXIbus devices.
- When the reference source is changed, the unit will require approximately 50 msec to re-establish lock for other unit reference sources derived from the 10 MHz reference.

<b>COMMAND PARAMETERS</b>	<src>	desired reference source
		INTernal      unit on-board 10 MHz reference VXI              VXIbus CLK10 10 MHz reference
<b>RESPONSE PARAMETERS</b>	<src>	current reference source
		INT              unit on-board 10 MHz reference VXI              VXIbus CLK10 10 MHz reference
<b>DEFAULT VALUE</b>	INT	
<b>EXAMPLES</b>	ROSC:SOUR VXI	select the VXIbus 10 MHz reference
	ROSC:SOUR?	query current reference source, example response: INT
<b>STATUS EVENTS</b>	-141	invalid reference source name selected, command was ignored
<b>AFFECTIVITY</b>	None	
<b>RELATED COMMANDS</b>	None	

### 1.3.70 Sense Sweep Mode (SENSe:SWEep:MODE)

<b>COMMAND SYNTAX</b>	[SENSe:]SWEep:MODE <sw>	full
	SWEep:MODE <sw>	minimum
<b>QUERY SYNTAX</b>	[SENSe:] SWEep:MODE?	full
	SWEep:MODE?	minimum

Sets or queries the sweep subsystem triggering mode. The following considerations apply when using the sweep mode command:

- When AUTO mode is selected, the sweep subsystem will wait for a trigger event for a maximum of 100 msec before forcing a trigger event.
- When NORM mode is selected the sweep subsystem will wait indefinitely for a trigger event
- A trigger event may be forced using the \*TRG command

<b>COMMAND PARAMETERS</b>	<sw>	desired triggering mode AUTO wait a maximum of 100 msec for a trigger event NORM wait indefinitely for a trigger event
<b>RESPONSE PARAMETERS</b>	<sw>	desired smoothing filter state
<b>DEFAULT VALUE</b>	AUTO	
<b>EXAMPLES</b>	SWE:MODE NORM	wait indefinitely for a trigger event
	SWE:MODE?	query trigger mode, example response: NORM (or wait indefinitely for a trigger event)
<b>STATUS EVENTS</b>	-141	invalid triggering mode name selected, command was ignored
<b>AFFECTIVITY</b>	None	
<b>RELATED COMMANDS</b>	*TRG	

### 1.3.71 Sense Sweep Offset Time (SENSe:SWEEp:OFFSet:TIME)

<b>COMMAND SYNTAX</b>	[SENSe]:SWEEp:OFFSet:TIME <time>	full
	SWEEp:OFFSet:TIME <time>	minimum
<b>QUERY SYNTAX</b>	[SENSe]:SWEEp:OFFSet:TIME?	full
	SWEEp:OFFSet:TIME?	minimum

Sets or queries the time between the sweep offset reference and the trigger event. The following considerations apply when using the sweep offset time command:

- the offset time is the time between the trigger event and the offset reference, refer to the SENSe:SWEEp:OREFERENCE:LOCation command for more information on the offset reference
- positive offset times move the offset reference after the trigger event
- negative offset times move the offset reference before the trigger event
- the time of the last sample taken may be calculated as:  

$$\text{Last Sample Time} = \text{offset time} + \text{sweep time} * (100 - \text{offset reference location})/100$$
- the most positive offset time allowed will result in a Last Sample Time of 656 seconds
- the most negative offset time allowed will result in a Last Sample Time of 0 seconds
- all channels use the same sweep offset time

<b>COMMAND PARAMETERS</b>	<time>	desired sweep offset time
<b>RESPONSE PARAMETERS</b>	<time>	current sweep offset time
<b>DEFAULT VALUE</b>	0.0	
<b>EXAMPLES</b>	SWEE:OFFS:TIME 1 MS	move the offset reference 1 msec after the trigger event
	SWEE:OFFS:TIME?	query offset time, example response: 1.0E-3 (or move the offset reference 1 msec after the trigger event)
<b>STATUS EVENTS</b>	-222	attempted to set offset time to an illegal value, value was coerced in range
<b>AFFECTIVITY</b>	None	
<b>RELATED COMMANDS</b>	SWEEp:OREFERENCE:LOCation	

### 1.3.72 Sense Sweep Offset Reference Location (SENSe:SWEep:OREFERENCE:LOCation)

<b>COMMAND SYNTAX</b>	[SENSe]:SWEep:OREFERENCE:LOCation <loc>	full
	SWEep:OREFERENCE:LOCation <loc>	minimum
<b>QUERY SYNTAX</b>	[SENSe]:SWEep:OREFERENCE:LOCation?	full
	SWEep:OREFERENCE:LOCation?	minimum

Sets or queries the waveform record offset reference location. The following considerations apply when using the offset reference location command:

- The waveform offset reference can be considered a “handle” on the record. The waveform offset reference location is the location of the handle relative to the start of the waveform record.
- The offset reference is used by the SENSe:SWEep:OFFSet:TIME to move the record relative to the trigger event.
- An offset reference value of 0.0 places the handle at the first point of the record; a value of 50.0 selects the mid point; and a value of 100.0 selects the last point.
- When the offset time is set to 0.0, a reference location of 0.0 will place the trigger event at the waveform record start, a reference location of 100.0 will place the trigger event at the waveform record end
- All captured waveforms use the same offset reference

<b>COMMAND PARAMETERS</b>	<loc>	desired offset reference location 0.0 to 100.0      reference location in percent
<b>RESPONSE PARAMETERS</b>	<loc>	current offset reference location in percent
<b>DEFAULT VALUE</b>	50.0	
<b>EXAMPLES</b>	:SWE:OREF:LOC 25	place the reference 25% of the way from the record start
	:SWE:OREF:LOC?	query current offset reference location example response: 2.5E+1
<b>STATUS EVENTS</b>	-222	attempted to set offset reference location to an illegal value, the value was coerced in range
<b>AFFECTIVITY</b>	None	
<b>RELATED COMMANDS</b>	SWEep:OFFSet:TIME	

# 1

## 1.3.73 Sense Sweep Points (SENSe:SWEep:POINts)

<b>COMMAND SYNTAX</b>	[SENSe]:SWEep:POINts <pts>	full
	SWEep:POINts <pts>	minimum
<b>QUERY SYNTAX</b>	[SENSe]:SWEep:POINts?	full
	SWEep:POINts?	minimum

Sets or queries the number of samples in a waveform record. The range of points varies with the size of the installed digitizer memory. The following considerations apply when using the sweep points command:

- The minimum record length is 256 data points
- The maximum record length is the size of the digitizer memory, refer to the SYSTem:MEMory command to determine the digitizer memory size.. The maximum record size is further limited by the number of active channels.
- When in 2 channel modes, each channel may use up to the full digitizer memory size.
- When in 4 channel mode, each channel may use up to ½ of the full digitizer memory size. Refer to the SENSe:CHANnel:MODE for more information on selecting the number of active channels.
- The selected record length must be a multiple of 2, i.e. 256, 258, 260, ...
- Because of the finite number of sample rates, not all points -to-time ratios are valid. In the case of a ratio resulting in an invalid sample rate, the sweep time is increased until a valid ratio is achieved and a warning will be issued. Refer to The Installation, Operation. Programming, & Diagnostics Manual for a list of valid sample rates.
- Use the SENSe:SWEep:TINterval? query to read the current sample interval in seconds per sample. Use the SENSe:SWEep:SRATe? query to read the current sample rate in samples per second.
- If both sweep points and time are to be changed, set the sweep points before setting the sweep time to prevent large changes in the requested sweep time, refer to the SENSe:SWEep:TIME command for information on setting the sweep time.
- All active channels share the same record length setting.
- The waveform record size directly affects the number of waveforms that may be captured in a multiple record capture sequence, refer to the :TRIGger:COUNt command for more information
- Large changes in sweep points will often result in coercion warnings because the sample rate or sweep time is forced out of range.

<b>COMMAND PARAMETERS</b>	<pts>	desired waveform record size
		256 to N where N is the digitizer memory size
	MAXimum	set the record size to the available digitizer memory size
	MINimum	set the record size to 256
<b>RESPONSE PARAMETERS</b>	<pts>	current waveform record size
<b>DEFAULT VALUE</b>	1024	

---

<b>EXAMPLES</b>	SWE:POIN MIN	set the record size to 256
	SWE:POIN?	query the current record size, example response: 256
<b>STATUS EVENTS</b>	-222	attempted to set record length to an illegal value, the value was coerced in range
<b>AFFECTIVITY</b>		When setting the number of points, the sweep time will be adjusted to provide a valid sample rate.
<b>RELATED COMMANDS</b>	SENSe:SWEep:TIME	
	SENSe:SWEep:TINTerval?	
	SENSe:SWEep:SRATe?	

### 1.3.74 Sense Sweep Sample Rate (SENSe:SWEep:SRATe)

**COMMAND SYNTAX** n/a

<b>QUERY SYNTAX</b>	[SENSe]:SWEep:SRATe?	full
	SWEep:SRATe?	minimum

Queries the unit sample rate. The sample interval is defined as the ratio of the sweep points to the sweep time and is given in samples per second.

**COMMAND** n/a

**PARAMETERS**

<b>RESPONSE PARAMETERS</b>	<rate>	current sample rate in samples per second
----------------------------	--------	-------------------------------------------

**DEFAULT VALUE** 1.0E+8

**EXAMPLES** SWE:SRAT? query the current sample rate, example response: 1.0E+8

**STATUS EVENTS** None

**AFFECTIVITY** None

**RELATED COMMANDS** SENSe:SWEep:POINts

SENSe:SWEep:TIME

### 1.3.75 Sense Sweep Time (SENSe:SWEEp:TIME)

<b>COMMAND</b>	[SENSe]:SWEEp:TIME <time>	full
<b>SYNTAX</b>	SWEEp:TIME <time>	minimum
<b>QUERY SYNTAX</b>	[SENSe]:SWEEp:TIME?	full
	SWEEp:TIME?	minimum

Sets or queries the time span or duration of the waveform acquisition gate. The following considerations apply when using the sweep time command:

- the minimum sweep time is 256 nsec, (minimum record size) / (maximum sample rate)
- the maximum sweep time is 656 sec, defined by the waveform acquisition circuitry.
- The minimum and maximum sweep times are further limited by the selected record size.
- Because of the finite number of sample rates, not all points -to-time ratios are valid. In the case of a ratio resulting in an invalid sample rate, the sweep time is increased until a valid ratio is achieved and a warning will be issued. Refer to The Installation, Operation, Programming, & Diagnostics Manual for a list of valid sample rates.
- Use the SENSe:SWEEp:TINteRval? query to read the current sample interval in seconds per sample. Use the SENSe:SWEEp:SRATE? query to read the current sample rate in samples per second.
- If both sweep points and time are to be changed, set the sweep points before setting the sweep time to prevent large changes in the requested sweep time, refer to the SENSe:SWEEp:POINts command for information on setting the sweep points.
- All channels share the same sweep time span
- Large changes in sweep time will often result in coercion warnings because the sample rate is forced out of range.

<b>COMMAND PARAMETERS</b>	<time>	desired sweep time
	number	sweep time is seconds
	MAXimum	set the sweep time to its maximum value as defined by the current record size and available sample rates
	MINimum	set the sweep time to its minimum value as defined by the current record size and available sample rates
<b>RESPONSE PARAMETERS</b>	<time>	current sweep time
<b>DEFAULT VALUE</b>	1.024E-6	
<b>EXAMPLES</b>	SWEE:TIME 10 US	set the sweep time to 10 usec
	SWEE:TIME?	query current sweep time, example response: 1.0e-5 (or 10 usec)
<b>STATUS EVENTS</b>	-222	attempted to set time span to an illegal value, the value was coerced in range
<b>AFFECTIVITY</b>		Changing SWEEp:TIME will sometimes result in sample rate changes.
<b>RELATED COMMANDS</b>	SENSe:SWEEp:POINts	
	SENSe:SWEEp:TINteRval?	
	SENSe:SWEEp:SRATE?	

### 1.3.76 Sense Sweep Time Interval (SENSe:SWEep:TINTerval?)

**COMMAND SYNTAX** n/a

<b>QUERY SYNTAX</b>	[SENSe]:SWEep:TINTerval?	full
	SWEep:TINTinterval?	minimum

Queries the time interval between acquired data points. The sample interval is defined as the ratio of the sweep time to the sweep points and is given in seconds per sample.

**COMMAND PARAMETERS** n/a

<b>RESPONSE PARAMETERS</b>	<interval>	current sampling interval in seconds per sample
----------------------------	------------	-------------------------------------------------

**DEFAULT VALUE** 1.0E-8

<b>EXAMPLES</b>	SWE:TINT?	query the current sample interval, example response: 1.0E-8 (or 10 nsec)
-----------------	-----------	--------------------------------------------------------------------------

**STATUS EVENTS** None

**AFFECTIVITY** None

**RELATED COMMANDS**

- SENSe:SWEep:POINts
- SENSe:SWEep:TIME
- SENSe:SWEep:SRATe

### 1.3.77 Sense Voltage Range Offset (SENSe:VOLTage:RANGe:OFFSet?)

<b>COMMAND</b>	n/a	
<b>SYNTAX</b>		
<b>QUERY SYNTAX</b>	[SENSe:]VOLTage<n>:RANGe:OFFSet?	full
	VOLTage<n>:RANGe:OFFSet?	minimum

Queries the specified input channel voltage offset. The input offset voltage for the ZT432VXI is fixed at 0.0 V. Refer to the SENSe:VOLTage:RANGe:PTPeak command to set the input channel voltage range.

<b>COMMAND</b>	<n>	input channel to read
<b>PARAMETERS</b>		1 to 4 input channel number
<b>RESPONSE</b>	<offs>	input channel offset voltage
<b>PARAMETERS</b>		0.0 fixed offset voltage
<b>DEFAULT VALUE</b>	0.0E+0	
<b>EXAMPLES</b>	VOLT:RANG:OFFS?	query input offset voltage, response: 0.0 (or 0.0 V offset)
<b>STATUS EVENTS</b>	-241	attempt to read from a non-existent input channel
<b>AFFECTIVITY</b>	None	
<b>RELATED COMMANDS</b>	SENSe:VOLTage:RANGe:PTPeak	

### 1.3.78 Sense Voltage Range Peak-to-Peak (SENSe:VOLTage:RANGe:PTPeak)

<b>COMMAND SYNTAX</b>	[SENSe:]VOLTage<n>:RANGe:PTPeak <volt>	full
	VOLTage<n>:RANGe:PTPeak <volt>	minimum
<b>QUERY SYNTAX</b>	[SENSe:]VOLTage<n>:RANGe:PTPeak?	full
	VOLTage<n>:RANGe:PTPeak?	minimum

Sets or queries the specified input channel full-scale voltage range. The following considerations apply when using the SENSe:VOLTage:RANGe:PTPeak command:

- the allowed values for input channel full-scale range are dependent upon the input impedance setting, refer to The Installation, Operation, Programming, & Diagnostics Manual for the valid input range settings and the SENSe:INPut:IMPedance command for information on setting impedance.
- The input channel offset voltage is fixed at 0.0 V.
- External signal attenuation can be accounted for using the SENSe:INPut:ATTenuation command.

<b>COMMAND PARAMETERS</b>	<n>	<ul style="list-style-type: none"> <li>• input channel number, defaults to 1 if omitted           <ul style="list-style-type: none"> <li>○ 1 to 4 input channel number</li> </ul> </li> </ul>
	<volt>	<ul style="list-style-type: none"> <li>• desired full-scale voltage range           <ul style="list-style-type: none"> <li>○ 10, 5, 2, 1, 0.5, 0.25, 0.2, or 0.1 50 Ω input impedance</li> <li>○ 50, 25, 10, 5, 2.5, 1.25, 1, or 0.5 1 MΩ input impedance</li> <li>○ MAXimum maximum range for the current input impedance</li> <li>○ MINimum minimum range for the current input impedance</li> </ul> </li> </ul>
<b>RESPONSE PARAMETERS</b>	<volt>	current full-scale voltage range
<b>DEFAULT VALUE</b>	5.0E+0	
<b>EXAMPLES</b>	VOLT:RANG:PTP 1.0	set the input voltage to 1.0 Vpp
	VOLT:RANG:PTP?	query current input voltage range, example response: 1.0e+0 (or 1.0 Vpp input range)
<b>STATUS EVENTS</b>	-222	attempt to set the input voltage range to an illegal value, value coerced in range
	-241	attempt to set the input voltage range for a non-existent channel, command was ignored)
<b>AFFECTIVITY</b>	None	
<b>RELATED COMMANDS</b>	SENSe:INPut:ATTenuation	
	SENSe:INPut:IMPedance	

---

SENSe:VOLTage:RANGe:OFFSet

# 1

## 1.3.79 Status Operation Event (STATus:OPERation:EVENT?)

<b>COMMAND SYNTAX</b>	n/a	
<b>QUERY SYNTAX</b>	STATus:OPERation[:EVENT]?	full
	STATus:OPERation?	minimum

Queries the contents of the Operation Event Status Register. The Operation Event Status Register identifies unit processes that have been run, such as waveform acquisition. Refer to The Installation, Operation, Programming, & Diagnostics Manual for more information on the Operation Status Event Register. The following considerations apply when using the operation event status query:

- the Operation Status Event Register records the history of the processes that have been run since the previous operation event status query, use the :STATus:OPERation:CONDition? query to identify currently running processes
- the STATus:OPERation:EVENT? query clears the Operation Status Event Register after returning the current register contents
- Operation Status Event Register bits may be summarized in the Status Byte, refer to the STATus:OPERation:ENABle command for more information.
- In order to identify which processes have run between two times, ex. acquisition start and later status check, the Operation Status Event Register must be cleared by reading it before the processes are run.

<b>COMMAND PARAMETERS</b>	n/a	
<b>RESPONSE PARAMETERS</b>	<status>	current operation event status
<b>DEFAULT VALUE</b>	n/a	
<b>EXAMPLES</b>	STAT:OPER?	Query current operation event status, example response: 112 (or waveform acquisition, wait for trigger, and wait for arm processes have been run, indicating that an INITiate command was issued)
<b>STATUS EVENTS</b>	None	
<b>AFFECTIVITY</b>		The STATus:OPERation:EVENT? query clears the Operation Status Event Register after reading it.
<b>RELATED COMMANDS</b>	STATus:OPERation:CONDition? STATus:OPERation:ENABle	

### 1.3.80 Status Operation Condition (STATus:OPERation:CONDition?)

**COMMAND SYNTAX** n/a

**QUERY SYNTAX** STATus:OPERation:CONDition?

Queries the contents of the Operation Status Condition Register. The Operation Status Condition Register identifies currently running processes, such as waveform acquisition. Refer to The Installation, Operation, Programming, & Diagnostics Manual for more information on the Operation Status Condition Register. The following considerations apply when using the operation condition status query:

- the Operation Status Condition Register identifies current running processes, use the :STATus:OPERation:EVENT? query to identify a history of which processes have run since the last operation event status check
- the STATus:OPERation:CONDition? query does not clear the Operation Status Condition Register

**COMMAND PARAMETERS** n/a

**RESPONSE PARAMETERS** <status> current operation status

**DEFAULT VALUE** n/a

**EXAMPLES** STAT:OPER:COND? query the current operation status, example response: 112 (or acquiring waveforms, waiting for trigger, and waiting for arm processes are running, indicating that an INITiate command was issued)

**STATUS EVENTS** None

**AFFECTIVITY** None

**RELATED COMMANDS** STATus:OPERation:EVENT?

### 1.3.81 Status Operation Enable (STATus:OPERation:ENABle)

**COMMAND SYNTAX** STATus:OPERation:ENABle <mask>

**QUERY SYNTAX** STATus:OPERation:ENABle?

Sets or queries the contents of the Operation Status Enable Register. The Operation Status Enable Register enables the reporting of operation events to the Status Byte. Refer to The Installation, Operation, Programming, & Diagnostics Manual for more information on the Operation Status Enable Register. The following considerations apply when using the operation event reporting enable command:

- The Operation Status Enable Register is a bit mask that allows selected operation status events to be reported to the Status Byte.
- Only low to high (inactive to active) Operation Status Event Register bit transitions are reported, refer to the STATus:OPERation:EVENT? query.
- Operation status events are report in bit 7 of the Status Byte, refer to the \*STB? query.
- The STATus:PRESet sets all register bits to zero (0) which disables all operation event reporting.

<b>COMMAND PARAMETERS</b>	<mask>	desired event reporting mask 0 to 65535 bit 15 is ignored
<b>RESPONSE PARAMETERS</b>	<mask>	current event reporting mask 0 to 32767 bit 15 is always 0
<b>DEFAULT VALUE</b>	0	
<b>EXAMPLES</b>	STAT:OPER:ENAB 112	enable reporting of event register bits 4, 5, and 6
	STAT:OPER:ENAB?	query event reporting mask, example response: 112
<b>STATUS EVENTS</b>	None	
<b>AFFECTIVITY</b>	None	
<b>RELATED COMMANDS</b>	*STB?	
	STATus:OPERation:EVENT?	
	STATus:PRESet	

**1.3.82 Status Preset (STATus:PRESet)**

**COMMAND** STATus:PRESet

**SYNTAX**

**QUERY SYNTAX** n/a

Disables event reporting from the Standard Event Status, Operation Status Event, and Questionable Status Event Registers, i.e. the reporting enable registers are set to 0. Refer to the \*ESE, STATus:OPER:ENABLE, and STATus:QUEStionable:ENABLE commands for more information. Current condition and event register contents are not affected.

**COMMAND** None

**PARAMETERS**

**RESPONSE** None

**PARAMETERS**

**DEFAULT VALUE** n/a

**EXAMPLES** STAT:PRES disable event reporting

**STATUS EVENTS** None

**AFFECTIVITY** The Operation Status Enable, Questionable Status Enable, and Standard Event Enable Registers are set to 0.

**RELATED** STATus:OPERation:ENABLE  
**COMMANDS** STATus:QUEStionable:ENABLE  
 \*ESE

### 1.3.83 Status Questionable Event (STATus:QUEStionable?)

**COMMAND SYNTAX** n/a

**QUERY SYNTAX** STATus: QUEStionable [:EVENT]? full  
STATus: QUEStionable? minimum

Queries the Questionable Status Event Register. The Questionable Event Status Register identifies unit processes that have completed with questionable results, such as self test errors. Refer The Installation, Operation, Programming, & Diagnostics Manual for more information on the Questionable Status Event Register. The following considerations apply when using the questionable event status query:

- the Questionable Status Event Register records the history of the questionable process results generated since the previous questionable event status query, use the :STATus:QUEStionable:CONDition? query to identify the current questionable status results.
- the STATus:QUEStionable:EVENT? query clears the Questionable Status Event Register after returning the current register contents.
- Questionable Status Event Register bits may be summarized in the Status Byte, refer to the STATus:QUEStionable:ENABLE command for more information.
- In order to identify questionable results from a particular process, the Questionable Status Event Register must be cleared by reading it before the processes are run.

**COMMAND PARAMETERS** n/a

**RESPONSE PARAMETERS** <status> current questionable event status

**DEFAULT VALUE** n/a

**EXAMPLES** STAT:QUES? query questionable event status, example response: 256 (or questionable self-calibration results)

**STATUS EVENTS** None

**AFFECTIVITY** The STATus:QUEStionable:EVENT? query clears the Questionable Status Event Register after reading it.

**RELATED COMMANDS** STATus:QUEStionable:CONDition?  
STATus:QUEStionable:ENABLE

**1.3.84 Status Questionable Condition (STATus:QUEStionable:CONDition?)**

**COMMAND** n/a  
**SYNTAX**  
**QUERY SYNTAX** STATus:QUEStionable:CONDition?

Queries the contents of the Questionable Status Condition Register. The Questionable Status Condition Register identifies current questionable results from running processes, such as self-test. Refer to The Installation, Operation, Programming, & Diagnostics Manual for more information on the Questionable Status Condition Register. The following considerations apply when using the questionable condition status query:

- the Questionable Status Condition Register identifies current questionable results from running processes, use the :STATus:QUEStionable:EVENT? query to identify which questionable results generated since the last questionable event status check.
- the STATus:QUEStionable:CONDition? query does not clear the Questionable Status Condition Register

**COMMAND** n/a  
**PARAMETERS**  
**RESPONSE** <status> current questionable status  
**PARAMETERS**  
**DEFAULT VALUE** n/a

**EXAMPLES** STAT:QUES:COND? query the current questionable status, example response: 256 (or questionable self-calibration results)

**STATUS EVENTS** None  
**AFFECTIVITY** None

**RELATED** STATus:QUEStionable:EVENT?  
**COMMANDS**

### 1.3.85 Status Questionable Enable (STATus:QUEStionable:ENABle)

**COMMAND SYNTAX** STATus:QUEStionable:ENABle <mask>

**QUERY SYNTAX** STATus:QUEStionable:ENABle?

Sets or queries the contents of the Questionable Status Enable Register. The Questionable Status Enable Register enables the reporting of questionable events to the Status Byte. Refer to The Installation, Operation, Programming, & Diagnostics Manual for more information on the Questionable Status Enable Register. The following considerations apply when using the questionable event reporting enable command:

- The Questionable Status Enable Register is a bit mask that allows selected operation status events to be reported to the Status Byte.
- Only low to high (inactive to active) Questionable Status Event Register bit transitions are reported, refer to the STATus:QUEStionable:EVENT? query.
- Questionable status events are report in bit 3 of the Status Byte, refer to the \*STB? query.
- The STATus:PRESet sets all register bits to zero (0) which disables all questionable event reporting.

<b>COMMAND PARAMETERS</b>	<mask>	desired event reporting mask 0 to 65535 bit 15 is ignored
<b>RESPONSE PARAMETERS</b>	<mask>	current event reporting mask 0 to 32767 bit 15 is always 0
<b>DEFAULT VALUE</b>	0	
<b>EXAMPLES</b>	STAT:QUES:ENAB 1	enable reporting of event register bit 0
	STAT:QUES:ENAB?	query event reporting mask, example response: 1 (or report unit PLL un-lock events)
<b>STATUS EVENTS</b>	None	
<b>AFFECTIVITY</b>	None	
<b>RELATED COMMANDS</b>	*STB?	
	STATus: QUEStionable:EVENT?	
	STATus:PRESet	

### 1.3.86 System Error All (SYSTem:ERRor:ALL?)

**COMMAND SYNTAX** n/a

**QUERY SYNTAX** SYSTem:ERRor:ALL?

Returns the list of all events stored in the unit Status Queue. The following considerations apply when using the status queue query:

- each status event message contains the event code, refer to The Installation, Operation, Programming, & Diagnostics Manual, Appendix B for event code descriptions.
- reading the Status Queue when it is empty does not cause an error.
- event messages are returned in a comma separated list
- use the SYSTem:ERRor:NEXT? query to read one queue entry
- use the SYSTem:ERRor:COUNT? query to read the number of queue entries
- this query will result in the Status Queue being cleared
- the Status Queue is also cleared upon power up and receipt of the \*CLS command

**COMMAND PARAMETERS** n/a

**RESPONSE PARAMETERS** <code>,<code>,... list of event code separated by commas  
 > 0 informational codes  
 0 queue was empty  
 < 0 error and warning codes

**DEFAULT VALUE** n/a

**EXAMPLES** SYST:ERR:ALL? query all entries from the Status Queue, example response:  
 -113,-222,-141

**STATUS EVENTS** -350 the Status Queue has overflowed

**AFFECTIVITY** This command will clear the status queue after returning its contents.

**RELATED COMMANDS** SYSTem:ERRor:COUNT?  
 SYSTem:ERRor:NEXT?  
 \*CLS

# 1

---

## 1.3.87 System Error Count (SYSTem:ERRor:COUNt?)

**COMMAND SYNTAX** n/a

**QUERY SYNTAX** SYSTem:ERRor:COUNt?

Returns the number of entries in the Status Queue.

**COMMAND  
PARAMETERS** n/a

**RESPONSE  
PARAMETERS** <cnt> current number of entries in the Status Queue

**DEFAULT VALUE** n/a

**EXAMPLES** SYST:ERR:COUN? query the number of entries in the Status Queue, example  
response: 3 (or there are 3 entries in the queue)

**STATUS EVENTS** None

**AFFECTIVITY** None

**RELATED  
COMMANDS** None

### 1.3.88 System Error Next (SYSTem:ERRor:NEXT?)

**COMMAND SYNTAX** n/a

<b>QUERY SYNTAX</b>	SYSTem:ERRor[:NEXT]?	full
	SYSTem:ERRor?	minimum

Queries the next entry from the unit Status Queue. The following considerations apply when using the status queue query:

- the status event message contains the event code, refer to The Installation, Operation, Programming, & Diagnostics Manual, Appendix B for event code description
- reading the Status Queue when it is empty does not cause an error.
- reading an event removes it from the queue
- use the SYSTem:ERRor:ALL? query to read all queue entries at once
- use the SYSTem:ERRor:COUNT? query to read the number of queue entries
- the Status Queue is also cleared upon power up, receipt of the \*CLS command, or reading of the last queue entry

**COMMAND PARAMETERS** n/a

<b>RESPONSE PARAMETERS</b>	<code>	event code
		> 0 informational codes
		0 queue was empty
		< 0 error and warning codes

**DEFAULT VALUE** n/a

<b>EXAMPLES</b>	:SYST:ERR?	query the next Status Queue entry, example response: -113 (or unrecognized command)
-----------------	------------	-------------------------------------------------------------------------------------

<b>STATUS EVENTS</b>	-350	the Status Queue has overflowed
----------------------	------	---------------------------------

**AFFECTIVITY** None

**RELATED COMMANDS**

SYSTem:ERRor:ALL?

SYSTem:ERRor:COUNT?

\*CLS



### 1.3.90 System Version (SYSTem:VERSion?)

**COMMAND SYNTAX** n/a

**QUERY SYNTAX** SYSTem:VERSion?

Returns the SCPI version supported by the waveform analyzer.

**COMMAND PARAMETERS** n/a

**RESPONSE PARAMETERS** <vers> SCPI version supported by the unit  
1997.0

**DEFAULT VALUE** n/a

**EXAMPLES** SYST:VERS? query SCPI version, example response: 1997.0

**STATUS EVENTS** None

**AFFECTIVITY** None

**RELATED COMMANDS** None

---

### 1.3.91 Test Report (TEST:REPort?)

**COMMAND SYNTAX** n/a

**QUERY SYNTAX** TEST:REPort?

Queries the results of the most recent self test. Refer to The Installation, Operation, Programming, & Diagnostics Manual, Appendix B for information on self-test report codes. A self-test cycle may be initiated using the IEEE-488 \*TST? command.

**COMMAND PARAMETERS** n/a

**RESPONSE PARAMETERS** <code> calibration results report code  
> 0 information codes  
0 successful self-test  
< 0 self-test failure

**DEFAULT VALUE** n/a

**EXAMPLES** TEST:REP? query the calibration results code, example response 0 (or successful calibration)

**STATUS EVENTS** None

**AFFECTIVITY** None

**RELATED COMMANDS** \*TST?

### 1.3.92 Trace Address (TRACe:ADDRess?)

**COMMAND** n/a  
**SYNTAX**  
**QUERY SYNTAX** TRACe:ADDRess? <src>,<num>

Queries the address and size of a waveform source in VXIbus A32 space. The following considerations apply when using the trace address query:

- the TRACe:ADDRess? query returns the A32 address and waveform size in bytes
- reference waveforms and waveform math outputs permanently reside in A32 accessible memory
- input channel waveforms must be moved from digitizer memory into A32 space, this can be accomplished using the TRACe:DOWNload:IMMediate command or by enabling automatic download after waveform capture using the TRACe:DONWload:STATe setting.
- The address returned by this query defines the waveform start address. The associated waveform preamble may be accessed starting at the address 40 bytes before the waveform start address.
- The <num> parameter provides a means to access records from a multiple waveform capture sequence. If present for non-input channel sources, <num> must be set to 1.
- The <src> and <num> selections are position sensitive.

<b>COMMAND PARAMETERS</b>	<src>	waveform source designator CALC1 or CALC2 waveform math sources INP1 to INP4 input signal waveform sources REF1 to REF6 reference waveform sources
	<num>	waveform number designator 1 to N waveform number for multiple input channel waveform capture sequences 1 non-input channel waveform sources
<b>RESPONSE PARAMETERS</b>	<addr>,<size>	waveform A32 start address and size in bytes
<b>DEFAULT VALUE</b>	n/a	
<b>EXAMPLES</b>	TRAC:ADDR? INP1,1	read the input 1, waveform 1, A32 start address and size, example response: 40,1044 (or the waveform starts at byte offset 40 and is 1044 bytes long)
<b>STATUS EVENTS</b>	-141	attempted to access data from an invalid waveform source name, command was ignored
	-222	invalid waveform number selected, value was coerced in range
<b>AFFECTIVITY</b>		
<b>RELATED COMMANDS</b>	TRACe:DOWNload:IMMediate TRACe:DOWNload:STATe	

### 1.3.93 Trace Data (TRACe:DATA)

<b>COMMAND SYNTAX</b>	TRACe[:DATA] <ref>	full
	TRACe <ref>	minimum
<b>QUERY SYNTAX</b>	TRACe[:DATA]? <src>,<num>	full
	TRACe? <src>,<num>	minimum

Reads or writes waveform data to the unit using the word-serial interface. The following considerations apply when using the word-serial waveform load command

- The command form writes waveform data to the unit reference waveform store. Two types of reference waveform store are available. Refer to The Installation, Operation, Programming, & Diagnostics Manual for more information on reference waveforms.
 

non-volatile storage waveforms	REF1 to REF4	REF5 and REF6
volatile storage waveforms		
- Reference waveforms written using the command form will overwrite previous reference waveform information.
- Reference waveform preamble information may be written over the word-serial interface using the TRACe:PREamble command.
- The query form reads waveform data from the unit. Three waveform sources are available:
 

Input channel waveforms, INP1 to INP4
Reference waveforms, REF 1 to REF 6, and
Waveform math functions, CALC1 and CALC2
- Waveform preamble information may be read over the word-serial interface using the TRACe:PREamble? query.
- The <num> parameter provides a means to access records from a multiple waveform capture sequence. If present for non-input channel sources, <num> must be set to 1.
- The <src> and <num> selections are position sensitive.
- Word-serial waveform transfers use the low address area of the unit A32 memory, waveform data in this area will be overwritten.
- For large waveforms, word serial waveform transfers are a slow process, but have the advantage of being broadly applicable. Word serial transfers are especially useful when using chassis controllers without A32 access capability.
- For faster waveform transfers, refer to the The Installation, Operation, Programming, & Diagnostics Manual for register based VXIbus A32 waveform loading information.
- Waveform data is always transferred as 16 bit integer values. Refer to the The Installation, Operation, Programming, & Diagnostics Manual for information on encoding and decoding waveform time and voltage values.
- The 16 bit integer values may be transferred high byte first or low byte first, refer to the FORMat:BORDER command for byte order setting.
- The TRACe:DATA command and query cause the word-serial interface to change from SCPI mode to binary transfer mode, no further SCPI commands or queries are allowed until the waveform transfer is complete. Refer to The Installation, Operation, Programming, & Diagnostics Manual for more information on waveform transfers.

<b>COMMAND PARAMETERS</b>	<p>&lt;ref&gt;</p> <p>&lt;src&gt;</p> <p>&lt;num&gt;</p>	<p>destination reference waveform designator, this designator must be followed by the waveform data as described in The Installation, Operation, Programming, &amp; Diagnostics Manual</p> <p>REF1 to REF6      destination reference waveform</p> <p>waveform source designator</p> <p>CALC1 or CALC2      waveform math sources</p> <p>INP1 to INP4      input signal waveform sources</p> <p>REF1 to REF6      reference waveform sources</p> <p>waveform number designator</p> <p>1 to N      waveform number for multiple input</p> <p>sequences</p> <p>1      channel waveform capture</p> <p>sources      non-input channel waveform</p>
<b>RESPONSE PARAMETERS</b>	n/a	
<b>DEFAULT VALUE</b>	n/a	
<b>EXAMPLES</b>	<p>TRAC REF1 &lt;waveform data&gt;</p> <p>TRAC? INP1,3</p>	<p>write waveform data to reference waveform 1</p> <p>read the waveform data from input channel 1, waveform 3, example response: #42000&lt;2000 bytes of waveform data&gt;</p>
<b>STATUS EVENTS</b>	<p>-141</p> <p>-222</p> <p>-223</p>	<p>attempted to access an invalid waveform source, command was ignored</p> <p>invalid waveform number selected, value was coerced in range.</p> <p>attempted to send a data record larger than 33554432 points, download aborted.</p>
<b>AFFECTIVITY</b>		<p>The TRACe:DATA command and query cause the word-serial interface to enter its binary transfer mode, no further SCPI command interface reads or writes are allowed until the binary transfer is complete.</p> <p>The beginning of A32 memory is over-written by this command.</p>
<b>RELATED COMMANDS</b>	<p>FORMat:BORDER</p> <p>TRACe:DOWNload:IMMediate</p> <p>TRACe:PREamble</p>	

### 1.3.94 Trace Download Immediate (TRACe:DOWNload:IMMediate)

**COMMAND SYNTAX** TRACe:DOWNload[:IMMediate] <src>,<num> full  
 TRACe:DOWNload <src>,<num> minimum

**QUERY SYNTAX** n/a

Causes the unit to immediately copy a selected waveform from digitizer memory to VXIbus A32 accessible memory. The following considerations apply when using the trace copy command:

- New waveforms are not captured until the download is complete.
- The selected waveforms and associated preamble are copied to beginning of VXIbus A32 memory, the preamble will begin at byte address  $0_{16}$  and waveform data will begin at byte address  $40_{16}$ .
- The waveform size can be determined by multiplying the :SWEep:POINts value by 2.
- The previous low address A32 memory contents are over-written.
- Digitizer waveforms may be automatically transferred to A32 space after a capture using the TRACe:DOWNload:STATe command
- Word-serial waveform transfers, such as TRACe:DATA, overwrite the low address area of the unit A32 memory, waveform data in this area must be re-downloaded using the TRACe:DOWNload:IMMediate command before being accessed from A32 space.
- This command may be used when the unit is not capturing data, running self-test, or performing a calibration cycle.

<b>COMMAND PARAMETERS</b>	<src>	waveform source designator CALC1 or CALC2 waveform math sources INP1 to INP4 input signal waveform sources REF1 to REF6 reference waveform sources
	<num>	waveform number designator 1 to N waveform number for multiple input channel waveform capture sequences 1 non-input channel waveform sources
<b>RESPONSE PARAMETERS</b>	n/a	
<b>DEFAULT VALUE</b>	n/a	
<b>EXAMPLES</b>	TRAC:DOWN	copy captured waveforms to A32 memory
<b>STATUS EVENTS</b>	-222	invalid waveform number selected, value was coerced in range.
<b>AFFECTIVITY</b>	None	
<b>RELATED COMMANDS</b>	TRACe:ADDReSS	
	TRACe:DATA	
	TRACe:DOWNload:STATe	



# 1

## 1.3.96 Trace Flags State (TRACe:FLAGs:STATe)

**COMMAND SYNTAX** TRACe:FLAGs:STATe <sw>

**QUERY SYNTAX** TRACe:FLAGs:STATe?

Sets or queries digitizer flag reporting. The digitizer saves a set of status flags along with the sample data, these flags indicate trigger, start of data, and overflow status. The following considerations apply when using the flags reporting enable command:

- The digitizer data format is given below:

Bits 15 to 4	MSBs	12 bit signed integer waveform sample
Bit 3		digitizer input over-range condition, 1 = over-range
Bit 2		detected trigger event marker, 1 = trigger detected
Bit 1		raw sampled trigger, 1 = input signal above threshold
Bit 0	LSB	start-of-data marker, 1 = start of data
- For capture sequences with long delays between the trigger event and end of capture, the detected trigger event marker, bit 2, may be overwritten.
- For capture sequences with long delays between the arm event and end of capture, the start-of-data marker, bit 0, may be overwritten.
- Bits 3 and 1 are always valid.
- The status flags are only maintained for input channel signals
- When disabled, the status flags are replaced by signal bits resulting from enabled signal processing, such as smoothing, which may result in increased significant digits.
- All channels use the same flag reporting setting

**COMMAND PARAMETERS** <sw> desired digitizer flags reporting state  
1 or ON enable flag reporting  
0 or OFF disable flag reporting

**RESPONSE PARAMETERS** <sw> current digitizer flags reporting state  
1 flag reporting enabled  
0 flag reporting disabled

**DEFAULT VALUE** 0

**EXAMPLES** TRAC:FLAG:STAT ON enable flag reporting  
TRAC:FLAG:STAT? read the flag reporting setting, example response: 1  
(or flags reporting is enabled)

**STATUS EVENTS** -141 invalid flag reporting setting name, command was ignored

**AFFECTIVITY** When enabled, the status flags replace the lower 4 bits of the 16 bit data sample word.

**RELATED COMMANDS** None

### 1.3.97 Trace Preamble (TRACe:PREamble)

<b>COMMAND SYNTAX</b>	TRACe:PREamble <ref>,<preamble>	full
	TRACe:PREamble <ref>,<preamble>	minimum
<b>QUERY SYNTAX</b>	TRACe:PREamble <src>,<num>	full
	TRACe:PREamble? <src>,<num>	minimum

Reads or writes waveform preambles to the unit using the word-serial interface. The following considerations apply when using the TRACe:PREamble command

- <preamble> is an ordered, comma separated list of preamble values. The list of values and numeric formats are shown below. Refer to The Installation, Operation, Programming, & Diagnostics Manual for additional information.

format,type,count,xsize,xinc,xoffset,xref,ysize,yinc,yoffset,yref

where the fields are

format	waveform data format, fixed at 3 = 16-bit integer	u16
type	waveform type, 1 = normal waveform, 2 = averaged waveform, 3 = envelope waveform high/low pair	u16
count	number of waveform processed to generate output, number of waveform for multiple capture sequence or number of waveforms examined for averaged and enveloped waveforms	u16
xsize	number of samples in the waveform	u32
xinc	time interval between samples	f32
xoffset	time of first sample relative to the trigger event	f32
xref	trigger event timestamp	f32
ysize	number of waveforms in waveform record	u32
yinc	voltage interval between sample codes	f32
yoffset	voltage offset of waveform	f32
yref	waveform number in multiple waveform capture	u32

- The command form writes a waveform preamble to the unit reference waveform store. Two types of reference waveform store are available. Refer to The Installation, Operation, Programming, & Diagnostics Manual for more information on reference waveforms.
  - non-volatile storage waveforms REF1 to REF4
  - volatile storage waveforms REF5 and REF6
- Reference waveform preambles written using the command form will overwrite previous reference waveform information
- Waveform data for the reference waveforms may be written over the word-serial interface using the TRACe:DATA command.
- The query form reads a waveform preamble from the unit. Three waveform sources are available:
  - Input channel waveforms, INP1 to INP4
  - Reference waveforms, REF 1 to REF 6, and
  - Waveform math functions, CALC1 and CALC2
- Waveform data information for the waveform sources may be read over the word-serial interface using the TRACe:DATA? query.
- The <num> parameter provides a means to access records from a multiple waveform capture sequence. If present for non-input channel sources, <num> must be set to 1.

# 1

---

- The <src>, <num>, and <preamble> selections are position sensitive.

<b>COMMAND PARAMETERS</b>	<p>&lt;ref&gt;</p> <p>&lt;preamble&gt;</p> <p>&lt;src&gt;</p> <p>&lt;num&gt;</p>	<p>destination reference waveform designator, this designator must be followed by the waveform data as described in The Installation, Operation. Programming, &amp; Diagnostics Manual REF1 to REF6 destination reference waveform</p> <p>comma separated list of preamble values, see the description above and The Installation, Operation. Programming, &amp; Diagnostics Manual for more information on preamble values.</p> <p>waveform source designator  CALC1 or CALC2 waveform math sources  INP1 to INP4 input signal waveform sources  REF1 to REF6 reference waveform sources</p> <p>waveform number designator  1 to N waveform number for multiple input channel waveform capture sequences  1 non-input channel waveform sources</p>
<b>RESPONSE PARAMETERS</b>	<p>&lt;preamble&gt;</p>	<p>comma separated list of preamble values, see the description above and The Installation, Operation. Programming, &amp; Diagnostics Manual for more information on preamble values.</p>
<b>DEFAULT VALUE</b>	<p>n/a</p>	
<b>EXAMPLES</b>	<p>TRAC REF1,&lt;preamble&gt;</p> <p>TRAC? INP1,3</p>	<p>write preamble information to reference waveform 1</p> <p>read the preamble from input channel 1, waveform 3, example response: 3,1,1,1024,1.0e-8,-5.12e-6,3.2e-1,1,1,1.5259e-4,0.0e+0,1</p>
<b>STATUS EVENTS</b>	<p>-141</p> <p>-222</p>	<p>attempted to access data from an invalid waveform source name</p> <p>invalid waveform number selected, value was coerced in range.</p>
<b>AFFECTIVITY</b>	<p>None</p>	
<b>RELATED COMMANDS</b>	<p>TRACe:DATA</p>	

### 1.3.98 Trace Reference Copy (TRACe:REFerence:COpy)

**COMMAND SYNTAX** TRACe:REFerence<n>:COpy <src>,<num>

**QUERY SYNTAX** n/a

Copies a waveform from a selected source to a reference waveform. Both the waveform samples and preamble are copied. The following considerations apply when using the reference copy command:

- waveforms copied to REF1 to REF4 will be saved in non-volatile memory, waveforms for these references are limited to 32748 points, i.e. (waveform storage size – preamble size) or (32768 words – 20 words).
- Waveforms copied to REF 5 or REF6 will be saved in volatile memory, waveforms for these references are limited to the maximum SENSE:SWEp:POINts size.
- Previous contents of the reference waveform storage location are lost.

<b>COMMAND PARAMETERS</b>	<n>	reference waveform number 1 to 4 non-volatile waveforms 5 to 6 volatile waveforms
	<src>	waveform source designator CALC1 or CALC2 waveform math sources INP1 to INP4 input signal waveform sources REF1 to REF6 reference waveform sources
	<num>	waveform number designator 1 to N waveform number for multiple input channel waveform capture sequences 1 non-input channel waveform sources
<b>RESPONSE PARAMETERS</b>	n/a	
<b>DEFAULT VALUE</b>	n/a	
<b>EXAMPLES</b>	TRAC:REF3:COpy INP3,2	copy input channel 3, waveform 2 to reference channel 3
<b>STATUS EVENTS</b>	-141	Attempted to copy data from an invalid trace name, command was ignored
	-222	invalid waveform number selected, value was coerced in range.
	-241	attempted to save data to an invalid reference waveform, command was ignored
	-362	waveform too large for storage resource, waveform truncated
<b>AFFECTIVITY</b>	None	
<b>RELATED COMMANDS</b>	SENSe:SWEp:POINts	

### 1.3.99 Trace Time Stamp (TRACe:TIMestamp?)

**COMMAND SYNTAX** n/a

**QUERY SYNTAX** TRACe:TIMestamp? <num>

Queries the trigger event time stamp associated with a captured waveform. The time stamp is based on a free-running timer with 100 nsec resolution and 1 second rollover period. Time stamps are only available for analog input channels.

**COMMAND PARAMETERS** <num> waveform number, used to access waveforms in a multiple waveform capture sequence  
1 to N waveform number

**RESPONSE PARAMETERS** <time> waveform trigger event timestamp in seconds

**DEFAULT VALUE** n/a

**EXAMPLES** TRAC:TIM? 3 query the timestamp for waveform 3: 1.4532e-1

**STATUS EVENTS** -222 invalid waveform number selected, value was coerced in range.

**AFFECTIVITY** None

**RELATED COMMANDS** None

### 1.3.100 Trigger A Event Count (TRIGger:A:ECOUNT)

<b>COMMAND SYNTAX</b>	TRIGger[:A]:ECOUNT <cnt>	full
	TRIGger:ECOUNT <cnt>	minimum
<b>QUERY SYNTAX</b>	TRIGger[:A]:ECOUNT?	full
	TRIGger:ECOUNT?	minimum

Sets or queries the number of events that Trigger A must skip before it enables other waveform capture functions. The following considerations apply when using the trigger event count command:

- the event count affects the capture cycle as follows
  - arm
  - skip (event count) trigger events
  - on the next trigger event capture a waveform
  - if a multiple waveform capture cycle, re-start the event count and trigger detection
  - else, end the capture cycle and begin post-capture processing
- setting the event count to 0 disables trigger event counting
- the event counter uses the output of the trigger qualifier as event source, refer to the TRIGger:A:PATTERN,PULSE,SLOPE,SOURCE, and TYPE commands for information on configuring trigger qualifier parameters.
- when the event count is reached, Trigger A will
  - enable Trigger B detection, or if Trigger B is disabled, see TRIGger:B:STATE,
  - enable the sweep timer, see SWEep:OFFSet:TIME

<b>COMMAND PARAMETERS</b>	<cnt>	desired number of events to skip 0 to 65535
<b>RESPONSE PARAMETERS</b>	<cnt>	current number of events to skip
<b>DEFAULT VALUE</b>	0	
<b>EXAMPLES</b>	TRIG:ECO 2	skip 2 trigger events
	TRIG:ECO?	query the current number of events to count, example response: 2 (or skip 2 trigger events before generating a Trigger A event output)
<b>STATUS EVENTS</b>	None	
<b>AFFECTIVITY</b>	None	
<b>RELATED COMMANDS</b>	SWEep:OFFSet:TIME	
	TRIGger:A:PATTERN	
	TRIGger:A:PULSE	
	TRIGger:A:SLOPE	

### 1.3.101 Trigger A Pattern Mask (TRIGger:A:PATtern:MASK)

<b>COMMAND SYNTAX</b>	TRIGger[:A]:PATtern:MASK <mask>	full
	TRIGger:PATtern:MASK <mask>	minimum
<b>QUERY SYNTAX</b>	TRIGger[:A]:PATtern:MASK?	full
	TRIGger:PATtern:MASK?	minimum

Sets or queries the active bits used by the trigger pattern detector. The following considerations apply when using the TRIGger:PATtern:MASK command:

- there are 5 bits in the trigger pattern detector, defined as follows:
  - bits 15-5 (msbs) ignored
  - bit 4 VXIbus ECLTrg1
  - bit 3 VXIbus ECLTrg0
  - bit 2 front panel TRG IN
  - bit 1 CH2 OR CH4, where OR is the logical “or” operation
  - bit 0 (lsb) CH1 OR CH3, where OR is the logical “or” operation
- setting a mask bit to ‘1’ enables the associated source, setting the bit to ‘0’ disables the source
- refer to the TRIGger:PATtern:TRUTH command for information on setting the source state to be detected.
- the pattern detector output becomes a selectable trigger source, refer to the trigger source commands.
- pattern detection uses direct signal source outputs, before trigger source selection.

<b>COMMAND PARAMETERS</b>	<mask>	desired source detection mask 0 to 31 where 0 disables all sources and 31 enables all sources
<b>RESPONSE PARAMETERS</b>	<mask>	current source detection mask
<b>DEFAULT VALUE</b>	0	
<b>EXAMPLES</b>	TRIG:PATT:MASK 25	enable ECLT0, ECLT1, and CH1 for detection
	TRIG:PATT:MASK?	query the current detection mask, example response: 25
<b>STATUS EVENTS</b>	None	
<b>AFFECTIVITY</b>	None	
<b>RELATED COMMANDS</b>	TRIGger:A:PATtern:TRUTH TRIGger:A:SOURce TRIGger:B:SOURce	

# 1

## 1.3.102 Trigger A Pattern Truth (TRIGger:A:PATtern:TRUTh)

<b>COMMAND SYNTAX</b>	TRIGger[:A]:PATtern:TRUTh <pattern>	full
	TRIGger:PATtern:TRUTh <mask>	minimum
<b>QUERY SYNTAX</b>	TRIGger[:A]:PATtern:TRUTh?	full
	TRIGger:PATtern:TRUTh?	minimum

Sets or queries the active bits used by the trigger pattern detector. The following considerations apply when using the TRIGger:PATtern:TRUTh command:

- there are 5 bits in the trigger pattern detector, defined as follows:
 

bits 15-5 (msbs)	ignored
bit 4	VXIbus ECLTrg1
bit 3	VXIbus ECLTrg0
bit 2	front panel TRG IN
bit 1	CH2 OR CH4, where OR is the logical "or"
operation	
bit 0 (lsb)	CH1 OR CH3, where OR is the logical "or"
operation	
- setting a pattern bit to '1' sets the pattern detector to watch for a '1' from the associated trigger source, setting the bit to '0' sets the detector to watch for a '0'.
- bits that are not enabled for detection are ignored, refer to the TRIGger:PATtern:MASK command for information on setting the pattern detection mask.
- the pattern detector output becomes a selectable trigger source, refer to the trigger source commands.
- pattern detection uses direct signal source outputs, before trigger source selection.

<b>COMMAND PARAMETERS</b>	<pattern>	desired source detection pattern 0 to 31 where 0 detects all sources = 0 and 31 enables all sources = 1
<b>RESPONSE PARAMETERS</b>	<pattern>	current source detection pattern
<b>DEFAULT VALUE</b>	0	
<b>EXAMPLES</b>	TRIG:PATT:TRUT 25	detect ECLT0, ECLT1, and CH1 = '1' and all other sources = '0'
	TRIG:PATT:TRUT?	query the current detection pattern, example response: 25
<b>STATUS EVENTS</b>	None	
<b>AFFECTIVITY</b>	None	
<b>RELATED COMMANDS</b>	TRIGger:A:PATtern:MASK TRIGger:A:SOURce TRIGger:B:SOURce	

### 1.3.103 Trigger A Pulse High Limit (TRIGger:A:PULSe:HLIMit)

<b>COMMAND SYNTAX</b>	TRIGger[:A]:PULSe:HLIMit <limit>	full
	TRIGger:PULSe:HLIMit <limit>	minimum
<b>QUERY SYNTAX</b>	TRIGger[:A]:PULSe:HLIMit?	full
	TRIGger:PULSe:HLIMit?	minimum

Sets or queries the longer of the two pulse detection time limits. The following considerations apply when using the pulse high limit command:

- there are two time limits used by the pulse detection circuit, the high time limit and the low time limit. The high limit is the longer time of the two.
- the high time limit must be greater than the low time limit, refer to TRIGger:PULSe:LLIMit for information on setting the low limit.
- the high time limit setting is ignored for “edge” and “pulse less than” trigger types, refer to the TRIGger:A:TYPE command for information on selecting the trigger type.
- the trigger detection type must be set to PULSe for this command to have an effect, refer to the TRIGger:A:TYPE command for information on setting the trigger type.

<b>COMMAND PARAMETERS</b>	<limit>	desired pulse high time limit in seconds, where limit must lie within the following bounds: limit ≥ 15 nsec limit < 327 usec
<b>RESPONSE PARAMETERS</b>	<limit>	current pulse high time limit
<b>DEFAULT VALUE</b>	0.0	
<b>EXAMPLES</b>	TRIG:PULS:HLIM 11usec	set pulse detect high limit to 11 usec
	TRIG:PULS:HLIM?	query the current pulse detect high limit, example response: 1.1e-5 (or the pulse detect high limit = 11 usec)
<b>STATUS EVENTS</b>	None	
<b>AFFECTIVITY</b>	None	
<b>RELATED COMMANDS</b>	TRIGger:A:LLIMit	
	TRIGger:A:TYPE	

### 1.3.104 Trigger A Pulse Low Limit (TRIGger:A:PULSe:LLIMit)

<b>COMMAND SYNTAX</b>	TRIGger[:A]:PULSe:LLIMit <limit>	full
	TRIGger:PULSe:LLIMit <limit>	minimum
<b>QUERY SYNTAX</b>	TRIGger[:A]:PULSe:LLIMit?	full
	TRIGger:PULSe:LLIMit?	minimum

Sets or queries the longer of the two pulse detection time limits. The following considerations apply when using the pulse low limit command:

- there are two time limits used by the pulse detection circuit, the high time limit and the low time limit. The low limit is the shorter time of the two.
- the low time limit must be less than the high time limit, refer to TRIGger:PULSe:HLIMit for information on setting the high limit.
- the low time limit setting is ignored for “edge” and “pulse greater than” trigger types, refer to the TRIGger:A:TYPE command for information on selecting the trigger type.
- the trigger detection type must be set to PULSe for this command to have an effect, refer to the TRIGger:A:TYPE command for information on setting the trigger type.

<b>COMMAND PARAMETERS</b>	<limit>	desired pulse low time limit in seconds, where limit must lie within the following bounds: limit < 262 usec limit ≥ 10 nsec
<b>RESPONSE PARAMETERS</b>	<limit>	current pulse low time limit
<b>DEFAULT VALUE</b>	0.0	
<b>EXAMPLES</b>	TRIG:PULS:LLIM 11usec	set pulse detect low limit to 11 usec
	TRIG:PULS:LLIM?	query the current pulse detect low limit, example response: 1.1e-5 (or the pulse detect low limit = 11 usec)
<b>STATUS EVENTS</b>	None	
<b>AFFECTIVITY</b>	None	
<b>RELATED COMMANDS</b>	TRIGger:A:HLIMit	
	TRIGger:A:TYPE	

### 1.3.105 Trigger A Slope (TRIGger:A:SLOPe)

<b>COMMAND SYNTAX</b>	TRIGger[:A]:SLOPe <slope>	full
	TRIGger:SLOPe <slope>	minimum
<b>QUERY SYNTAX</b>	TRIGger[:A]:SLOPe?	full
	TRIGger:SLOPe?	minimum

Sets or queries the Trigger A active edge. The following considerations apply when setting the Trigger A slope:

- All trigger detection is performed on active high signals, i.e. an analog input is above its threshold or VXIbus ECLTrg<n> and TTLTrg<n> signals are at a logic '1' state.
- The Trigger A active slope is set by a selectable inverter applied after the trigger source has been selected and before any enabled pulse qualification.
- The NEGATIVE slope setting causes the selected trigger to be inverted before being used by subsequent trigger functions, this results in a trigger signal high-to-low transition as the active edge
- The POSITIVE slope setting causes the trigger to be passed without inversion, this results in a trigger signal low-to-high transition as the active edge
- Trigger A detection must be complete before the Trigger B detector or sweep offset timer are enabled

<b>COMMAND PARAMETERS</b>	<slope>	desired trigger active edge
		POSitive low-to-high transitions
		NEGative high-to-low transitions

<b>RESPONSE PARAMETERS</b>	<slope>	current trigger active edge
		POS low-to-high transitions
		NEG high-to-low transitions

**DEFAULT VALUE** POS

<b>EXAMPLES</b>	TRIG:SLOP NEG	selects high-to-low transitions as the active edge
	TRIG:SLOP?	query trigger active edge, example response: POS

**STATUS EVENTS** -141 invalid slope name selected, command was ignored

**AFFECTIVITY** None

**RELATED COMMANDS** None

# 1

## 1.3.106 Trigger A Source (TRIGger:A:SOURce)

<b>COMMAND SYNTAX</b>	TRIGger[:A]:SOURce <source>	full
	TRIGger:SOURce <source>	minimum
<b>QUERY SYNTAX</b>	TRIGger[:A]:SOURce?	full
	TRIGger:SOURce?	minimum

Sets or queries the Trigger A signal source. The following considerations apply when using setting the Trigger A source:

- All sources are assumed to low-to-high transitioning signals when active. If a source becomes active on a high-to-low transition, use the TRIGger:A:SLOPe command to select NEGative slope.
- Trigger A detection must be complete before the Trigger B detector or sweep offset timer are enabled

<b>COMMAND PARAMETERS</b>	<source>	desired Trigger A signal source
	ARM	front panel ARM IN signal
	ECLTrg<n>	VXIbus ECL trigger line, where <n> may be 0 or 1
	IMMediate	bypass arm detection
	INP<n>	analog input, where <n> may be 1, 2, 3, or 4
	NONE	*TRG or word-serial TRIGGER commands only
	TTLTrg<n>	VXIbus TTL trigger line, where <n> may be 0, 1, 2, ... or 7

<b>RESPONSE PARAMETERS</b>	<source>	current Trigger A signal source
	ARM	front panel ARM IN signal
	ECLT<n>	VXIbus ECL trigger line, where <n> may be 0 or 1
	INP<n>	analog input, where <n> may be 1, 2, 3, or 4
	NONE	*TRG or word-serial TRIGGER commands only
	TTLT<n>	VXIbus TTL trigger line, where <n> may be 0, 1, 2, ... or 7

**DEFAULT VALUE** INP1

**EXAMPLES** TRIG:SOUR NONE trigger on the \*TRG or word-serial TRIGGER commands only  
 TRIG:SOUR? query current Trigger A signal source, example response: PATT

**STATUS EVENTS** -141 invalid trigger signal source name selected, command was ignored  
 -221 trigger settings conflict, inactive channel selected as source, command was ignored

**AFFECTIVITY** None

**RELATED COMMANDS** TRIGger:A:SLOPe

### 1.3.107 Trigger A Type (TRIGger:A:TYPE)

<b>COMMAND SYNTAX</b>	TRIGger[:A]:TYPE <type>	full
	TRIGger:TYPE <type>	minimum
<b>QUERY SYNTAX</b>	TRIGger[:A]:TYPE?	full
	TRIGger:TYPE?	minimum

Sets or queries the Trigger A detection type. The following considerations apply when using setting the Trigger A type:

- five trigger types are accepted: edge, pulse inside limits, pulse outside limits, pulse less than limit, and pulse greater than limit.
- EDGE triggering looks for a rising or falling edge from the selected signal source, refer to the TRIGger:A:SLOPe and TRIGger:A:SOURce respectively for more information on selecting the active edge and signal source.
- pulse inside limits or PIN looks for a pulse width greater than a low time limit and less than or equal to a high time limit, refer to the TRIGger:A:LLIMit and TRIGger:A:HLIMit for more information on setting the low and high limits.
- pulse outside limits or POUT looks for a pulse width less than a low time limit or greater than a high time limit, refer to the TRIGger:A:LLIMit and TRIGger:A:HLIMit for more information on setting the low and high limits.
- pulse less than or PLTH looks for a pulse width less than a low time limit, refer to the TRIGger:A:LLIMit for more information on setting the low limit.
- pulse greater than or PGTH looks for a pulse width greater than a high time limit, refer to the TRIGger:A:HLIMit for more information on setting the low limit.
- Trigger A detection must be complete before the Trigger B detector or sweep offset timer are enabled

<b>COMMAND PARAMETERS</b>	<source>	desired Trigger A detection type
		EDGE detector watches for a active edge from the trigger source
		PGTH detector watches for a pulse width greater than a time limit
		PIN detector watches for a pulse width inside a time window
		PLTH detector watches for a pulse width less than a time limit
		POUT detector watches for a pulse width outside a time window

**RESPONSE PARAMETERS**      <source>      current Trigger A type

**DEFAULT VALUE**      EDGE

# 1

---

<b>EXAMPLES</b>	TRIG:TYPE POUT TRIG:TYPE?	trigger on pulse width outside the selected time window query current Trigger A type, example response: POUT
<b>STATUS EVENTS</b>	-141	invalid trigger type name selected, command was ignored
<b>AFFECTIVITY</b>	None	
<b>RELATED COMMANDS</b>	TRIGger:A:HLIMit TRIGger:A:LLIMit TRIGger:A:SLOPe TRIGger:A:SOURce	

**1.3.108 Trigger B Slope (TRIGger:B:SLOPe)**

**COMMAND SYNTAX** TRIGger:B:SLOPe <slope>

**QUERY SYNTAX** TRIGger:B:SLOPe?

Sets or queries the Trigger B active edge. The following considerations apply when setting the Trigger B slope:

- All trigger detection is performed on active high signals, i.e. an analog input is above its threshold or VXIbus ECLTrg<n> and TTLTrg<n> signals are at a logic '1' state.
- The Trigger B active slope is set by a selectable inverter applied after the trigger source has been selected.
- The NEGATIVE slope setting causes Trigger B to be recognized on a high-to-low signal transition
- The POSITIVE slope setting causes Trigger B to be recognized on a low-to-high signal transaction
- Trigger B detection must be enabled for this command to have an effect, refer to the TRIGger:B:STATE command

<b>COMMAND PARAMETERS</b>	<slope>	desired trigger active edge POSitive low-to-high transitions NEGative high-to-low transitions
<b>RESPONSE PARAMETERS</b>	<slope>	current trigger active edge POS low-to-high transitions NEG high-to-low transitions
<b>DEFAULT VALUE</b>	POS	
<b>EXAMPLES</b>	TRIG:B:SLOP NEG TRIG:B:SLOP?	selects high-to-low transitions as the active edge query trigger active edge, example response: POS
<b>STATUS EVENTS</b>	-141	invalid slope name selected
<b>AFFECTIVITY</b>	None	
<b>RELATED COMMANDS</b>	TRIGger:B:STATE	

### 1.3.109 Trigger B Source (TRIGger:B:SOURce)

**COMMAND SYNTAX** TRIGger:B:SOURce <source>

**QUERY SYNTAX** TRIGger:B:SOURce?

Sets or queries the Trigger B signal source. The following considerations apply when using setting the Trigger B source:

- All sources are assumed to low-to-high transitioning signals when active. If a source becomes active on a high-to-low transition, use the TRIGger:B:SLOPe command to select NEGative slope.
- Trigger B detection may be bypassed by selecting the IMMEDIATE source.
- Trigger A detection must be complete before the Trigger B detector is enabled
- Trigger B detection must be complete before the sweep offset timer is enabled
- Trigger B detection must be enabled for this command to have an effect, refer to the TRIGger:B:STATe command

<b>COMMAND PARAMETERS</b>	<source>	desired Trigger B signal source
		ARM front panel ARM IN signal
		ECLTrg<n> VXIbus ECL trigger line, where <n> may be 0 or 1
		IMMEDIATE bypass Trigger B detection
		INP<n> analog input, where <n> may be 1, 2, 3, or 4
		NONE *TRG or word-serial TRIGGER commands only
		PATTERN pattern detector output
		TTLTrg<n> VXIbus TTL trigger line, where <n> may be 0, 1, 2, ... or 7
<b>RESPONSE PARAMETERS</b>	<source>	current Trigger B signal source
		ARM front panel ARM IN signal
		ECLT<n> VXIbus ECL trigger line, where <n> may be 0 or 1
		INP<n> analog input, where <n> may be 1, 2, 3, or 4
		NONE *TRG or word-serial TRIGGER commands only
		PATT pattern detector output
		TTLT<n> VXIbus TTL trigger line, where <n> may be 0, 1, 2, ... or 7
<b>DEFAULT VALUE</b>	IMM	
<b>EXAMPLES</b>	TRIG:B:SOUR PATT	generate a Trigger B event when a pattern match is detected
	TRIG:B:SOUR?	query current Trigger B signal source, example response: PATT
<b>STATUS EVENTS</b>	-141	invalid trigger signal source name selected
	-221	trigger settings conflict, inactive channel selected as source, command was ignored
<b>AFFECTIVITY</b>	None	
<b>RELATED COMMANDS</b>	TRIGger:B:SLOPe	
	TRIGger:B:STATe	

### 1.3.110 Trigger B State (TRIGger:B:SOURce)

**COMMAND SYNTAX** TRIGger:B:STATe <sw>

**QUERY SYNTAX** TRIGger:B:STATe?

Sets or queries the Trigger B detector enable. The following considerations apply when using setting the Trigger B source:

- When Trigger B is disabled, the SWEep:OFFSet:TIME timer is started immediately after the Trigger A event is recognized.
- Trigger A detection must be complete before the Trigger B detector is enabled
- Trigger B detection must be complete before the sweep offset timer is enabled

<b>COMMAND PARAMETERS</b>	<sw>	desired Trigger B dectector state 1 or ON enable detector 0 or OFF disable detector
---------------------------	------	-------------------------------------------------------------------------------------------

<b>RESPONSE PARAMETERS</b>	<sw>	current Trigger B detector state 1 or ON enable detector 0 or OFF disable detector
----------------------------	------	------------------------------------------------------------------------------------------

**DEFAULT VALUE** 0

<b>EXAMPLES</b>	TRIG:B:STAT ON	enable the Trigger B detector
	TRIG:B:STAT?	query current Trigger B detector enable, example response: 1 (or Trigger B detection is enabled)

<b>STATUS EVENTS</b>	-141	invalid trigger state name selected, command was ignored
----------------------	------	----------------------------------------------------------

**AFFECTIVITY** None

**RELATED COMMANDS** None

# 1

---

## 1.3.111 Trigger Complete (TRIGger:COMplete)

**COMMAND SYNTAX** n/a

**QUERY SYNTAX** TRIGger:COMplete?

Queries the number of trigger events recognized since the last INITiate.

**COMMAND** n/a

**PARAMETERS**

**RESPONSE** <num> number of trigger events recognized since the last INITiate

**PARAMETERS**

**DEFAULT VALUE** n/a

**EXAMPLES** TRIG:COMP? query current number of trigger events, example response: 3

**STATUS EVENTS** None

**AFFECTIVITY** None

**RELATED** INITiate  
**COMMANDS**

**1.3.112 Trigger Count (TRIGger:COUNT)**

**COMMAND SYNTAX** TRIGger:COUNT <cnt>

**QUERY SYNTAX** TRIGger:COUNT?

Sets or queries the number of trigger events to collect during the next waveform capture cycle. The following considerations apply when setting the trigger count:

- setting the trigger count to a value greater than 1 will cause the next capture cycle to be a multiple waveform capture.
- the maximum trigger count can be determined using the TRIGger:MCOunt query.
- waveform averaging is disabled when the trigger count is set to a value greater than 1.

<b>COMMAND PARAMETERS</b>	<cnt>	desired number of trigger events to collect during the next capture cycle 1 to 1024
<b>RESPONSE PARAMETERS</b>	<cnt>	current number of trigger events to be collected by the next capture cycle
<b>DEFAULT VALUE</b>	1	
<b>EXAMPLES</b>	TRIG:COUN 10 TRIG:COUN?	capture 10 trigger events during the next capture cycle query current number of trigger events to be captured, example response: 10 (or capture 10 waveforms after the next INITiate command is issued)
<b>STATUS EVENTS</b>	-222	invalid trigger count selected, value coerced in range
<b>AFFECTIVITY</b>		Waveform averaging is disabled when the trigger count is set to a value greater than 1.
<b>RELATED COMMANDS</b>	TRIGger:MCOunt	

# 1

## 1.3.113 Trigger External Level (TRIGger:EXtErnal:LEVel)

**COMMAND SYNTAX** TRIGger:EXtErnal:LEVel <volts>

**QUERY SYNTAX** TRIGger:EXtErnal:LEVel?

Sets or queries the front panel TRG IN trigger level. The TRG IN signal has a fixed peak-to-peak and trigger level range of  $\pm 1$  V. The logic level out of the threshold detector is reported to the Trigger A and B multiplexers for selection as the EXtErnal trigger source.

**COMMAND PARAMETERS** <volts> desired TRG IN signal detection level  
 $-1\text{ V} \leq \text{volts} \leq +1\text{ V}$

**RESPONSE PARAMETERS** <volts> current TRG IN signal detection level

**DEFAULT VALUE** 0.0

**EXAMPLES** TRIG:EXT:LEV 0.5 select signal ranges over 0.5 V to be reported as active  
TRIG:EXT:LEV? query current signal detection level, example response: 5.0e-1

**STATUS EVENTS** -222 invalid trigger level selected, value coerced in range.

**AFFECTIVITY** None

**RELATED COMMANDS** None

**1.3.114 Trigger Holdoff (TRIGger:HOLDoff)**

**COMMAND SYNTAX** TRIGger:HOLDoff <time>

**QUERY SYNTAX** TRIGger:HOLDoff?

Sets or queries the minimum time between trigger events in a multiple waveform capture cycle. The following considerations apply when setting the trigger count:

- Trigger detection is disabled when the hold-off is active
- The effective minimum time between trigger events will be the larger of: the unit re-arm time, approximately 35 usec, or the trigger hold-off time.

**COMMAND PARAMETERS** <time> desired trigger hold-off time  
0.0 to 64.0 sec

**RESPONSE PARAMETERS** <time> current trigger hold-off time

**DEFAULT VALUE** 0.0

**EXAMPLES** TRIG:HOLD 50 usec set trigger hold-off to 50 usec  
TRIG:HOLD? query current trigger hold-off, example response: 5.0e-5

**STATUS EVENTS** -222 invalid trigger hold-off, value coerced in range.

**AFFECTIVITY** None

**RELATED COMMANDS** None

### 1.3.115 Trigger Input Level (TRIGger:INPut:LEVel)

**COMMAND SYNTAX** TRIGger:INPut<n>:LEVel <volts>

**QUERY SYNTAX** TRIGger:INPut<n>:LEVel?

Sets or queries an analog input channel trigger level. The following considerations apply when setting the input trigger level:

- The trigger level has the same range as the input signal.
- Each input channel has an independent trigger level setting but input channels 1 & 3 and channels 2 & 4 share a single trigger detector. In order to resolve possible conflicts, Trigger A source settings have priority over Trigger B source settings, i.e. if Trigger A uses INPut1 and Trigger B uses INPut3, then the INPut 1 trigger level will be used.
- The logic level out of the threshold detector is reported to the Trigger A and B multiplexers for selection as the INP<n> trigger sources.

<b>COMMAND PARAMETERS</b>	<n>	input channel number 1 to 4
	<volts>	desired TRG IN signal detection level $-1\text{ V} \leq \text{volts} \leq +1\text{ V}$
<b>RESPONSE PARAMETERS</b>	<volts>	current analog input signal detection level
<b>DEFAULT VALUE</b>	0.0	
<b>EXAMPLES</b>	TRIG:INP2:LEV 22	select signal ranges over 22 V to be reported as active
	TRIG:INP2:LEV?	query current signal detection level, example response: 2.2e+1
<b>STATUS EVENTS</b>	-222	invalid trigger level selected, value coerced in range.
<b>AFFECTIVITY</b>	None	
<b>RELATED COMMANDS</b>	None	

### 1.3.116 Trigger Maximum Count (TRIGger:MCOut)

**COMMAND SYNTAX** n/a

**QUERY SYNTAX** TRIGger:MCOut?

Queries the maximum allowed value for the TRIGger:COUNT command. The maximum trigger count is dependent on the installed digitizer memory size and the sweep points size.

**COMMAND PARAMETERS** n/a

**RESPONSE PARAMETERS** <cnt> maximum number of trigger events that can be collected

**DEFAULT VALUE** n/a

**EXAMPLES** TRIG:MCO? query maximum number of trigger events to be captured, example response: 1024

**STATUS EVENTS** None

**AFFECTIVITY** None

**RELATED COMMANDS** :SENSe:SWEep:POINTs

