

ARTISAN®
TECHNOLOGY GROUP

Your **definitive** source for quality pre-owned equipment.

Artisan Technology Group
(217) 352-9330 | sales@artisantg.com | artisantg.com

Full-service, independent repair center
with experienced engineers and technicians on staff.

We buy your excess, underutilized, and idle equipment
along with credit for buybacks and trade-ins.

Custom engineering
so your equipment works exactly as you specify.

- Critical and expedited services
- In stock / Ready-to-ship
- Leasing / Rentals / Demos
- ITAR-certified secure asset solutions

Expert team | Trust guarantee | 100% satisfaction

All trademarks, brand names, and brands appearing herein are the property of their respective owners.

Visit our website - *Click HERE*

Command Headers

```
[CHx]:<COMMAND HEADER>[?] [<ARGUMENT>];
```

The command header specifies what function the command is going to perform, or what parameter it affects. In its simplest form, command program headers are a single word, as shown here.

EXAMPLE:

```
INIT
```

This command initializes the instrument to its power up settings.

Compound Headers

Standard IEEE 488.2 requires that hierarchical commands be formatted as compound command headers, with the separate words linked with colons. Spaces are not allowed before or after the colons. Each layer of the header builds on the one to its right.

EXAMPLE:

```
CH2:TRIGGER:LEVEL 1.0
```

This command sets the trigger level of channel 2 to 1.0 volts. The header LEVEL pertains to the TRIGGER. This linking is required because there are several parameters which control the trigger, and each is a unique command.

Compound headers can be linked with as many layers as necessary.

EXAMPLE:

```
CH1:TRIGGER:EXT:COUPLING DC
```

This command sets the coupling of the channel 1 external trigger input to DC. The header COUPLING pertains to the header EXT, when it is specified as the source for TRIGGER.

Query Commands

```
[CHx]:<COMMAND HEADER>[?]
```

Query program headers are used to query the instrument for a response. The query program header always ends with a question mark"?". Spaces are not allowed between the end of the header and the question mark. Some commands only have the query format, like the one in this example.

EXAMPLE:

```
ID?
```

The ID? command returns information about the instrument's identification. The command ID without the question mark is not valid and will generate an error. In general, all commands sent with a parameter argument can also be sent in the query form by adding the question mark to the end of the command and omitting the argument.

EXAMPLE:

```
CH1:OFFSET 3.0
```

This command programs the input offset of channel 2 to 3.0 volts. The query form of the command is:

```
CH1:OFFSET?
```

Sending this command queries the instrument as to what the channel 1 offset voltage is. The VX4250 responds:

```
CH1:OFFSET 3.000
```

Using the query form of settings commands is useful during the debugging of the test system. The response from settings commands queries is always in a form that is valid for sending back as a setting command. This allows programmers to write routines that capture query responses, and then use them to build instrument settings commands that can later be retransmitted to the instrument.

Argument Delimiter

```
[CHx]:<COMMAND HEADER[?]_[ARGUMENT];
```

A space character (ASCII 32) must be used to delimit the command header from the argument. The space character must be used. (In the example here, an underscore character is shown for clarity). Multiple spaces are ignored.

EXAMPLE:

```
CH1:RANGE 0.4
```

This command sets the full scale range of channel 1 to 400 millivolts. If this command was sent without the space as:

```
CH1:RANGE0.4
```

An error would be generated and the command ignored.

Some commands have more than one argument. The multiple arguments are separated with a comma (,).

EXAMPLE:

```
CURVE 14,12,11,10,10,9,8,.....
```

This command is used to download a waveform into the VX4250. There may be up to 2044 arguments used with this command.

Command Unit Separators

```
[CHx]:<COMMAND HEADER>[?] [ARGUMENT];:<NEXT COMMAND UNIT>
```

A semicolon followed by a colon (;:) is used as a delimiter between separate commands when multiple commands are being sent in the same message. Linking several commands in the same message saves execution time because the communication channel is already established for the remaining commands.

EXAMPLE:

```
CH1:RANGE 40;:CH1:SAMPLE:INTERVAL 10E-9;:CH1:START
```

This command string sets the channel 1 full scale to 40 volts, the sample interval to 1 nanosecond, and starts the acquisition.

The use of a semicolon delimiter without the colon has special meaning when used with compound command headers. In certain instances, this format reduces the amount of text required in command headers. The effective use of this format requires a thorough understanding of command header tree structures. For more detailed information on this, refer to the Command Header Tree Structures topic in Appendix A, *IEEE Standard 488.2 Reference.*

IEEE 488.2 Standard Commands

The IEEE 488.2 Standard defines a set of recommended common instrument commands. These commands can be easily identified because their first character is an asterisk "*".

EXAMPLE:

```
*TST?
```

This command invokes the instrument self tests which return a result.

For a complete list of the IEEE 488.2 common commands implemented in the VX4250, see the System Commands topic at the end of Section 6, *Command Dictionary,* in this manual.

This page intentionally left blank.

Section 6 – Command Dictionary

Table of Contents

Introduction 6-1
Alphabetical Command Listing 6-2
Functional Command Listing 6-4
Acquisition Sequence Setup Commands . . 6-4
Acquisition Sequence Control Commands . . 6-5
Acquired Data Access Commands 6-5
Post Acquisition Data Manipulation Commands 6-6
Low Level Shared Memory Commands . . . 6-8
Error and Event Reporting Commands . . . 6-8
Instrument Maintenance Command 6-8
Miscellaneous Commands 6-8
IEEE-488.2 Support Commands 6-9
Power-Up Default Control Settings 6-10
Command Dictionary Layout 6-11

Channel Commands 6-11
CHx:ACQUIRE:LOC 6-12
CHx:ACQUIRE:MODE 6-13
CHx:ACQUIRE:NUM 6-14
CHx:AUTO 6-15
CHx:COMPARE:HIGH 6-16
CHx:COMPARE:LOC 6-17
CHx:COMPARE:LOW 6-18
CHx:COMPARE:NOMINAL 6-19
CHx:COMPARE:TYPE 6-20
CHx:CONNECT:TRIGGEROUT 6-22
CHx:COUPLING 6-23
CHx:OFFSET 6-24
CHx:PRETRIGGER 6-25
CHx:RANGE 6-26
CHx:RECALL 6-27
CHx:RESULT?, CHx:RESULT:LONG? . . 6-28
CHx:SAMPLE:INTERVAL 6-29
CHx:SAMPLE:WINDOW 6-30
CHx:SAVE 6-31
CHx:START 6-32
CHx:STATS? 6-33
CHx:STATUS? 6-34
CHx:STEP:COUNT? 6-35
CHx:STEP:EXIT 6-36
CHx:STEP:MODE 6-37
CHx:STEP:NUMLOOP 6-38
CHx:STOP 6-39
CHx:TRIGGER:EXT:COUPLING 6-40
CHx:TRIGGER:HFR 6-41
CHx:TRIGGER:LEVEL 6-42
CHx:TRIGGER:SLOPE 6-43
CHx:TRIGGER:SOURCE 6-44
CHx:WAIT? 6-45
CH1? 6-46
CH2? 6-47

Instrument Commands 6-49
ADD 6-51
AMPLITUDE? 6-52
AREA? 6-53
BASE? 6-54
BMODE 6-55
BOVER? 6-56
BTLINES 6-57
CURVE 6-58
DATA:COUNT 6-59
DATA:ENCDG 6-60
DATA:LOC 6-61
DATA:START 6-62
DESE, DESE? 6-63
DUTY? 6-64
ENERGY? 6-65
EVENT? 6-66
EVMSG? 6-67
EXTMODE 6-68
FALL? 6-69
FDCBASE? 6-70
FDCLOAD, FDCLOAD? 6-71
FDCSIZE? 6-72
FREQUENCY? 6-73
GAIN? 6-74
ID? 6-75
INIT 6-76
MARGINS 6-77
MAX? 6-78
MEAN? 6-79
MID? 6-80
MIN? 6-81
PERIOD? 6-82
PHASE? 6-83
PK2PK? 6-84
REFCRS? 6-85
REFLINES 6-86
REF:LOC 6-87
REFUNITS 6-88
REFVALUES? 6-89
RISE? 6-90
RMS? 6-91
SET? 6-92
SUB 6-93
TMODE 6-94
TOP? 6-95
TOVER? 6-96
TST? 6-97
WFMCMP? 6-98
WFMPRE? 6-99
WIDTH? 6-100
WINDOW 6-101

System Commands 6-103
*ESE, *ESE? 6-104
*ESR? 6-105
*IDN? 6-106
*OPC, *OPC? 6-107
*RCL 6-108
*SAV 6-109
*SRE, *SRE? 6-110
*STB? 6-111
*TST? 6-112
*WAI 6-113

Section 6 – Command Dictionary

Table of Contents

Introduction 6-1
Alphabetical Command Listing 6-2
Functional Command Listing 6-4
Acquisition Sequence Setup Commands . . 6-4
Acquisition Sequence Control Commands . . 6-5
Acquired Data Access Commands 6-5
Post Acquisition Data Manipulation Commands 6-6
Low Level Shared Memory Commands . . . 6-8
Error and Event Reporting Commands . . . 6-8
Instrument Maintenance Command 6-8
Miscellaneous Commands 6-8
IEEE-488.2 Support Commands 6-9
Power-Up Default Control Settings 6-10
Command Dictionary Layout 6-11

Channel Commands 6-11
CHx:ACQUIRE:LOC 6-12
CHx:ACQUIRE:MODE 6-13
CHx:ACQUIRE:NUM 6-14
CHx:AUTO 6-15
CHx:COMPARE:HIGH 6-16
CHx:COMPARE:LOC 6-17
CHx:COMPARE:LOW 6-18
CHx:COMPARE:NOMINAL 6-19
CHx:COMPARE:TYPE 6-20
CHx:CONNECT:TRIGGEROUT 6-22
CHx:COUPLING 6-23
CHx:OFFSET 6-24
CHx:PRETRIGGER 6-25
CHx:RANGE 6-26
CHx:RECALL 6-27
CHx:RESULT?, CHx:RESULT:LONG? . . 6-28
CHx:SAMPLE:INTERVAL 6-29
CHx:SAMPLE:WINDOW 6-30
CHx:SAVE 6-31
CHx:START 6-32
CHx:STATS? 6-33
CHx:STATUS? 6-34
CHx:STEP:COUNT? 6-35
CHx:STEP:EXIT 6-36
CHx:STEP:MODE 6-37
CHx:STEP:NUMLOOP 6-38
CHx:STOP 6-39
CHx:TRIGGER:EXT:COUPLING 6-40
CHx:TRIGGER:HFR 6-41
CHx:TRIGGER:LEVEL 6-42
CHx:TRIGGER:SLOPE 6-43
CHx:TRIGGER:SOURCE 6-44
CHx:WAIT? 6-45
CH1? 6-46
CH2? 6-47

Instrument Commands 6-49
ADD 6-51
AMPLITUDE? 6-52
AREA? 6-53
BASE? 6-54
BMODE 6-55
BOVER? 6-56
BTLINES 6-57
CURVE 6-58
DATA:COUNT 6-59
DATA:ENCDG 6-60
DATA:LOC 6-61
DATA:START 6-62
DESE, DESE? 6-63
DUTY? 6-64
ENERGY? 6-65
EVENT? 6-66
EVMSG? 6-67
EXTMODE 6-68
FALL? 6-69
FDCBASE? 6-70
FDCLOAD, FDCLOAD? 6-71
FDCSIZE? 6-72
FREQUENCY? 6-73
GAIN? 6-74
ID? 6-75
INIT 6-76
MARGINS 6-77
MAX? 6-78
MEAN? 6-79
MID? 6-80
MIN? 6-81
PERIOD? 6-82
PHASE? 6-83
PK2PK? 6-84
REFCRS? 6-85
REFLINES 6-86
REF:LOC 6-87
REFUNITS 6-88
REFVALUES? 6-89
RISE? 6-90
RMS? 6-91
SET? 6-92
SUB 6-93
TMODE 6-94
TOP? 6-95
TOVER? 6-96
TST? 6-97
WFMCMP? 6-98
WFMPRE? 6-99
WIDTH? 6-100
WINDOW 6-101

System Commands 6-103
*ESE, *ESE? 6-104
*ESR? 6-105
*IDN? 6-106
*OPC, *OPC? 6-107
*RCL 6-108
*SAV 6-109
*SRE, *SRE? 6-110
*STB? 6-111
*TST? 6-112
*WAI 6-113

Section 6
Command Dictionary

Introduction

This section presents the instrument command set. It begins with an alphabetical list of the instrument specific commands, with a brief description of each command. Next, the functional commands are listed, showing the associations of commands with arguments. The functional commands are grouped in these categories:

- Acquisition sequence setup commands
- Acquisition sequence control commands
- Acquired data access commands
- Post acquisition data manipulation commands
- Low level shared memory commands
- Error and event reporting commands
- Instrument maintenance commands
- Miscellaneous commands

The next three topics present:

- The IEEE-488.2 support commands
- The power-up default control settings
- The command dictionary layout

The command dictionary has three major subsections:

- The channel commands
- The instrument commands
- The system commands

Each channel, instrument, and system command is presented, one to a page, with a complete description in conventional command dictionary layout. Refer to the Command Dictionary Layout topic on page 6-11 for an explanation of the layout.

Alphabetical Command Listing

This section provides an alphabetical list and description of each instrument specific command. In the list that follows, the space between the CHx: and the command is included for readability only. When these commands are entered, there is no space between the colon (:) and the rest of the command.

CHx:	ACQUIRE:LOC	Sets where acquired data will be stored
CHx:	ACQUIRE:MODE	Sets how an acquisition will be made
CHx:	ACQUIRE:NUM	Sets the number of acquisitions for averaging and enveloping
CHx:	AUTO	Automatically determines setups for the current signal
	CH1?	Returns all current settings for channel 1
	CH2?	Returns all current settings for channel 2
CHx:	COMPARE:HIGH	Sets the high value for parametric compare
CHx:	COMPARE:LOC	Sets the waveform location for envelope compare
CHx:	COMPARE:LOW	Sets the nominal value for parametric compare
CHx:	COMPARE:NOMINAL	Sets the low value for parametric compare
CHx:	COMPARE:TYPE	Sets the type of comparison (template or parametric)
CHx:	CONNECT:TRIGGEROUT	Routes the internal trigger signal to the VXIbus
CHx:	COUPLING	Sets the input signal coupling (AC or DC)
	CURVE?	Transmits waveform data
	DATA:ENCDG	Sets the encoding format for CURVE?
	DATA:LOC	Sets the data waveform location accessed with CURVE?
	EVENT?	Returns error and event information
	EVMSG?	Returns error and event information
	FDCBASE?	Returns the location of physical shared memory
	FDCLOAD?	Allows the user to access shared memory
	FDCSIZE?	Returns the size of physical shared memory
	ID?	Returns the model, manufacturer, serial number, and firmware version
	INIT	Sets all current setting to power-up default
	MARGINS	Adjusts the envelope size for use as a template

CHx:	OFFSET	Sets the input signal offset
CHx:	PRETRIGGER	Sets the pre-trigger data value
CHx:	RANGE	Sets the input signal full scale range
CHx:	RECALL	Recalls channel sequence from a stored setting location
CHx:	RESULT?	Returns the results of channel sequence compare
CHx:	SAMPLE:INTERVAL	Sets the sample interval for acquisition
CHx:	SAMPLE:WINDOW	Sets the time window for acquisition
CHx:	SAVE	Saves the current channel sequence state in a stored setting location
	SET?	Returns complete instrument setups
CHx:	START	Starts execution of a channel sequence
	STATS?	Returns statistics on tests passed and failed
CHx:	STATUS?	Returns the status of the current channel sequence
CHx:	STEP:COUNT?	Returns the number of completed channel sequences
CHx:	STEP:EXIT	Sets the test step terminating condition
CHx:	STEP:MODE	Sets the channel sequence termination condition
CHx:	STEP:NUMLOOP	Sets the number of channel sequences to be performed
CHx:	STOP	Forces the current channel sequence to terminate
CHx:	TRIGGER:EXT:COUPLING	Sets the external trigger signal coupling
CHx:	TRIGGER:HFR	Enables trigger high frequency reject
CHx:	TRIGGER:LEVEL	Sets the trigger level
CHx:	TRIGGER:SLOPE	Sets the trigger slope
CHx:	TRIGGER:SOURCE	Selects the trigger source
	TST?	Invokes Extended Self-test
CHx:	WAIT?	Waits (for a limited time) for the channel sequence to complete
	WFMCMP?	Compares a waveform to an envelope and returns the results
	WFMPRE?	Returns CURVE? scaling information

Functional Command Listing

This section provides a functional list showing associations of commands with arguments.

Acquisition Sequence Setup Commands

The following commands control all the channel sequence options.

CHx:ACQUIRE:LOC	<waveform location>
CHx:ACQUIRE:MODE	{NORMAL | SMOOTH | AVERAGE | ENVELOPE}
CHx:ACQUIRE:NUM	<real>
CHx:COMPARE:HIGH	<real>
CHx:COMPARE:LOC	<waveform location>
CHx:COMPARE:LOW	<real>
CHx:COMPARE:NOMINAL	<real>
CHx:COMPARE:TYPE	{NONE | TEMPLATE | MAX | MIN | AVE | PERIOD | .....}
CHx:COUPLING	{AC | DC}
CHx:PRETRIGGER	<real>
CHx:OFFSET	<real>
CHx:RANGE	<real>
CHx:SAMPLE:INTERVAL	<real>
CHx:SAMPLE:WINDOW	<real>
CHx:STEP:EXIT	{IMMEDIATE | NEVER | LOOPCOUNT | COMPAREFAIL | COMPAREPASS}
CHx:STEP:MODE	{SINGLE | RUN | LOOP}
CHx:STEP:NUMLOOP	<real> (the number of times to loop)
CHx:TRIGGER:EXT:COUPLING	{AC | DC}
CHx:TRIGGER:HFR	{ON | OFF}
CHx:TRIGGER:LEVEL	<real>
CHx:TRIGGER:SLOPE	{PLUS | MINUS}
CHx:TRIGGER:SOURCE	{CH1 | CH2 | EXTERNAL | SYSTRIG}

Acquisition Sequence Control Commands

The following commands provide control and status information about the execution of the channel sequence.

CHx:START

CHx:STATUS?

CHx:STOP

CHx:WAIT? <timeout to abort>

Acquired Data Access Commands

The following commands and queries provide access to raw data.

CURVE?

CURVE <waveform data>

DATA:COUNT

DATA:ENCDG {ASCII | RPB | RIB}

DATA:LOC <waveform location>

DATA:START

WFMPRE?

Post Acquisition Data Manipulation Commands

The following commands and queries provide access to the measurement of data.

AMPLITUDE?	Returns the amplitude of a waveform
AREA?	Returns the area of a waveform in volts/second
BASE?	Returns the base magnitude of a waveform
BMODE	Sets how base is determined
BTLINES	Sets the base or top value of a waveform
DUTY?	Returns the duty cycle of a waveform
ENERGY?	Returns the energy of a waveform
EXTMODE	Sets how RMS, mean, area and energy are calculated
FALL?	Returns the falltime of a waveform
FREQUENCY?	Returns the frequency of a waveform
GAIN?	Returns the gain difference between two waveforms
MAX?	Returns the max voltage of a waveform
MARGINS	Modifies an existing envelope
MEAN?	Returns the mean voltage of a waveform
MID?	Returns the mid voltage of a waveform
MIN?	Returns the min voltage of a waveform
PERIOD?	Returns the period of a waveform
PHASE?	Returns the phase difference between two waveforms
PK2PK?	Returns the peak-to-peak voltage of a waveform
REFCRS?	Returns the sample point in the waveform segment for the selected proximal, mesial, and distal points
REFLINES	Sets proximal, mesial, and distal points
REFUNITS	Sets units of measure for REFCRS and REFUNITS

	REFVALUES?	Returns proximal, mesial, and distal values
CHx:	**RESULT?**	Returns the results of the comparison in the last channel sequence.
	RISE?	Returns the risetime of a waveform
	RMS?	Returns the RMS voltage of a waveform
	TMODE	Sets how top is determined
	TOP?	Returns top magnitude of a waveform
	WIDTH?	Returns pulse width of a waveform
	WINDOW	Sets waveform location and segment
	WFMCMP?	Returns number of samples outside of envelope

Low Level Shared Memory Commands

The following queries allow fast access to stored settings and data.

FDCBASE?	{ DATA | ALLSET | ALL }
FDCLOAD?	{ DATA | ALLSET | ALL }
FDCSIZE?	{ DATA | ALLSET | ALL }

Error and Event Reporting Commands

The following queries provide error and event information.

EVENT?

EVMSG?

Instrument Maintenance Command

The following query supports instrument self-test.

***TST?**

Miscellaneous Commands

The following commands and queries provide global instrument control and information.

	CH1?	Returns all current channel 1 settings
	CH2?	Returns all current channel 2 settings
CHx:	**CONNECT:TRIGGEROUT**	<TTLTRG0 . . . TTLTRG7>
	ID?	Returns instrument identification
	INIT	Resets instrument to power up settings
	SET?	Returns all current instrument settings

IEEE-488.2 Support Commands

The following 488.2 "Common Commands" support system level programming. System commands are recognized by the * (asterisk) designating them as the 488.2 type.

Commands for Instrument Identification

***IDN?**	Instrument identification

Commands for Synchronization

***OPC**	Waits for operation to complete
***OPC?**	Waits for operation to complete and returns a "1"
***WAI**	Holds up execution of commands until all pending operations are complete

Commands for Error & Event Reporting

***ESE**	Sets the event status enable register
***ESE?**	Queries the setting of the event status enable register
***ESR?**	Queries the setting of the event status register
***SRE**	Sets the service request enable register
***SRE?**	Queries the service request enable register
***STB?**	Queries the status byte register

Power-Up Default Control Settings

The following list summarizes the instrument condition after power-up or receiving the `INIT` command.

Note:
Defaults apply to both channel 1 and 2.

RANGE	40
OFFSET	0
COUPLING	DC
SAMPLE:INTERVAL	100E-9
TRIGGER:SOURCE	INT
TRIGGER:EXT:COUPLING	DC
TRIGGER:LEVEL	0
PRETRIGGER	128
ACQUIRE:MODE	Normal
ACQUIRE:NUM	1
ACQUIRE:LOC	(1 for CH1, 2 for CH2)
COMPARE:TYPE	None
COMPARE:LOW	Off
COMPARE:NOMINAL	0
COMPARE:HIGH	Off
COMPARE:LOC	(3 for CH1, 4 for CH2)
STEP:EXIT	Single
STEP:NUMLOOP	1
DATA:LOC	1
DATA:ENCDG	RIB
CONNECT:TRIGGEROUT	OFF

Command Dictionary Layout

The FDCLOAD command is shown here as an example of the way the pages in this section are laid out:

FDCLOAD, FDCLOAD?
(COMMAND NAME)

Purpose:	Allows user to perform valid shared memory accesses.
Type:	Operational
Command Syntax:	FDCLOAD
Command Parameters:	(not applicable)
Default Value:	None
Query Syntax:	FDCLOAD?
Query Parameters:	None
Query Response:	The response will always be "1".
Description:	Allows movement of FDC information.
Examples:	Command: FDCLOAD? Response: 1;
Related Commands:	FDCBASE?, FDCSIZE?

Not every item in the list applies to every command; only the relevant items apply. For example, if the command is a query only, there are no command syntax and command parameters, and these items do not appear in the list. In the same way, if the command has no query form, no query syntax, query parameters, or query response is listed.

Commands to, and responses from, the VX4250 are shown in this typeface, e.g., FDCLOAD, to indicate they are actual commands or responses.

Channel Commands

The following is an alphabetically arranged summary of the commands that apply to the operation of each independent channel.

Channel commands are easily recognized by their header of CH1:<command> or CH2:<command>.

CHx:ACQUIRE:LOC

Purpose: Specifies which waveform location will hold the acquired waveform data after the channel sequence has completed.

Type: Setting

Command Syntax: `CHx:ACQUIRE:LOC <destination>`

Command Parameters: <destination> specifying the waveform location to hold the acquired data. The valid range is 1 to 8.

Default Value: 1 for CH1, 2 for CH2

Query Syntax: `CHx:ACQUIRE:LOC?`

Query Parameters: None

Query Response: The response will be <INTEGER> representing the waveform location.

Description: The `ACQUIRE:LOC` command allows the user to specify where the acquired data will be stored.

The `ACQUIRE:LOC?` query returns the waveform location in the current settings, where the waveform will be stored.

Examples:

Command: `CH1:ACQUIRE:LOC 1` — *Store channel 1 data in waveform location 1*

Command: `CH2:ACQUIRE:LOC 3` — *Store channel 2 data in waveform location 3*

Command: `CH2:ACQUIRE:LOC?`
Response: `CH2:ACQUIRE:LOC 3;`

Related Commands: `ACQUIRE:MODE`

CHx:ACQUIRE:MODE

Purpose: Sets how an acquisition will be made; either single, smoothed, averaged or envelope.

Type: Setting

Command Syntax: `CHx:ACQUIRE:MODE { NORMAL | SMOOTH | AVERAGE | ENVELOPE | OFF }`

Command Parameters:

NORMAL - performs a single acquisition of the input signal

SMOOTH - takes a single acquisition and performs a three point smooth algorithm (ACQUIRE:NUM) times.

AVERAGE - takes (ACQUIRE:NUM) acquisitions and computes the average value for each sample

ENVELOPE - takes (ACQUIRE:NUM) acquisitions and saves the maximum and minimum for each sample.

OFF - no acquisition is performed. This allows multiple tests to be performed on a single acquisition within the context of the channel sequence.

Default Value: NORMAL

Query Syntax: `CHx:ACQUIRE:MODE?`

Query Parameters: None

Query Response: The response will be either; "NORMAL", "SMOOTH", "AVERAGE", "ENVELOPE" or "OFF".

Description: Sets how an acquisition will be made. The options include; a single acquisition, signal averaging and signal envelope generation.

If SMOOTH is selected, then ACQUIRE:NUM specifies the number of times to smooth the waveform.

If AVERAGE is selected, then ACQUIRE:NUM specifies the number of averages to perform.

If ENVELOPE is selected, then ACQUIRE:NUM specifies the number of acquisitions to make.

Examples:

Command: `CH1:ACQUIRE:MODE AVERAGE;:CH1:ACQUIRE:NUM 10;:CH1:ACQUIRE:MODE?`
Response: `CH1: ACQUIRE:MODE AVERAGE;`

Related Commands: `ACQUIRE:NUM`

CHx:ACQUIRE:NUM

Purpose: Sets the number of acquisitions for averaging, enveloping, or number of times a single acquisition is smoothed.

Type: Setting

Command Syntax: `CHx:ACQUIRE:NUM <INTEGER>`

Command Parameters: <INTEGER> will specify the number of acquisitions to perform. The valid range is 1 to 256.

Default Value: 1

Query Syntax: `CHx:ACQUIRE:NUM?`

Query Parameters: None

Query Response: The response will be <INTEGER> and will specify the number of averages to perform.

Description: Sets the number of times to acquire the waveform when in ACQUIRE:MODE normal, average or envelope.

When in ACQUIRE:MODE smooth, NUM describes the number of times to apply the smooth algorithm to a single acquisition.

Examples: Command: `CH1:ACQUIRE:NUM 10;:CH1:ACQUIRE:NUM?`
Response: `CH1:ACQUIRE:NUM 10;`

Related Commands: `ACQUIRE:MODE`

CHx:AUTO

Purpose: Automatically determines the range, offset, timebase, and trigger level for the current input signal.

Type: Operational

Command Syntax: `CHx:AUTO`

Description: This command attempts to automatically determine setups to capture the current input signal. This is done by making repeated acquisitions with various setups on a trial-and-error basis. The command will typically complete within three seconds. The signal must be repetitive and within the range of instrument capabilities.

Invoking this command will result in changes to channel settings for RANGE, COUPLING, OFFSET, SAMPLE:INTERVAL, PRETRIGGER, TRIGGER:SOURCE, AND TRIGGER:LEVEL. WINDOW:LOC will be made the same as ACQUIRE:LOC, if it is not already. COUPLING will always be set to AC.

This command invokes an algorithm that attempts to capture one full cycle of the signal and position it for maximum voltage resolution. Due to unknown signal complexity, it is possible that the algorithm will fail to find a good fit. The user may adjust pulse analysis methods to assist the algorithm: WINDOW:START and WINDOW:STOP will influence the number of cycles captured and TMODE and BMODE will help for different waveshapes.

Both channels must be stopped before sending the `AUTO` command. If not, the channels will continue to run but the `AUTO` command will terminate and generate an error.

Examples: Command: `CH1:AUTO;STATUS?`
Response: `COMPLETE;`

Related Commands: None

CHx:COMPARE:HIGH

Purpose: Sets the upper boundary value for a parametric measurement compare.

Type: Setting

Command Syntax: `CHx:COMPARE:HIGH { REAL | OFF }`

Command Parameters: <REAL> is the upper bound for a parametric compare. The units are determined by the type of parametric measurement set in COMPARE:TYPE. The range and resolution are determined by hardware settings. COMPARE:HIGH is disabled using the OFF parameter.

Default Value: OFF

Query Syntax: `CHx:COMPARE:HIGH?`

Query Parameters: None

Query Response: The response will be <REAL> representing the upper bound that must not be exceeded. Query responses will always have a five digit mantissa.

Description: This command provides access to the upper bound criterion for a parametric type test. The parameter describes the value that must not be exceeded, else the compare will fail.

To disable COMPARE:HIGH, set the parameter to OFF.

Examples: Command: `CH1:COMPARE:HIGH 1.02;:CH1:COMPARE:HIGH?`
Response: `CH1:COMPARE:HIGH 1.0200E+00;`

Related Commands: `COMPARE:LOW, COMPARE:TYPE`

CHx:COMPARE:LOC

Purpose: Sets the waveform location to be used when envelope comparing the acquired waveform.

Type: Setting

Command Syntax: `CHx:COMPARE:LOC <INTEGER>`

Command Parameters: <INTEGER> is the waveform location that contains envelope data. The range for waveform location is 1 to 8 inclusive.

Default Value: 3 for CH1, 4 for CH2

Query Syntax: `CHx:COMPARE:LOC?`

Query Parameters: None

Query Response: Response will be <INTEGER> representing the waveform location containing envelope data.

Description: This command provides access to the waveform location to be used in an envelope comparison.

Examples: Command: `CH1:COMPARE:LOC 3;:CH1:COMPARE:LOC?`
Response: `CH1:COMPARE:LOC 3;`

Related Commands: `COMPARE:TYPE`

CHx:COMPARE:LOW

Purpose: Sets the lower boundary value for a parametric measurement compare.

Type: Setting

Command Syntax: `CHx:COMPARE:LOW { REAL | OFF }`

Command Parameters: <REAL> is the lower bound for a parametric compare. The units are determined by the type of parametric measurement set in COMPARE:TYPE. The range and resolution are determined by hardware settings.

COMPARE:LOW may be disabled with OFF.

Default Value: OFF

Query Syntax: `CHx:COMPARE:LOW?`

Query Parameters: None

Query Response: The response will be <REAL> indicating the lower parametric test bound, not to be exceeded. Query responses will always have a five digit mantissa.

Description: This command provides access to the lower bound criterion for a parametric type test. The parameter describes the value that must not be exceeded, else the compare will fail.

To disable COMPARE:LOW, set the parameter to OFF.

Examples:
Command: `CH1:COMPARE:LOW .562;:CH1:COMPARE:LOW?`
Response: `CH1:COMPARE:LOW 562.00E-03;`

Related Commands: `COMPARE:NOMINAL, COMPARE:HIGH, COMPARE:TYPE`

CHx:COMPARE:NOMINAL

Purpose: Sets the expected value for a parametric measurement compare. This is information only.

Type: Setting

Command Syntax: `CHx:COMPARE:NOMINAL { REAL | OFF }`

Command Parameters: <REAL> is the expected value. The units are determined by the type of parametric measurement set in COMPARE:TYPE. The range and resolution are determined by hardware settings.

COMPARE:NOMINAL is not used for pass/fail decision making. It is available to record the expected value and thus derive the ± margins using COMPARE:HIGH and COMPARE:LOW.

Default Value: None

Query Syntax: `CHx:COMPARE:NOMINAL?`

Query Parameters: None

Query Response: The response will be <REAL> indicating the expected value to be measured. Query responses will always have a five digit mantissa.

Description: This command allows the user to record the expected value to be measured.

Examples: Command: `CH1:COMPARE:NOMINAL 1500;:CH1:COMPARE:NOMINAL?`
Response: `CH1:COMPARE:NOMINAL 1.500E+03;`

Related Commands: `COMPARE:LOW, COMPARE:HIGH, COMPARE:TYPE`

CHx:COMPARE:TYPE

Purpose: Sets the type of comparison to be performed in an channel sequence; template or parametric.

Type: Setting

Command Syntax:
```
CHx:COMPARE:TYPE { NONE | TEMPLATE | MAX | MIN | MID | PK2PK
| RMS | PERIOD | WIDTH | FREQ | DUTY | RISE | FALL | TOP |
BASE | AMPLITUDE | TIMEAB | GAIN | PHASE | TOVER | BOVER |
AREA | ENERGY | MEAN }
```

Command Parameters:

NONE	- no comparison will be performed
TEMPLATE	- a template compare will be performed
MAX	- compares the computed maximum voltage of the waveform
MIN	- compares the computed minimum voltage of the waveform
MID	- compares the computed midpoint voltage of the waveform
PK2PK	- compares the computed peak-peak voltage of the waveform
RMS	- compares the computed root-mean-squared voltage of the waveform
PERIOD	- compares the computed period of the waveform
WIDTH	- compares the computed width of the waveform
FREQ	- compares the computed frequency of the waveform
DUTY	- compares the computed duty factor of the waveform
RISE	- compares the computed risetime of the first positive pulse edge found in the waveform
FALL	- compares the computed falltime of the first negative pulse edge found in the waveform
TOP	- compares the computed TOP
BASE	- compares the computed BASE
AMPLITUDE	- compares the computed AMPLITUDE
TOVER	- Top overshoot
BOVER	- Base overshoot
GAIN	- compares the computed the gain difference between CH1:ACQUIRE:LOC and CH2:ACQUIRE:LOC
PHASE	- compares the computed the phase difference between CH1:ACQUIRE:LOC and CH2:ACQUIRE:LOC
AREA	- area under the curve
ENERGY	- energy of the waveform

Default Value: NONE

Query Syntax:
```
CHx:COMPARE:TYPE?
```

Query Parameters: None

Query Response: The response will be any of the valid command parameters.

Description: This command allows the selection of the type of comparison to be performed on the acquired data.

CHx:COMPARE:TYPE (continued)

Examples:

Command: `CH1:COMPARE:TYPE TEMPLATE;:CH1:COMPARE:TYPE?`
Response: `CH1:COMPARE:TYPE TEMPLATE;`

Related Commands:

`COMPARE:PARAM, COMPARE LOC, COMPARE:HIGH, COMPARE:LOW`

CHx:CONNECT:TRIGGEROUT

Purpose: Routes the internal channel trigger signal to the VXIbus.

Type: Setting

Command Syntax: `CHx:CONNECT:TRIGGEROUT { OFF | TTLTRG0 | TTLTRG1 | ... | TTLTRG7 }`

Command Parameters: OFF - no trigger signals are routed
TTLTRGx - the trigger signal for the specified channel is routed to the specified TTLTRG line.

Default Value: OFF

Query Syntax: `CHx:CONNECT:TRIGGEROUT?`

Query Parameters: None.

Query Response: The response will be OFF or TTLTRGx.

Description: This command allows the trigger signal for the specified channel to be routed to the VXIbus. Only one channel can be selected to source a VXI trigger at a time.

Examples:
Command: `CH1:CONNECT:TRIGGEROUT TTLTRG5;:`
`CH1:CONNECT:TRIGGEROUT?`
Response: `CH1:CONNECT:TRIGGEROUT TTLTRG5;`

Related Commands: `TRIGGER:SOURCE`

CHx:COUPLING

Purpose: Allows the user to AC or DC couple the input signal to the measurement hardware.

Type: Setting

Command Syntax: `CHx:COUPLING { AC | DC }`

Command Parameters: The parameter will be either "AC" or "DC".

Default Value: DC

Query Syntax: `CHx:COUPLING?`

Query Parameters: None

Query Response: The response will be either "AC" or "DC".

Description: Selects the signal coupling for the specified channel. The valid options are AC or DC coupling. A command error will be generated if neither option is found.

Examples: Command: `CH1:COUPLING AC;:CH1:COUPLING?`
Response: `CH1:COUPLING AC;`

Related Commands: `RANGE, OFFSET`

Note:

The CHANNEL 1 and CHANNEL 2 inputs have an impedance of 22 nF in series with 1 Meg Ω when AC coupling is selected.

CHx:OFFSET

Purpose: OFFSET defines the input signal voltage that appears "center screen" (the middle value for the A/D converter).

Type: Setting

Command Syntax: `CHx:OFFSET <REAL>`

Parameters: <REAL> represents the offset voltage

Default Value: 0

Query Syntax: `CHx:OFFSET?`

Query Parameters: None

Query Response: The response will be <REAL> representing the offset voltage.

Description: Selects the waveform offset for the specified channel. The valid option range is ± 50 % of full scale range with a setting resolution of .5%. A command error is generated if the absolute value of the option exceeds 50% of range.

The current parameter arqument of this setting may change when the input range is changed. A warning message is generated when this occurs. Refer to Table 1-4, VX4250 Range, Offset, and Trigger Capabilities for more information.

This command is useful for bringing a biased signal into the A/D converter's range for best resolution.

Examples: Command: `CH1:OFFSET 10;:CH1:OFFSET?`
Response: `CH1:OFFSET 10.00E+00;`

Related Commands: `RANGE, COUPLING`

CHx:PRETRIGGER

Purpose: Sets the amount of pretrigger information.

Type: Setting

Command Syntax: `CHx:PRETRIGGER { 0 | 128 | 1024 | 1920 }`

Command Parameters: The parameter specifies the number of samples to be held as pretrigger data. Valid options include 0 , 128 (1/15 of the record), 1024 (1/2 of the record), 1920 (15/16 of the record).

PRETRIGGER 0 is only valid in equivalent time modes (sample intervals less than 5 ns).

PRETRIGGER 128, 1024 and 1920 are only valid for real-time sampling modes (10 ns and above).

Default Value: 128

Query Syntax: `CHx:PRETRIGGER?`

Query Parameters: None

Query Response: The response is <INTEGER> representing the number of pretrigger samples.

Description: Selects the amount of pretrigger information saved for the specified channel.

Note that pretrigger data is not available if the SAMPLE:INTERVAL is faster than 10 ns (10E-9). Please refer to Table 1-5. VX4250 Timebase Capabilities.

Examples: Command: `CH1:PRETRIGGER 1024;:CH1:PRETRIGGER?`
Response: `CH1:PRETRIGGER 1024;`

Related Commands: None

CHx:RANGE

Purpose: Allows the user to adapt the measurement hardware to the signal amplitude.

Type: Setting

Command Syntax: `CHx:RANGE <REAL>`

Command Parameters: <REAL> represents the maximum voltage range at the input.

Default Value: 40

Query Syntax: `CHx:RANGE?`

Query Parameters: None

Query Response: <REAL> will be returned, indicating the current range setting.

Description: Selects the full scale range for the specified channel. The valid options for <REAL> are; 40, 16, 8, 4, 1.6, .8, .4, .16, .08 volts.

If the specified value exceeds any of the above, the next highest value is selected. No warning will be generated by this adjustment. A command error will be generated if 40 is exceeded.

The side-effects of changing range are: (1) Trigger level and (2) offset voltages.

Please refer to Table 1-4. VX4250 Range, Offset and Trigger Capabilities.

Examples: Command: `CH1:RANGE 2;:CH1:RANGE?`
Response: `CH1:RANGE 4.000E+00;`

Related Commands: None

CHx:RECALL

Purpose: Restores a previously saved channel sequence setup.

Type: Operational

Command Syntax: `CHx:RECALL <INTEGER>`

Command Parameters: <INTEGER> represents the stored setting location for that channel. The range is 1 to 8 inclusive.

Description: This command allows the user to restore the channel sequence to a previous state.

Examples: Command: `CH1:RECALL 2`

Related Commands: `SAVE`

CHx:RESULT?, CHx:RESULT:LONG?

Purpose: Returns the details of the comparison in the last channel sequence.

Type: Query only

Default Value: None

Query Syntax:
```
RESULT? [step]

RESULT:LONG? [step]
```

Query Parameters: [step] optionally recalls the results from the execution of a test step.

Query Response: The RESULT? response will indicate (1) if the last compare passed/failed, (2) the measured value.

The RESULT:LONG? response will indicate (1) the measurement type, (2) if the last compare passed/failed, (3) the measured value, (4) the upper bound, (5) the lower bound, (6) the acquisition mode.

Description: This query allows details of the last comparison to be obtained.

If an argument is provided, then the results of the test step are obtained. For this information to be available, the channel sequence must have been run with STEP:MODE RUN.

Examples:
```
Command: CH1:RESULT?
Response: PASS, 1.010E+03;
Command: CH1:RESULT:LONG? 1
Response: FREQ, FAIL, 1.010E+03, 2.000E+03, 1.0500E+03,
          SMOOTH;
```

Related Commands: COMPARE:TYPE, COMPARE:HIGH, COMPARE:LOW

CHx:SAMPLE:INTERVAL

Purpose: Sets the value for the measurement hardware timebase.

Type: Setting

Command Syntax: `CHx:SAMPLE:INTERVAL <REAL>`

Command Parameters: <REAL> represents the time between samples.

Default Value: 100E-9 (100 ns)

Query Syntax: CHx:SAMPLE:INTERVAL?

Query Parameters: None

Query Response: The response will indicate the current sample interval.

Description: This command allows the user to select the acquisition rate of the instrument. INTERVAL sets the time between samples. WINDOW sets the total acquisition time for 2044 samples. Thus, WINDOW indirectly sets INTERVAL and INTERVAL indirectly sets WINDOW.

Please refer to Table 1-5. VX4250 Timebase Capabilities.

Examples: Command: `CH1:SAMPLE:INTERVAL 1E-6;:CH1:SAMPLE:INTERVAL?`
Response: `CH1:SAMPLE:INTERVAL 1.000E-06;`

Related Commands: `SAMPLE:WINDOW`

CHx:SAMPLE:WINDOW

Purpose: Sets the value for the measurement hardware timebase.

Type: Setting

Command Syntax: `CHx:SAMPLE:WINDOW <REAL>`

Command Parameters: <REAL> is the time represented in the record.

Default Value: 204.4E-6 (204.4 µs)

Query Syntax: `CHx:SAMPLE:WINDOW?`

Query Parameters: None

Query Response: The response will indicate the elapsed time during an acquisition.

Description: This command allows the user to select the acquisition rate of the instrument. INTERVAL sets the time between samples. WINDOW sets the total acquisition time for 2044 samples. Thus, WINDOW indirectly sets INTERVAL and INTERVAL indirectly sets WINDOW.

If the sample window specified is not exactly one that exists, the next *largest* sample window is used. An error is not generated when this occurs.

Please refer to Table 1-5. VX4250 Timebase Capabilities.

Examples: Command: `CH1:SAMPLE:WINDOW 1E-3;:CH1:SAMPLE:WINDOW?`
Response: `CH1:SAMPLE:WINDOW 1.022E-03;`

Related Commands: `SAMPLE:INTERVAL`

CHx:SAVE

Purpose: Stores a channel sequence setup (step) in a stored setting location.

Type: Operational

Command Syntax: `CHx:SAVE <INTEGER>`

Command Parameters: <INTEGER> represents the stored setting location for that channel. The range is 1 to 8 inclusive.

Description: This command allows the user to store the channel sequence for future recall.

Examples: Command: `CH1:SAVE 2`

Related Commands: `RECALL`

CHx:START

Purpose: Starts the channel sequence.

Type: Operational

Command Syntax: `CHx:START`

Command Parameters: None

Description: This command starts the channel sequence.

Examples: Command: `CH1:START`

Related Commands: `STOP`

CHx:STATS?

Purpose: Returns pass/fail tally statistics.

Type: Query only

Default Value: None

Query Syntax: `CHx:STATS? [step]`

Query Parameters: [step] optionally recalls the tally from the execution of a test step.

Query Response: The STATS? response will indicate (1) the number of passed tests, (2) the number of results that were under the COMPARE:LOW boundary and (3) the number of results that were over the COMPARE:HIGH boundary.

Description: This query returns pass/fail tally information. If an argument is provided, then the tally of the test step are obtained. For this information to be available, the channel sequence must have been run with STEP:MODE RUN.

If the channel sequence has not been run, the results will be 0,0,0.

Examples: Command: `CH1:STATS?`
Response: `103,2,1;` *103 in bounds, 2 under, 1 over*

Related Commands: `COMPARE:HIGH, COMPARE:LOW`

CHx:STATUS?

Purpose: Returns the state of the channel sequence.

Type: Operational, Query only

Command Syntax: `CHx:STATUS?`

Command Parameters: None

Description: This query returns the state of the current channel sequence.

If the sequence is waiting for a trigger then the response will be WAITING.

If the sequence is still in process, the response will be RUNNING.

If the sequence has terminated, the response will be COMPLETE.

Examples:

Command: `CH1:STATUS?`
Response: `WAITING;`

Command: `CH1:STATUS?`
Response: `RUNNING;`

Command: `CH1:STATUS?`
Response: `COMPLETE;`

Related Commands: `WAIT?`

CHx:STEP:COUNT?

Purpose: Returns the number of passes through the channel sequence since the last START.

Type: Query only

Command Syntax: `CHx:STEP:COUNT?`

Command Parameters: None

Description: This query returns the number of passes through the acquisition sequence.

Examples: Command: `CH1:STEP:COUNT?`
Response: `10;`

Related Commands: None

CHx:STEP:EXIT

Purpose: Sets the condition for channel step termination.

Type: Setting

Command Syntax: `CHx:STEP:EXIT { PASS | FAIL | NPASS | NFAIL | TERMCOUNT | SINGLE | NEVER }`

Command Parameters:

PASS - run the step until the pass condition is true.

FAIL - run the step until the fail condition is true.

NPASS - run the step until the pass condition is true or for NUMLOOP times.

NFAIL - run the step until the fail condition is true or for NUMLOOP times.

TERMCOUNT - run the step for NUMLOOP times.

SINGLE - run the step once.

NEVER - run the step until commanded to stop with STOP.

Default Value: SINGLE

Query Syntax: `CHx:STEP:EXIT?`

Query Parameters: None

Query Response:

Description: This command selects the channel step termination condition. When the step ends, the last acquired waveform is in the location specified by ACQUIRE:LOC.

PASS and FAIL refers to the COMPARE step.

NPASS and NFAIL allow a preset number of compares to be performed.

TERMCOUNT refers to the STEP:NUMLOOP specification which sets the number of times to run the channel step.

SINGLE and NEVER refer to special cases of the loop control that override NUMLOOP.

Examples: Command: `CH1:STEP:EXIT NPASS;:CH1:STEP:NUMLOOP 100`

Related Commands: `STEP:NUMLOOP`

CHx:STEP:MODE

Purpose: Sets the channel sequence to either single, run or loop.

Type: Setting

Command Syntax: `CHx:STEP:MODE { SINGLE | RUN | LOOP }`

Command Parameters: SINGLE - run the current test step only

RUN - run all steps at once

LOOP - loop on all steps continuously

Default Value: SINGLE

Description: This command offers two states for execution of the channel sequence; single or run.

If single is selected, when the channel is started, only the current settings of the channel sequence will be used.

If run is selected, when the channel is started, stored setting number 1 will be recalled, executed, and when complete, stored setting number 2 will be executed, and so on until all eight stored settings steps have been executed, after which the channel will enter the stopped state.

If loop is selected, the channel sequence will run continuously until the `STOP` command is received.

Examples: Command: `CH1:STEP:MODE RUN;:CH1:START;CH1:WAIT?`
Response: `COMPLETE;`

Related Commands: None

CHx:STEP:NUMLOOP

Purpose: Sets the number of channel steps to perform before stopping.

Type: Setting

Command Syntax: `CHx:STEP:NUMLOOP <INTEGER>`

Command Parameters: <INTEGER> is the number of channel steps to perform. The valid range is 1 to 65,535.

Default Value: 1

Query Syntax: `CHx:STEP:NUMLOOP?`

Query Parameters: None

Query Response: The response will be <INTEGER> indicating the setting for NUMLOOP.

Description: This command allows access to the number of times through the channel step.

Examples: Command: `CH1:STEP:NUMLOOP 100`

Related Commands: `STEP:EXIT`

CHx:STOP

Purpose: Stops a currently active channel sequence.

Type: Operational

Command Syntax: `CHx:STOP`

Command Parameters: None

Description: This command forces the current channel sequence to stop.

If STOP is executed during an acquisition, the resulting waveform stored in ACQUIRE:LOC will be from the previous acquisition.

Examples: Command: `CH1:STOP`

Related Commands: `START`

CHx:TRIGGER:EXT:COUPLING

Purpose: Sets the external trigger signal coupling.

Type: Setting

Command Syntax: `CHx:TRIGGER:EXT:COUPLING { AC | DC }`

Command Parameters: AC - "AC" couples the external trigger signal

DC - "DC" couples the external trigger signal

Default Value: DC

Query Syntax: `CHx:TRIGGER:EXT:COUPLING?`

Query Parameters: None

Query Response: The response will be either "AC" or "DC"

Description: Selects the coupling of the external trigger signal for the specified channel.

Examples:
Command: `CH1:TRIGGER:EXT:COUPLING AC;:`
`CH1:TRIGGER:EXT:COUPLING?`
Response: `CH1:TRIGGER:EXT:COUPLING AC;`

Related Commands: `TRIGGER:LEVEL, TRIGGER:SOURCE, TRIGGER:SLOPE`

Note:

The EXT TRIG 1 and EXT TRIG 2 inputs have an impedance of 100 nF in series with 1 Meg Ω when AC coupling is selected.

CHx:TRIGGER:HFR

Purpose: Sets the trigger circuitry to high frequency reject.

Type: Setting

Command Syntax: `CHx:TRIGGER:HFR { ON | OFF }`

Command Parameters: ON - turns high frequency reject on
OFF - turns high frequency reject off

Default Value: OFF

Query Syntax: `CHx:TRIGGER:HFR?`

Query Parameters: None

Query Response: The response will be either "ON" or "OFF".

Description: Selects the high frequency reject trigger circuit. When on, sensitivity of signals greater than 1 MHz is significantly reduced.

Examples: Command: `CH1:TRIGGER:HFR ON;:HFR?;`
Response: `CH1:TRIGGER:HFR ON;`

Related Commands: `TRIGGER:LEVEL, TRIGGER:SOURCE, TRIGGER:SLOPE`

CHx:TRIGGER:LEVEL

Purpose: Sets the trigger level.

Type: Setting

Command Syntax: `CHx:TRIGGER:LEVEL <REAL>`

Command Parameters: <REAL> sets the trigger level in voltage. The acceptable values for <REAL> vary with the RANGE selected.

Default Value: 0

Query Syntax: `CHx:TRIGGER:LEVEL?`

Query Parameters: None

Query Response: The response will be <REAL> representing the trigger level as a voltage.

Description: Selects the level of the trigger signal for the specified channel in volts.

The TRIGGER:LEVEL command is valid for internal or external trigger sources. For internal sources, the acceptable range for the level is ± 50 % of full scale input signal, in .5% steps resolution. With external sources, the valid range is ± 5 V and the resolution is 40 mV.

The current parameter argument of this setting may change when the input range is changed. If this occurs, a warning message will be generated. Refer to Table 1-4. VX4250 Range, Offset, and Trigger Capabilities for more information.

Examples: Command: `CH1:TRIGGER:LEVEL 1.10;:CH1:TRIGGER:LEVEL?`
Response: `CH1:TRIGGER:LEVEL 1.100E+00;`

Related Commands: `TRIGGER:SLOPE, TRIGGER:SOURCE, TRIGGER:EXT:COUPLING`

CHx:TRIGGER:SLOPE

Purpose: Selects the slope of the trigger signal.

Type: Setting

Command Syntax: `CHx:TRIGGER:SLOPE { PLUS | MINUS }`

Command Parameters: Valid parameters are: PLUS or MINUS.

Default Value: PLUS

Query Syntax: `CHx:TRIGGER:SLOPE?`

Query Parameters: None

Query Response: The response will be PLUS or MINUS.

Description: Selects the slope of the trigger signal. "PLUS" refers to a positive going transition and "MINUS" refers to a negative going transition.

Examples: Command: `CH1:TRIGGER:SLOPE PLUS;:CH1:TRIGGER:SLOPE?`
Response: `CH1:TRIGGER:SLOPE PLUS;`

Related Commands: `TRIGGER:LEVEL, TRIGGER:EXT:COUPLING, TRIGGER:SOURCE`

CHx:TRIGGER:SOURCE

Purpose: Selects the trigger source for the channel.

Type: Setting

Command Syntax: `CHx:TRIGGER:SOURCE { INT | EXT | XCH | FREERUN | TTLTRGx | ECLTRGx }`

Command Parameters: Valid parameters are: INT, EXT, XCH, FREERUN, TTLTRGx and ECLTRGx. INT is the default after power-up.

INT selects the trigger source from the channel's input signal;
EXT selects the trigger source from the channel's external BNC input.

XCH selects the trigger source from the other channel's trigger (whatever is selected for it).

FREERUN continuously (and asynchronously) triggers the acquisition hardware.

TTLTRGx selects VXI system triggers (TTLTRG0—TTLTRG7);
ECLTRGx selects VXI system triggers (ECLTRG0 or ECLTRG1)

Default Value: INT

Query Syntax: `CHx:TRIGGER:SOURCE?`

Query Parameters: None

Query Response: The response will be one of the valid command parameters.

Description: Selects the trigger source for the specified channel. The default is for the internal trigger from the same channel—that is, the input signal for that channel.

INT and XCH sources inherit the coupling characteristic (AC or DC) as selected by COUPLING for that channel. Coupling for the EXT source may be separately selected with the TRIGGER:EXT:COUPLING command.

XCH cannot be specified as a source for both channels at the same time.

FREERUN means the acquisition hardware is always triggered.

Note that TRIGGER:LEVEL does not affect digital triggers. However, TRIGGER:SLOPE does affect digital triggers with some limitations.

Examples: Command: `CH1:TRIGGER:SOURCE INT;:CH1:TRIGGER:SOURCE?`
Response: `CH1:TRIGGER:SOURCE INT;`

Related Commands: `TRIGGER:LEVEL, TRIGGER:EXT:COUPLING, TRIGGER:SLOPE`

CHx:WAIT?

Purpose: Holds off execution of more commands until acquisition is complete or the timeout value is reached.

Type: Operational, Query only

Default Value: None

Query Syntax: `CHx:WAIT? [timeout value]`

Query Parameters: [timeout value] is an <INTEGER> that specifies the number of milliseconds to wait before forcing a response. The valid range is 1 to 100,000.

Query Response: The returned message will be "COMPLETE" if timeout is not exceeded and acquisition is done.

The returned message will be "WAITING" if timeout is exceeded and no trigger has been received.

The returned message will be "RUNNING" if timeout is exceeded and acquisition is still in progress (usually for slow sample rates).

Description: This command is used to synchronize communications activity. Its primary use is to postpone further commands until channel sequence has completed.

If timeout is not specified, the default value will be 0 (and will function like STATUS?).

Examples: Command: `CH1:START;:CH1:WAIT? 500;:CURVE?`
Response: `COMPLETE;CURVE . . .;`

Related Commands: `STATUS?`

CH1?

Purpose: Returns all current settings for channel 1.

Type: Query only

Query Syntax: `CH1?`

Query Parameters: None

Query Response: The response will be all channel sequence setting commands, with arguments reflecting the current state of CH1.

Description: This query is used to return all information relating to channel 1 channel sequence setup.

Examples: Command: `CH1?`
Response: `:CH1:ACQUIRE:NUM 1;LOC 1;MODE NORMAL . . .;`

Related Commands: `CH2?`

CH2?

Purpose: Returns all current settings for channel 2.

Type: Query only

Query Syntax: `CH2?`

Query Parameters: None

Query Response: The response will be all channel sequence setting commands, with arguments reflecting the current state of CH2.

Description: This query is used to return all information relating to channel 2 channel sequence setup.

Examples: Command: `CH2?`
Response: `:CH2:ACQUIRE:NUM 1;LOC 1;MODE NORMAL . . .;`

Related Commands: `CH1?`

This page intentionally left blank.

Instrument Commands

This section provides an alphabetically arranged description of instrument commands—those that apply globally to the operation of both channels, data analysis and data access.

Instrument commands do not use the **CHx:** header.

In general, instrument level commands that access the data (like CURVE? and MAX?) should *not* be used during an acquisition, as the data may be changing while the commands are executing. Detailed information on how data analysis parameters are calculated can be found in Section 4, *Advanced Operation*, and Appendix B, *Pulse Measurement Definitions*.

AMPLITUDE?	Returns the amplitude of a waveform
ADD	Algebraically adds two waveforms
AREA?	Returns the area of a waveform in volts/second
BASE?	Returns the base magnitude of a waveform
BMODE	Sets how base is determined
BTLINES	Sets the base or top value of a waveform
CURVE	Allows access to the waveform location data
DATA:COUNT	Sets the length of the waveform segment to be used with CURVE? and CURVE
DATA:ENCDG	Sets the encoding format for CURVE command/query access
DATA:LOC	Sets the waveform location to be used with CURVE? and CURVE
DATA:START	Sets the start of the waveform segment to be used with CURVE? and CURVE
DESE, DESE?	Controls the setting and reading of the Device Event Status Register
DUTY?	Returns the duty cycle of a waveform
ENERGY?	Returns the energy of a waveform
EXTMODE	Sets how RMS, mean, area, and energy are calculated
FALL?	Returns the falltime of a waveform
FREQUENCY?	Returns the frequency of a waveform

GAIN?	Returns the gain difference between two waveforms
MAX?	Returns the max voltage of a waveform
MEAN?	Returns the mean voltage of a waveform
MID?	Returns the mid voltage of a waveform
MIN?	Returns the min voltage of a waveform
PERIOD?	Returns the period of a waveform
PHASE?	Returns the phase difference between two waveforms
PK2PK?	Returns the peak-to-peak voltage of a waveform
REF:LOC	Specifies the reference waveform location used in two channel measurements (GAIN and PHASE)
REFCRS?	Returns the sample point in the waveform segment for the selected proximal, mesial, and distal points
REFLINES?	Sets proximal, mesial, and distal values
REFUNITS	Sets the units of measure for REFCRS and REFUNITS
REFVALUES?	Returns proximal, mesial, and distal values
RISE?	Returns the risetime of a waveform
RMS?	Returns the RMS voltage of a waveform
SUB	Algebraically subtracts two waveforms
TMODE	Sets how top is determined
TOP?	Returns top magnitude of a waveform
WIDTH?	Returns pulse width of a waveform
WINDOW:LOC	Defines the location of the waveform segment to be used
WINDOW:START	Defines the start of the waveform segment to be used
WINDOW:STOP	Defines the end of the waveform segment to be used

ADD

Purpose: Algebraically adds two waveforms.

Type: Operational

Command Syntax: `ADD <source1>, <source2>, <destination>`

Command Parameters:
<source1> = specifies the first source waveform location (1—8)
<source2> = specifies the second source waveform location (1—8)
<destination> = specifies the result waveform location

Description: This command allows two waveforms to be algebraically added in raw data values.

Examples: Command: `ADD 1,2,3`

Related Commands: `SUB`

AMPLITUDE?

Purpose: Returns the AMPLITUDE of the waveform segment described by WINDOW.

Type: Query only

Query Syntax: `AMPLITUDE?`

Query Parameters: None

Query Response: The measured value

Description: This query will return the AMPLITUDE of a waveform segment.
Amplitude is the top – base amplitude.
A waveform segment is described by WINDOW:LOC, WINDOW:START and WINDOW:STOP.

Amplitude is calculated from TOP–BASE. TOP and BASE utilize histograms to determine their value. PK2PK is calculated from MAX-MIN. For many waveforms, amplitude gives more accurate results because it ignores noise. With some waveforms, such as triangles, amplitude will give erroneous results. Refer to TOP and BASE determination in Section 4, *Advanced Operation*, for more information.

Examples: Command: `AMPLITUDE?`
Response: `AMPLITUDE 1.302E+00;`

Related Commands: `PK2PK, WINDOW:LOC, WINDOW:START, WINDOW:STOP`

AREA?

Purpose: Returns the AREA of the waveform segment described by WINDOW.

Type: Query only

Query Syntax: `AREA?`

Query Parameters: None

Query Response: The measured value

Description: This query will return the AREA of a waveform segment.
Area is the area contained under a waveform.
A waveform segment is described by WINDOW:LOC, WINDOW:START and WINDOW:STOP.

Examples: Command: `AREA?`
Response: `AREA 13.78E-03;`

Related Commands: `EXTMODE, WINDOW:LOC, WINDOW:START, WINDOW:STOP`

BASE?

Purpose: Returns the BASE of the waveform segment described by WINDOW.

Type: Query only

Query Syntax: `BASE?`

Query Parameters: None

Query Response: The measured value

Description: This query will return the BASE of a waveform segment.
Base is the baseline voltage representing the nominal base of the waveform.
A waveform segment is described by WINDOW:LOC, WINDOW:START and WINDOW:STOP.

Examples: Command: `BASE?`
Response: `BASE 253.7E-03;`

Related Commands: `WINDOW:LOC, WINDOW:START, WINDOW:STOP`

BMODE

Purpose: Specifies how the BASE voltage will be determined.

Type: Setting

Command Syntax: `BMODE:<arg1> [arg2]`

Command Parameters:

<arg1> = { ABSOLUTE | HISTOGRAM | MINMAX }

ABSOLUTE - Selects the base value through BTLINES

MINMAX - Sets the base value as the minimum waveform value

HISTOGRAM - Selects the histogram distribution method to set the base value. Note that MEAN or MODE must be specified in the following parameter.

[arg2] = { MEAN | MODE }

MEAN - Sets the BASE value of the waveform to the MEAN value of the histogram distribution

MODE - Sets the BASE value of the waveform to the MODE value (most frequently occurring value) of the histogram distribution

Default Value: HISTOGRAM MODE

Description: This command allows you to change the way BASE voltage is determined. Along with TMODE, this allows control over the way pulse analysis functions and thus obtains reliable measurements from complex waveforms.

Examples:

Command: `BMODE:HISTOGRAM MEAN;`
Command: `BMODE:HISTOGRAM MODE;`
Command: `BMODE:MINMAX;`
Command: `BMODE:ABSOLUTE;:REFUNITS VOLTS;:BTLINES:BASELINE -1.0;`

Related Commands: `TMODE, REFUNITS, BTLINES`

BOVER?

Purpose: Returns the BASE OVERSHOOT of the waveform segment described by WINDOW.

Type: Query only

Query Syntax: `BOVER?`

Query Parameters: None

Query Response: The measured value

Description: This query will return the BOVER of a waveform segment.
Bover is the percentage of base overshoot.
A waveform segment is described by WINDOW:LOC, WINDOW:START and WINDOW:STOP.

Examples: Command: `BOVER?`
Response: `BOVER 312.7E-03;`

Related Commands: `WINDOW:LOC, WINDOW:START, WINDOW:STOP`

BTLINES

Purpose: Specifies the TOP or BASE value of a waveform when TMODE or BMODE is set to ABSOLUTE.

Type: Setting

Command Syntax: `BTLINES:<arg1> <arg2>`

Command Parameters:
<arg1> = { TOPLINE | BASELINE }
<arg2> = <REAL> - the absolute voltage level assigned to TOP or BASE

Description: This command allows an absolute voltage level to be assigned to a waveform TOP or BASE.

Examples:
Command: `TMODE:ABSOLUTE;:BTLINES:TOPLINE 1.0;`
Command: `BMODE:ABSOLUTE;:BTLINES:BASELINE 0.5;`

Related Commands: `TMODE, BMODE, REFUNITS`

DATA:ENCDG

Purpose: Sets the encoding format for CURVE command/query access.

Type: Setting

Command Syntax: `DATA:ENCDG { ASCII | RIB | RPB }`

Command Parameters: ASCII - data will be passed as three digit numbers separated by commas

RIB - data will be passed in arbitrary binary blocks (+127 to -128)

RPB - data will be passed in arbitrary binary blocks (0 to 255)

Default Value: RIB

Query Syntax: `DATA:ENCDG?`

Query Parameters: None

Query Response: `DATA:ENCDG RPB`

Description: This command specifies the format of the data passed with the `CURVE` and `CURVE?` commands.

Examples: Command: `DATA:ENCDG RIB;:DATA:ENCDG?`
Response: `DATA:ENCDG RIB;`

Related Commands: `CURVE, CURVE?`

DATA:LOC

Purpose: Sets the waveform location to be used with CURVE? and CURVE.

Type: Setting

Command Syntax: `DATA:LOC <waveform location>`

Command Parameters: <waveform location> is an integer specifying the source for CURVE? The valid range is 1—8.

Default Value: 1

Query Syntax: `DATA:LOC?`

Query Parameters: None

Query Response: `DATA:LOC 3`

Description: This command specifies the waveform location to be accessed with the `CURVE?` command.

Examples: Command: `DATA:LOC 1`

Related Commands: `CURVE, CURVE?`

DATA:START

Purpose: Sets the start of the waveform segment to be used with CURVE? and CURVE. This, with DATA:COUNT, allows a portion smaller than the whole waveform to be transferred.

Type: Setting

Command Syntax: `DATA:START <first sample>`

Command Parameters: <first sample> is an integer that specifies the starting point for CURVE? The valid range is 1—2044.

Default Value: 1

Query Syntax: `DATA:START?`

Query Parameters: None

Query Response: `DATA:START 128`

Description: This command sets the start of the waveform segment to be accessed with the `CURVE?` and `CURVE` commands. This, with DATA:COUNT, allows a portion smaller than the whole waveform to be transferred.

Examples: Command: `DATA:START 1100;COUNT 5;LOC 1;:CURVE?`
Response: (returns the waveform segment from location 1, samples 1100 to 1104)

Related Commands: `DATA:COUNT, CURVE, CURVE?`

DESE, DESE?

Purpose: DESE controls the setting and reading of the Device Event Status Enable Register.
DESE? reads the Device Event Status Enable Register.

Type: Setting

Command Syntax: DESE <register data>

Command Parameters: <register data> An integer in the range 0—255. When expressed in binary, this integer represents the bit values of the Device Event Status Register.

Default Value: None

Query Syntax: DESE?

Query Parameters: None

Query Response: DESE 255; An integer in the range of 0—255

Description: The Device Event Status Register is an 8-bit register with bit definitions corresponding to the Device Events Status Enable Register. Bits in the Device Event Status Register (set using the DESE command) allow reporting through the Event Reporting Mechanism. Bits not set in the Device Event Status Register disable reporting of that event.

The response to the DESE? query is an integer in the range of 0—255. When this integer is expressed in binary form, it corresponds to the bit values of the Device Event Status Enable Register.

Note:

For further information regarding the bit definitions of the Event Status Enable Register, refer to Figure 7-1. Register Bit Definitions.

Examples: Command: DESE 255
Command: DESE?
Response: DESE 255;

Related Commands: None

DUTY?

Purpose: Returns the DUTY of the waveform segment described by WINDOW.

Type: Query only

Query Syntax: `DUTY?`

Query Parameters: None

Query Response: The measured value

Description: This query will return the DUTY of a waveform segment.
Duty is the percentage of time the waveform exceeds its mesial value.
A waveform segment is described by WINDOW:LOC, WINDOW:START and WINDOW:STOP.

Examples: Command: `DUTY?`
Response: `DUTY 278.9E-06;`

Related Commands: `WINDOW:LOC, WINDOW:START, WINDOW:STOP`

ENERGY?

Purpose: Returns the ENERGY of the waveform segment described by WINDOW.

Type: Query only

Query Syntax: `ENERGY?`

Query Parameters: None

Query Response: The measured value

Description: This query will return the ENERGY of a waveform segment.
Energy is the energy contained within a waveform.
A waveform segment is described by WINDOW:LOC, WINDOW:START and WINDOW:STOP.

Examples: Command: `ENERGY?`
Response: `ENERGY 2.718E-06;`

Related Commands: `WINDOW:LOC, WINDOW:START, WINDOW:STOP`

EVENT?

Purpose: Returns error and event information.

Type: Query only

Query Syntax: `EVENT?`

Query Parameters: None

Query Response: The response is the error number of the last error in the error queue.

Description: This command will access error queue information. The messages in the error buffer occur at any time, asynchronously to any communications.

Examples: Command: `EVENT?`
Response: `EVENT 100;`

Related Commands: `EVMSG?`

EVMSG?

Purpose: Returns error and event information.

Type: Query only

Query Syntax: `EVMSG?`

Query Parameters: None

Query Response: Error number, "<Description>"

Description: This command will access error queue information. The messages in the error buffer occur at any time, asynchronously to any communications.

Examples: Command: `EVMSG?`
Response: `EVMSG 100;, "Command Error"`

Related Commands: `EVENT?`

EXTMODE

Purpose: Provides horizontal windowing of waveform data for MEAN, RMS, AREA and ENERGY.

Type: Setting

Command Syntax: `EXTMODE { PERIOD | WINDOW }`

Command Parameters: PERIOD - Selects one period of the waveform at the MESIAL crossings

WINDOW - Selects the portion of the waveform according to WINDOW:START and WINDOW:STOP

Default Value: PERIOD

Description: This command allows measurement of MEAN, RMS, AREA and ENERGY over a period or window.

Examples: Command: `EXTMODE WINDOW;`
Command: `EXTMODE PERIOD;`

Related Commands: `MEAN, RMS, AREA, ENERGY`

FALL?

Purpose: Returns the FALL of the waveform segment described by WINDOW.

Type: Query only

Query Syntax: `FALL?`

Query Parameters: None

Query Response: The measured value

Description: This query will return the FALL time of a waveform segment.
Fall is the time between falling proximal and distal points.
A waveform segment is described by WINDOW:LOC, WINDOW:START and WINDOW:STOP.

Examples: Command: `FALL?`
Response: `FALL 1.479E-09;`

Related Commands: `WINDOW:LOC, WINDOW:START, WINDOW:STOP`

FDCBASE?

Purpose: Returns the base (beginning) address of a specific shared memory buffer.

Type: Operational

Default Value: None

Query Syntax: `FDCBASE?[{ DATA | ALLSET | ALL }]`

Query Parameters: DATA - The base address of the shared memory buffer containing waveform locations 1—8

ALLSET - The base address of the stored settings buffer

ALL - The base address of all buffers

Query Response: The response of < INTEGER> will indicate the VME A24 base address of the specified item in physical shared memory. Note the response is in hexadecimal.

Description: Returns the shared memory base address in VME A24 space. This allows the controller to access the shared memory in the instrument. If no argument is given, the beginning address of shared memory is returned.

Examples:

Command: `FDCBASE?` *where is the physical (A24) start of memory?*

Response: `#H200000;` *hexadecimal response*

Related Commands: `FDCSIZE?, FDCLOAD?`

FDCLOAD, FDCLOAD?

Purpose: This allows the user to perform valid shared memory accesses.

Type: Operational

Command Syntax: `FDCLOAD`

Default Value: None

Query Syntax: `FDCLOAD?`

Query Parameters: None

Query Response: The response will always be "1".

Description: Allows movement of waveform data, scaling information, and stored instrument settings using FDC. FDCLOAD is sent by the controller to prepare the instrument for receiving data. FDCLOAD? is sent by the controller to cause the instrument to load the shared memory with data. A response of "1" indicates that the data is in the shared memory and is valid.

Examples: Command: `FDCLOAD?`
Response: `1;`

Related Commands: `FDCBASE?, FDCSIZE?`

FDCSIZE?

Purpose: Returns the size of a specific shared memory buffer. Along with FDCBASE?, this facilitates valid shared memory accesses.

Type: Operational

Query Syntax: `FDCSIZE?[{ DATA | ALLSET | ALL }]`

Query Parameters: DATA - The size of the shared memory buffer containing waveform locations 1—8

ALLSET - The size of the stored settings buffer

ALL - The size of all buffers

Query Response Number of bytes in decimal notation.

Description: The normal response will indicate to the controller, the number of bytes representing the requested information (i.e., the size of the buffer to be read).

A "0" will be returned on unsuccessful completion. If no argument is given, the usable size of shared memory is returned in decimal.

Examples: Command: `FDCSIZE?`
Response: `65528;`

Related Commands: `FDCBASE?, FDCLOAD?`

FREQUENCY?

Purpose: Returns the FREQUENCY of the waveform segment described by WINDOW.

Type: Query only

Query Syntax: `FREQUENCY?`

Query Parameters: None

Query Response: The measured value

Description: This query will return the FREQUENCY of a waveform segment.
Frequency is the inverse of the period.
A waveform segment is described by WINDOW:LOC, WINDOW:START and WINDOW:STOP.

Examples: Command: `FREQUENCY?`
Response: `FREQUENCY 27.62E+03;`

Related Commands: `WINDOW:LOC, WINDOW:START, WINDOW:STOP`

GAIN?

Purpose: Returns the GAIN of the waveform segment described by WINDOW and REF:LOC.

Type: Query only

Query Syntax: `GAIN?`

Query Parameters: None

Query Response: The measured value

Description: This query will return the GAIN of a waveform segment. The source waveform is described by WINDOW:LOC and the reference waveform by REF:LOC. Gain is the amplitude gain between two waveforms.
A waveform segment is described by WINDOW:LOC, WINDOW:START and WINDOW:STOP. The reference waveform is described in REF:LOC.

Examples: Command: `GAIN?`
Response: `GAIN 13.25E+00;`

Related Commands: `WINDOW:LOC, WINDOW:START, WINDOW:STOP, REF:LOC`

ID?

Purpose: Returns the manufacturer, model, and firmware revision number and date.

Type: Query only

Default Value: None

Query Syntax: `ID?`

Query Parameters: None

Query Response: Returns "ID TEK/VX4250,<firmware version>,<date>".

Description: This query returns identity information about the instrument.

Examples: Command: `ID?`
Response: `ID TEK/VX4250,V1.0, Wed, Apr 24 11:03:34 PDT 1991;`

Related Commands: `*IDN?`

INIT

Purpose: Resets all current settings to their power-up default condition.

Type: Operational

Command Syntax: `INIT`

Command Parameters: None

Description: This command will reset all current settings to their power-up default condition, as listed here.

Note:
Defaults apply to both channel 1 and 2.

RANGE	40
OFFSET	0
COUPLING	DC
SAMPLE:INTERVAL	100E-9
TRIGGER:SOURCE	INT
TRIGGER:EXT:COUPLING	DC
TRIGGER:LEVEL	0
PRETRIGGER	128
ACQUIRE:MODE	Normal
ACQUIRE:NUM	1
ACQUIRE:LOC	(1 for CH1, 2 for CH2)
COMPARE:TYPE	None
COMPARE:LOW	0
COMPARE:HIGH	0
COMPARE:LOC	(3 for CH1, 4 for CH2)
STEP:EXIT	Single
STEP:NUMLOOP	1
DATA:LOC	1
DATA:ENCDG	RIB
CONNECT:TRIGGEROUT	OFF

Examples: Command: `INIT`

Related Commands: None

MARGINS

Purpose: Creates a new envelope from an existing envelope.

Type: Operational

Command Syntax: `MARGINS <waveform location#>, <add to upper>, <subtract from lower>`

Command Parameters: <waveform location> specifies the WAVEFORM to "margin"

<add to upper> is an integer that specifies the voltage change to the UPPER bound.

<subtract from lower> is an integer that specifies the voltage change to the LOWER bound.

Description: This command allows acquired envelopes to be modified in the voltage dimension.

Examples: Command: `MARGINS 1,5,-5`

Related Commands: None

MAX?

Purpose: Returns the MAX of the waveform segment described by WINDOW.

Type: Query only

Query Syntax: `MAX?`

Query Parameters: None

Query Response: The measured value

Description: This query will return the MAX of a waveform segment.
Max is the most positive voltage of a waveform.
A waveform segment is described by WINDOW:LOC, WINDOW:START and WINDOW:STOP.

Examples: Command: `MAX?`
Response: `MAX 279.5E-03;`

Related Commands: `WINDOW:LOC, WINDOW:START, WINDOW:STOP`

MEAN?

Purpose: Returns the MEAN of the waveform segment described by WINDOW.

Type: Query only

Query Syntax: `MEAN?`

Query Parameters: None

Query Response: The measured value

Description: This query will return the MEAN of a waveform segment.
Mean is the average magnitude of a waveform.
A waveform segment is described by WINDOW:LOC, WINDOW:START and WINDOW:STOP.

Examples: Command: `MEAN?`
Response: `MEAN 7.293E+00;`

Related Commands: `WINDOW:LOC, WINDOW:START, WINDOW:STOP`

MID?

Purpose: Returns the MID of the waveform segment described by WINDOW.

Type: Query only

Query Syntax: `MID?`

Query Parameters: None

Query Response: The measured value

Description: This query will return the MID of a waveform segment.
Mid is the midpoint voltage of a waveform.
A waveform segment is described by WINDOW:LOC, WINDOW:START and WINDOW:STOP.

Examples: Command: `MID?`
Response: `MID 910.3E-03;`

Related Commands: `WINDOW:LOC, WINDOW:START, WINDOW:STOP`

MIN?

Purpose: Returns the MIN of the waveform segment described by WINDOW.

Type: Query only

Query Syntax: `MIN?`

Query Parameters: None

Query Response: The measured value

Description: This query will return the MIN of a waveform segment.
Min is the most negative voltage of a waveform.
A waveform segment is described by WINDOW:LOC, WINDOW:START and WINDOW:STOP.

Examples: Command: `MIN?`
Response: `MIN 26.79E+00;`

Related Commands: `WINDOW:LOC, WINDOW:START, WINDOW:STOP`

PERIOD?

Purpose: Returns the PERIOD of the waveform segment described by WINDOW.

Type: Query only

Query Syntax: `PERIOD?`

Query Parameters: None

Query Response: The measured value

Description: This query will return the PERIOD of a waveform segment.
Period is the time between the first and third qualified mesial points.
A waveform segment is described by WINDOW:LOC, WINDOW:START and WINDOW:STOP.

Examples: Command: `PERIOD?`
Response: `PERIOD 7.695E-06;`

Related Commands: `WINDOW:LOC, WINDOW:START, WINDOW:STOP`

PHASE?

Purpose: Returns the PHASE of the waveform segment described by WINDOW.

Type: Query only

Query Syntax: `PHASE?`

Query Parameters: None

Query Response: The measured value

Description: This query will return the PHASE of a waveform segment.
Phase is the phase angle difference between two waveforms.
A waveform segment is described by WINDOW:LOC, WINDOW:START and WINDOW:STOP.

Examples: Command: `PHASE?`
Response: `PHASE 79.23E+00;`

Related Commands: `WINDOW:LOC, WINDOW:START, WINDOW:STOP`

PK2PK?

Purpose: Returns the PK2PK of the waveform segment described by WINDOW.

Type: Query only

Query Syntax: `PK2PK?`

Query Parameters: None

Query Response: The measured value

Description: This query will return the PK2PK of a waveform segment.
PK2PK is the difference between the maximum and minimum values.
A waveform segment is described by WINDOW:LOC, WINDOW:START and WINDOW:STOP.

Examples: Command: `PK2PK?`
Response: `PK2PK 1.3;`

Related Commands: `WINDOW:LOC, WINDOW:START, WINDOW:STOP`

REFCRS?

Purpose: Returns the sample point in the waveform segment for the selected proximal, mesial, and distal points.

Type: Query only

Default Value: None

Query Syntax: `REFCRS[:{ FDISTAL | FMESIAL | FPROXIMAL | RDISTAL | RMESIAL | RPROXIMAL | PMESIAL }]?`

Query Parameters:
FDISTAL - the FALLING distal crossing point
FMESIAL - the FALLING mesial crossing point
FPROXIMAL - the FALLING proximal crossing point
DISTAL - the RISING distal crossing point
RMESIAL - the RISING mesial crossing point
RPROXIMAL - the RISING proximal crossing point
PMESIAL - the PERIOD mesial crossing point

Query Response: The measured value interpolated between sample points

Description: This query allows access to the waveform points that have been determined as PROXIMAL, MESIAL and DISTAL.

Examples:
Command: `REFCRS:PMESIAL?`
Response: `REFCRS:PMESIAL 497.6E+00;`

Related Commands: `REFLINES`

REFLINES

Purpose: Allows PROXIMAL, MESIAL and DISTAL points to be set as a percentage or absolute voltage.

Type: Setting

Command Syntax: `REFLINES:<arg1> <arg2>`

Command Parameters:
<arg1> = { PROXIMAL | MESIAL | DISTAL }
<arg2> = <REAL>

Default Value: None

Query Syntax: `REFLINES:<arg1>?`

Query Parameters: <arg1> = { PROXIMAL | MESIAL | DISTAL }

Query Response: The selected value

Description: This command allows PROXIMAL, MESIAL and DISTAL points to be set as a percentage (between TOP and BASE) or as an absolute voltage. This allows flexible control over the way time measurements are calculated for complex waveforms.

Examples: Command: `REFLINES:MESIAL 50`

Related Commands: `REFCRS?, REFUNITS`

REF:LOC

Purpose: Specifies the reference waveform location used in two channel measurements (GAIN and PHASE).

Type: Setting

Command Syntax: `REF:LOC <waveform location>`

Command Parameters: <waveform location> specifies the reference waveform location. This is an integer argument within the range of 1—8 inclusive.

Description: This command assigns the reference location to be used in gain and phase measurements.

Examples: Command: `REF:LOC 3`

Related Commands: `GAIN, PHASE`

REFUNITS

Purpose: Selects the units applied to BTLINES and REFLINES commands.

Type: Setting

Command Syntax: `REFUNITS <arg1>`

Command Parameters: <arg1> = { PERCENT | VOLTS }

Default Value: None

Query Syntax: `REFUNITS?`

Query Parameters: None

Query Response: The measured value in percent or volts

Description: This command sets the units applied to the BTLINES and REFLINES commands.

Examples:
Command: `REFUNITS PERCENT; REFUNITS?`
Response: `REFUNITS PERCENT;`
Command: `REFUNITS VOLTS; REFUNITS?`
Response: `REFUNITS VOLTS;`

Related Commands: `BTLINES, REFLINES`

REFVALUES?

Purpose: Returns the voltage value of a specified waveform crossing point.

Type: Query only

Default Value: None

Query Syntax: `REFVALUES:<arg1>?`

Query Parameters: <arg1> = { DISTAL | MESIAL | PROXIMAL }

Query Response: The measured value for distal, mesial, or proximal points

Description: Returns the voltage value of a specified waveform crossing point.

Examples: Command: `REFVALUES:MESIAL?`
Response: `REFVALUES:MESIAL 1.357E+00;`

Related Commands: `REFCRS?`

RISE?

Purpose: Returns the RISE of the waveform segment described by WINDOW.

Type: Query only

Query Syntax: `RISE?`

Query Parameters: None

Query Response: The measured value

Description: This query will return the RISE of a waveform segment.
Rise is the time between the rising proximal and distal points.
A waveform segment is described by WINDOW:LOC, WINDOW:START and WINDOW:STOP.

Examples: Command: `RISE?`
Response: `RISE 47.26E-06;`

Related Commands: `WINDOW:LOC, WINDOW:START, WINDOW:STOP`

Artisan Technology Group is an independent supplier of quality pre-owned equipment

Gold-standard solutions

Extend the life of your critical industrial, commercial, and military systems with our superior service and support.

We buy equipment

Planning to upgrade your current equipment? Have surplus equipment taking up shelf space? We'll give it a new home.

Learn more!

Visit us at **artisantg.com** for more info on price quotes, drivers, technical specifications, manuals, and documentation.

Artisan Scientific Corporation dba Artisan Technology Group is not an affiliate, representative, or authorized distributor for any manufacturer listed herein.

We're here to make your life easier. How can we help you today?

(217) 352-9330 | sales@artisantg.com | artisantg.com

ARTISAN® TECHNOLOGY GROUP