

**Tektronix  
BitAlyzer/BERTScope  
Remote Control Guide**

[www.tektronix.com](http://www.tektronix.com)



077-0696-01

**Tektronix**

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

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

TEKTRONIX and TEK are registered trademarks of Tektronix, Inc.

## **Contacting Tektronix**

Tektronix, Inc.  
14150 SW Karl Braun Drive  
P.O. Box 500  
Beaverton, OR 97077  
USA

For product information, sales, service, and technical support:

- In North America, call 1-800-833-9200.
- Worldwide, visit [www.tektronix.com](http://www.tektronix.com) to find contacts in your area.

---

# Table of Contents

Preface .....	v
Theory of operation .....	v
Getting started .....	1
Remote control overview .....	1
Set up the instrument for remote control operation .....	2
Command description .....	2
Command syntax .....	3
Features .....	3
Operations .....	4
Parameters .....	4
Command status .....	5
Command error messages and codes .....	5
Communications timeouts .....	6
Command listing .....	7
Generator .....	11
Generator Data Generator .....	17
Generator Pattern Start .....	19
Generator Spread Spectrum Clock .....	20
Generator Clock Input .....	22
Generator Clock Output Positive .....	24
Generator Clock Output Negative .....	27
Generator Clock Output Pos/Neg .....	30
Generator Data Output Positive .....	31
Generator Data Output Negative .....	34
Generator Data Output Pos/Neg .....	36
Generator Trigger .....	37
Generator Error Inject .....	38
Generator Stress Module (GSM) - Stressed Eye .....	41
GSM - Stressed Eye Configure/Enable .....	41
GSM - Stressed Eye Setup .....	42
Detector .....	51
Detector Error Detector .....	54
Detector Start Detect .....	58
Detector Clock Input .....	59
Detector Data Input .....	61
Detector Blank Input .....	63
Detector Trigger .....	64
Detector Results .....	65
Detector Symbol Filtering .....	67

Detector Convenience.....	68
Detector Auto Align Results .....	68
Detector Optical Mode.....	69
Stress Combiner Option.....	71
Stress Combiner command summary .....	72
Stress Combiner commands .....	74
Clock Recovery Option .....	79
Clock recovery command summary .....	81
CRService : Clock Recovery Service .....	84
CRControl : Clock Recovery Control.....	85
Clock Recovery Loop Response .....	94
Clock Recovery SSC Waveform .....	96
Clock Recovery Jitter Spectrum .....	99
Clock Recovery Miscellaneous .....	103
Clock Recovery: Sample Program .....	106
Digital Pre-Emphasis Processor (DPP) Option.....	113
DPP command summary.....	114
DPP remote control commands .....	116
Lightwave Test Set (LTS) option .....	131
LTS remote control setup .....	132
LTS command summary .....	135
LTS Remote Control Commands .....	136
Analysis Engine.....	145
Basic BER.....	149
Block Errors.....	150
Burst Length .....	153
Correlation.....	155
Error Free Interval .....	158
Error Map.....	160
FEC Emulation.....	163
Pattern Sensitivity .....	169
Strip Chart.....	172
Physical Layer Test.....	175
Eye Diagram .....	176
Eye: CleanEye .....	187
Eye: Enable Eye Measurement Overlay .....	196
Eye: Data Collecting .....	206
Eye: Single Value Waveform.....	207
Eye: Optical Mode .....	208
Eye: Read Eye Measurement .....	212
BER Contour.....	215

---

BER Contour: Optical Mode.....	221
Jitter Map.....	222
Jitter Peak.....	234
Jitter Tolerance.....	242
Mask Test.....	247
Q-Factor.....	253
Mainframe.....	261
System View.....	269
RAM Capture.....	271
System Event Log.....	273
Configuration.....	275
Status Queries.....	277
Common Commands.....	281
File Transfer.....	283
Appendix A: Sample Remote Programming.....	287
Connect the cables to the instrument.....	287
Send commands to the instrument.....	287
Run a test and read back the measurements.....	291
Index	



---

# Preface

This document contains a comprehensive listing of remote control commands for the Tektronix BERTScope family of products. It includes commands for the different platforms. Restrictions and separate parameters or ranges are noted for each of those commands.

## Theory of operation

The remote control features enable operation of instrument processes remotely via an IEEE-488 communications connection. Remote control is implemented using a text-oriented command protocol that enable you to set and query system parameters, and to operate the analyzer in different modes, including live error analysis and offline playback of previously recorded error data.

The text-oriented commands follow a basic three-part structure, consisting of one word identifying the major feature being addressed, another word identifying a specific operation or parameter within that feature, and optional parameters. The analyzer receives and operates on each command immediately. Command execution sets a status variable that may be queried by the user to determine if the previous command was successful. Alternatively, a mode can be selected in which these statuses are automatically returned after each command execution.

This document describes how to interface with the analyzer using the IEEE-488 connection. The Analyzer is an IEEE-488 bus peripheral only; it is not an IEEE-488 bus controller and cannot produce SRQ signals. The IEEE-488 Setup panel. In the System View, touch the Tools tab, then the Remote button, then IEEE Address. Enter the unique address in the GPIB Address field.

For more technical assistance in programming your remote control applications, please contact your local Tektronix representative.





---

# Getting started

## Remote control overview

The BERTScope analyzer and any connected BERTScope instruments can be controlled remotely via either an IEEE-488 or a TCP/IP communications connection. The remote control software runs on either a BERTScope analyzer, or a host PC..

The remote control software accepts text-oriented commands from a remote computer via either an IEEE-488 or TCP/IP connection. It then routes the commands to one or more BERTScope instruments, connected to the BERTScope analyzer or host PC via USB.

If the local host is a BERTScope, then the BERTScope software gets a chance to handle the command first. Any commands the BERTScope software doesn't understand are sent to the BERTScope instrument software, which then controls the BERTScope instruments.

When a BERTScope instrument is connected to a BERTScope, you can inter-mix BERTScope and BERTScope instrument commands in the same script, program, or interactive session. If the host is just an ordinary PC, then the commands are routed to the remote control software directly. In this case, using the BERTScope-specific commands will result in an error.

For most applications, a single BERTScope instrument is connected to a BERTScope or host PC. In this case, the BERTScope instrument is automatically detected and connected when the remote control software is started, and automatically disconnected when the remote control software is stopped. There is no need in this case to use the OPEN or CLOSE commands described later in this section.

If more than one BERTScope instrument is connected (such as a DPP125C and one or more Clock Recovery instruments) to the BERTScope or host computer, the remote control software will not connect automatically. In this case, the Remote computer must issue a NAMES? query to discover the IDs of the connected BERTScope instruments, and OPEN the one desired before issuing control commands. If the Remote computer needs to control multiple BERTScope instruments, it would OPEN, control, then CLOSE one, then OPEN, control, and CLOSE another. The device that is OPEN is referred to as the current device throughout this document.

Remote control is implemented using a text-oriented command protocol described in the following pages of this document. These commands enable you to set and query the system parameters of the BERTScope instruments, and to retrieve measurements made by the instrument.

## Set up the instrument for remote control operation

Remote setup is accessed from the BERTScope Analyzer System View.

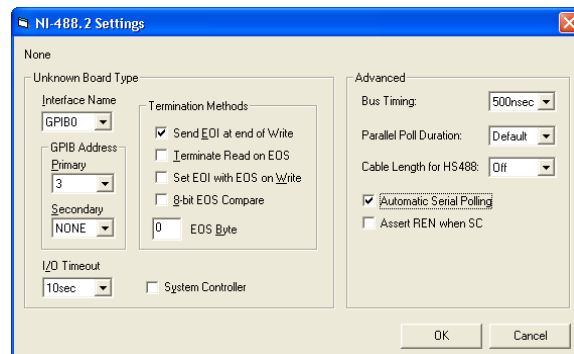
1. In the System view, under the Tools tab Utilities heading, click the **Remote** button to access the configuration panel.
2. Before beginning a remote control session, it is necessary to start the Remote Client application, *RemoteClient.exe*, on the host computer.

If the host is a BERTScope, click the **Remote** button from the System View Tools page. If the host computer is a normal PC, then just execute the *RemoteClient.exe*, file.

3. If you are using GPIB to control the BERTScope, enable the IEEE Address button.

If the button is not enabled, click the **Disconnect** button; it might take a few seconds to enable the IEEE Address button.

4. Click the **IEEE Address** button and enter the required information into the NI-488.2 Settings dialog box.



## Command description

Remote control communications are exchanged as ASCII strings over a LAN telnet (TCP port 23) between the host computer and the analyzer. Select a communications port to be used for access to the instrument's remote control operations. If the port is set to NONE, then remote control is disabled.

The input values of commands are checked against the same ranges as the user interface. If an input is received that is outside of this legal range, the input will be clipped and recorded in the status queue.

It is recommended that you request the status of a command by sending **SYStemERRor?** after each command. Besides telling you what has happened with the command, this will also synchronize command transmission and handling.

## Command syntax

Remote Control Command Lines are defined as ASCII text strings ending with ‘\r’ or ‘\n’.

The commands follow a basic three-part structure, consisting of one word identifying the feature of the BERTScope instrument being addressed, another word identifying a specific operation within that feature, and optional parameters. The feature and operation are separated by a colon, and the operation and parameter (if any), are separated by a space. The operation can be Double, Integer, or Boolean numbers; a file path name; a special word; or a data type. For a Boolean data type, the input can be ON, OFF, 1, or 0. The returned response will be either 1 for true or 0 for false. A question-mark character (?) is added at the end of a command, without a space, to create a query.

**FEATURE:OPERATION PARAMETER**

Commands and parameters are generally case-insensitive, with the notable exception of those requiring a case-sensitive parameter for the Standard names.

Most commands have both a “long version” and “short version,” indicated by capitalized versus lowercase letters. For example: **GENERator:PATtern?** can be typed out exactly as shown, or as **GEN:PATT?** The capital letters and the colon (:) are required; the lowercase letters are optional – commands are not case-sensitive.

The short form is convenient during interactive control (using a telnet session, for example). You might want to use the long form in scripts or programs, to increase code clarity.

When command setup data uses a string as a parameter (a filename or path name, for instance), enclose the string in double-quotes (“string”).

For every command processed, an error code will be returned. If the command has a question mark (a query), the setup information is also returned.

## Features

The BERTScope remote control commands are divided into feature groups. The number of features per each group depend on the BERTScope product.

In the first group are commands that handle device discovery and connection. You can use these commands anytime, even if a current device has not been selected yet (with the OPEN command). The NAMES, OPEN, and CLOSE commands are all part of this group.

A second group consists of commands for controlling and monitoring a particular BERTScope instruments. The current device must have been specified (using the OPEN command) before to using any commands in this group.

In addition to the various feature groups, several miscellaneous features exist to control the host computer display (VIEW, PTFILE, GUILOCKOUT), the run state (RSTATE, RDURATION), and the overall program configuration (SCONFIGURATION, RCONFIGURATION, ISSTATUS, RSSTATUS). These commands primarily implement the functionality provided by the GUI Console buttons.

## Operations

### Read-only properties

Query a readable property by appending a question mark to it (with no space between the command and the question mark). For example:

```
CRS:OPEN?
```

returns the device name of the currently connected BERTScope CR. In most cases, the returned values are all capitalized.

### Read-write properties

Query a read-write property by appending a question mark after it.

Set a read-write property by adding an appropriate parameter. For example:

```
DPP:OPEN "DPP_1234"
```

selects the BERTScope DPP with the device name DPP\_1234 as the current device, while

```
DPP:OPEN?
```

returns name of the device previously OPENed.

### Write-only operation

Question marks are illegal for write-only operation, and parameters may or may not be required, depending upon the particular operation. For example:

```
DPP:CLOSE
```

requires no parameters, but

```
DPP:RCONFIGDISK "D:\BitAlyzer\Configurations\mycfg.DPP"
```

requires the complete pathname of the BERTScope DPP configuration file.

## Parameters

A parameter can be a double or integer number, Boolean (0 or 1), an enumerated data type, or a string.

When a command uses a string as a parameter, the string is enclosed in double quotes ("string"). Filenames always require the absolute path, enclosed within double quotes.

Parameters are generally case-insensitive, with the notable exception of case-sensitive Standard names. For example:

```
CRC:STANDARD "123abc"
```

and

```
CRC:DELETESTANDARD "123ABC"
```

are specifying different user-defined clock recovery standards.

## Command status

The BERTScope instrument receives and operates on each command immediately.

The input values of commands are checked. If an input is received that is outside of the legal range, the input will be clipped and recorded in the status queue.

Command execution sets a status variable that can be queried by the user to determine if the previous command was successful.

Request the status of a command by sending:

```
SYSTEM:ERROR? or SYST:ERR?
```

This command retrieves items listed in the error status queue, in the format:

```
< n, "event/string" >
```

The status queue is first in, first out. It can contain up to 100 error messages. If the status queue overflows, the last error/event in the queue is replaced with Error -350, "Queue overflow". When all errors/events have been read from the queue, further error/event queries will return 0, "NO ERROR".

To clear the status queue all at once, send:

```
*CLS
```

## Command error messages and codes

**Error messages** The following error messages might be returned:

- "TOO MANY PARAMETERS"
- "UNRECOGNIZED COMMAND"
- "WRONG PARAMETER TYPE"
- "TOO LITTLE PARAMETERS"
- "WRONG PARAMETER FORMAT"
- "WRONG PARAMETER NUMBER"
- "WRONG ACTION-ONLY"

- "WRONG QUERY-ONLY"
- "REQUESTED <QUERY PARAMETER> IS NOT SUPPORTED ON THE PLATFORM"

**Error codes** The following error codes might be returned:

- 0 "NO ERROR"
- 350 "Queue Overflow"
- 10 File transfer error
- 20 System error or event
- 30 Command error, see list of error messages
- 40 No software option

## Communications timeouts

Most commands issued to the BERTScope/BitAnalyzer are processed quickly and return within immediately. However a handful of commands might require considerable processing and take longer, including the following:

- Detector delay calibration — 8 seconds (approximately)
- Generator delay calibration — 7 seconds
- Detector auto align — 5 seconds
- Generator data signal enabling (BERTScope only) — 7 seconds
- Run state enable (Physical layer tests that perform initial auto-align) — 6 seconds

As a result, programming the communications architecture with a two-second timeout for most command, and a 10-second timeout for the above-listed commands is one approach. Another approach is to set all communication timeouts to be 10 seconds. Of course, commands that don't require this much time will not be affected by the longer timeout. This ensures that all commands can complete without triggering communications timeouts.

# Command listing

Generator (See page 11, <i>Generator</i> .)	
Data Generator (See page 17, <i>Generator Data Generator</i> .)	Clock Output +/- (See page 30, <i>Generator Clock Output Pos/Neg</i> .)
Pattern Start (See page 19, <i>Generator Pattern Start</i> .)	Data Output + (See page 31, <i>Generator Data Output Positive</i> .)
Spread Spectrum Clock (See page 20, <i>Generator Spread Spectrum Clock</i> .)	Data Output – (See page 34, <i>Generator Data Output Negative</i> .)
Clock Input (See page 22, <i>Generator Clock Input</i> .)	Data Output +/- (See page 36, <i>Generator Data Output Pos/Neg</i> .)
Clock Output + (See page 24, <i>Generator Clock Output Positive</i> .)	Trigger (See page 37, <i>Generator Trigger</i> .)
Clock Output – (See page 27, <i>Generator Clock Output Negative</i> .)	Error Inject (See page 38, <i>Generator Error Inject</i> .)
Generator Stress Module (GSM) – Stressed Eye (See page 41, <i>Generator Stress Module (GSM) - Stressed Eye</i> .)	
GSM Configure/Enable (See page 41, <i>GSM - Stressed Eye Configure/Enable</i> .)	GSM Stress Setup (See page 42, <i>GSM - Stressed Eye Setup</i> .)
Detector (See page 51, <i>Detector</i> .)	
Error Detector (See page 54, <i>Detector Error Detector</i> .)	Trigger (See page 64, <i>Detector Trigger</i> .)
Start Detect (See page 58, <i>Detector Start Detect</i> .)	Results (See page 65, <i>Detector Results</i> .)
Clock Input (See page 59, <i>Detector Clock Input</i> .)	Auto Align Results (See page 68, <i>Detector Auto Align Results</i> .)
Data Input (See page 61, <i>Detector Data Input</i> .)	Optical Mode (See page 69, <i>Detector Optical Mode</i> .)
Blank Input (See page 63, <i>Detector Blank Input</i> .)	Convenience (See page 68, <i>Detector Convenience</i> .)
Symbol Filtering (See page 67, <i>Detector Symbol Filtering</i> .)	
Stress Combiner (See page 71, <i>Stress Combiner Option</i> .)	
Stress Combiner command summary (See page 72, <i>Stress Combiner command summary</i> .)	Stress Combiner commands (See page 74, <i>Stress Combiner commands</i> .)
Clock Recovery (See page 79, <i>Clock Recovery Option</i> .)	
CR Service (See page 84, <i>CRService</i> : <i>Clock Recovery Service</i> .)	CR Jitter Spectrum (See page 99, <i>Clock Recovery Jitter Spectrum</i> .)
CR Control (See page 85, <i>CRControl</i> : <i>Clock Recovery Control</i> .)	CR Miscellaneous (See page 103, <i>Clock Recovery Miscellaneous</i> .)
CR Loop Response (See page 94, <i>Clock Recovery Loop Response</i> .)	CR Sample Program (See page 106, <i>Clock Recovery: Sample Program</i> .)

CR SSC Waveform (See page 96, <i>Clock Recovery SSC Waveform.</i> )	
Digital Pre-Emphasis Processor (DPP) (See page 113, <i>Digital Pre-Emphasis Processor (DPP) Option.</i> )	
DPP command summary (See page 114, <i>DPP command summary.</i> )	DPP Interface(See page 116, <i>DPP remote control commands.</i> )
Lightwave Test Set LTS option (See page 131, <i>Lightwave Test Set (LTS) option.</i> )	
LTS remote control setup (See page 132, <i>LTS remote control setup.</i> )	RC LTS command summary (See page 135, <i>LTS command summary.</i> )
RC LTS Remote Control Commands (See page 136, <i>LTS Remote Control Commands.</i> )	
Analysis Engine (See page 145, <i>Analysis Engine.</i> )	
Basic BER (See page 149, <i>Basic BER.</i> )	2D Error Map (See page 160, <i>Error Map.</i> )
Block Errors (See page 150, <i>Block Errors.</i> )	FEC Emulation (See page 163, <i>FEC Emulation.</i> )
Burst Length (See page 153, <i>Burst Length.</i> )	Pattern Sensitivity (See page 169, <i>Pattern Sensitivity.</i> )
Correlation (See page 155, <i>Correlation.</i> )	Strip Chart (See page 172, <i>Strip Chart.</i> )
Error Free Interval (See page 158, <i>Error Free Interval.</i> )	
Physical Layer Test (See page 175, <i>Physical Layer Test.</i> )	
Eye Diagram (See page 176, <i>Eye Diagram.</i> )	Contour Optical Mode (See page 221, <i>BER Contour: Optical Mode.</i> )
Eye: CleanEye (See page 187, <i>Eye: CleanEye.</i> )	Mask Test (See page 247, <i>Mask Test.</i> )
Eye: Single Value Waveform (See page 207, <i>Eye: Single Value Waveform.</i> )	Jitter Map (See page 222, <i>Jitter Map.</i> )
Eye: Data Collecting (See page 206, <i>Eye: Data Collecting.</i> )	Jitter Peak (See page 234, <i>Jitter Peak.</i> )
Eye: Optical Mode (See page 208, <i>Eye: Optical Mode.</i> )	Jitter Tolerance (See page 242, <i>Jitter Tolerance.</i> )
Eye: Read Eye Measurement (See page 212, <i>Eye: Read Eye Measurement.</i> )	Q-Factor (See page 253, <i>Q-Factor.</i> )
BER Contour (See page 215, <i>BER Contour.</i> )	
Mainframe (See page 261, <i>Mainframe.</i> )	
System (See page 269, <i>System View.</i> )	
System Event Log (See page 273, <i>System Event Log.</i> )	
Configuration (See page 275, <i>Configuration.</i> )	



---

Status Queries (See page 277, *Status Queries*.)

---

Common Commands (See page 281, *Common Commands*.)

---



# Generator

Complete command list (See page 7, *Command listing*.)

## **DELAY:GENRecal?**

Retrieve the information that indicates whether Generator delay needs recalibration. Query only. Note: Since monitoring for these calibrations is suspended while Physical Layer tests are running, using this command under those circumstances may not return an accurate result.

Returns:	< 1 >	Needs delay recalibration.
	< 0 >	Does not need delay recalibration.

## **GENerator:CREference <EXTernal | INTernal> GENerator:CREference?**

BERTScope Analyzer only. Set or retrieve the Generator input clock reference.

Params:	<EXTernal>	External clock reference
	<INTernal>	Internal clock reference

Returns:	< EXTERNAL   INTERNAL >
----------	-------------------------

## **GENerator:CSElect <INTernal | EXTernal> GENerator:CSElect?**

Set or retrieve the clock used by the Generator.

Params:	<INTernal>	Generator uses the internal clock synthesizer
	<EXTernal>	Generator uses the external clock

Returns:	< INTERNAL   EXTERNAL >
----------	-------------------------

**GENerator:CUTOFFKHZ?**

Retrieve the frequency at which the Generator switches to the DDR mode. Applies to BERTScope BSA175C. Query only.

**GENerator:DDEFs <numeric>****GENerator:DDEFs?**

Set or retrieve the Generator data delay in femto-seconds.

Params:	<numeric>	Set Generator data delay in fs.
Returns:	<numeric>	

---

**GENerator:DDELay <numeric>****GENerator:DDELay?**

Set or retrieve the Generator data delay in picoseconds.

Params:	<numeric>	Generator data delay. BitAlyzer: Input out of range will be clipped and recorded in the status queue. When the BitAlyzer is operating within range [982 to 1,500 MHz], Data Delay is within range [0 to 5,000 ps]. BERTScope Analyzer: Input out of range will be clipped and recorded in the status queue. When the Analyzer is operating within range [1,100 to 12,500 MHz], Data Delay is within range [0 to 3,000 ps].
Returns:	<numeric>	

---

**GENerator:DINVert <bool>****GENerator:DINVert?**

Set or retrieve data inversion state for the Generator.

Params:	<bool>	On = 1, Off = 0
Returns:	< 1 >	Data inversion is on
	< 0 >	Data inversion is off

---

**GENerator:DRATE?**

Retrieve the Generator data rate. Query only.

Returns:	<numeric>	Generator data rate in bits/sec. "0" means "No Clock."
----------	-----------	--

---

**GENerator:EXTPagmode <SWITCHRISE | SWITCHFALL | AORBRISE | AORBFALL | OFF>****GENerator:EXTPagmode?**

Set or retrieve the Generator External Page Select mode.

Params:	<SWITCHRISE>	Switch momentarily to the other page on rising edge
	<SWITCHFALL>	Switch momentarily to the other page on falling edge
	<AORBRISE>	Switch to the other page on a rising edge
	<AORBFALL>	Switch to the other page on a falling edge
	<OFF>	External page select is off

Returns: < SWITCHRISE | SWITCHFALL | AORBRISE | AORBFALL | OFF >

**GENerator:FULLRATEclock <bool>****GENerator:FULLRATEclock?**

Set or retrieve Full Rate Clock property of the Generator.

Returns:	< 1 >	Generator is in Full Rate Clock mode
	< 0 >	Generator is in Half Rate Clock mode

**GENerator:ICLock <numeric>****GENerator:ICLock?**

Set or retrieve the internal clock synthesizer frequency of the Generator. May require some delay to complete.

Params:	<numeric>	Generator internal clock synthesizer frequency in Hz BitAlyzer BA1500: Range [1 to 1,500,000] BitAlyzer BA1600: Range [1 to 1,600,000] BERTScope BSA7500: Range [100,000,000 to 7,500,000,000] BERTScope BSA125C: Range [100,000,000 to 12,500,000,000] BERTScope BSA175C: Range [500,000,000 to 17,500,000,000] Input out of range will be clipped and recorded in the status queue.
---------	-----------	---

Returns: <numeric>

**GENerator:OFFsetfreq <numeric>**  
**GENerator:OFFsetfreq?**

Set or retrieve Generator's Frequency Offset in PPM.

Params:	<numeric>	Generator's Frequency Offset in PPM. Range [-500 to +500]
Returns:	<numeric>	

**GENerator:PCALibration**

Perform Generator delay calibration. Action only. May require some delay to complete.

**GENerator:PMMOD:DEViation <numeric>**  
**GENerator:PMMOD:DEViation?**

Set or retrieve Phase Modulation Deviation in UI. Range is from 0 to the value dependent on PM.

Frequency:

>40 kHz to 160 kHz	Decreases 30 dB/decade to 37.5/ n UI at 165 kHz
40 kHz	300/ n UI
10 Hz to <40 kHz	Increases at 20 dB/decade to 6000/ n UI at 2 kHz (fixed 6000/ n UI from 2 kHz to 10 Hz)

where:

- n = 1 for 6 GHz = clk = 12.5 GHz
- n = 2 for 3 GHz = clk < 6 GHz
- n = 4 for 1.5 GHz = clk < 3 GHz
- n = 8 for 750 MHz = clk < 1.5 GHz
- n = 16 for 375 MHz = clk < 750 MHz
- n = 32 for 187.5 MHz = clk < 375 MHz
- n = 64 for 100 MHz = clk < 187.5 MHz

Params:	<numeric>	Range: 0 to the upper limit as dependent on the PM Frequency (see GEN:PMMOD:FREQ)
Returns:	<numeric>	

**GENerator:PMMOD:ENABLE <bool>**  
**GENerator:PMMOD:ENABLE?**

Enable or disable Phase Modulation.

Params:	<bool>	On = 1, Off = 0
Returns:	< 1 >	Generator Phase Modulation is enabled
	< 0 >	Generator Phase Modulation is disabled

**GENerator:PMMOD:FREQuency <numeric>**  
**GENerator:PMMOD:FREQuency?**

Set or retrieve Phase Modulation Frequency.

Params:	<numeric>	Range [10 Hz to 160 kHz]
Returns:	<numeric>	

**GENerator:REFIN:ENABLE <bool>**  
**GENerator:REFIN:ENABLE?**

Enable or disable the Generator Reference In.

Returns:	< 1 >	Generator Reference In is enabled
	< 0 >	Generator Reference In is disabled

**GENerator:REFIN:FREQuency <numeric>**  
**GENerator:REFIN:FREQuency?**

Set or retrieve the Generator Reference In Frequency in Hz.

Params:	<numeric>	Generator Reference In frequency in Hz.
		Possible values are:
		10,000,000
		100,000,000
		106,250,000
		156,250,000
		133,330,000
		166,670,000
		200,000,000

Returns:	<numeric>
----------	-----------

**GENerator:REFOUT:ENABLE <bool>**  
**GENerator:REFOUT:ENABLE?**

Enable or disable the Generator Reference Out.

Params:	<bool>	On = 1, Off = 0
Returns:	< 1 >	Generator Reference Out is enabled
	< 0 >	Generator Reference Out is disabled

**GENerator:REFOUT:FREQuency <numeric>  
GENerator:REFOUT:FREQuency?**

Set or retrieve the Generator Reference Out Frequency in Hz.

Params:	<numeric>	Generator Reference Out frequency in Hz.
		Possible values are:
		10,000,000
		100,000,000
		106,250,000
		156,250,000
		133,330,000
		166,670,000
		200,000,000

---

Returns: <numeric>

---

**GENerator:SYNFM:PRESent?**

Retrieve whether SynFM module is present. Query only.

Params:	<bool>	On = 1, Off = 0
---------	--------	-----------------

Returns:	< 1 >	SynFM module is present
	< 0 >	SynFM module is not present

---

Complete command list (See page 7, *Command listing*.)



# Generator Data Generator

Complete command list (See page 7, *Command listing*.)

## **GENerator:ISERror**

Inject a single error. Action only.

## **GENerator:LUPFilename?**

Retrieve user pattern file that has been loaded to the Generator. Query only.

Returns: < "filename" >      Filename is enclosed in double quotes.

---

## **GENerator:PATtern <PN7 | PN11 | PN15 | PN20 | PN23 | PN31 | UStart | UCYCLE | ZERO | PASSthrough>**

### **GENerator:PATtern?**

Set or retrieve the Generator data type. PRBS-11 is available only on BERTScope Analyzers. The Passthrough type is available only on BSA85C, BSA175C, and BSA286C.

Params:	<PN7   PN11   PN15   PN20   PN23   PN31>	Generator pseudo-random pattern data type. PN11 pattern is available only on the BERTScope Analyzer.
	<UStart>	User pattern (See GEN:UPLoad command to load a user pattern file)
	<UCYCLE>	User pattern cycle
	<ZERO>	All-zeros pattern
	<PASSthrough>	When corresponding option is purchased, Detector Pass-Through mode outputs whatever the Detector has just received. Available only on BSA85C, BSA175C, and BSA286C.

---

Returns: < PN7 | PN11 | PN15 | PN20 | PN23 | PN31 | USTART | UCYCLE | ZERO | PASSTHROUGH >

---

## **GENerator:PSElect <A | B>**

### **GENerator:PSElect?**

Set or retrieve the selected page of the user pattern loaded to the Generator.

Params:	<A>	User pattern page A
	<B>	User pattern page B

---

Returns: < PAGEA | PAGEB >

---

## **GENerator:PSWitch**

Switch the page of the user pattern loaded to the Generator. Action only.

**GENerator:UPLoad <"filename">**

Load user pattern from the specified file. Action only.

Params:	<"filename">	Use the absolute path name and enclose the string in double quotes.
---------	--------------	---

---

**GENerator:UPLPercent?**

Retrieve the user pattern loading percentage completion value. Query returns 100 if the loading is complete. Query only.

Returns:	< number >	The percentage of user pattern loaded into the Generator
----------	------------	--

---

**GENerator:UPLTracking<bool>**  
**GENerator:UPLTracking?**

Set or retrieve whether the Generator tracks the Detector when loading a user pattern. If the tracking is true, the command GEN:UPL <"filename"> loads the user pattern into both the Generator and Detector.

Params:	<bool>	On = 1, Off = 0
Returns:	< 1 >	Generator tracks Detector when loading
	< 0 >	Generator does not track Detector when loading

---

**GENerator:UPWLen?**

Retrieve the word count from the Generator RAM. Query only.

Returns:	< number >	The number of words in RAM
----------	------------	----------------------------

---

Complete command list (See page 7, *Command listing*.)

## Generator Pattern Start

Complete command list (See page 7, *Command listing*.)

**GENerator:PSStart:LFAMily** <LVTTTL | CML | ECL | PECL | LVPECL>  
**GENerator:PSStart:LFAMily?**

Set or retrieve the pattern start input signal logic family. Command available only on the BitAlyzer.

Params:	<LVTTTL   CML   ECL   PECL   LVPECL>	The pattern start input logic family
---------	--	--------------------------------------

Returns:	< LVTTTL   CML   ECL   PECL   LVPECL >
----------	--

**GENerator:PSStart:TAC** <bool>  
**GENerator:PSStart:TAC?**

Set or retrieve the termination AC of the Generator pattern start input.

Params:	<bool>	On = 1, Off = 0
---------	--------	-----------------

Returns:	< 1 >	Generator pattern start input termination AC is on
	< 0 >	Generator pattern start input termination AC is off

**GENerator:PSStart:THReshold** <numeric>  
**GENerator:PSStart:THReshold?**

Set or retrieve the threshold of the Generator pattern start input. Command is available only on BitAlyzer.

Params:	<numeric>	Generator pattern start input threshold in mV. Range [-2,000 to 4,000]. Input out of range will be clipped and recorded in the status queue.
---------	-----------	--

Returns:	<numeric>
----------	-----------

**GENerator:PSStart:TMODe** <REDGe | FEDGe | DISabled>  
**GENerator:PSStart:TMODe?**

Set or retrieve the trigger mode of the Generator pattern start input.

Params:	<REDGe>	Generator pattern start trigger at the rising edge
	<FEDGe>	Generator pattern start trigger at the falling edge
	<DISabled>	Trigger is disabled.

Returns:	< REDGE   FEDGE   DISABLED >
----------	------------------------------

**GENerator:PSStart:TVOLTage<numeric>**  
**GENerator:PSStart:TVOLTage?**

Set or retrieve the termination voltage of the Generator clock input.

Params: <numeric> Generator clock input termination voltage in mV. Range [-2,000 to 3,300]. Input out of range will be clipped and recorded in the status queue.

---

Returns: <numeric>

---

Complete command list (See page 7, *Command listing*.)

## Generator Spread Spectrum Clock

---

**NOTE.** *SSC commands apply to the BERTScope only.*

---

Complete command list (See page 7, *Command listing*.)

**GENerator:SSCMOD:DEVIation <numeric>**  
**GENerator:SSCMOD:DEVIation?**

Set or retrieve the SSC modulation deviation. Argument range is different for first generation hardware (Option SSC) and second generation (Option XSSC).

Params: <numeric> SSC modulation deviation in ppm.  
 Option SSC: Range [0 to 10,000] in Down spread or Up spread mode; [0 to 5,000] in Center spread mode.  
**Option XSSC:** Range is a function of data rate. Refer to BERTScope Analyzer Technical Specifications.

---

Returns: <numeric>

---

**GENerator:SSCMOD:ENABLE <bool>**  
**GENerator:SSCMOD:ENABLE?**

Set or retrieve whether the SSC/XSSC option is enabled (option must be present in order to enable; see GEN:SSCMOD:PRES command below).

Params: <bool> On = 1, Off = 0

---

Returns: < 1 > SSC is enabled  
 < 0 > SSC is disabled

---

**GENerator:SSCMOD:FREQuency <numeric>**  
**GENerator:SSCMOD:FREQuency?**

Set or retrieve the SSC modulation frequency. Argument range is different for first generation hardware (Option SSC) and second generation (Option XSSC).

Params:	<numeric>	SSC modulation frequency in Hz. Option SSC: Range [25,000 to 35,000] Option XSSC: Range [20,000 to 160,000]. Note that modulation deviation is uncalibrated for modulation frequencies >40 kHz.
---------	-----------	---

---

Returns:	<numeric>
----------	-----------

---

**GENerator:SSCMOD:PRESent <bool>**  
**GENerator:SSCMOD:PRESent?**

Set or retrieve whether the SSC/XSSC option is present.

Params:	<bool>	On = 1, Off = 0
---------	--------	-----------------

---

Returns:	< 1 >	SSC is present.
----------	-------	-----------------

---

	< 0 >	SSC is not present.
--	-------	---------------------

---

**GENerator:SSCMOD:SIGnal <TRIANGLE | SINUSOID>**  
**GENerator:SSCMOD:SIGnal?**

Set or retrieve the SSC/XSSC modulation signal type.

Params:	<TRIANGLE>	Triangular-shaped modulation
---------	------------	------------------------------

---

	<SINUSOID>	Sinewave-shaped modulation
--	------------	----------------------------

---

Returns:	< TRIANGLE   SINUSOID >
----------	-------------------------

---

**GENerator:SSCMOD:SIGNALSETTABLE?**

Returns whether the SSC/XSSC modulation signal is selectable. First-generation SSC hardware only modulates in TRIANGLE mode. Query only.

Returns:	< 1 >	Modulation is selectable (TRIANGLE or SINUSOID) (XSSC hardware)
----------	-------	---

---

	< 0 >	Only modulates in TRIANGLE mode (1st generation SSC hardware)
--	-------	---

---

**GENerator:SSCMOD:TYPE<UP | DOWN | CENTER>**  
**GENerator:SSCMOD:TYPE?**

Set or retrieve the SSC/XSSC modulation type.

Params:	<UP>	Modulation deviation ranges from the nominal frequency to the nominal frequency plus the deviation amount.
	<DOWN>	Modulation deviation is centered around the nominal frequency.
	<CENTER>	Modulation deviation ranges from the nominal frequency minus the deviation amount to the nominal frequency.
Returns:	< UP   DOWN   CENTER >	

Complete command list (See page 7, *Command listing*.)

## Generator Clock Input

Complete command list (See page 7, *Command listing*.)

**GENerator:CINPut:LFAMily <LVTTTL | CML | ECL | PECL | LVPECL>**  
**GENerator:CINPut:LFAMily?**

Set or retrieve the Generator Clock input logic family. This command is available only on the BitAlyzer platform.

Params:	<LVTTTL   CML   ECL   PECL   LVPECL>	Set the Generator Clock input logic family
Returns:	< LVTTTL   CML   ECL   PECL   LVPECL >	

**GENerator:CINPut:TAC <bool>**  
**GENerator:CINPut:TAC?**

Set or retrieve the termination AC of the Generator Clock input.

Params:	<bool>	On = 1, Off = 0
Returns:	< 1 >	Generator Clock input termination AC is on
	< 0 >	Generator Clock input termination AC is off

**GENerator:CINPut:THREshold <numeric>**  
**GENerator:CINPut:THREshold?**

Set or retrieve the threshold of the Generator Clock input. This command is available only on the BitAlyzer platform.

Params: <numeric> Generator Clock input threshold in mV. Range [-1,200 to +800]. Input out of range will be clipped and recorded in the status queue.

---

Returns: <numeric>

---

**GENerator:CINPut:TVOLTage <numeric>**  
**GENerator:CINPut:TVOLTage?**

Set or retrieve the termination voltage of the Generator Clock input.

Params: <numeric> Generator Clock input termination voltage in mV. Range [-2,000 to +3,300]. Input out of range will be clipped and recorded in the status queue.

---

Returns: <numeric>

---

**GENerator:CLockOPTimize**

BitAlyzer only. Initiates procedure for optimizing the location of the Generator external clock input threshold. May take seconds to complete. Locates the mid-point of the clock envelope. Action only.

**GENerator:CLockOPTimizeResult?**

BitAlyzer only. Returns the located threshold in millivolts. If optimization fails, returns -99999999 as a sentinel value. Query only.

**GENerator:SUBRatediv <1 | 2 | 4 | 8 | 16 | 32 | 64 | 128>**

**GENerator:SUBRatediv?**

Set or retrieve the sub-rate clock output divider for the Generator's internal clock synthesizer.

Params: <1 | 2 | 4 | 8 | 16 | 32 | 64 | 128> "1" is full rate.

---

Returns: < 1 | 2 | 4 | 8 | 16 | 32 | 64 | 128 >

---

Complete command list (See page 7, *Command listing*.)

## Generator Clock Output Positive

Complete command list (See page 7, *Command listing*.)

### **GENerator:COPositive:CLIPped?**

Retrieve whether the Generator Clock+ output setting is clipped. Query only.

Returns:	< 1 >	The Generator Clock+ output setting is clipped
	< 0 >	The Generator Clock+ output setting is not clipped

### **GENerator:COPositive:ENABLE <bool>**

#### **GENerator:COPositive:ENABLE?**

Set or retrieve whether the Generator Clock+ output is enabled. May require some delay to complete.

Params:	<bool>	On = 1, Off = 0
Returns:	< 1 >	Generator Clock+ output is enabled
	< 0 >	Generator Clock+ output is disabled

### **GENerator:COPositive:IMPedance <numeric>**

#### **GENerator:COPositive:IMPedance?**

Set or retrieve the impedance of the Generator Clock output positive. May require some delay to complete.

Params:	<numeric>	Generator Clock+ output impedance in Ohms. Range [30 to 100,000]. Input out of range will be clipped and recorded in the status queue.
Returns:	<numeric>	

### **GENerator:COPositive:LFAMILY <LVTTL | CML | ECL | PECL | LVPECL | LVDS | SCFL>**

#### **GENerator:COPositive:LFAMILY?**

Set or retrieve the Generator Clock+ output logic family. PECL is not available on the BERTScope Analyzer.

Params:	<LVTTL   CML   ECL   PECL   LVPECL   LVDS   SCFL>	The Generator Clock+ output logic family
Returns:	< LVTTL   CML   ECL   PECL   LVPECL   LVDS   SCFL >	



**GENerator:COPositive:SLAMplitude <numeric>**  
**GENerator:COPositive:SLAMplitude?**

Set or retrieve the signal level amplitude of the Generator Clock+ output. May require some delay to complete.

Params: <numeric> Generator Clock+ output signal level amplitude in mV. Input out of range will be clipped and recorded in the status queue.  
 BitAlyzer: Range [70 to 2,000].  
 BERTScope Analyzer: Range [250 to 2,000].

---

Returns: <numeric> Generator Clock+ output signal level amplitude

---

**GENerator:COPositive:SLOffset <numeric>**  
**GENerator:COPositive:SLOffset?**

Set or retrieve the signal level offset of the Generator Clock+ output. May require some delay to complete.

Params: <numeric> Generator Clock+ output signal level offset in mV. Input out of range will be clipped and recorded in the status queue.  
 BitAlyzer: Range [-2,000 to +4,000].  
 BERTScope Analyzer: Range [-2,000 to +2,000].

---

Returns: <numeric> Generator Clock+ output signal level offset

---

**GENerator:COPositive:SLVHigh <numeric>**  
**GENerator:COPositive:SLVHigh?**

Set or retrieve the signal level  $V_H$  of the Generator Clock+ output. May require some delay to complete.

Params: <numeric> Generator Clock+ output signal level  $V_H$  in mV. Range [-1,750 to +3,000]. Input out of range will be clipped and recorded in the status queue.

---

Returns: <numeric>

---

**GENerator:COPositive:SLVLow <numeric>**  
**GENerator:COPositive:SLVLow?**

Set or retrieve the signal level  $V_{LOW}$  of the Generator Clock+ output. May require some delay to complete.

Params: <numeric> Generator Clock+ output signal level  $V_{LOW}$  in mV. Range [-2,250 to +1,000]. Input out of range will be clipped and recorded in the status queue.

---

Returns: <numeric>

---

**GENerator:COPositive:TAC <bool>**  
**GENerator:COPositive:TAC?**

Set or retrieve the termination AC of the Generator Clock+ output. May require some delay to complete.

Params:	<bool>	On = 1, Off = 0
Returns:	< 1 >	Generator Clock+ output termination AC is on
	< 0 >	Generator Clock+ output termination AC is off

**GENerator:COPositive:TVOLTage <numeric>**  
**GENerator:COPositive:TVOLTage?**

Set or retrieve the termination voltage of the Generator Clock output positive. May require some delay to complete.

Params:	<numeric>	Generator Clock+ output signal level offset in mV. Range [-2,000 to +2,000]. Input out of range will be clipped and recorded in the status queue.
Returns:	<numeric>	

Complete command list (See page 7, *Command listing*.)

# Generator Clock Output Negative

Complete command list (See page 7, *Command listing*.)

## **GENerator:CONegative:CLIPped?**

Retrieve whether the Generator Clock– output setting is clipped. Query only.

Returns:	< 1 >	The Generator Clock– output setting is clipped
	< 0 >	The Generator Clock– output setting is not clipped

## **GENerator:CONegative:ENABLE <bool>**

### **GENerator:CONegative:ENABLE?**

Set or retrieve whether the Generator Clock– output is enabled. May require some delay to complete.

Params:	<bool>	On = 1, Off = 0
Returns:	< 1 >	Generator Clock– output is enabled
	< 0 >	Generator Clock– output is disabled

## **GENerator:CONegative:IMPedance <numeric>**

### **GENerator:CONegative:IMPedance?**

Set or retrieve the impedance of the Generator Clock– output. May require some delay to complete.

Params:	<numeric>	Generator Clock– output impedance in Ohms. Range [30 to 100,000]. Input out of range will be clipped and recorded in the status queue.
Returns:	<numeric>	

## **GENerator:CONegative:LFAMILY <LVTTTL | CML | ECL | PECL | LVPECL | LVDS | SCFL>**

### **GENerator:CONegative:LFAMILY?**

Set or retrieve the Clock– output logic family. PECL is not available on the BERTScope Analyzers.

Params:	<LVTTTL   CML   ECL   PECL   LVPECL   LVDS   SCFL>	The Clock– output logic family
Returns:	<"string">	

**GENerator:CONegative:SLAMplitude <numeric>**  
**GENerator:CONegative:SLAMplitude?**

Set or retrieve the signal level amplitude of the Generator Clock– output. May require some delay to complete.

Params: <numeric> Generator Clock– output signal level amplitude in mV. Range [250 to 2,000]. Input out of range will be clipped and recorded in the status queue.

---

Returns: <numeric>

---

**GENerator:CONegative:SLOffset <numeric>**  
**GENerator:CONegative:SLOffset?**

Set or retrieve the signal level offset of the Generator Clock– output. May require some delay to complete.

Params: <numeric> Generator Clock– output signal level offset in mV. Range [-2,000 to +2,000]. Input out of range will be clipped and recorded in the status queue.

---

Returns: <numeric>

---

**GENerator:CONegative:SLVHigh <numeric>**  
**GENerator:CONegative:SLVHigh?**

Set or retrieve the signal level V H of the Generator Clock– output. May require some delay to complete.

Params: <numeric> Generator Clock– output signal level V H in mV. Range [-1,750 to +3,000]. Input out of range will be clipped and recorded in the status queue.

---

Returns: <numeric>

---

**GENerator:CONegative:SLVLow <numeric>**  
**GEN:CON:SLVL <numeric>**

Set or retrieve the signal level V L of the Generator Clock– output. May require some delay to complete.

Params: <numeric> Generator Clock– output signal level V L in mV. Range [-2,250 to +1,000]. Input out of range will be clipped and recorded in the status queue.

---

Returns: <numeric>

---

**GENerator:CONegative:TAC <bool>**  
**GENerator:CONegative:TAC?**

Set or retrieve the termination AC of the Generator Clock– output. May require some delay to complete.

Params:	<bool>	On = 1, Off = 0
Returns:	< 1 >	Generator Clock– output termination AC is on
	< 0 >	Generator Clock– output termination AC is off

**GENerator:CONegative:TVOLtage <numeric>**  
**GENerator:CONegative:TVOLtage?**

Set or retrieve the termination voltage of the Generator Clock– output. May require some delay to complete.

Params:	<numeric>	Generator Clock– output termination voltage in mV. Range [–2,000 to +2,000]. Input out of range will be clipped and recorded in the status queue.
Returns:	<numeric>	

Complete command list (See page 7, *Command listing*.)

## Generator Clock Output Pos/Neg

Complete command list (See page 7, *Command listing*.)

**GENerator:CLKDIVider <long>**

**GENerator:CLKDIVider?**

Set or retrieve the value of the clock divider.

Params:	<long>	Value of clock divider
---------	--------	------------------------

Returns:	< long >
----------	----------

**GENerator:COUtput:LPNSignals <bool>**

**GENerator:COUtput:LPNSignals?**

Set or retrieve whether the Generator Clock output positive and negative are linked. May require some delay to complete.

Params:	<bool>	On = 1, Off = 0
---------	--------	-----------------

Returns:	< 1 >	Clock outputs are linked.
----------	-------	---------------------------

	< 0 >	Clock outputs are not linked.
--	-------	-------------------------------

**GENerator:SUBRatediv <1 | 2 | 4 | 8 | 16 | 32 | 64 | 128>**

**GENerator:SUBRatediv?**

Set or retrieve the sub-rate clock output divider for the Generator's internal clock synthesizer.

Params:	<1   2   4   8   16   32   64   128>	Generator internal clock synthesizer subrate clock output divider. "1" is full rate.
---------	--------------------------------------	--

Returns:	< 1   2   4   8   16   32   64   128 >
----------	--

**GENerator:SUBrate:STress:CLKMODE <STressed | SUBRate>**

**GENerator:SUBrate:STress:CLKMODE?**

Set or retrieve the subrate clock mode.

Params:	<STressed>	Clock Mode is Stressed.
---------	------------	-------------------------

	<SUBRate>	Clock Mode is Subrate.
--	-----------	------------------------

Returns:	< STRESSED   SUBRATE >
----------	------------------------

Complete command list (See page 7, *Command listing*.)

# Generator Data Output Positive

Complete command list (See page 7, *Command listing*.)

## **GENerator:DOPositive:CLIPped?**

Retrieve whether the Generator Data+ output setting is clipped. Query only.

Returns:	< 1 >	The Generator Data+ setting is clipped
	< 0 >	The Generator Data+ setting is not clipped

## **GENerator:DOPositive:ENABLE <bool>**

### **GENerator:DOPositive:ENABLE?**

Set or retrieve whether the Generator Data+ output is enabled. May require some delay to complete.

Params:	<bool>	On = 1, Off = 0
Returns:	< 1 >	Generator Data+ output is enabled
	< 0 >	Generator Data+ output is disabled

## **GENerator:DOPositive:IMPedance <numeric>**

### **GENerator:DOPositive:IMPedance?**

Set or retrieve the impedance of the Generator Data+ output. May require some delay to complete.

Params:	<numeric>	Generator Data+ output impedance in Ohms. Range [30 to 100,000] Input out of range will be clipped and recorded in the status queue.
---------	-----------	---

Returns:	<numeric>
----------	-----------

## **GENerator:DOPositive:LFAMily <LVTTTL | CML | ECL | PECL | LVPECL | LVDS | SCFL>**

### **GENerator:DOPositive:LFAMily?**

Set or retrieve the Generator Data+ output logic family. PECL is not available on the BERTScope Analyzer.

Params:	<LVTTTL   CML   ECL   PECL   LVPECL   LVDS   SCFL>	Generator Data+ output logic family
---------	--	-------------------------------------

Returns:	<"string">
----------	------------

**GENerator:DOPositive:SLAMplitude <numeric>**  
**GENerator:DOPositive:SLAMplitude?**

Set or retrieve the signal level amplitude of the Generator Data+ output. May require some delay to complete.

Params:        <numeric>                    Generator Data+ output signal level amplitude in mV. Range [250 to 2,000]. Input out of range will be clipped and recorded in the status queue.

---

Returns:       <numeric>

---

**GENerator:DOPositive:SLOffset <numeric>**  
**GENerator:DOPositive:SLOffset?**

Set or retrieve the signal level offset of the Generator Data+ output. May require some delay to complete.

Params:        <numeric>                    Generator Data+ output signal level offset in mV. Range [-2,000 to +2,000]. Input out of range will be clipped and recorded in the status queue.

---

Returns:       <numeric>

---

**GENerator:DOPositive:SLVHigh <numeric>**  
**GENerator:DOPositive:SLVHigh?**

Set or retrieve the signal level  $V_H$  of the Generator Data+ output. May require some delay to complete.

Params:        <numeric>                    Generator Data+ output signal level  $V_H$  in mV. Range [-1,750 to +3,000]. Input out of range will be clipped and recorded in the status queue.

---

Returns:       <numeric>

---

**GENerator:DOPositive:SLVLow <numeric>**  
**GENerator:DOPositive:SLVLow?**

Set or retrieve the signal level  $V_L$  of the Generator Data+ output. May require some delay to complete.

Params:        <numeric>                    Generator Data+ output signal level  $V_L$  in mV. Range [-2,250 to +1,000]. Input out of range will be clipped and recorded in the status queue.

---

Returns:       <numeric>

---



**GENerator:DOPositive:SYMMetry <numeric>**  
**GENerator:DOPositive:SYMMetry?**

Set or retrieve the symmetry adjustment of the Generator Data+ output.

Params: <numeric> Generator Data+ output symmetry, in percent. Range [25 to 75]. Input out of range will be clipped and recorded in the status queue.

---

Returns: <numeric>

---

**GENerator:DOPositive:TAC <bool>**  
**GENerator:DOPositive:TAC?**

Set or retrieve the termination AC of the Generator Data+ output. May require some delay to complete.

Params: <bool> On = 1, Off = 0

---

Returns: < 1 > Generator Data+ output termination AC is on

---

< 0 > Generator Data+ output termination AC is off

---

**GENerator:DOPositive:TVOLTage <numeric>**  
**GENerator:DOPositive:TVOLTage?**

Set or retrieve the termination voltage of the Generator Data+ output. May require some delay to complete.

Params: <numeric> Generator Data+ output termination voltage in mV. Range [-2,000 to +2,000]. Input out of range will be clipped and recorded in the status queue.

---

Returns: <numeric>

---

Complete command list (See page 7, *Command listing*.)

## Generator Data Output Negative

Complete command list (See page 7, *Command listing*.)

### **GENerator:DONegative:CLIPped?**

Retrieve whether the Generator Data– output setting is clipped. Query only.

Returns:	< 1 >	The Generator Data– output setting is clipped
	< 0 >	The Generator Data– output setting is not clipped

### **GENerator:DONegative:ENABLE <bool>**

#### **GENerator:DONegative:ENABLE?**

Set or retrieve whether the Generator Data– output is enabled. May require some delay to complete.

Params:	<bool>	On = 1, Off = 0
Returns:	< 1 >	Generator Data– output is enabled
	< 0 >	Generator Data– output is disabled

### **GENerator:DONegative:IMPedance <numeric>**

#### **GENerator:DONegative:IMPedance?**

Set or retrieve the impedance of the Generator Data– output. May require some delay to complete.

Params:	<numeric>	Generator Data– output impedance in Ohms. Range [30 to 100,000]. Input out of range will be clipped and recorded in the status queue.
Returns:	<numeric>	

### **GENerator:DONegative:LFAMily <LVTTTL | CML | ECL | PECL | LVPECL | LVDS | SCFL>**

#### **GENerator:DONegative:LFAMily?**

Set or retrieve the Generator Data– output logic family. PECL is not available on the BERTScope Analyzer.

Params:	<LVTTTL   CML   ECL   PECL   LVPECL   LVDS   SCFL>	The Generator Data– output logic family
Returns:	< LVTTTL   CML   ECL   PECL   LVPECL   LVDS   SCFL >	

**GENerator:DONegative:SLAMplitude <numeric>**  
**GENerator:DONegative:SLAMplitude?**

Set or retrieve the signal level amplitude of the Generator Data– output. May require some delay to complete.

Params: <numeric> Generator Data– output signal level amplitude in mV. Range [250 to 2,000]. Input out of range will be clipped and recorded in the status queue.

Returns: <numeric>

**GENerator:DONegative:SLOffset <numeric>**  
**GENerator:DONegative:SLOffset?**

Set or retrieve the signal level offset of the Generator Data– output. May require some delay to complete.

Params: <numeric> Generator Data– output signal level offset in mV. Range [–2,000 to +2,000]. Input out of range will be clipped and recorded in the status queue.

Returns: <numeric>

**GENerator:DONegative:SLVHigh <numeric>**  
**GENerator:DONegative:SLVHigh?**

Set or retrieve the signal level  $V_H$  of the Generator Data– output. May require some delay to complete.

Params: <numeric> Generator Data– output signal level  $V_H$  in mV. Range [–1,750 to +3,000]. Input out of range will be clipped and recorded in the status queue.

Returns: <numeric>

**GENerator:DONegative:SLVLow <numeric>**  
**GENerator:DONegative:SLVLow?**

Set or retrieve the signal level  $V_L$  of the Generator Data– output. May require some delay to complete.

Params: <numeric> Generator Data– output signal level  $V_L$  in mV. Range [–2,250 to +1,000mV]. Input out of range will be clipped and recorded in the status queue.

Returns: <numeric>

**GENerator:DONegative:SYMmetry <numeric>**  
**GENerator:DONegative:SYMmetry?**

Set or retrieve the symmetry adjustment of the Generator Data– output.

Params:	<numeric>	Range [25% to 75%]. Input out of range will be clipped and recorded in the status queue.
Returns:	<numeric>	

**GENerator:DONegative:TAC <bool>**  
**GENerator:DONegative:TAC?**

Set or retrieve the termination AC of the Generator Data– output. May require some delay to complete.

Params:	<bool>	On = 1, Off = 0
Returns:	< 1 >	Generator Data– output termination AC is on
	< 0 >	Generator Data– output termination AC is off

**GENerator:DONegative:TVOLTage <numeric>**  
**GENerator:DONegative:TVOLTage?**

Set or retrieve the termination voltage of the Generator Data– output. May require some delay to complete.

Params:	<numeric>	Generator Data– output termination voltage in mV. Range [–2,000 to +2,000]. Input out of range will be clipped and recorded in the status queue.
Returns:	<numeric>	

Complete command list (See page 7, *Command listing*.)

## Generator Data Output Pos/Neg

Complete command list (See page 7, *Command listing*.)

**GENerator:DOUtpuT:LPNSignals <bool>**  
**GENerator:DOUtpuT:LPNSignals?**

Set or retrieve whether the Generator Data+/- outputs are linked. May require some delay to complete.

Params:	<bool>	On = 1, Off = 0
Returns:	< 1 >	Generator Data+/- outputs are linked
	< 0 >	Generator Data+/- outputs are not linked

# Generator Trigger

Complete command list (See page 7, *Command listing*.)

**GENerator:TOFFset <numeric>**

**GENerator:TOFFset?**

Set or retrieve the Generator Trigger offset.

Params:	<numeric>	Generator Trigger offset BitAlyzer: in 32-bit words. Range [0 to (Pattern Size – 1)] for all the PRBS patterns. For others, the range is [0 to 0]. Input out of range will be clipped and recorded in the status queue. BERTScope Analyzer: in 128-bit words. Range [0 to (Pattern Size – 1)] for all the PRBS patterns. For others, the range is [0 to 0]. Input out of range will be clipped and recorded in the status queue.
---------	-----------	--

---

Returns: <numeric>

---

**GENerator:TOMethod <PCYClE | CLOcK>**

**GENerator:TOMethod?**

Set or retrieve the Generator Trigger out method.

Params:	<PCYClE>	Generator Trigger out method is pattern cycle
	<CLOcK>	BitAlyzer: Generator Trigger out method is a divided-down clock/64 BERTScope Analyzer: Generator trigger out method is a divided-down clock/256

---

Returns: < PCYClE | CLOcK >

---

Complete command list (See page 7, *Command listing*.)

## Generator Error Inject

Complete command list (See page 7, *Command listing*.)

**GENerator:EIEType <1BIT | 2BITs | 4BITs | 8BITs | 16Bits | 32Bits | 64Bits | 128Bits>**

**GENerator:EIEType?**

Set or retrieve the Generator Error Inject type. May require some delay to complete.

Params:        <1Bit | 2Bits | 4Bits        Generator error inject type is: 1, 2, 4, 8, 16, 32, 64, or  
                 | 8Bits | 16Bits |        128 bit(s)  
                 32Bits | 64Bits |  
                 128Bits>

---

Returns:       < 1BIT | 2BITs | 4BITs | 8BITs | 16Bits | 32Bits | 64Bits | 128Bits >

---

**GENerator:EIInterval <numeric>**

**GENerator:EIInterval?**

Set or retrieve the Generator Error Inject interval. May require some delay to complete.

Params:        <numeric>                Generator Error Inject interval in bits  
                                                BitAlyzer: Range [32 to 16,777,184]; must be modulo  
                                                32. Input out of range will be clipped and recorded in  
                                                the status queue.  
                                                BERTScope Analyzer : Range [16,384 to 16,777,184];  
                                                must be modulo 128. Input out of range will be clipped  
                                                and recorded in the status queue.

---

Returns:        <numeric>

---

---

**GENerator:EIMode <CONTInuous | MANual | EXTernal | OFF>**  
**GENerator:EIMode?**

Set or retrieve the Generator Error Inject mode. May require some delay to complete.

Params:	<CONTInuous>	Generator Error Inject mode is continuous
	<MANual>	Generator Error Inject mode is manual
	<EXTernal>	Generator is set to inject a single error on receipt of an external signal
	<OFF>	Generator Error Inject is off
Returns:	< CONTINUOUS   MANUAL   EXTERNAL   OFF >	

**GENerator:IBER?**

Retrieve the injected BER of the Generator. Query only.

Returns:	<numeric>	Generator injected bit error rate
----------	-----------	-----------------------------------

Complete command list (See page 7, *Command listing*.)





---

# Generator Stress Module (GSM) - Stressed Eye

Complete command listing (See page 7, *Command listing*.)

All GSM commands apply only to the BERTScope family of analyzers. Before using the GSM commands, send **GSM:STress:ENABLE 1** to enable all stress features (see GSM:STress).

GSM Configure/Enable(See page 41, *GSM - Stressed Eye Configure/Enable*.)

GSM Stress Setup(See page 42, *GSM - Stressed Eye Setup*.)

## GSM - Stressed Eye Configure/Enable

Complete command listing (See page 7, *Command listing*.)

All GSM commands apply only to the BERTScope family of analyzers. Before using the GSM commands, send **GSM:STress:ENABLE 1** to enable all stress features (see **GSM:STress** below).

**GSM:RCONfiguration** <“filename”>

Restore Stress configuration. Action only.

Params:	<“filename”>	Configuration filename used to restore. Filename is enclosed in double quotes.
---------	--------------	---

---

Returns:	< filename >
----------	--------------

---

**GSM:SCONfiguration** <“filename”>

Save Stress configuration. Action only.

Params:	<“filename”>	Configuration filename used to save. Filename is enclosed in quotes.
---------	--------------	---

---

Returns:	< filename >
----------	--------------

---

**GSM:STress:ENABLE <bool>**  
**GSM:STress:ENABLE?**

Set or retrieve whether the entire global stress feature is enabled. The entire stress feature includes jitter insertion of sinusoidal jitter (SJ), external sinusoidal jitter (EXSJ), random jitter (RJ), PRBS jitter (BUJ) and external high frequency jitter (EXHF).

There are commands to enable each individual stress insertion, such as GSM:SJitter:ENABLE; however, unless the Stress Feature is enabled using this command, those individual jitter insertion enabling commands do not enable any jitter insertion. For example, if one sent a **GSM:SJ:ENAB 1** without enabling the stress feature (query **GSM:STR:ENAB?** returns 0), the system merely caches the SJ Enable value. The next time **GSM:STR:ENAB 1** is sent, the SJ is then enabled. After the stress feature is enabled, one can still enable or disable individual types of jitter insertion by sending the corresponding ENABLE command.

This command corresponds to the Clock control’s “Jitter Insertion Enabled” menu item in the Generator view.

Params:	<bool>	On = 1, Off = 0
Returns:	< 1 >	All stress enabled
	< 0 >	All stress disabled

## GSM - Stressed Eye Setup

Complete command listing (See page 7, *Command listing*.)

**GSM:BUJitter:AMPUi <number>**  
**GSM:BUJitter:AMPUi?**

Set or retrieve the PRBS jitter amplitude in percent of UI.

Params:	<number>	Range [0 to 50%]. The amplitude limits change with regard to the analyzer’s operating frequency.
Returns:	< number >	

**GSM:BUJitter:ENABLE <bool>**  
**GSM:BUJitter:ENABLE?**

Set or retrieve whether the PRBS jitter on the clock is enabled.

Params:	<bool>	On = 1, Off = 0
Returns:	< 1 >	PRBS jitter on the clock is enabled
	< 0 >	PRBS jitter on the clock is disabled

**GSM:BUJitter:FREQuency <number>**  
**GSM:BUJitter:FREQuency?**

Set or retrieve the PRBS jitter frequency in Hz.

Params:	<number>	Range [100,000,000 to 2,500,000,000].
Returns:	< number >	

**GSM:BUJitter:TYPE <PRBS7 | PRBS10 | PRBS11>**  
**GSM:BUJitter:TYPE?**

Set or retrieve the PRBS jitter type.

Params:	<PRBS7>	Insert PRBS7 jitter
	<PRBS10>	Insert PRBS10 jitter
	<PRBS11>	Insert PRBS11 jitter.
Returns:	< PRBS7   PRBS10   PRBS11 >	

**GSM:EXHFrequency:ENABle <bool>**  
**GSM:EXHFrequency:ENABle?**

Set or retrieve whether the external high frequency jitter is enabled.

Params:	<bool>	On = 1, Off = 0
Returns:	< 1 >	External high frequency jitter is enabled
	< 0 >	External high frequency jitter is disabled

**GSM:EXSJitter:AMPUI <number>**  
**GSM:EXSJitter:AMPUI?**

Set or retrieve the external sinusoidal jitter amplitude in percent of UI.

Params:	<number>	Range [0 to Maximum allowed %UI]. The amplitude limits change with regard to the analyzer operating frequency.
Returns:	< number >	

**GSM:EXSJitter:MODE**

Command is obsolete; see GSM:EXSJitter:INVErt and GSM:EXSJitter:ENABle.

**GSM:EXSJitter:ENABle <bool>**  
**GSM:EXSJitter:ENABle?**

Enable/disable External Sinusoidal Jitter.

Params:	<bool>	On = 1, Off = 0
Returns:	< 1 >	EXS Jitter is enabled
	< 0 >	EXS Jitter is disabled

**GSM:EXSJitter:INVErt <bool>**  
**GSM:EXSJitter:INVErt?**

Set or retrieve invert state of EXS Jitter.

Params:	<bool>	On = 1, Off = 0
Returns:	< 1 >	EXS Jitter is inverted
	< 0 >	EXS Jitter is not inverted

**GSM:F2Jitter:AMPUI <number>**  
**GSM:F2Jitter:AMPUI?**

Set or retrieve the F/2 Jitter amplitude.

Params:	<number>	F/2 Jitter amplitude in %UI
Returns:	< number >	

**GSM:F2Jitter:AVailable?**

Checks to see if the F/2 Jitter option is available. Query only.

Returns:	< 1 >	F/2 Jitter is available
	< 0 >	F/2 Jitter is not available

**GSM:F2Jitter:ENABLE <bool>**  
**GSM:F2Jitter:ENABLE?**

Set or retrieve whether F/2 Jitter is enabled.

Params:	<bool>	On = 1, Off = 0
Returns:	< 1 >	F/2 Jitter is enabled
	< 0 >	F/2 Jitter is disabled

**GSM:F2Jitter:MAXAmpui?**

Returns the maximum %UI for the current F/2 Jitter synthesizer frequency. Query only.

Returns	<number>	Maximum %UI for F/2 Jitter synthesizer frequency
---------	----------	--

**GSM:LFRJ:AMPUI <number>**  
**GSM:LFRJ:AMPUI?**

Set or retrieve the low frequency random jitter amplitude in percent of UI.

Params:	<number>	LFR Jitter amplitude in %UI
Returns:	< number >	

**GSM:LFRJ:ENABLE <bool>**  
**GSM:LFRJ:ENABLE?**

Set or retrieve whether low frequency random jitter is enabled.

Params:	<bool>	On = 1, Off = 0
Returns:	< 1 >	LFR Jitter is enabled
	< 0 >	LFR Jitter is disabled

**GSM:LFSJ:AMPPS <number>**  
**GSM:LFSJ:AMPPS?**

Set or retrieve the low frequency sinusoidal jitter amplitude in picoseconds.

Params:	<number>	LFS Jitter amplitude in ps
Returns:	< number >	

**GSM:LFSJ:ENABLE <bool>**  
**GSM:LFSJ:ENABLE?**

Set or retrieve whether low frequency sinusoidal jitter is enabled.

Params:	<bool>	On = 1, Off = 0
Returns:	< 1 >	LFS Jitter is enabled
	< 0 >	LFS Jitter is disabled

**GSM:LFSJ:FREQ <number>**  
**GSM:LFSJ:FREQ?**

Set or retrieve the low frequency sinusoidal jitter frequency in Hz.

Params:	<number>	LFSJ frequency in Hz
Returns:	< number >	

**GSM:RCONfiguration <"filename">**

Restore Stress configuration. Action only.

Params:	<"filename">	Configuration filename used to restore. Filename is enclosed in double quotes.
Returns:	< filename >	

**GSM:RJitter:AMPUI <number>**  
**GSM:RJitter:AMPUI?**

Set or retrieve the random jitter amplitude in percent of UI.

Params:	<number>	Range [0 to 50%]. The amplitude limits change with regard to the analyzer operating frequency.
Returns:	< number >	

**GSM:RJitter:ENABle <bool>**  
**GSM:RJitter:ENABle?**

Set or retrieve whether random jitter on the clock is enabled.

Params:	<bool>	On = 1, Off = 0
Returns:	< 1 >	Random jitter on the clock is enabled
	< 0 >	Random jitter on the clock is disabled

**GSM:RJ:TYPE <1GHZ | 100MHZ>**  
**GSM:RJ:TYPE?**

Set or retrieve the random jitter bandwidth type.

Params:	<1GHZ>	RJ bandwidth type 1 GHz
	<100MHZ>	RJ bandwidth type 100 MHz
Returns:	< 1GHZ   100MHZ >	

**GSM:SCONfiguration <"filename">**

Save Stress configuration. Action only.

Params:	<"filename">	Configuration filename used to save. Filename is enclosed in quotes.
Returns:	< filename >	

**GSM:SJitter:AMPUI <number>**  
**GSM:SJitter:AMPUI?**

Set or retrieve the sinusoidal jitter amplitude in percent of UI.

Params:	<number>	Range [0 to Maximum allowed %UI]. The amplitude limits change with regard to the analyzer operating frequency.
Returns:	< number >	

**GSM:SJitter:ENABLE <bool>**  
**GSM:SJitter:ENABLE?**

Set or retrieve whether sinusoidal jitter on the clock is enabled.

Params:	<bool>	On = 1, Off = 0
Returns:	< 1 >	Sinusoidal jitter on the clock is enabled
	< 0 >	Sinusoidal jitter on the clock is disabled

**GSM:SJitter:FREQuency <number>**  
**GSM:SJitter:FREQuency?**

Set or retrieve the sinusoidal jitter frequency in Hz.

Params:	<number>	Range [20,000 to 80,000,000]
Returns:	< number >	

**GSM:SInterference:AMPLitude <number>**  
**GSM:SInterference:AMPLitude?**

Set or retrieve the Data+/- sinusoidal interference amplitude in mV.

Params:	<number>	Range [0 to 400]. The amplitude limits change with regard to the analyzer operating frequency.
Returns:	< number >	

**GSM:SInterference:ENABLE <bool>**  
**GSM:SInterference:ENABLE?**

Set or retrieve whether sinusoidal interference on Data+/- output is enabled.

Params:	<bool>	On = 1, Off = 0
Returns:	< 1 >	Sinusoidal interference on Data+/- is enabled
	< 0 >	Sinusoidal interference on Data+/- is disabled

**GSM:SInterference:FREQuency <number>**  
**GSM:SInterference:FREQuency?**

Set or retrieve the sinusoidal interference frequency in Hz.

Params:	<number>	Sinusoidal interference frequency in Hz. Range [100,000,000 to 2,500,000,000]
Returns:	< number >	

**GSM:SInterference:MODE <INPhase | OUTPhase | EXTErnal>  
GSM:SInterference:MODE?**

Set or retrieve the internal sinusoidal interference mode.

Params:	<INPhase>	Sinusoidal interference is in phase
	<OUTPhase>	Sinusoidal interference is out of phase
	<EXTErnal>	Sinusoidal interference is External
Returns:	< INPHASE   OUTPHASE   EXTERNAL >	

**GSM:SINEgative:AMPLitude <number>  
GSM:SINEgative:AMPLitude?**

Set or retrieve the Data- output sinusoidal interference amplitude in mV.

Params:	<number>	Range [0 to 400]. The amplitude limits change with regard to the analyzer operating frequency.
Returns:	< number >	

**GSM:SINEgative:ENABLE <bool>  
GSM:SINEgative:ENABLE?**

Set or retrieve whether sinusoidal interference on the Data- output is enabled.

Params:	<bool>	On = 1, Off = 0
Returns:	< 1 >	Sinusoidal interference on Data- is enabled
	< 0 >	Sinusoidal interference on Data- is disabled

**GSM:SIPositive:AMPLitude <number>  
GSM:SIPositive:AMPLitude?**

Set or retrieve the Data+ output sinusoidal interference amplitude in mV.

Params:	<number>	Range [0 to 400]. The amplitude limits change with regard to the analyzer operating frequency.
Returns:	< number >	

**GSM:SIPositive:ENABLE <bool>  
GSM:SIPositive:ENABLE?**

Set or retrieve whether sinusoidal interference on the Data+ output is enabled.

Params:	<bool>	On = 1, Off = 0
Returns:	< 1 >	Sinusoidal interference on Data+ is enabled
	< 0 >	Sinusoidal interference on Data+ is disabled



**GSM:STress:ENABLE <bool>**  
**GSM:STress:ENABLE?**

Set or retrieve whether the entire global stress feature is enabled. The entire stress feature includes jitter insertion of sinusoidal jitter (SJ), external sinusoidal jitter (EXSJ), random jitter (RJ), PRBS jitter (BUJ) and external high frequency jitter (EXHF).

There are commands to enable each individual stress insertion, such as GSM:SJitter:ENABLE; however, unless the Stress Feature is enabled using this command, those individual jitter insertion enabling commands do not enable any jitter insertion. For example, if one sent a “GSM:SJ:ENAB 1” without enabling the stress feature (query “GSM:STR:ENAB?” returns 0), the system merely caches the SJ Enable value. The next time “GSM:STR:ENAB 1” is sent, the SJ is then enabled. After the stress feature is enabled, one can still enable or disable individual types of jitter insertion by sending the corresponding ENABLE command.

This command corresponds to the Clock control “Jitter Insertion Enabled” menu item in the Generator view.

Params:	<bool>	On = 1, Off = 0
Returns:	< 1 >	All stress enabled
	< 0 >	All stress disabled



# Detector

Complete command listing (See page 7, *Command listing*.)

**DETECTOR:ATTENFACTOR <numeric>**  
**DETECTOR:ATTENFACTOR?**

BERTScope Analyzer only. Set or retrieve the Detector Data Input Attenuation Factor.

---

**NOTE.** *The format used for parameter and return value is determined by the current attenuation factor mode setting. See the DET:ATTENFM command.*

---

Params:	<numeric>	Valid ranges: 0.001:1 to 1000:1 or -60 dB to +60 dB
Returns:	<numeric>	

---

**DETECTOR:ATTENFACTORMODE <DECIBEL | RATIO>**  
**DETECTOR:ATTENFACTORMODE?**

BERTScope Analyzer only. Set or retrieve Detector attenuation factor mode: Decibel or Ratio.

Params:	<DECIBEL>	The BERTScope Analyzer interprets and displays the Detector Data Input Attenuation Factor in dB. (Example: 20 dB)
	<RATIO>	BERTScope Analyzer interprets and displays the Detector Data Input Attenuation Factor as a ratio. (Example: 20:1)
Returns:	< DECIBEL   RATIO >	

---

**DETECTOR:BINVERT <bool>**  
**DETECTOR:BINVERT?**

Set or retrieve the blank invert of the Detector.

Params:	<bool>	On = 1, Off = 0
Returns:	< 1 >	Detector Blank is inverted
	< 0 >	Detector Blank is not inverted

---

**DETECTOR:BRM <numeric>**  
**DETECTOR:BRM?**

Set or retrieve the Bit Rate Multiplier.

Params: <numeric> Range [1 to 8]. A value of -1 calculates the Bit Rate Multiplier value automatically.

---

Returns: <numeric>

---

**DETECTOR:CINVert <bool>**  
**DET:CINV <bool>**  
**DETECTOR:CINVert?**  
**DET:CINV?**

Set or retrieve the clock invert of the Detector.

Params: <bool> On = 1, Off = 0

---

Returns: < 1 > Detector Clock is inverted

---

< 0 > Detector Clock is not inverted

---

**DETECTOR:DDEFs <numeric>**  
**DETECTOR:DDEFs?**

Set or retrieve the Detector data delay in femto-seconds.

Params: <numeric> Set Detector data delay in fs

---

Returns: <numeric>

---

**DETECTOR:DDELay <numeric>**  
**DETECTOR:DDELay?**

Set or retrieve the data delay of the Detector (in picoseconds).

Params: <numeric> Detector Data delay.  
 BitAlyzer: When operating within the range [1 to 981 MHz], Detector Data Delay is within a range [0 to 33,000 ps]. Input out of range will be clipped and recorded in the status queue.  
 When operating within the range [982 to 1,500 MHz], the Detector Data Delay is within a range [0 to 5,000 ps].  
 BERTScope Analyzer: When the Analyzer is operating within the range [500 to 1,000 MHz], the Detector Data Delay is within a range [0 to 32,000 ps]. Input out of range will be clipped and recorded in the status queue.  
 When the Analyzer is operating within the range [1,100 to 12,500 MHz], the Detector Data Delay is within a range [0 to 3,000 ps].

---

Returns: <numeric>

---

**DELAY:DETRecal?**

Retrieve the information that indicates whether Detector delay needs recalibration. Query only.

**NOTE.** *Since monitoring for these calibrations is suspended while Physical Layer tests are running, using this command under those circumstances may not return an accurate result.*

Returns:	< 1 >	Needs Delay Recalibration
	< 0 >	Does not need a Delay Recalibration

**DETECTOR:DINVert <bool>****DETECTOR:DINVert?**

Set or retrieve the data invert of the Detector.

Params:	<bool>	On = 1, Off = 0
Returns:	< 1 >	Detector Data is inverted
	< 0 >	Detector Data is not inverted

**DETECTOR:DRATE?**

Retrieve the data rate of the Detector. Query only.

Returns:	<numeric>	Detector Data rate in Bits/s. 0 = No Clock.
----------	-----------	---

**DETECTOR:FULLRATEclock <bool>****DETECTOR:FULLRATEclock?**

Set or retrieve the Full Rate Clock property of the Detector.

Params:	<bool>	On = 1, Off = 0
Returns:	< 1 >	Detector is in Full Rate Clock mode
	< 0 >	Detector is in Half Rate Clock mode

**DETECTOR:OPERmode <ELEC | OPTIC>****DETECTOR:OPERmode?**

BERTScope Analyzer only. Set or retrieve Detector operating mode, Electrical or Optical.

Params:	<ELEC>	BERTScope Analyzer operates in Electrical mode
	<OPTIC>	BERTScope Analyzer operates in Optical mode
Returns:	< ELEC   OPTIC >	

**DEtector:PCALibration**

Perform Detector delay calibration. Action only. May require some delay to complete.

**DEtector:RSNumber?**

Retrieve the result serial number of the Detector. Query only.

Returns:      <numeric>                      Detector's result serial number

---

## Detector Error Detector

Complete command listing (See page 7, *Command listing*.)

**DEtector:ARENable <bool>**

**DET:AREN <bool>**

Set or retrieve the automatic resync setting of the Detector.

Params:      <bool>                      On = 1, Off = 0

---

Returns:      < 1 >                      Detector automatic resync is on

---

                 < 0 >                      Detector automatic resync is off

---

**DEtector:ARTHreshold <numeric>**

**DEtector:ARTHreshold?**

Set or retrieve the threshold of the Detector automatic resync.

Params:      <numeric>                      Detector automatic resync threshold in number of errors.  
Range [3 to 100,000]. Input out of range will be clipped  
and recorded in the status queue.

---

Returns:      <numeric>

---

**DEtector:DPATtern?**

Retrieve detected pattern of the Detector. The PRBS-11 pattern is available only on BERTScope Analyzers. Query only.

Returns:	GRABNGO	Grab and Go
	NO_CLOCK	No clock
	NONE	No pattern
	NOT_AVAILABLE	Detector is in Live Data mode
	ONE	All ones
	IPN7   IPN11   IPN15   IPN20   IPN23   IPN31	Pseudo-random pattern: PRBS-7, PRBS-11 <sup>1</sup> , PRBS-15, PRBS-20, PRBS-23, or PRBS-31
	RAM_SHIFTNSYNC	Shift and Sync
	RAM_TRIGGER	Trigger
	IGRABNGO	Inverted Grab and Go
	IPN7   IPN11   IPN15   IPN20   IPN23   IPN31	Inverted pseudo-random pattern: PRBS-7, PRBS-11 <sup>1</sup> , PRBS-15, PRBS-20, PRBS-23, or PRBS-31
	IRAM_SHIFTNSYNC	Inverted Shift and Sync
	IRAM_TRIGGER	Inverted Trigger
	ZERO	All zeros

<sup>1</sup> \*PN11 pattern is available only on the BERTScope Analyzer.

**DEtector:GLENgtH <integer>****DEtector:GLENgtH?**

Set or retrieve the Detector capture length. The action causes the machine to capture the input number of words into the Detector. The query returns the amount of captured RAM in words.

Params:	<integer>	Number of words to capture
Returns:	< integer >	Number of words captured

**DEtector:ISYNc?**

Retrieve whether the Detector is in sync. Query only.

Returns:	< 1 >	Detector is in sync
	< 0 >	Detector is not in sync

**DEtector:LUPFilename?**

Retrieve user pattern file that has been loaded to the Detector. Query only.

Returns:	< filename >	User pattern file that has been loaded to the Detector.
----------	--------------	---

**DETECTOR:NALArm?**

Retrieve the negative input protection alarm. Query only.

Returns:	< 1 >	Negative input protection alarm is on
	< 0 >	No negative input protection alarm

**DETECTOR:PALArm?**

Retrieve the positive input protection alarm state. Query only.

Returns:	< 1 >	Positive input protection alarm is on
	< 0 >	No positive input protection alarm

**DETECTOR:PATtern <PN7 | PN11 | PN15 | PN20 | PN23 | PN31 | UStart | UGRab | USHift | AUTomatic | ALLZERO>  
DETECTOR:PATtern?**

Set or retrieve the data type of the Detector. PN11 pattern is available only on BERTScope Analyzers.

Params:	<PN7   PN11   PN15   PN15   PN20   PN23   PN31>	Pseudo-random data types: PRBS-7, PRBS-11 <sup>1</sup> , PRBS-15, PRBS-20, PRBS-23, PRBS-31
	<UStart>	User pattern (See GEN:UPLoad to load a user pattern file.)
	<UGRab>	Grab and Go
	<USHift>	Shift and Sync
	<AUTomatic>	Automatic detection
	<ALLZERO>	All-zeros pattern
Returns:	< PN7   PN11   PN15   PN20   PN23   PN31   UStart   UGRab   USHift   AUTomatic   ALLZERO >	

<sup>1</sup> The PN11 pattern is available only on BERTScope Analyzers.

**DETECTOR:RESEtall**

Reset the Detector results AND Error Analysis results. Action only.

**DETECTOR:RRESults**

Reset the Detector results. Action only.



**DEtector:SETHreshold <integer>**  
**DEtector:SETHreshold?**

Set or retrieve the Detector Shift Error Tolerance. This action causes the machine to capture the input number of words into the detector.

Params: <integer>                      Number of errors permitted for synchronizing

---

Returns: < integer >

---

**DEtector:UPLoad<"filename">**

Load user pattern from the specified file. Action only.

Params: <"filename">                      Use the absolute path name and enclose the string in quotes.

---

**DEtector:UPLPercent?**

Retrieve the user pattern loading percentage completion value. It returns 100 if the loading is complete. Query only.

Returns: < number >                      The percentage of user pattern loaded into the Detector

---

**DEtector:UPWLen?**

Retrieve the word count from the Generator RAM. Query only.

Returns: < number >                      The number of words in RAM

---

## Detector Start Detect

Complete command listing (See page 7, *Command listing*.)

**DETECTOR:SDETect:TAC <bool>**

**DETECTOR:SDETect:TAC?**

Set or retrieve the termination AC of the Detector Start Detect.

Params:	<bool>	On = 1, Off = 0
Returns:	< 1 >	Detector Start Detect Termination AC is on
	< 0 >	Detector Start Detect Termination AC is off

**DETECTOR:SDETect:THREShold <numeric>**

**DETECTOR:SDETect:THREShold?**

Set or retrieve the threshold of the Detector Start Detect in mV.

Params:	<numeric>	Range [ 2,000 to +4,000]. Input out of range will be clipped and recorded in the status queue.
Returns:	<numeric>	

**DETECTOR:SDETect:TMODe <REDGe | FEDGe | DISabled>**

**DETECTOR:SDETect:TMODe?**

Set or retrieve the trigger mode of the Detector Start Detect.

Params:	<REDGe>	Trigger at the rising edge
	<FEDGe>	Trigger at the falling edge
	<DISabled>	Trigger disabled
Returns:	< REDGE   FEDGE   DISABLED >	

**DETECTOR:SDETect:TVOLtage <numeric>**

**DETECTOR:SDETect:TVOLtage?**

Set or retrieve the termination voltage of the Detector Start Detect in mV.

Params:	<numeric>	Range [-2,000 to +3,300]. Input out of range will be clipped and recorded in the status queue.
Returns:	<numeric>	

# Detector Clock Input

Complete command listing (See page 7, *Command listing*.)

## **DETECTOR:CINPut:CLIPped?**

Retrieve whether the Detector Clock input setting is clipped. Query only.

Returns:	< 1 >	Detector Clock input setting is clipped
	< 0 >	Detector Clock input setting is not clipped

## **DETECTOR:CINPut:IMODE <DIFFerential | SENDEd | ISENDEd> DETECTOR:CINPut:IMODE?**

Set or retrieve the interface mode of the Detector Clock input. May require some delay to complete.

Params:	<DIFFerential>	Differential interface
	<SENDEd>	Single-ended interface
	<ISENDEd>	Inverted single-ended interface
Returns:	< Differential   SingleEnded   SingleEndedInverted >	

## **DETECTOR:CINPut:LFAMILY <LVTTL | CML | ECL | PECL | LVPECL | LVDS | SCFL | DLVTTL | DCML | DECL | DPECL | DLVPECL | DLVDS> DETECTOR:CINPut:LFAMILY?**

Set or retrieve the clock input logic family. PECL and DPECL are not available on the BERTScope Analyzer.

Params:	<LVTTL   CML   ECL   PECL   LVPECL   LVDS   SCFL   DLVTTL   DCML   DECL   DPECL   DLVPECL   DLVDS>	The input logic family
Returns:	< LVTTL   CML   ECL   PECL   LVPECL   LVDS   SCFL   DLVTTL   DCML   DECL   DPECL   DLVPECL   DLVDS >	

## **DETECTOR:CINPut:TAC <bool> DETECTOR:CINPut:TAC?**

Set or retrieve the termination AC of the Detector Clock input. May require some delay to complete.

Params:	<bool>	On = 1, Off = 0
Returns:	< 1 >	Detector Clock input termination AC is on
	< 0 >	Detector Clock input termination AC is off

**DETECTOR:CINPut:THReshold <numeric>**  
**DETECTOR:CINPut:THReshold?**

Set or retrieve the threshold of the Detector Clock input in mV. May require some delay to complete. Available only on BitAlyzer.

Params: <numeric> Range [-2,000 to +4,000]. Input out of range will be clipped and recorded in the status queue.

---

Returns: <numeric>

---

**DETECTOR:CINPut:TVOLTage <numeric>**  
**DETECTOR:CINPut:TVOLTage?**

Set or retrieve the termination voltage of the Detector Clock input in mV. May require some delay to complete.

Params: <numeric> Range [-2,000 to +3,300]. Input out of range will be clipped and recorded in the status queue.

---

Returns: <numeric>

---

**DETECTOR:CLocKOPTimize**

BitAlyzer only. Initiates procedure for optimizing the location of the Detector Clock input threshold. Locates mid-point of clock envelope. May take seconds to complete. Action only.

**DETECTOR:CLocKOPTResult?**

BitAlyzer only. Returns the located threshold in millivolts. If optimization fails, returns -99999999 as a sentinel value. Query only.

## Detector Data Input

Complete command listing (See page 7, *Command listing*.)

**DETECTOR:ATTENFACTOR <numeric>**

**DETECTOR:ATTENFACTOR?**

Set or retrieve the Detector Data Input Attenuation Factor.

---

**NOTE.** *The format used for parameter and return value is determined by the current attenuation factor mode setting.*

---

Params:	<numeric>	Valid ranges [0.001:1 to 1000:1] or [-60 dB to +60 dB]
---------	-----------	--

Returns:	<numeric>
----------	-----------

---

**DETECTOR:ATTENFACTORMODE <DECIBEL | RATIO>**

**DETECTOR:ATTENFACTORMODE?**

BERTScope Analyzer only. Set or retrieve Detector attenuation factor mode, Decibel or Ratio.

Params:	<DECIBEL>	BERTScope Analyzer interprets and displays the Detector Data Input Attenuation Factor in dB. ( Example: 20 dB)
---------	-----------	--

	<RATIO>	BERTScope Analyzer interprets and displays the Detector Data Input Attenuation Factor as a ratio. ( Example: 20:1)
--	---------	--

Returns:	< DECIBEL   RATIO >
----------	---------------------

---

**DETECTOR:DINPUT:CLIPPED?**

Retrieve whether the Detector Data input setting is clipped. Query only.

Returns:	< 1 >	Detector Data input setting is clipped
----------	-------	--

	< 0 >	Detector Data input setting is not clipped
--	-------	--

---

**DETECTOR:DINPUT:IMODE <DIFFERENTIAL | SENDDED | ISENDDED>**

**DETECTOR:DINPUT:IMODE?**

Set or retrieve the interface mode of the Detector Data input. May require some delay to complete.

Params:	<DIFFERENTIAL>	Differential interface
---------	----------------	------------------------

	<SENDDED>	Single-ended interface
--	-----------	------------------------

	<ISENDED>	Inverted single-ended interface
--	-----------	---------------------------------

Returns:	< Differential   SingleEnded   SingleEndedInverted >
----------	--

---

**DETECTOR:DINPut:LFAMILY <LVTTL | CML | ECL | PECL | LVPECL | LVDS | SCFL | DLVTTL | DCML | DECL | DPECL | DLVPECL | DLVDS>  
DETECTOR:DINPut:LFAMILY?**

Set or retrieve the Data input logic family. PECL and DPECL are not available on the BERTScope Analyzer.

Params:	<LVTTL   CML   ECL   PECL   LVPECL   LVDS   SCFL   DLVTTL   DCML   DECL   DPECL   DLVPECL   DLVDS>	The data input logic family
---------	--	-----------------------------

---

Returns:	< LVTTL   CML   ECL   PECL   LVPECL   LVDS   SCFL   DLVTTL   DCML   DECL   DPECL   DLVPECL   DLVDS >
----------	--

---

**DETECTOR:DINPut:TAC <bool>  
DETECTOR:DINPut:TAC?**

Set or retrieve the termination AC of the Detector Data input in mV. May require some delay to complete. This command is available only on the BitAlyzer.

Params:	<bool>	On = 1, Off = 0
---------	--------	-----------------

---

Returns:	< 1 >	Detector Data input termination AC is on
	< 0 >	Detector Data input termination AC is off

---

**DETECTOR:DINPut:THRESHOLD <numeric>  
DETECTOR:DINPut:THRESHOLD?**

Set or retrieve the threshold of the Detector Data input in mV. May require some delay to complete.

Params:	<numeric>	Range [-2,400 to +2,500]. Input out of range will be clipped and recorded in the status queue.
---------	-----------	--

---

Returns:	<numeric>
----------	-----------

---

**DETECTOR:DINPut:TVOLTAGE <numeric>  
DETECTOR:DINPut:TVOLTAGE?**

Set or retrieve the termination voltage of the Detector Data input in mV. May require some delay to complete.

Params:	<numeric>	Range [-2,000 to +3,300]. Input out of range will be clipped and recorded in the status queue.
---------	-----------	--

---

Returns:	<numeric>
----------	-----------

---

## Detector Blank Input

Complete command listing (See page 7, *Command listing*.)

**DETECTOR:BINPut:CDBLank <bool>**

**DETECTOR:BINPut:CDBLank?**

Set or retrieve the “Count during blank” setting of the Detector Blank input.

Params:	<bool>	On = 1, Off = 0
---------	--------	-----------------

Returns:	< 1 >	Detector blank input count during blank is on
----------	-------	---

	< 0 >	Detector blank input count during blank is off
--	-------	--

**DETECTOR:BINPut:ROEND <bool>**

**DETECTOR:BINPut:ROEND?**

Set or retrieve the “Resync on end” setting of the Detector Blank input.

Params:	<bool>	On = 1, Off = 0
---------	--------	-----------------

Returns:	< 1 >	Detector blank input resync on end is on
----------	-------	--

	< 0 >	Detector blank input resync on end is off
--	-------	---

## Detector Trigger

Complete command listing (See page 7, *Command listing*.)

**DETECTOR:TOFFset <numeric>**

**DETECTOR:TOFFset?**

Set or retrieve the Detector Trigger offset.

Params:      <numeric>            Detector Trigger offset.  
BitAlyzer in 32-bit words. Range [0 to (Pattern Size 1) if the detected pattern is one of the PRBS patterns. For other patterns, the range is [0 to 0]. Input out of range will be clipped and recorded in the status queue.  
BERTScope Analyzer: in 128-bit words. Range [0 to (Pattern Size - 1)] if the detected pattern is one of the PRBS patterns. For others, the range is [0 to 0]. Input out of range will be clipped and recorded in the status queue.

---

Returns:      <numeric>

---

**DETECTOR:TOMethod <PCYCLE | CLOCK>**

**DETECTOR:TOMethod?**

Set or retrieve the Detector Trigger out method.

Params:      <PCYCLE>            Detector Trigger out method is pattern cycle  
                 <CLOCK>            BitAlyzer: Detector trigger out method is clock/64

---

Returns:      < PATTERNCYCLE | CLOCK/64 >

---



## Detector Results

Complete command listing (See page 7, *Command listing*.)

**DETECTOR:BDMODE <TACCUMULATION | INTerval>**  
**DETECTOR:BDMODE?**

Set or retrieve the Detector BER display mode.

Params:	<TACCumulation>	Display the total accumulation BER
	<INTerval>	Display the BER of the interval

Returns:	< TACCUMULATION   INTERVAL >
----------	------------------------------

**DETECTOR:BEDFORMAT <DECimal | SCientific>**  
**DETECTOR:BEDFORMAT?**

Set or retrieve the Detector's display format for bits and errors.

Params:	<DECimal>	Display the bits and errors as decimal
	<SCientific>	Display the bits and errors in scientific format

Returns:	< DECIMAL   SCIENTIFIC >
----------	--------------------------

**DETECTOR:BER?**

Retrieve the current bit error rate. Query only.

**DETECTOR:BITS?**

Retrieve how many bits the Detector has received. Query only.

**DETECTOR:EFBits?**

Retrieve the latest count of error free bits. Query only.

**DETECTOR:EFTime?**

Retrieve the latest error free time. Query only.

**DETECTOR:ETIME?**

Retrieve the elapsed time since last reset. Query only.

**DETECTOR:ERRors?**

Retrieve how many errors the Detector has detected. Query only.

**DETECTOR:RESEtall**

Reset the Detector AND Error Analysis results. Action only.

**DETECTOR:RRESults**

Reset the Detector results. Action only.

**DETECTOR:RESYNCS?**

Retrieve how many resyns the Detector has tried. Query only.

**DETECTOR:RRESULTS**

Reset the Detector results. Action only.

**DETECTOR:RUINTERVAL <numeric>**

**DETECTOR:RUINTERVAL?**

Set or retrieve the Detector's results update interval.

Params:      <numeric>                      Detector results update interval in seconds. Range [1 to 3,600]. Input out of range will be clipped and recorded in the status queue.

---

Returns:      <numeric>

---

## Detector Symbol Filtering

Complete command listing (See page 7, *Command listing*.)

### **DETECTOR:SYMBOLFILTERING:ENABLE <bool>** **DETECTOR:SYMBOLFILTERING:ENABLE?**

Set or retrieve whether Detector Symbol Filtering option is enabled (the corresponding option must be purchased). Available only on BERTScope 17.5G and 26G analyzers.

Params:	<bool>	On = 1, Off = 0
Returns:	< 1 >	Detector Symbol Filtering is on
	< 0 >	Detector Symbol Filtering is off

### **DETECTOR:SYMBOLFILTERING:PRESET <USB3 | SATA | PCIE | PCIESWITCH | SASALIGN | SASPRIMITIVES>**

Sets Detector Symbol Filtering according to a preset. The presets are equivalent to setting the symbols as shown below. Available only on BERTScope 17.5G and 26G analyzers. Action only.

Params:	<USB3 >	Set symbols to: K28.1, K28.1
	<SATA>	Set symbols to: K28.1, D10.2, D10.2, D27.3
	<PCIE>	Set symbols to: K28.0, K28.0
	<PCIESWITCH>	Set symbols to: K28.0
	<SASALIGN>	
	<SASPRIMITIVES>	
Returns:	< USB3   SATA   PCIE   PCIESWITCH   SASALIGN   SASPRIMITIVES >	

### **DETECTOR:SYMBOLFILTERING:SYMBOLS** **<sym0>[,<sym1>[,<sym2>[,<sym3>]]]** **DETECTOR:SYMBOLFILTERING:SYMBOLS?**

Set or retrieve symbol sequence for Detector Symbol Filtering. User can specify up to four symbols to be filtered. Available only on BERTScope 17.5G and 26G analyzers. Usage examples:

```
DET:SYMFIL:SYMBOLS K28.1,K28.1
DET:SYMFIL:SYMBOLS K28.1,D10.2,D10.2,D27.3
```

Params:	<sym0>[,<sym1>[,<sym2>[,<sym3>]]]	Set up to 4 symbols to be filtered
Returns:	< sym0 >[,< sym1 >[,< sym2 >[,< sym3 >]]]	>

## Detector Convenience

Complete command listing (See page 7, *Command listing*.)

### **DETECTOR:PDARKcalibration**

Perform Dark Calibration. Action only. May require some delay to complete.

### **DETECTOR:PDCenter**

Perform Data Centering. This command is equivalent to the Auto Align function on the local control interface. Action only. May require some delay to complete.

### **DETECTOR:MRESync**

Perform manual resync. Action only.

## Detector Auto Align Results

Complete command listing (See page 7, *Command listing*.)

### **DETECTOR:DCAMv?**

#### **DET:DCAM?**

Retrieve the data center amplitude in mV. Query only.

### **DETECTOR:DCHMv?**

#### **DET:DCHM?**

Retrieve the data center height in mV. Query only.

### **DETECTOR:DCSuccess?**

#### **DET:DCS?**

Retrieve whether Detector Data Centering (Auto Align) succeeds or not. Query only.

Returns:	< 1 >	Data centering is successful
	< 0 >	Data centering is not successful

---

### **DETECTOR:DCTime?**

#### **DET:DCT?**

Retrieve the data center time. Query only.

### **DETECTOR:DCUinterval?**

#### **DET:DCU?**

Retrieve the data center unit interval. Query only.

**DETECTOR:DCUMv?****DET:DCUM?**

Retrieve the data center Ua in mV. Query only.

**DETECTOR:DCWidth?****DET:DCW?**

Retrieve the data center width. Query only.

**DETECTOR:IVALue?****DET:IVAL?**

Retrieve the illegal value. Query only. Before data centering is successful, all the result queries will return this illegal value. The user can read the value to know what it is.

## Detector Optical Mode

Complete command listing (See page 7, *Command listing*.)

---

**NOTE.** *In Electrical mode, values are displayed in mV or percent of amplitude. In Optical mode, choose between mW or dBm.*

---

**DETECTOR:OPERmode <ELEC | OPTIC>****DETECTOR:OPERmode?**

BERTScope Analyzer only. Set or retrieve Detector operating mode, Electrical or Optical.

Params:	<ELEC>	BERTScope Analyzer operates in Electrical mode
	<OPTIC>	BERTScope Analyzer operates in Optical mode

---

Returns: < ELEC | OPTIC >

---

**OPTIC:ADDRcv <“string”,long,double,double,bool>**

Add a new optical Receiver. Action only.

Params:	<“Name”>	String of (maximum) 15 characters. String is enclosed in double quotes.
	<Wavelength>	Range [500 to 2000]
	<Gain>	Range [1 to 3000]
	<Offset>	Range [500 to 2000]
	<ACCoupled>	Boolean: True (1) for AC Coupled, False (0) for DC Coupled

---

**OPTIC:DELRev <"string">**

Delete an optical Receiver.

Params: <"Name"> String of (maximum) 15 characters. String is enclosed in double quotes. There should be at least one valid optical receiver.

---

**OPTIC:PIKRev <"string">**

**OPTIC:PIKRev?**

Select or retrieve the selected optical Receiver.

Params: <"Name"> String of (maximum) 15 characters. String is enclosed in double quotes. There should be at least one valid optical receiver.

---

Returns: <"string">

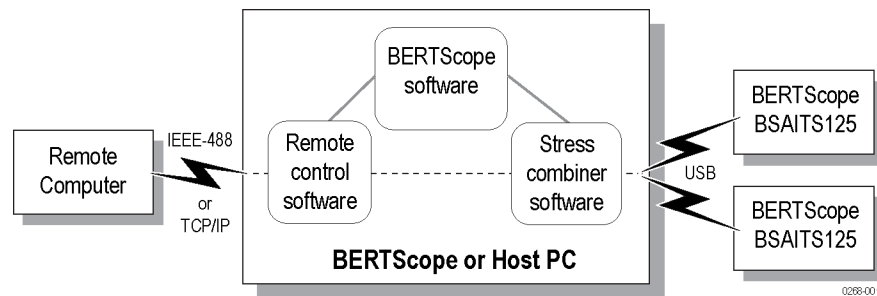
---

# Stress Combiner Option

Complete command listing (See page 7, *Command listing*.)

Software is available to control the BSAITS125 Interference test set through the BERTScope Analyzer interface. The following commands apply only to a BERTScope Analyzer with the BSAITS125 or BSAITS125 EXP capability installed. The remote control software runs on either a BERTScope Analyzer, or a host PC running Microsoft Windows 2000 or XP operating systems.

The remote control software accepts text-oriented commands from a remote computer via either an IEEE-488 or TCP/IP connection. It then routes the commands to the BSAITS125, connected to the BERTScope Analyzer or to a host PC through a USB connection.



If the local host is a BERTScope Analyzer, then the Analyzer software gets a chance to handle the command first. Any commands the Analyzer software doesn't understand are sent to the stress combiner software, which then controls the BERTScope BSAITS125.

When a BSAITS125 is connected to a BERTScope Analyzer, you can intermix BERTScope Analyzer and stress combiner commands in the same script, program, or interactive session. If the host is just an ordinary PC, then the commands are routed to the stress combiner software directly. In this case, using the BERTScope-specific commands will result in an error.

For most applications, the BSAITS125 is connected to a BERTScope Analyzer or host PC. In this case, the BSAITS125 is automatically detected and connected when the remote control software is started, and automatically disconnected when the remote control software is stopped. There is no need in this case to use the OPEN or CLOSE commands described below.

If more than one instrument is connected to the BERTScope Analyzer or host computer, the remote control software will not connect automatically. In this case, the Remote computer must issue a NAMES? query to discover the IDs of the connected BERTScope instruments, and OPEN the one desired before issuing control commands. If the Remote computer needs to control multiple BERTScope instruments, it would OPEN, control, then CLOSE one, then OPEN, control, and CLOSE another. The device that is OPEN is referred to as the current device throughout this document.

Remote control is implemented using a text-oriented command protocol described in the following pages of this document. These commands enable you to set and query the system parameters of the BSAITS125, and to retrieve measurements made by the instrument.

## Stress Combiner command summary

Complete command listing (See page 7, *Command listing*.)

---

**NOTE.** *Commands can only be sent to a BSAITS125 instrument when the instrument is connected to the system. You can send a query if the device is connected using the CONNECTED query and start using other Read/Write properties. Currently, only a single BSAITS125 instrument can be connected to either a BERTScope instrument or to a laptop for control.*

---

<b>Dynamic Read-Only Properties</b>	ATTACHDEVCOUNT or ADCT ATTACHDEVNAMES or NAMES TOTALDEVCOUNT or TDCT
<b>Read-Write Settings</b>	OPEN
<b>Write-Only Operations</b>	CLOSE
<b>Static Read-Only Identification and Capabilities</b>	DATAHZAUTO or DATA DEVICENAME or DN ENABVARISI or ENAB ISSECPRES or ISSE LINKPRISEC or LINK SERIALNUM or SN USEDPP or USED
<b>Dynamic Read-Only Properties</b>	CONNECTED or CONN
<b>Read-Write Settings</b>	CABLOSS or CABL CMBISI or CMBI DPPISI or DPPI EXT1 EXT2 OUTINF or OUTI



PRIISI  
REFREQ or REFF  
SECISI  
VARISI or VARI

## Stress Combiner commands

Complete command listing (See page 7, *Command listing*.)

### **STRCmb:AttachDevCount?**

Retrieves the number of unique BSAITS125s (unique serial numbers) that are currently responding. Query only.

Returns: < numeric >

---

### **STRCmb:attachDevNAMES?**

Retrieves a comma-separated list of all BSAITS125s that are currently connected. Query only.

Each BSAITS125 is identified by its device name.

Returns: < "string" >                      Comma-separated list of BERTScope BSAITS125s found, or "" if none found.

---

### **STRCmb:CABLoss <numeric>**

#### **STRC:CABLoss?**

Sets or retrieves the cable loss in dB.

Params: < numeric >                      Cable loss in dB

---

Returns: < numeric >

---

### **STRCmb:CLOSE**

Close the communication connection to the current BSAITS125. Action only.

---

**NOTE.** *Closing the current BSAITS125 requires that a new BSAITS125 be selected (opened) before issuing any more device-specific commands.*

---

**STRCmb:CMBIsi < PASSTHROUGH | -3.9dB | -5.3dB | -6.6dB | -8dB | -9.3dB | -10.7dB | -12dB | -13.4dB | -14.1dB | -14.7dB | -15.4dB | -16.1dB | -16.8dB | -17.4dB | -18.1dB | -18.8dB | -19.5dB | -20.1dB | -21.5dB | -22.8dB | -24.2dB | -25.5dB | EXT1 | EXT2 >**  
**STRCmb:CMBIsi?**

Set or retrieve the combined ISI setting.

Params: < PASSTHROUGH | -3.9dB | -5.3dB | -6.6dB | -8dB | -9.3dB | -10.7dB | -12dB | -13.4dB | -14.1dB | -14.7dB | -15.4dB | -16.1dB | -16.8dB | -17.4dB | -18.1dB | -18.8dB | -19.5dB | -20.1dB | -21.5dB | -22.8dB | -24.2dB | -25.5dB | EXT1 | EXT2 >

Returns: < PASSTHROUGH | -3.9dB | -5.3dB | -6.6dB | -8dB | -9.3dB | -10.7dB | -12dB | -13.4dB | -14.1dB | -14.7dB | -15.4dB | -16.1dB | -16.8dB | -17.4dB | -18.1dB | -18.8dB | -19.5dB | -20.1dB | -21.5dB | -22.8dB | -24.2dB | -25.5dB | EXT1 | EXT2 >

### **STRCmb:CONNEcted?**

Retrieves the connection status of the currently selected BSAITS125. Query only.

Params: < 0 > Device is open, but not found by the host computer.  
 < 1 > Device is open and available for communication.

Returns: <0 | 1>

### **STRCmb:DATAhzauto <0 | 1>** **STRC:DATAhzauto?**

Set or determine if the reference frequency will be found automatically or entered manually.

Params: < 0 > Reference frequency will not be found automatically;  
 < 1 > frequency will be entered manually.  
 Reference frequency will be found automatically.

Returns: <0 | 1>

### **STRCmb:DeviceName?**

Retrieves the device name of the currently selected BSAITS125. Query only.

Each BSAITS125s is identified by its device name.

Returns: < "string" > Name of the current device. The device name can be from 1 to 11 characters.

**STRCmb:DPPIsi <numeric>**  
**STRC:DPPIsi?**

Sets or retrieves the DPP ISI in dB.

Params:	< numeric >	DPP ISI in dB
Returns:	< numeric >	

**STRCmb:ENABvarisi <0 | 1>**  
**STRC:ENABvarisi?**

Enable or disable the variable ISI feature.

Params:	< 0 >	Disable the variable ISI feature.
	< 1 >	Enable the variable ISI feature.
Returns:	<0   1>	

**STRCmb:EXT1 <numeric>**  
**STRC:EXT1?**

Sets or retrieves the External Input 1 loss in dB.

Params:	< numeric >	External 1 loss in dB
Returns:	< numeric >	

**STRCmb:EXT2 <numeric>**  
**STRC:EXT2?**

Sets or retrieves the External Input 2 loss in dB.

Params:	< numeric >	External 2 loss in dB
Returns:	< numeric >	

**STRCmb:ISSEcpres?**

Determine if the secondary ISI module is present. Query only.

Params:	< 0 >	Secondary ISI module is not present.
	< 1 >	Secondary ISI module is present.
Returns:	<0   1>	

**STRCmb:LINKprisec <0 | 1>**  
**STRC:LINKprisec?**

Sets or retrieves the primary and secondary link state.

Params:	< 0 >	Clear the primary and secondary link state.
	< 1 >	Set the primary and secondary link state.
Returns:	<0   1>	

**STRCmb:OPEN <"string">**  
**STRCmb:OPEN?**

Selects or Returns the device name of the current BSAITS125.

You can query an instrument using STRCmb:attachDevNAMES and pass on the device name to open. An empty string can be passed to select the first (usually only) BSAITS125 found.

Params:	< "string" >	Serial number, device name, or ""
Returns:	< "string" >	Current device name, or "NONE"

**STRCmb:OUTInf < SINGLE | DIFFERENTIAL >**  
**STRCmb:OUTInf?**

Set or retrieve the Output Interface.

Params:	< SINGLE > < DIFFERENTIAL >
Returns:	< SINGLE   DIFFERENTIAL >

**STRCmb:PRIIsi < PASSTHROUGH | -9.7dB | -13dB | -17.7dB | EXT1 | EXT2 >**  
**STRCmb:PRIIsi?**

Set or retrieve the Primary ISI setting.

Params:	< PASSTHROUGH > < -9.7dB > < -13dB > < -17.7dB > < EXT1 > < EXT2 >
Returns:	< PASSTHROUGH   -9.7dB   -13dB   -17.7dB   EXT1   EXT2 >

**STRCmb:REFFreq <numeric>**  
**STRC:REFFreq?**

Sets or retrieves the reference frequency in MHz.

Params:	< numeric >	Reference frequency in MHz
Returns:	< numeric >	

**STRCmb:SECIsi < PASSTHROUGH | -2.4dB | -3.8dB | -5.1dB | -6.5dB >  
STRCmb:SECIsi?**

Set or retrieve the Secondary ISI setting.

Params: < PASSTHROUGH >  
< -2.4dB >  
< -3.8dB >  
< -5.1dB >  
< -6.5dB >

---

Returns: < PASSTHROUGH | -2.4dB | -3.8dB | -5.1dB | -6.5dB >

---

**STRCmb:SerialNum?**

Retrieves the serial number string of the currently selected BSAITS125. Query only.

Returns: < "string" > Serial number string of the current device

---

**STRCmb:TotalDevCount?**

Retrieves the total number of unique BSAITS125s (unique serial numbers) found. Query only.

---

**NOTE.** *This number may include devices that are no longer connected to the host computer, or are powered off.*

---

**STRCmb:USEDpp <0 | 1>  
STRC:USEDpp?**

Set or determine if a DPP is used as an ISI source.

Params: < 0 > DPP is not used as an ISI source.  
< 1 > PP is used as an ISI source.

---

Returns: <0 | 1>

---

**STRCmb:VARIsi <numeric>  
STRC:VARIsi?**

Sets or retrieves the variable ISI in dB.

Params: < numeric > Variable ISI in dB

---

Returns: < numeric >

---

---

# Clock Recovery Option

Complete command listing (See page 7, *Command listing*.)

Software is available to control the BERTScope Clock Recovery instrument through the BERTScope Analyzer interface. The following commands apply only to a BERTScope Analyzer with the Clock Recovery Option capability installed.

Clock Recovery command summary (See page 81, *Clock recovery command summary*.)

Clock Recovery Service (See page 84, *CRService : Clock Recovery Service*.)

Clock Recovery Control (See page 85, *CRControl : Clock Recovery Control*.)

CR Loop Response (See page 94, *Clock Recovery Loop Response*.)

SSC Waveform (See page 96, *Clock Recovery SSC Waveform*.)

CR Jitter Spectrum – CRJ (See page 99, *Clock Recovery Jitter Spectrum*.)

CR Miscellaneous (See page 103, *Clock Recovery Miscellaneous*.)

Sample Program for Clock Recovery Remote Control (See page 106, *Clock Recovery: Sample Program*.)

The characteristics that define a standard are:

- Standard Name (up to 11 characters; may contain spaces)
- Nominal Frequency (150 MHz to 12.5 GHz)
- Loop Bandwidth (0.02 MHz to 12 MHz)
- Peaking (0.0 dB to 6.0 dB)
- Lock Range (10 MHz to 500 MHz)
- Nominal Edge Density (10.0% to 100%)

The following industry standards are pre-programmed as defined at this time. Note that the definitions of the standards are subject to change by the establishing committees.

All parameters can be manually adjusted, and custom configurations can be created. This provision also makes it possible to delete a predefined standard — if this happens, you will have to manually re-create its settings.

Standard	Nominal Frequency	Loop Bandwidth	Peaking	Lock Range	Nominal Edge Density
	Hertz	Hertz	dB	Hertz	Percent
FBDIMM1 3.2	3.20000E+09	1.10000E+07	1	5.00E+07	50
FBDIMM1 4.0	4.00000E+09	1.10000E+07	1	5.00E+07	50
FBDIMM1 4.8	4.80000E+09	1.10000E+07	1	5.00E+07	50
FBDIMM2 4.8	4.80000E+09	1.10000E+07	1	5.00E+07	50
FBDIMM2 6.4	6.40000E+09	1.10000E+07	1	5.00E+07	50
FBDIMM2 8.0	8.00000E+09	1.10000E+07	1	5.00E+07	50
FBDIMM2 9.6	9.60000E+09	1.10000E+07	1	5.00E+07	50
FC1X	1.06250E+09	6.38000E+05	0.1	5.00E+07	50
FC2X	2.12500E+09	1.27500E+06	0.1	5.00E+07	50
FC4X	4.25000E+09	2.55000E+06	0.1	5.00E+07	50
FC8X	8.50000E+09	5.10000E+06	0.1	5.00E+07	50
GbE1	1.25000E+09	6.37000E+05	0.1	5.00E+07	50
GbE10	1.03125E+10	4.00000E+06	0.1	5.00E+07	50
OC12	6.22000E+08	2.50000E+05	0.1	5.00E+07	50
OC48	2.48800E+09	1.00000E+06	0.1	5.00E+07	50
OC192	9.95000E+09	4.00000E+06	0.1	5.00E+07	50
OIF CEI 6G+	6.00000E+09	4.00000E+06	0.1	5.00E+07	50
OIF CEI 11G+	1.10000E+10	8.00000E+06	0.1	5.00E+07	50
PCIe-1(2.5)	2.50000E+09	2.50000E+06	1.5	5.00E+07	50
PCIe-1(10.0)	2.50000E+09	10.0000E+06	2.2	5.00E+07	50
PCIe-2(6.0)	5.00000E+09	6.00000E+06	0.5	5.00E+07	50
PCIe-2(10.0)	5.00000E+09	10.0000E+06	2.2	5.00E+07	50
PCIe-3(2.5)	8.00000E+09	2.50000E+06	1.5	5.00E+07	50
PCIe-3(4.5)	8.00000E+09	4.50000E+06	0.5	5.00E+07	50
SATA1(0.9)	1.50000E+09	9.00000E+05	2.09	5.00E+07	50
SATA1(1.98)	1.50000E+09	1.98000E+06	0	5.00E+07	100
SATA1(6.0)	1.50000E+09	6.00000E+06	2.09	5.00E+07	100
SATA2(1.8)	3.00000E+09	1.80000E+06	2.09	5.00E+07	50
SATA2(1.98)	3.00000E+09	1.98000E+06	0	5.00E+07	100



SATA2(6.0)	3.00000E+09	6.00000E+06	2.09	5.00E+07	100
XAUI	3.12500E+09	1.87500E+06	0.1	5.00E+07	50
XFP/XFI RX	9.95000E+09	8.00000E+06	0.1	5.00E+07	50
XFP/XFI TX	9.95000E+09	4.00000E+06	0.1	5.00E+07	50

## Clock recovery command summary

Complete command listing (See page 7, *Command listing*.)

CRService Commands RC CRService Clock Recovery Service(See page 84, *CRService : Clock Recovery Service*.) (usablebefore OPEN)

### Dynamic Read-Only Properties

TOTALDEVCOUNT or TDCT  
ATTACHDEVCOUNT or ADCT  
ATTACHDEVNAMES or NAMES

### Read-Write Settings

OPEN

### Write-Only Operations

CLOSE

CRControl Commands (requires prior OPEN)

### Static Read-Only Identification and Capabilities

SERIALNUM or SN  
DEVICENAME or DN  
SWREV or SW  
HWREV or HW  
FPGAREV or FPGA  
EXPREV or EXP  
STANDARDLIST or STANDARDS  
SUBCLOCKDIVLIST or SUBDIVS

**Dynamic Read-Only Properties**

CONNECTED or CONN  
BUSY  
LOCKSTATE or LOCKST  
LOCKCOUNT or LOCKCT  
DATARATE  
DUTYCYCLEDISTORTION or DCD  
EDGEDENSITY or EDGED  
MAXLBW or MAXCALLOOPWIDTH  
MINLBW or MINCALLOOPWIDTH  
PHASEERRP2P or PHERR  
PHASEERRRMS or PHERMS

**Read-Write Settings**

STANDARD  
NOMFREQUENCY or NOMFREQ  
LOCKRANGE or RANGE  
LOOPBANDWIDTH or LOOPBW  
PEAKING  
LOCKMODE  
EDGEDENSITYMODE or EDGEDMODE  
NOMEDGEDENSITY or NOMEDGED  
PHASEERRLIMIT or PHERRLMT  
CLOCKOUTPUT or CLKOUT  
CLOCKAMPLITUDE or CLKAMPL  
SUBCLOCKOUTPUT or SUBOUT  
SUBCLOCKAMPLITUDE or SUBAMPL  
SUBCLOCKDIV or SUBDIV  
AUTOSCONFIGDEVICE or AUTOSDEV

**Write-Only Operations**

RELOCK or LOCK  
RESETLOCKCOUNT or RESETLOCKCT  
SCONFIGDISK or SDISK  
RCONFIGDISK or RDISK  
RCONFIGURATION or RCONFIG  
SCONFIGDEVICE or SDEV  
SCONFIGURATION or SCONFIG  
RCONFIGDEVICE or RDEV  
CREATESTANDARD  
DELETESTANDARD

CRLoopResponse Commands Clock Recovery Loop Response (See page 94, *Clock Recovery Loop Response*.) (requires prior OPEN)

**Read-Write Settings**      AVGSETTING or AVGS  
 CHARTTYPE  
 DERIVATIVEDELTA  
 MBARSEL  
 PLOTTYPE  
 SHIFT

**Dynamic Read-Only Measurements**      BANDWIDTH  
 MTHREEDBCAL  
 PEAKING  
 PEAKFREQ  
 MTHREEDBCAL

crSSCWaveform Commands Clock Recovery SSC Waveform (See page 96, *Clock Recovery SSC Waveform*.) (requires prior OPEN)

**Dynamic Read-Only Properties and Measurements**      AVGCOUNT or AVGC  
 NUMSCANS or SCANS  
 NUMHISTOGRAMS or HISTOS  
 HISTOGRAMPOINTS or HISTOPTS  
 DEVMIN\_PPM or DEVMIN  
 DEVMIN\_HZ  
 DEVMAX\_PPM or DEVMAX  
 DEVMAX\_HZ  
 MAXDER  
 MODFREQ  
 NOMFREQ

## CRService : Clock Recovery Service

Complete command listing (See page 7, *Command listing*.)

### CRService:AttachDevCount?

Retrieves the number of unique BERTScope CRs (unique serial numbers) that are currently communicating with the Clock Recovery Service, whether or not the remote computer has OPENed a session to any of them. Query only.

---

Returns:	<numeric>	Number of CRs found
----------	-----------	---------------------

---

### CRService:attachdevNAMES?

Retrieves a comma-separated list of all BERTScope CRs currently communicating with the Clock Recovery Service, whether or not the remote computer has OPENed a session to any of them. Query only.

Each BERTScope CR is identified by its user-assignable device name.

---

Returns:	<"string">	Comma-separated list of BERTScope CRs found, or "" if none
----------	------------	--

---

### CRService:CLOSE

Close the communication connection to the current BERTScope CR. Action only.

Note that closing the current BERTScope CR requires that a new current BERTScope CR be selected (or opened) prior to issuing any more device-specific commands.

### CRService:OPEN <"string">

#### CRService:OPEN?

Selects or Returns the device name of the current BERTScope CR.

The user-assigned device name or serial number string may be passed to select a particular BERTScope CR. An empty string may be passed to select the first (usually only) BERTScope CR found. A BERTScope CR can be opened by multiple clients.

---

Params:	<"string">	Serial number, device name, or "" if none
Returns:	<"string">	Current device name, or "NONE"

---

**CRService:TotalDevCount?**

Retrieves the total number of unique BERTScope CRs (unique serial numbers) that the Clock Recovery Service has encountered since last restarted. Query only.

**NOTE.** *This number may include devices that are no longer connected to the host computer, or are powered off.*

Returns:	<numeric>	Number of CRs encountered
----------	-----------	---------------------------

## CRControl : Clock Recovery Control

Complete command listing (See page 7, *Command listing*.)

**CRControl:AUTOSconfigDEvice <bool>****CRControl:AUTOSconfigDEvice?**

Sets or retrieves the “Setup Auto-Save” flag state.

This flag determines whether or not the BERTScope CR restores its shutdown state at power-on.

Params:	<bool>	0 or 1
Returns:	< 0 >	Off – Shutdown state is not restored at power-on
	< 1 >	On – Shutdown state is restored at power-on

**CRControl:BUSY?**

Retrieves the busy status of the currently selected BERTScope CR. Query only.

Returns:	< 0 >	Device is for available for communication
	< 1 >	Device is temporarily unavailable for communication

**CRControl:CLockOUTput <bool>****CRControl:CLockOUTput?**

Sets or retrieves whether the clock output is enabled.

Params:	<bool>	0 or 1
Returns:	< 0 >	Clock output disabled
	< 1 >	Clock output enabled

**CRControl:CLOCKAMPLitude <numeric>**  
**CRControl:CLOCKAMPLitude?**

Sets or retrieves the clock output amplitude in mV.

Params: <numeric> Range [250 mV to 900 mV]

Returns: <numeric> Current clock output amplitude in mV

---

**CRControl:CONNECTed?**

Retrieves the connection status of the currently selected BERTScope CR. Query only.

Returns: < 0 > Device is opened, but not found by the host computer

< 1 > Device is opened and available for communication

---

**CRControl:CREATESTANDARD <"string">**

Saves a new Clock Recovery Standard to the non-volatile memory of the BERTScope CR, using the current settings for NOMFREQUENCY, LOCKRANGE, LOOPBANDWIDTH, PEAKING, and NOMEDGEDENSITY. Action only.

---

**NOTE.** *The Standard name parameter is case-sensitive, and may contain space characters.*

---

Params: <"string"> Name of the Standard to select

---

**CRControl:DATARATE?**

Retrieves the measured data rate in Hertz. Query only.

Returns: <numeric> Measured data rate in Hz

---

**CRControl:DELETESTANDARD <"string">**

Removes a Clock Recovery Standard from the BERTScope CR's non-volatile memory. Action only.

---

**NOTE.** *The Standard name parameter is case-sensitive, and may contain space characters.*

---



---

**NOTE.** *The factory-supplied Standards are treated no differently than user-created Standards, and can be deleted. If accidentally deleted, a Standard can be manually re-created by pre-setting the desired values for NOMFREQUENCY, LOCKRANGE, LOOPBANDWIDTH, PEAKING, and NOMEDGEDENSITY, then sending CRC:CREATESTANDARD with the desired Standard name.*

---

Params:      <"string">                      Name of the Standard to delete

---

**CRControl:DeviceName <"string">****CRControl:DeviceName?**

Changes or retrieves the device name of the currently selected BERTScope CR. A device name can be from 1 to 11 characters in length. Note that the BERTScope CR device name is saved in non-volatile memory.

Params:      <"string">                      New device name

Returns:     <"string">                      Device name of current device

---

**CRControl:DutyCycleDistortion?**

Retrieves the Duty Cycle Distortion measurement in % Unit Interval. Query only.

Note: The DCD measurement is only valid for the period of acquisition, so a new "CRC:MDCD" should be called prior to each "CRC:DCD?" query.

Returns:     <numeric>                      DCD measurement in %UI

---

**CRControl:EDGEDensity?**

Retrieves the measured edge density in percent. Query only.

Returns:     <numeric>                      Measured edge density in %

---

**CRControl:EDGEDensityMODE <enum>**  
**CRControl:EDGE DENSITYMODE?**

Sets or retrieves the edge density mode.

Params:	<NOMINAL>	Use NOMEDGE DENSITY setting
	<ON_LOCK>	Use edge density found when first locked
Returns:	< NOMINAL   ON_LOCK >	

**CRControl:EQUALization <numeric>**  
**CRControl: EQUALization?**

Sets or retrieves the equalization value. BERTScope Si 25000C only.

Params:	<numeric>	Range [0 to 10]
Returns:	<numeric>	Current setting

**CRControl:EXPrev?**

Retrieves the expansion board revision string of the currently selected BERTScope CR. Query only.

Note that if no expansion board is installed in the BERTScope CR, then “xxxx” is returned.

Returns:	<"string">	Expansion board revision string.
----------	------------	----------------------------------

**CRControl:FPGArev?**

Retrieves the FPGA revision string of the currently selected BERTScope CR. Query only.

Returns:	<"string">	FPGA revision string
----------	------------	----------------------

**CRControl:HWrev?**

Retrieves the hardware revision string of the currently selected BERTScope CR. Query only.

Returns:	<"string">	Hardware revision string
----------	------------	--------------------------

**CRControl:LOCKCount?**

Retrieves the number of times the device has relocked since last reset. Query only.

Returns:	<numeric>	Number of times the device has relocked
----------	-----------	---



**CRControl:LOCKMODE <enum>**  
**CRControl:LOCKMODE?**

Sets or retrieves the lock mode of the currently selected BERTScope CR.

Params:	<MANUAL>	Lock mode is manual
	<AUTOMATIC>	Lock mode is automatic
	<NARROW>	Lock mode is narrow
Returns:	< MANUAL   AUTOMATIC   NARROW >	

**CRControl:lockRANGE <numeric>**  
**CRControl:lockRANGE?**

Sets or retrieves the +/- lock range in Hertz, about the nominal frequency.

Params:	<numeric>	Range [10E6. Hz to 500E6. Hz]
Returns:	<numeric>	Current lock range setting in Hz

**CRControl:LOCKSTate?**

Retrieves the lock state of the currently selected BERTScope CR. Query only.

Returns:	< UNLOCKED >	Unlocked
	< ACQUIRINGLOCK >	Acquiring lock
	< LOCKED >	Locked
	< HIGHJITTER >	High jitter locked

**CRControl:LOOPBandWidth <numeric>**  
**CRControl:LOOPBandWidth?**

Sets or retrieves the loop bandwidth in Hz.

Params:	<numeric>	Range is 100E3 Hz to 12E6 Hz
Returns:	<numeric>	Current setting in Hz

**CRControl:MAXcalLoopBWidth?**

Retrieves the maximum calibrated loop bandwidth in Hz. Query only.

Returns:	<numeric>	Maximum calibrated Loop Bandwidth in Hz
----------	-----------	---

**CRControl:MeasureDCD**

Initiate the Duty Cycle Distortion measurement. Action only.

Notes:

- The measurement takes approximately 1 second.
- The CRJ will lose lock during the measurement.
- An application should poll until the CRC:BUSY? query returns 0, before attempting to read the DCD value.
- The DCD measurement is only valid for the period of acquisition, so a new CRC:MDCD should be called prior to each CRC:DCD? query.

**CRControl:MINcalLoopBWidth?**

Retrieves the minimum calibrated loop bandwidth in Hz. Query only.

Returns:     <numeric>                   Minimum calibrated Loop Bandwidth in Hz

---

**CRControl:NOMEDGEDensity <numeric>**

**CRControl:NOMEDGEDensity?**

Sets or retrieves the nominal edge density in percent.

Params:     <numeric>                   Range [10% to 100%]

---

Returns:     <numeric>                   Current edge density setting in %

---

**CRControl:NOMFREQUENCY <numeric>**

**CRControl:NOMFREQUENCY?**

Sets or retrieves the nominal frequency in Hz.

Params:     <numeric>                   Range is 150E6 Hz to 12.5E9 Hz

---

Returns:     <numeric>                   Current setting in Hz

---

**CRControl:PEAKING <numeric>**

**CRControl:PEAKING?**

Sets or retrieves the peaking in dB.

Params:     <numeric>                   Range is to 0 dB to 6 dB

---

Returns:     <numeric>                   Current setting in dB

---

**CRControl:PHaseERRLiMiT <numeric>**

**CRControl:PHaseERRLiMiT?**

Sets or retrieves the the phase error limit in percent unit interval.

Params:     <numeric>                   Range is 10 %UI to 90 %UI

---

Returns:     <numeric>                   Current setting in %UI

---

**CRControl:PHASEERRP2P?**

Retrieves the phase error peak-to-peak measurement in percent unit interval. Query only.

Returns: <numeric>

---

**CRControl:PHASEERRRMS?**

Retrieves the phase error RMS measurement in percent unit intervals. Query only.

Returns: <numeric>

---

**CRControl:RconfigDEvice <enum>**

Restores BERTScope CR settings from a non-volatile Setup location in the BERTScope CR. Action only.

Params:	<POWER_ON>	Configuration used at power-on
	<SETUP_1>	
	<SETUP_2>	
	<SETUP_3>	
	<SETUP_4>	
	<FACTORY>	Restore to the "factory default" configuration

---

**CRControl:RconfigDISK <"string">**

Restores BERTScope CR settings from a file. The complete path must be specified in the parameter string. The path must be accessible to the host computer or BERTScope CR running the remote control software. Use the file extension ".cru" and enclose the path name in quotes. Action only.

Example: CRC:RDISK "D:\Bitalyzer\Configurations\mycfg.cru"

Params:	<"string.cru">	Path name of the configuration file
---------	----------------	-------------------------------------

---

**CRControl:reLOCK**

Initiates a search for the data rate. Action only.

---

**NOTE.** *This command may take some time to complete. An application program would typically call this, then repeatedly query LOCKSTATE until lock is achieved.*

---

**CRControl:RESETLOCKCount**

Resets the lock count to zero. Action only.

**CRControl:SconfigDEVICE <enum>**

Saves the current BERTScope CR settings to a non-volatile Setup location in the BERTScope CR. Action only.

Params:	<POWER_ON>	Configuration used at power-on
	<SETUP_1>	
	<SETUP_2>	
	<SETUP_3>	
	<SETUP_4>	

**CRControl:SconfigDISK <"string">**

Saves the current BERTScope CR settings to a file. The complete path must be specified in the parameter string. The path must be accessible to the host computer or BERTScope running the remote control software. Use the file extension “.cru” and enclose the path name in quotes. Action only.

Example: CRC:SDISK “D:\BitAnalyzer\Configurations\mycfg.cru”

Params:	<"string.cru">	Path name of the configuration file
---------	----------------	-------------------------------------

**CRControl:SerialNum?**

Retrieves the serial number string of the currently selected BERTScope CR. Query only.

Returns:	<"string">	Serial number string of current device.
----------	------------	---

**CRControl:STANDARD <"string">**

**CRControl:STANDARD?**

Changes or retrieves the name of the Clock Recovery Standard. Selecting a new Standard will change the BERTScope CR’s settings for NOMFREQUENCY, LOCKRANGE, LOOPBANDWIDTH, PEAKING, and NOMEDGEDENSITY.

**NOTE.** The Standard name parameter is case-sensitive, and may contain space characters.

Params:	<"string">	Name of the Standard to select
Returns:	<"string">	Currently selected Standard, or “None”

**CRControl:STANDARDliSt?**

Retrieves a comma-separated list of the available Standards. Query only.

---

**NOTE.** *The Standard names are case-sensitive, and may contain space characters.*

---

Returns: <"string"> Standards are comma-separated in the string.

---

**CRControl:SWrev?**

Retrieves the firmware revision string of the currently selected BERTScope CR. Query only.

Returns: <"string"> Firmware revision string

---

**CRControl:SUBclockAMPLitude <numeric>****CRControl:SUBclockAMPLitude?**

Sets or retrieves the sub-rate clock amplitude in mV.

Params: <numeric> Range is 250 mV to 900 mV

---

Returns: <numeric> Current sub-rate clock output amplitude in mV

---

**CRControl:SUBclockDIV <numeric>****CRControl:SUBclockDIV?**

Sets or retrieves the sub-rate clock divisor.

Params: <numeric> The divisor used by the sub-rate clock. (Full rate = 1)  
 1, 2, 4, 5, 6, 7, 8, 9, 10,12, 14, 16, 18, 20, 24, 25, 28, 30,  
 32, 35, 36, 40, 42, 45, 48, 49, 50, 54, 56, 60, 63, 64, 70,  
 72, 80, 81, 90, 100, 108, 112, 120, 126, 128, 140, 144,  
 160, 162, 168, 180, 192, 196, 200, 216, 224, 240, 252,  
 256, 280, 288, 320, 324, 336, 360, 384, 392, 432, 448,  
 504, 512, 576, 648

---

Returns: <numeric> Current sub-rate clock divisor setting

---

**CRControl:SUBclockDIVliSt?**

Retrieves a comma-separated list of available sub-rate clock divisors. Query only.

Returns: <"string"> Divisors are comma-separated in the string

---

**CRControl:SUBclockOUTPUT<bool>**  
**CRControl:SUBclockOUTPUT?**

Sets or retrieves sub-rate clock output enable state.

Params:	<bool>	0 or 1
Returns:	< 0 >	Disabled
	< 1 >	Enabled

## Clock Recovery Loop Response

Complete command listing (See page 7, *Command listing*.)

**CRLoopresponse:BANDWIDTH?**

Retrieves the measured Loop Response Bandwidth (Amplitude chart type only).  
 Query only.

Returns:	<numeric>	Bandwidth in Hertz
----------	-----------	--------------------

**CRLoopresponse:CHARTTYPE <enum>**  
**CRLoopresponse:CHARTTYPE?**

Sets or retrieves the Loop Response Chart Type.

Params:	<AMPLITUDE>	Amplitude chart type
	<PHASE>	Phase chart type
Returns:	< AMPLITUDE   PHASE >	

**CRLoopresponse:MBARSEL <enum>**  
**CRLoopresponse:MBARSEL?**

Set or retrieve the CR Loop Response Sidebar selection.

Params:	<OFF>	No Sidebar
	<MEASUREMENTS>	Show Measurement Sidebar
	<GENERATOR>	Show Generator Sidebar
	<STRESS>	Show Stress Sidebar (if Stress Option is enabled)
	<CRU>	Show CRU Sidebar
	<DETECTOR>	Show Detector Sidebar
	<DPP>	Show DPP Sidebar
Returns:	< OFF   MEASUREMENTS   GENERATOR   STRESS   CRU   DETECTOR   DPP >	

**CRLoopresponse:MTHREEDBCAL?**

Retrieves the measured Jitter transfer function frequency at which it crosses  $-3$  dB (Amplitude chart type only). Query only.

Returns: <numeric> Frequency in Hertz

---

**CRLoopresponse:PEAKFREQ?**

Retrieves the measured Loop Response Peak Frequency (Amplitude chart type only). Query only.

Returns: <numeric> Peak Frequency in Hz

---

**CRLoopresponse:PEAKING?**

Retrieves the measured Loop Response Peaking (Amplitude chart type only). Query only

Returns: <numeric> Peaking in dB

---

## Clock Recovery SSC Waveform

Complete command listing (See page 7, *Command listing*.)

### **crSSCWaveform:AVGCount?**

Retrieves the number of averages completed. Query only

Returns:	<numeric>	Averages completed
----------	-----------	--------------------

### **crSSCWaveform:AVGSetting <numeric>**

### **crSSCWaveform:AVGSetting?**

Sets or retrieves the number of running averages to perform.

Params:	<numeric>	Range [1 to 20]
---------	-----------	-----------------

Returns:	<numeric>	Current average setting
----------	-----------	-------------------------

### **crSSCWaveform:AXISYUNITS <enum>**

### **crSSCWaveform:AXISYUNITS?**

Sets or retrieves the Y axis units for the SSC plot.

Params:	<PPM>	Y axis units are ppm
---------	-------	----------------------

	<HZ>	Y axis units are Hz
--	------	---------------------

Returns:	< PPM   HZ >
----------	--------------

### **crSSCWaveform:DERIVATIVEDELTA <numeric>**

### **crSSCWaveform:DERIVATIVEDELTA?**

Sets or retrieves the time step size used to plot dF/dt.

Params:	<numeric>	Range [0.1 $\mu$ s to 10 $\mu$ s]
---------	-----------	-----------------------------------

Returns:	<numeric>	Current setting
----------	-----------	-----------------

### **crSSCWaveform:DEVMAX\_HZ?**

Retrieves the measured average of maximum frequency relative to the CR Nominal Frequency. Query only.

Returns:	<numeric>	Frequency deviation in Hz
----------	-----------	---------------------------

### **crSSCWaveform:DEVMAX\_ppm?**

Retrieves the measured average of maximum frequency relative to the CR Nominal Frequency. Query only.

Returns:	<numeric>	Frequency deviation in parts per million
----------	-----------	--



**crSSCWaveform:DEVMIN\_HZ?**

The measured average of minimum frequency relative to the CR Nominal Frequency. Query only.

Returns: <numeric> Frequency deviation in Hz

---

**crSSCWaveform:DEVMIN\_ppm?**

Retrieves the measured average of minimum frequency relative to the CR Nominal Frequency. Query only.

Returns: <numeric> Frequency deviation in parts per million (PPM)

---

**crSSCWaveform:HISTOgramPoinTS?**

Retrieves the number of points per histogram used when computing the average. Query only.

Returns: <numeric> Number of points per histogram

---

**crSSCWaveform:MAXDER?**

Retrieves the maximum absolute value of the dF/dt plot. Query only.

Returns: <numeric> Current maximum value

---

**crSSCWaveform:MODFREQ?**

Retrieves the measured modulation frequency of the SSC waveform. Query only.

Returns: <numeric> Frequency in Hz

---

**crSSCWaveform:NOMFREQ?**

Retrieves the measured nominal clock frequency in Hertz. Query only.

Returns: <numeric> Frequency in Hz

---

**crSSCWaveform:numHISTOgramS?**

Retrieves the number of histograms to use when computing the average. Query only.

Returns: <numeric> Number of histograms

---

**crSSCWaveform:numSCANS?**

Retrieves the total number of scans completed. Query only.

Returns: <numeric> Scans completed

---

**crSSCWaveform:PLOTTYPE <enum>**  
**crSSCWaveform:PLOTTYPE?**

Sets or retrieves the SSC plot type

Params:	<FT>	Plot F(t)
	<DFDT>	Plot dF/dt
Returns:	< FT   DFDT >	

**crSSCWaveform:RCONFIguration <"string">**

Restores the SSC Waveform settings from a file. The complete path must be specified in the parameter string. The path must be accessible to the host computer or BERTScope running the remote control software. Use the file extension “.ssc” and enclose the path name in quotes. Action only.

Example: SSCW:RCONFIG “D:\BitAnalyzer\Configurations\mycfg.ssc”

Params:	<"string">	Path name of the configuration file
---------	------------	-------------------------------------

**crSSCWaveform:SCONFIguration <"string">**

Saves the current SSC Waveform settings to a file. The complete path must be specified in the parameter string. The path must be accessible to the host computer or BERTScope running the remote control software. Use the file extension “.ssc” and enclose the path name in quotes. Action only.

Example: SSCW:SCONFIG “D:\BitAnalyzer\Configurations\mycfg.ssc”

Params:	<"string">	Path name of the configuration file
---------	------------	-------------------------------------

# Clock Recovery Jitter Spectrum

Complete command listing (See page 7, *Command listing*.)

## **crJitterSpectrum:AVGCount?**

Retrieves the number of averages completed. Query only.

Returns:	<numeric>	Averages completed
----------	-----------	--------------------

---

## **crJitterSpectrum:AVGSetting <numeric>**

### **crJitterSpectrum:AVGSetting?**

Sets or retrieves the number of running averages to perform.

Params:	<numeric>	Range [1 to 16]
---------	-----------	-----------------

---

Returns:	<numeric>	Current average setting
----------	-----------	-------------------------

---

## **crJitterSpectrum:CRINITmode <enum>**

### **crJitterSpectrum:CRINITmode?**

Sets or retrieves the way the BERTScope CR is initialized prior to a Jitter Spectrum session.

Params:	<AUTO>	First tries to use NO_SSC, then SSC_CAPABLE settings
---------	--------	--

---

	<NO_SSC>	Default Settings for signals without SSC
--	----------	--

---

	<SSC_CAPABLE>	Settings that will accommodate input signals with SSC
--	---------------	---

---

Returns:	< AUTO   NO_SSC   SSC_CAPABLE >
----------	---------------------------------

---

## **crJitterSpectrum:IJMAXHZ\_1 <double>**

### **crJitterSpectrum:IJMAXHZ\_1?**

Sets or retrieves the upper modulation frequency boundary for Integrated Jitter range 1.

Params:	<double>	Upper modulation frequency boundary in Hz
---------	----------	---

---

Returns:	Current upper modulation frequency boundary in Hz
----------	---

---

**crJitterSpectrum:IJMAXHZ\_2 <double>**  
**crJitterSpectrum:IJMAXHZ\_2?**

Sets or retrieves the upper modulation frequency boundary for Integrated Jitter range 2.

Params:	<double>	Upper modulation frequency boundary in Hz
Returns:	Current upper modulation frequency boundary in Hz	

**crJitterSpectrum:IJMAXHZ\_3 <double>**  
**crJitterSpectrum:IJMAXHZ\_3?**

Sets or retrieves the upper modulation frequency boundary for Integrated Jitter range 3.

Params:	<double>	Upper modulation frequency boundary in Hz
Returns:	Current upper modulation frequency boundary in Hz	

**crJitterSpectrum:IJMINHZ\_1 <double>**  
**crJitterSpectrum:IJMINHZ\_1?**

Sets or retrieves the lower modulation frequency boundary for Integrated Jitter range 1.

Params:	<double>	Lower modulation frequency boundary in Hz
Returns:	Current lower modulation frequency boundary in Hz	

**crJitterSpectrum:IJMINHZ\_2 <double>**  
**crJitterSpectrum:IJMINHZ\_2?**

Sets or retrieves the lower modulation frequency boundary for Integrated Jitter range 2.

Params:	<double>	Lower modulation frequency boundary in Hz
Returns:	Current lower modulation frequency boundary in Hz	

**crJitterSpectrum:IJMINHZ\_3 <double>**  
**crJitterSpectrum:IJMINHZ\_3?**

Sets or retrieves the lower modulation frequency boundary for Integrated Jitter range 3.

Params:	<double>	Lower modulation frequency boundary in Hz
Returns:	Current lower modulation frequency boundary in Hz	

**crJitterSpectrum:IJMODE\_1 <enum>**  
**crJitterSpectrum:IJMODE\_1?**

Sets or retrieves the RMS mode to be used for IJ measurements in IJ range 1.

Params:	<PK-PK>	Peak-to-Peak measurement mode
	<RMS>	Root-Mean-Squared measurement mode
Returns:	< PK-PK   RMS >	

**crJitterSpectrum:IJMODE\_2 <enum>**  
**crJitterSpectrum:IJMODE\_2?**

Sets or retrieves the measurement mode to be used for IJ measurements in IJ range 2.

Params:	<PK-PK>	Peak-to-Peak measurement mode
	<RMS>	Root-Mean-Squared measurement mode
Returns:	< PK-PK   RMS >	

**crJitterSpectrum:IJMODE\_3 <enum>**  
**crJitterSpectrum:IJMODE\_3?**

Sets or retrieves the mode to be used for IJ measurements in IJ range 3.

Params:	<PK-PK>	Peak-to-Peak measurement mode
	<RMS>	Root-Mean-Squared measurement mode
Returns:	< PK-PK   RMS >	

**crJitterSpectrum:IJUNITS <enum>**  
**crJitterSpectrum:IJUNITS?**

Sets or retrieves the units to be used for Jitter Spectrum Amplitude and IJ measurements.

Params:	<PERCENTUI>	Percent of Unit Interval
	<MILLIUI>	1/1000 of Unit Interval
	<UI>	Unit Interval
	<PICOSECS>	Pico-seconds
Returns:	< PERCENTUI   MILLIUI   UI   PICOSECS >	

**crJitterSpectrum:IntegratedJitter\_1?**

Retrieves the integrated jitter measurement for IJ range 1. Query only.

Returns:	<numeric>	Either %UI, milliUI, UI, or ps, depending upon JS:IJUNITS setting; either p-p or rms, depending on JS:IJMODE_1
----------	-----------	--

**crJitterSpectrum:IntegratedJitter\_2?**

Retrieves the integrated jitter measurement for IJ range 2. Query only.

Returns: <numeric>                      Either %UI, milliUI, UI, or ps, depending upon  
JS:IJUNITS setting; either p-p or rms, depending on  
JS:IJMODE\_2

---

**crJitterSpectrum:IntegratedJitter\_3?**

Retrieves the integrated jitter measurement for IJ range 3. Query only.

Returns: <numeric>                      Either %UI, milliUI, UI, or ps, depending upon  
JS:IJUNITS setting; either p-p or rms, depending on  
JS:IJMODE\_3

---

**crJitterSpectrum:numSCANS?**

Retrieves the total number of scans completed. Query only.

Returns: <numeric>                      Scans completed

---

**crJitterSpectrum:RCONFIGuration <"string">**

Restores the Jitter Spectrum settings from a file. The complete path must be specified in the parameter string. The path must be accessible to the host computer or BERTScope running the remote control software. Use the file extension ".cjs" and enclose the path name in quotes. Action only.

Example: JS:RCONFIG "D:\Bitalyzer\Configurations\mycfg.cjs"

Params: <"string.cjs">                      Path name of the configuration file

---

**crJitterSpectrum:SCANRESmode <enum>**

**crJitterSpectrum:SCANRESmode?**

Sets or retrieves the scan resolution for the Jitter Spectrum session.

Params: <AUTO>                              Chooses the NORMAL res if USB 1.x, HIGHRES  
if USB 2.0

---

<NORMAL>                                  Suitable for peak detection but not IJ measurement

---

<HIGHRES>                                  4x the resolution of NORMAL mode

---

Returns: < AUTO | NORMAL | HIGHRES >

---

**crJitterSpectrum:SCONFIGuration <"string">**

Saves the current Jitter Spectrum settings to a file. The complete path must be specified in the parameter string. The path must be accessible to the host computer or BERTScope running the remote control software. Use the file extension ".cjs" and enclose the path name in quotes. Action only.

Example: JS:SCONFIG "D:\BitAlyzer\Configurations\mycfg.cjs"

Params:	<"string.cjs">	Path name of the configuration file
---------	----------------	-------------------------------------

## Clock Recovery Miscellaneous

Complete command listing (See page 7, *Command listing*.)

**\*CLR>**

Clear the system status queue. Action only.

**\*IDN?**

Retrieves a string identifying the Host computer. Query only.

Returns:	<"string">	Host computer identity
----------	------------	------------------------

**GUILockout <bool>****GUILockout?**

Sets or retrieves the "GUI Lock-out" flag state. The user interface of the BERTScope CR software will not respond to input if this flag is set On.

Params:	<bool>	0 or 1
---------	--------	--------

Returns:	< 0 >	GUI is not locked
	< 1 >	GUI is locked

**ISSTatus?**

Retrieves a string identifying the version of the BERTScope CR software running on the Host computer or BERTScope. Query only.

Returns:	<"string">	"Instrument Server Version MM.mm" where: MM is the Major version number mm is the minor version number
----------	------------	---

**PTFile <"string">**

Save a screen capture of the BERTScope CR software to a file. The complete path must be specified in the parameter string. The path must be accessible to the Host computer or BERTScope running the remote control software. Use the file extension ".jpg" and enclose the path name in quotes. Action only.

Example: RCON "C:\mycfg.jpg"

Params:	<"string.jpg">	Path name of the screen capture image file
---------	----------------	--

**RCONfiguration <"string">**

Restores the complete BERTScope CR state from a file. The complete path must be specified in the parameter string. The path must be accessible to the Host computer or BERTScope running the remote control software. Use the file extension ".cfg" and enclose the filename in quotes. Action only.

Example: RCON "D:\BitAnalyzer\Configurations\mycfg.cfg"

Params:	<"string.cfg">	Path name of the configuration file
---------	----------------	-------------------------------------

**RCStatus?**

Retrieves a string identifying the version of the RemoteClient software running on the Host computer or BERTScope. Query only.

Returns:	<"string">	"Remote Client Version MM.mm" where :
		MM is the Major version number
		mm is the minor version number

**RDURATION <long>**

**RDURATION?**

Set or retrieve the run duration of the BERTScope.

Setting to '0' allows the run to go on 'forever.'

Params:	<long>	Range [1to 36E6]. Zero (0) sets an infinite duration.
Returns:	< long >	System run duration in seconds

**RState<bool>**

**RState?**

Sets or retrieves the run state of the BERTScope CR software.

Params:	<bool>	0 or 1
Returns:	< 0 >	System is not running
	< 1 >	System is running



**SCONfiguration <"string">**

Saves the complete state of the BERTScope CR to a file. The complete path must be specified in the parameter string. The path must be accessible to the host computer or BERTScope running the remote control software. Use the file extension ".cfg" and enclose the path name in quotes. Action only.

Example: SCON "D:\BitAnalyzer\Configurations\mycfg.cfg"

Params: <"string.cfg"> Path name of the configuration file

---

**SYSTEM:ERROR?**

Retrieve errors listed in the system status queue. Action only.

This command will not show in the list of trace messages displayed on the remote control client user interface. The status queue is first in, first out. It can contain up to 100 error messages. If the queue overflows, the last error/event in the queue is replaced with Error -350, "Queue overflow". When all errors/events have been read from the queue, further error/event queries will return 0, "NO error".

Returns: <n, string > n, "event/string"

---

**VIEW <enum>****VIEW?**

Sets or retrieves the current view displayed by the BERTScope CR software.

**NOTE.** *The choices below assume the Host computer is not a BERTScope. If the Host computer is a BERTScope, many more views are available.*

---

Params:	<CRControl> or <CRC>	Clock Recovery Control View
	<CRLoopResponse> or <CRLR>	Clock Recovery Loop Response View
	<crSSCWaveform> or <SSCW>	SSC Waveform View
	<crJitterSpectrum> or <JS>	Jitter Spectrum View (CRJ only)
Returns:	< CRControl> or <CRC> < CRLoopResponse > or <CRLR> < crSSCWaveform > or <SSCW> < crJitterSpectrum > or <JS> (CRJ only)	

---

## Clock Recovery: Sample Program

Complete command listing (See page 7, *Command listing*.)

```
// CR_RC.c : Commandline program to execute single command over TCP/IP
#include <windows.h>
#include <winsock.h>
#include <stdio.h>

#define PORTNUM 23 // Port number
#define HOSTNAME "localhost" // Server name string
SOCKET ServerSock = INVALID_SOCKET; // Socket bound to the
server
SOCKADDR_IN destination_sin; // Server socket address
PHOSTENT phostent = NULL; // HOSTENT structure of server
WSADATA WSADATA; // Contains details of the
// Winsocket implementation
static BOOL OpenSocket();
static BOOL CloseSocket();
static BOOL SendCommand( TCHAR * pszCmd );
static BOOL GetReply( TCHAR * pszReply );
int main( int argc, char* argv[] )
{
char szCmd[1024];
char szReply[1024];
memset( szCmd, 0, sizeof(szCmd) );
memset( szReply, 0, sizeof(szReply) );
if ( argc > 1 )
{
int i;
BOOL retVal = FALSE;
```

```
for ( i = 1; i < argc; i++ )
{
    strcat( szCmd, argv[i] );
    if ( i+1 < argc )
        strcat( szCmd, " " );
}
if ( !OpenSocket() )
    return FALSE;
if ( !SendCommand( "*CLS" ) ) // clear error queue
    goto Exit;
if ( !SendCommand( szCmd ) )
    goto Exit;
if ( strchr( szCmd, '?' ) != NULL )
{
    if ( !GetReply( szReply ) )
        goto Exit;
    else
    {
        printf( "%s", szReply );
        retVal = TRUE;
    }
}
else // not query, so check command status
{
    BOOL bErrs = FALSE;
    while(1)
    {
        if ( !SendCommand( "system:err?" ) )
            goto Exit;
        if ( !GetReply( szReply ) )
            goto Exit;
```

```
else
{
if ( strstr( szReply, "NO ERROR" ) == NULL )
{
printf( "%s", szReply );
bErrs = TRUE;
}
else // no more errors in queue
{
retval = !bErrs;
break;
}
}
}
}
Exit:
closeSocket();
return retval;
}
else // help
{
printf( "Expecting 'feature:operation parameter'\n" );
printf( "or 'feature:operation?'\n" );
}
return TRUE;
}
```

```
static BOOL OpenSocket()
{
    // Initialize winsocket.
    if (WSAStartup (MAKEWORD(1,1), &WSAData) != 0)
    {
        printf( "WSAStartup failed.  Error:  %d\n",
            WSAGetLastError() );
        return FALSE;
    }
    // Create a TCP/IP socket that is bound to the server.
    if ( (ServerSock = socket(AF_INET, SOCK_STREAM, 0) )
        == INVALID_SOCKET )
    {
        printf( "Allocating socket failed.  Error:  %d\n",
            WSAGetLastError() );
        return FALSE;
    }
    // Fill out the server socket address information.
    destination_sin.sin_family = AF_INET;
    // Retrieve the host information corresponding to the host
    name.
    if ((phostent = gethostbyname (HOSTNAME)) == NULL)
    {
        printf( "Unable to get the host name.  Error:  %d\n",
            WSAGetLastError() );
        closesocket (ServerSock);
        return FALSE;
    }
    // Assign the socket IP address.
    memcpy( (char FAR *)&(destination_sin.sin_addr),
        phostent->h_addr,
```

```
phostent->h_length);
// Convert to network ordering.
destination_sin.sin_port = htons (PORTNUM);
// Establish a connection to the server socket.
if ( connect( ServerSock,
(PSOCKADDR)&destination_sin,
sizeof(destination_sin)
) == SOCKET_ERROR )
{
printf( "Connecting to the server failed.  Error:  %d\n",
WSAGetLastError() );
closesocket (ServerSock);
return FALSE;
}
return TRUE;
}
static BOOL CloseSocket()
{
shutdown (ServerSock, 0x01); // Disable sending
shutdown (ServerSock, 0x00); // Disable receiving
closesocket (ServerSock);
WSACleanup ();
return TRUE;
}
static BOOL SendCommand( TCHAR * pszCmd )
{
// Send a string to the server.
if ( send( ServerSock, pszCmd, strlen(pszCmd) + 1, 0
) == SOCKET_ERROR )
{
printf( "Sending data to the server failed.  Error:  %d\n",
```

```
WSAGetLastError() );
return FALSE;
}
else
return TRUE;
}
static BOOL GetReply( TCHAR * pszReply )
{
// Receive data from the server socket.
int iReturn = recv( ServerSock, pszReply, 1024, 0 );
if ( iReturn == SOCKET_ERROR )
sprintf( pszReply, "No data is received, recv failed.
Error: %d",
WSAGetLastError() );
return iReturn != SOCKET_ERROR;
}
```



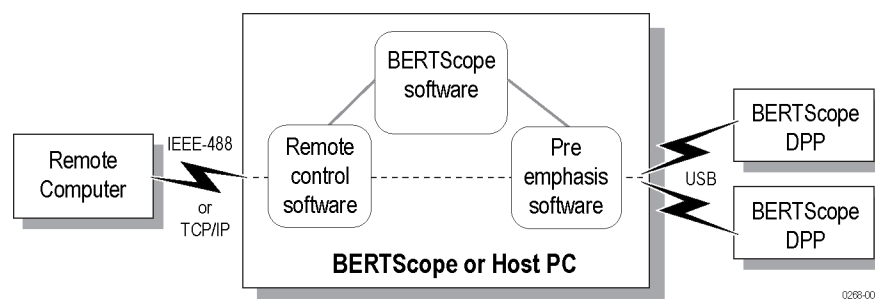


# Digital Pre-Emphasis Processor (DPP) Option

Complete command listing (See page 7, *Command listing*.)

The BERTScope DPP125 can be controlled remotely via either an IEEE-488 or a TCP/IP communications connection. The DPP125 remote control software runs on either a BERTScope Analyzer, or a host PC running Microsoft Windows 2000 or XP operating systems.

The remote control software accepts text-oriented commands from a remote computer via either an IEEE-488 or TCP/IP connection. It then routes the commands to one or more BERTScope DPPs, connected to the BERTScope Analyzer or host PC via USB.



If the local host is a BERTScope Analyzer, then the Analyzer software gets a chance to handle the command first. Any commands the Analyzer software doesn't understand are sent to the DPP software, which then controls the BERTScope DPP125.

When a BERTScope DPP is connected to a BERTScope Analyzer, you can intermix BERTScope Analyzer and BERTScope DPP commands in the same script, program, or interactive session. If the host is just an ordinary PC, then the commands are routed to the Pre-emphasis software directly. In this case, using the BERTScope-specific commands will result in an error.

For most applications, a single BERTScope DPP is connected to a BERTScope Analyzer or host PC. In this case, the BERTScope DPP is automatically detected and connected when the remote control software is started, and automatically disconnected when the remote control software is stopped. There is no need in this case to use the OPEN or CLOSE commands described below.

If more than one BERTScope DPP is connected to the BERTScope Analyzer or host computer, the remote control software will not connect automatically. In this case, the Remote computer must issue a NAMES? query to discover the IDs of the connected BERTScope DPPs, and OPEN the one desired before issuing control commands. If the Remote computer needs to control multiple BERTScope DPPs, it would OPEN, control, then CLOSE one, then OPEN, control, and CLOSE another. The device that is OPEN is referred to as the current device throughout this document.

Remote control is implemented using a text-oriented command protocol described in the following pages of this document. These commands enable you to set and query the system parameters of the BERTScope DPP, and to retrieve measurements made by the instrument.

## DPP command summary

Complete command listing (See page 7, *Command listing*.)

Lifecycle commands (usable before OPEN)

### Dynamic Read-Only Properties

ATTACHDEVCOUNT  
ATTACHDEVNAMES or NAMES  
DBL:INPUTMHZ or DBL:INPUT  
DBL:STATUS or DBL:STAT  
MULT:INPUTMHZ or MULT:INPUT  
MULT:STATUS or MULT:STAT  
TOTALDEVCOUNT

### Read-Write Settings

DBL:MODE or DBL:MD  
DBL:TERMMV or DBL:TERM  
MULT:FREQSEL or MULT:FREQ  
MULT:PLLMODESEL or MULT:PLL  
OPEN

### Operations

CLEAR or CLR  
CLOSE  
DPP:EQ:CALIBRATION or EQ:CAL  
EQ:DEEMPHASIS, or EQ:DEEMPH  
EQ:EQUALIZATIONDB, EQ:EQDB, or EQ:EQ  
IMPORTFREQ or IMPFREQ  
IMPORTIFREQ or IMPIFREQ  
IMPORTTAP or IMPTAP  
SETPREDEEMPH or SETPH  
SCONFIGDISK or SDISK  
RCONFIGDISK or RDISK

---

<b>Static Read-Only Identification and Capabilities (require prior OPEN)</b>	SERIALNUM or SN CAPABILITIES or CAP DEVICENAME or DN EXPREV or EXP FPGAREV or FPGA HWREV or HW INVALIDVALUE or IVALUE SWREV or SW
<b>Dynamic Read-Only Properties</b>	BUSY CONNECTED or CONN
<b>Read-Write Settings</b>	DATARATE DATARATESRC DEEMPHASIS or DEEMPH DISPLAY or DISP FIRCARTOON FIRBITAMPLFMT or BAFMT FIRBITAMPL0 or BA0 FIRBITAMPL1 or BA1 FIRBITAMPL2 or BA2 FIRBITAMPL3 or BA3 FIRTAP0 or TAP0 FIRTAP1 or TAP1 FIRTAP2 or TAP2 FIRTAP3 or TAP3 GAINLOSS, GAIN, or GN LINKOFFSETS or LINKOFFS MAXAMPLITUDE or MAXAMPL NEGOFFSET or NEGOFFS PCIEPRESET, PPRESET, or PPRE PRESHOOT or PRESH POSOFFSET or POSOFFS STAGE or STG

## DPP remote control commands

Complete command listing (See page 7, *Command listing*.)

### **DPP:AttachDevCount?**

Retrieves the number of unique BERTScope DPPs (unique serial numbers) that are currently responding, whether or not the remote computer has OPENed a session to any of them. Query only.

Returns: <numeric>

---

### **DPP:attachdevNAMES?**

Retrieves a comma-separated list of all BERTScope DPPs that are currently communicating with the Clock Recovery Service, whether or not the remote computer has OPENed a session to any of them. Query only.

Each BERTScope DPP is identified by its device name.

Returns:	<"string">	Comma-separated list of BERTScope DPPs found, or "" if none found
----------	------------	---

---

### **DPP:BUSY?**

Retrieves the busy status of the currently selected BERTScope DPP. Query only.

Returns:	< 0 >	Device is for available for communication
	< 1 >	Device is temporarily unavailable for communication

---

### **DPP:CAPabilities?**

DPP125C only. Retrieves an integer representing the bit field definitions. Query only.

Returns:	< integer >	B0 = Hardware present B1 = PIC32 Processor (otherwise legacy processor) B2 = 4th Tap present B3 = Multiplier feature available B4 = Doubler feature available B5 = Equalizer feature available B6 = Serial port feature avialable
----------	-------------	---

---

### **DPP:CLEaR**

Clears the BERTScope DPP data. Action only.

**DPP:CLOSE**

Close the communication connection to the current BERTScope DPP. Action only.

---

**NOTE.** Closing the current BERTScope DPP requires that a new current BERTScope DPP be selected (or opened) prior to issuing any more device-specific commands.

---

**DPP:CONNected?**

Retrieves the connection status of the currently selected BERTScope DPP. Query only.

Returns:	< 0 >	Device is open, but not found by the host computer
	< 1 >	Device is open and available for communication

---

**DPP:DATARATE <numeric>****DPP:DATARATE?**

Sets or retrieves the input data rate, in Hertz.

The DPP needs to know the input data rate to select the proper calibration values and to draw the frequency response X axis correctly.

NOTE: See DPP:DATARATESRC

Params:	<numeric>	Set the current data rate, in Hertz (If DPP.DATARATESRC == KEYPAD)
Returns:	<numeric>	Measured, or previously entered data rate, in Hertz

---

**DPP:DATARATESRC<AUTO | KEYPAD>****DPP:DATARATESRC?**

Sets or retrieves the method to be used to monitor the input data rate.

The DPP needs to know the input data rate to select the proper calibration values, and to draw the frequency response X axis correctly.

NOTE: See DPP: DATARATE

Params:	<AUTO >	Set the current input data rate using the BERTScope Analyzer setting
	<KEYPAD>	Specify input data rate using keypad
Returns:	< AUTO   KEYPAD >	Returns the previously set input data rate.

---

**DPP:DATASRC<GENCLK | SUBRATE>  
DPP:DATASRC?**

Sets or retrieves the setting indicating the generator output connector as the data source.

The DPP needs to know the generator output connector being used so that it can adjust the settings correctly if "Set Gen. Outputs for DPP" is selected.

NOTE: See DPP: DATARATE

Params:	<GENCLK>	Set the current data source to BERTScope Generator.
	<SUBRATE>	Set the current data source to BERTScope Analyzer Sub-Rate.

Returns: < GENCLK> | SUBRATE > Return the previously set data source.

**DPP:DBL:INPUTmhz?**

DPP125C only. Returns the input frequency reading presented to the Doubler circuits in MHz. Same as the input frequency provided to the DPP. Query only.

Returns: < number > Doubler input frequency in MHz

**DPP:DBL:MoDe<evaluation mode>  
DPP:DBL:MoDe?**

DPP125C only. Sets or returns the DPP Doubler mode.

Params:	< AUTO >	Doubler is set to the same DDR value as the BERTScope.
	< ON >	Doubler is forced on.
	< OFF >	Doubler is forced off.

Returns: < AUTO | ON | OFF >

**DPP:DBL:STATus?**

DPP125C only. Returns the status of the doubler. Query only.

Returns:	OK NOTAVAIL OUTOFRANGE NOCLOCK PASSTHROUGH	Status codes
----------	--	--------------

---

**DPP:DBL:TERMmv <number>****DPP:DBL:TERMmv?**

DPP125C only. Sets or retrieves the clock input termination value in mV.

Params:	< number >	Clock input termination in mV (range -2200 mV to 3750 mV)
---------	------------	--

---

Returns:	< number >
----------	------------

---

**DPP:DEEMPHasis <double>****DPP:DEEMPHasis?**

DPP125C only. Sets or retrieves the deemphasis value in dB.

Params:	< double >	Demphasis value in dB (range -12 to 0.00)
---------	------------	---

---

Returns:	< double >
----------	------------

---

**DPP:DeviceName?**

Retrieves the device name of the currently selected BERTScope DPP. Query only.

A device name can be from 1 to 11 characters in length.

Returns:	<"string">	Device name of current device.
----------	------------	--------------------------------

---

**DPP:DISPlay<enum>****DPP:DISPlay?**

Sets or retrieves the DPP display setting.

Params:	< GRAPHICAL >	Graphical display mode
---------	---------------	------------------------

---

	< NUMERIC >	Numeric display mode
--	-------------	----------------------

---

Returns:	< GRAPHICAL   NUMERIC >
----------	-------------------------

---

**DPP:EXPrev?**

Retrieves the expansion board revision string of the currently selected BERTScope DPP. Query only. Note that if no expansion board is installed in the BERTScope DPP, then “XXXX” is returned.

Returns:	<"string">	Expansion board revision string.
----------	------------	----------------------------------

---

**DPP:EQ:CALibration <bool>**

**DPP:EQ:CALibration?**

DPP125C only. Sets or determines if the equalizer is in calibration mode.

Params:	< bool >	On = 1, off = 0
---------	----------	-----------------

---

Returns:	< 1 >	Equalizer is in calibration mode.
----------	-------	-----------------------------------

	< 0 >	Equalizer is not in calibration mode.
--	-------	---------------------------------------

---

**DPP:EQ:DEEMPHasisDB <value>**

**DPP:EQ:DEEMPHasis?**

DPP125C only. Sets or retrieves the equalizer deemphasis value in dB; “value” is enclosed in double quotes.

Params:	< "value">	Deemphasis value in dB. Possible values are: "0.0DB", "1.5DB", "3.5DB", "5.0DB", "6.0DB", "8.0DB", "9.0DB", "12.0DB"
---------	------------	--

---

Returns:	< "value">
----------	------------

---



**DPP:EQ:EQualization <value>**  
**DPP:EQ:EQualization?**

DPP125C only. Sets or retrieves the equalizer equalization value; “value” is enclosed in double quotes and is a letter “A” through “P” or 0. The actual value depends on the equalizer frequency (see the DPP:EQ:FREQsel command/query). Refer to table for the exact equalization value depending on the frequency selection.

Params: <“value”> “A”, “B”, “C”, “D”, “E”, “F”, “G”, “H”, “I”, “J”,  
 “K”, “L”, “M”, “N”, “O”, “P”, 0

---

Returns: <“value”>

---

Equalization value	FREQSEL = 8000	FREQSEL = 5000	FREQSEL = 2500
A	4.9 dB	3.7 dB	2.1 dB
B	7.9 dB	5.8 dB	3.4 dB
C	9.9 dB	7.7 dB	4.8 dB
D	11.0 dB	8.9 dB	5.9 dB
E	14.3 dB	11.2 dB	7.2 dB
F	14.6 dB	11.4 dB	6.1 dB
G	17.0 dB	13.5 dB	8.8 dB
H	18.0 dB	15.0 dB	10.2 dB
I	18.5 dB	12.8 dB	7.5 dB
J	22.0 dB	17.4 dB	11.4 dB
K	24.4 dB	19.7 dB	13.0 dB
L	25.8 dB	21.1 dB	14.2 dB
M	27.4 dB	21.7 dB	13.8 dB
N	29.0 db	23.5 dB	15.6 dB
O	31.4 dB	25.8 dB	17.2 dB
P	32.7 dB	27.3 dB	18.4 dB

**DPP:EQ:FREQsel <value>**  
**DPP:EQ:FREQsel?**

Sets or retrieves the equalizer frequency in MHz.

Params: <value> Equalizer frequency. Values include: 2500,  
 5000, 8000

---

Returns: <value>

---

**DPP:FIRCARTOON<STEPRESP | TAPVALS>**

Sets or retrieves the method to be used to depict the DPP FIR filter settings.

Params:	<STEPRESP>	Shows computed bit amplitudes
	<TAPVALS>	Shows tap values
Returns:	< STEPRESP   TAPVALS >	

**DPP:firBitAmplFMT<PERCENT | DECIBEL>  
DPP:firBitAmplFMT?**

Sets or retrieves the bit amplitude units.

Params:	<PERCENT>	Units are in %
	<DECIBEL>	Units are in dB
Returns:	< PERCENT   DECIBEL >	

**DPP:firBitAmpl0 ... DPP:firBitAmpl#  
DPP:firBitAmpl0? ... DPP:firBitAmpl#?**

Sets or retrieves the normalized amplitude of the bit (bit0 ... bit#), where # = ( DPP:NUMTAPS -1)

---

**NOTE.** Value must be specified, and is retrieved, in the units specified with DPP:firBitAmplFMT

*Bit Amplitudes and Taps values are interdependent and normalized! Retrieve all bit amplitudes and tap values after changing any one of them!*

---

Params:	<numeric>	% (0 to 200%) dB (-75 to 75) NOTE: negative sign denotes bit inversion
Returns:	<numeric>	% (0 to 200%) dB (-75 to 75) NOTE: negative sign denotes bit inversion

**DPP:firTAP0 ... DPP:firTAP3 <numeric>**  
**DPP:firTAP0? ... DPP:firTAP3?**

Sets or retrieves the normalized tap value (tap0 ... tap3).

---

**NOTE.** *Bit Amplitudes and Taps values are interdependent and normalized!  
 Retrieve all bit amplitudes and tap values after changing any one of them!*

---

Params:	<numeric>	-1.0 to 1.0
		NOTE: 0 turns tap off
Returns:	<numeric>	-1.0 to 1.0
		NOTE: 0 means tap is turned off

---

**DPP:FPGArev?**

Retrieves the FPGA revision string of the currently selected BERTScope DPP.  
 Query only.

Returns:	<"string">	FPGA revision string.
----------	------------	-----------------------

---

**DPP:GAINloss <double>**  
**DPP:GAINloss?**

Sets or retrieves the DPP gain or loss value

Params:	< number >	Gain or loss value (range -12 to 12)
Returns:	< number >	

---

**DPP:HWrev?**

Retrieves the hardware revision string of the currently selected BERTScope DPP. Query only.

Returns: <"string"> Hardware revision string.

---

**DPP:IMPorTAP <fullpath>**

Sets the DPP import tap path name. The tap file is an ASCII contains the specific tap values in a comma-separated-vector (CSV) format.

---

**NOTE.** *If the TapCount or DataRate specified in the file do not match the current device settings a warning message displays and the numbers from the file will be ignored.*

---

Action only.

Params: <"string"> Use the absolute path name and enclose the string in double quotes

---

The tap file format is as follows:

- Any characters after a backslash “\” on a line are ignored.
- Each entry is a Key-Comma-Value.

The keys are: Version, TapCount, DataRate, t0, t1, t2, t3. The t3 is optional; if it isn't specified 0.0 is assumed. The Version is currently set to 1, but can change as needed.

- A sample file is shown below:

```
\ This is an example
\ File.

Version, 1
TapCount, 4
DataRateHz, 5e9
t0, 0.0
t1, 1.0
t2, 0.0
t3, 0.0 \ (optional, if not present, 0.0 is assumed)
```

**DPP:IMPortFREQ <fullpath>**

Sets the DPP import frequency path name. Action only.

Requires S2P or S4P Touchstone file format

Params:	<"string">	Use the absolute path name and enclose the string in double quotes
---------	------------	--

---

**DPP:IMPortIFREQ <fullpath>**

Sets the DPP import inverted frequency path name. Action only.

Requires S2P or S4P Touchstone file format

Params:	<"string">	Use the absolute path name and enclose the string in double quotes
---------	------------	--

---

**DPP:InvalidVALUE?**

Retrieves the "magic number" returned if a BERTScope DPP measurement cannot be made. Query only.

Returns:	<numeric>	Invalid value.
----------	-----------	----------------

---

**DPP:LINKOFFSets <bool>****DPP:LINKOFFSets?**

Specifies or retrieves whether the positive and negative data output offsets are linked or not.

Params:	<bool>	<0> or <1>
Returns:	< 0 >	Not Linked
	< 1 >	Linked

---

**DPP:MAXAMPLitude <numeric>****DPP:MAXAMPLitude?**

Specifies or retrieves the maximum data output amplitude, in mV.

Params:	<numeric>	200 to 900 mV
Returns:	<numeric>	Max amplitude, in mV

---

**DPP:MULT:FREQsel<enum>**  
**DPP:MULT:FREQsel?**

DPP125C only. Sets or retrieves the multiplier frequency.

Params:	< 2500>	Frequency multiplier is 2500
	< 5000>	Frequency multiplier is 5000
	< 8000 >	Frequency multiplier is 8000
Returns:	< 2500   5000   8000 >	

**DPP:MULT:INPUTmhz?**

DPP125C only. Returns the input frequency reading to the multiplier circuits in MHz. Query only.

Returns:	< number>	Read back doubler input frequency in MHz.
----------	-----------	---

**DPP:MULT:PLLmodesel<enum>**  
**DPP:MULT:PLLmodesel?**

DPP125C only. Sets or retrieves the multiplier peaking/bandwidth selection.

Params:	< LOW>	Low peaking and bandwidth
	< High>	High peaking and bandwidth
Returns:	< LOW   HIGH >	

**DPP:MULT:STATus?**

DPP125C only. Returns the status of the multiplier. Query only.

Returns:	OK NOTAVAIL UNLOCKED NOCLOCK OUTOFRANGE	Status codes
----------	---	--------------

**DPP:NEGOFFSet <numeric>**  
**DPP:NEGOFFSet?**

Specifies or retrieves the negative data output offset, in mV.

---

**NOTE.** *If DATA+ and DATA- offsets are linked, then this command will change DATA+ as well.*

---

See DPP:LINKOFFSets

Params:	<numeric>	-500 to 500 mV
Returns:	<numeric>	Offset, in mV

**DPP:NUMTAPS?**

Retrieves the number of taps supported by the currently selected BERTScope DPP. Query only.

Returns:	<numeric>	Number of taps
----------	-----------	----------------

**DPP:OPEN <"string">**  
**DPP:OPEN?**

Selects or returns the device name of the current BERTScope DPP.

The user-assigned device name or serial number string may be passed to select a particular BERTScope DPP. An empty string may be passed to select the first (usually only) BERTScope DPP found. A BERTScope DPP can be opened by multiple clients.

Params:	<"string">	Serial number, device name, or ""
Returns:	<"string">	Current device name, or "NONE"

**DPP:PciePREset<preset>**  
**DPP:PciePREset?**

Sets or retrieves the DPP PCIe presets.

Params:	< preset >	P0
		P1
		P2
		P3
		P4
		P5
		P6
		P7
		P8
		P9

---

Returns:	< preset >	May also return UNKNOWN
----------	------------	-------------------------

---

**DPP:POSOFFSet <numeric>**  
**DPP:POSOFFSet?**

Specifies or retrieves the positive data output\ offset, in mV.

---

**NOTE.** *If DATA+ and DATA- offsets are linked, then this command will change DATA- as well.*

---

See DPP:LINKOFFSets

Params:	<numeric>	-500 to 500 mV
Returns:	<numeric>	Offset, in mV

---

**DPP:PRESHoot<double>**  
**DPP:PciePRESHoot?**

Sets or retrieves the DPP Preshoot value in dB.

Params:	< number >	Preshoot value in dB (range 0 to 12)
Returns:	< number >	

---

**DPP:RconfigDISK<"string">**

Restores the BERTScope DPP settings from a file. The complete path must be specified in the parameter string using double quotes. The path must be accessible to the host computer or BERTScope Analyzer running the remote control software. The file extension should be .DPP. Action only.

Example: DPP:RCONFIGDISK  
 "D:\Bitalyzer\Configurations\mycfg.dpp"

Params:	<"string">	Path name of the configuration file
---------	------------	-------------------------------------

---



**DPP:SconfigDISK<"string">**

Saves the current BERTScope DPP settings to a file. The complete path must be specified in the parameter string using double quotes. The path must be accessible to the host computer or BERTScope Analyzer running the remote control software. The file extension should be .DPP. Action only.

Example: DPP:SCONFIGDISK  
"D:\BitAnalyzer\Configurations\mycfg.dpp"

Params:	<"string">	Path name of the configuration file
---------	------------	-------------------------------------

**DPP:SerialNum?**

Retrieves the serial number string of the currently selected BERTScope DPP. Query only.

Returns:	<"string">	Serial number string of current device.
----------	------------	---

**DPP:SETPreDeemph <double>, <double>**

Sets the DPP preshoot and deemphasis values in dB. Action only.

The comma is required between the two <double> values.

Params:	<PreshootDB>	Preshoot value in dB (range 0.0 to 12.0)
	<DeemphasisDB >	Deemphasis value in dB (range -12.0 to 0.0)

**DPP:STaGe <enum>****DPP:STaGe?**

Sets or retrieves the DPP stage settings.

Params:	BASELINE USERTAPS DEEMPH PRESHOOT GAINLOSS TOTAL	DPP stage settings
---------	---	--------------------

Returns:	< BASELINE   USERTAPS   DEEMPH   PRESHOOT   GAINLOSS   TOTAL >
----------	--

**DPP:SWrev?**

Retrieves the firmware revision string of the currently selected BERTScope DPP. Query only.

Returns:	<"string">	Firmware revision string
----------	------------	--------------------------

**DPP:TotalDevCount?**

Retrieves the total number of unique BERTScope DPPs (unique serial numbers) found. Query only.

---

**NOTE.** *This number may include devices that are no longer connected to the host computer, or are powered off.*

---

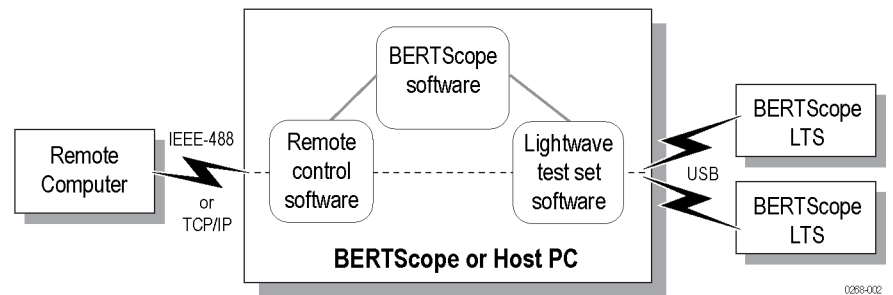
Returns:      <numeric>

---

## Lightwave Test Set (LTS) option

The BERTScope LTS can be controlled remotely through either an IEEE-488 or a TCP/IP communications connection. The BERTScope LTS remote control software runs on either a BERTScope Analyzer, or a host PC running Microsoft Windows 2000 or XP operating systems.

The remote control software accepts text-oriented commands from a remote computer via either an IEEE-488 or TCP/IP connection. It then routes the commands to one or more BERTScope LTSs, connected to the BERTScope Analyzer or host PC via USB.



If the local host is a BERTScope Analyzer, then the BERTScope software gets a chance to handle the command first. Any commands the Analyzer software doesn't understand are sent to the LTS software, which then controls the BERTScope LTS.

When a BERTScope LTS is connected to a BERTScope Analyzer, you can intermix BERTScope Analyzer and BERTScope LTS commands in the same script, program, or interactive session. If the host is just an ordinary PC, then the commands are routed to the LTS software directly. In this case, using the Analyzer-specific commands will result in an error.

For most applications, a single BERTScope LTS is connected to a BERTScope Analyzer or host PC. In this case, the BERTScope LTS is automatically detected and connected when the remote control software is started, and automatically disconnected when the remote control software is stopped. There is no need in this case to use the OPEN or CLOSE commands described below.

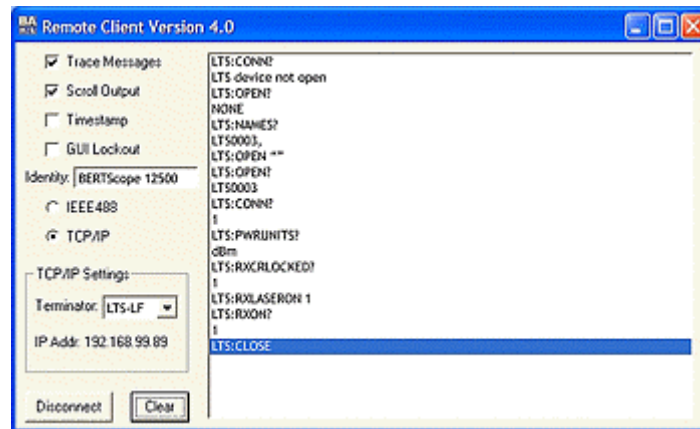
If more than one BERTScope LTS is connected to the BERTScope Analyzer or host computer, the remote control software will not connect automatically. In this case, the Remote computer must issue a NAMES? query to discover the IDs of the connected BERTScope LTSs, and OPEN the one desired before issuing control commands. If the Remote computer needs to control multiple BERTScope LTSs, it would OPEN, control, then CLOSE one, then OPEN, control, and CLOSE another. The device that is OPEN is referred to as the current device throughout this document.

Remote control is implemented using a text-oriented command protocol described in the following pages of this document. These commands enable you to set and query the system parameters of the BERTScope LTS, and to retrieve measurements made by the instrument.

## LTS remote control setup

Before beginning a remote control session, it is necessary to start the Remote Client application, `BitAlyzerRemoteClient.exe`, on the host computer.

If the host is a BERTScope, select the Remote button from the System View's Tools page. If the host computer is a normal PC, then just execute the `BitAlyzerRemoteClient.exe` file.



The Remote Client application allows you to select either a TCP/IP or GPIB communication channel.

Internally, the program handles routing of commands between the remote computer and the BERTScope LTS. Select the Trace Messages box to monitor the commands sent to the BERTScope LTS.

To test the Remote Client connection, use a TCP/IP terminal program like `telnet.exe` to interact with the BERTScope LTS directly. An example session is shown in the figure above.

### Command syntax

Remote Control Command Lines are defined as ASCII text strings ending with `'\r'` or `'\n'`.

The commands follow a basic three-part structure, consisting of one word identifying the feature of the BERTScope LTS being addressed, another word identifying a specific operation within that feature, and optional parameters. The feature and operation are separated by a colon, and the operation and parameter (if any), are separated by a space.

**FEATURE:OPERATION PARAMETER**

Commands and parameters are generally case-insensitive.

Most commands have a long form and a short form. For example, `LTS:CONNECTed?` can be typed out exactly as shown, or as `LTS:CONN?` or `lts:conn?` At minimum, the capitalized letters are required; lower-case letters are not required. The short form is convenient during interactive control (using a telnet session, for example). You might want to use the long form in scripts or programs, to increase code clarity.

**Features**

The BERTScope LTS commands are divided into feature groups. In the first group are commands that handle device discovery and connection. You can use these commands anytime, even if a current device has not been selected yet (with the `OPEN` command). The `NAMES`, `OPEN`, and `CLOSE` commands are all part of this group. A second group consists of commands for controlling and monitoring a particular BERTScope LTS. The current device must have been specified (using the `LTS:OPEN` command) prior to using any commands in this group.

In addition to these feature groups, several miscellaneous features exist to manipulate the host computer display (`VIEW`, `PTFILE`, `GUILOCKOUT`), the run state (`RSTATE`, `RDURATION`), and the overall program configuration (`SCONFIGURATION`, `RCONFIGURATION`, `ISSTATUS`, `RSSTATUS`). These commands primarily implement the functionality provided by the GUI Console buttons.

**Operations**

Operations fall into three subcategories:

- Read-Only properties.

Query a readable property by appending a question mark after it. For example,

```
LTS:DN?
```

returns the device name of the currently selected BERTScope LTS.

- Read-Write Properties

Query a read-write property by appending a question mark after it. Set a read-write property by adding an appropriate parameter. For example,

```
LTS:OPEN "LTS_1234"
```

selects the BERTScope LTS with the device name `LTS_1234` as the current device, while `LTS:OPEN?` returns name of the device previously OPENed.

- Write-Only operation

Question marks are illegal for write-only operation, and parameters may or may not be required, depending upon the particular operation. For example,

`LTS:CLOSE`

requires no parameters, but

`LTS:RCONFIGDISK "D:\BitAnalyzer\Configurations\mycfg.lts"`

requires the complete path name of the BERTScope LTS configuration file.

### Parameters

A parameter can be Double or Integer numbers, Boolean (0 or 1), an Enumerated Data Type, or a String. When a command uses a string as a parameter, the string is enclosed in double-quotes (“string”). Filenames always require the absolute path, enclosed within double-quotes. Parameters are generally case-insensitive.

### Command Status

The BERTScope LTS receives and operates on each command immediately. The input values of commands are checked. If an input is received that is outside of the legal range, the input will be clipped and recorded in the status queue. Command execution sets a status variable that may be queried by the user to determine if the previous command was successful. You request the status of a command by sending: `SYSTem:ERRor?` or `SYST:ERR?` This command retrieves items listed in the error status queue in the format: `< n, “event/string” >`.

The status queue is first in, first out. It can contain up to 100 error messages. If the status queue overflows, the last error/event in the queue is replaced with Error -350, “Queue overflow”. When all errors/events have been read from the queue, further error/event queries will return 0, “NO error”. To clear the status queue all at once, send: `*CLS`

### Tips

It is useful to have an understanding of the basic principles of the BERTScope LTS before starting the remote control programming. There is a high degree of similarity between the BERTScope graphical user interface and the remote control command protocol.

For more technical assistance in programming your remote control applications, please contact Tektronix Support.

# LTS command summary

---

**NOTE.** *Lifecycle commands (usable before OPEN)*

---

<b>Dynamic read-only properties</b>	TOTALDEVCOUNT or TDCT ATTACHDEVCOUNT or ADCT ATTACHDEVNAMES or NAMES
<b>Read-write settings</b>	OPEN
<b>Write-only operations</b>	CLOSE  (require prior OPEN)
<b>Static read-only identification and capabilities</b>	MODEL SERIALNUM or SN DEVICENAME or DN SWREV or SW HWREV or HW FPGAREV or FPGA EXPREV or EXP NUMTAPS INVALIDVALUE or IVALUE
<b>Dynamic read-only properties</b>	CONNECTED or CONN BUSY RXPOWER RXCRLOCKED
<b>Read-write settings</b>	OPTICALPOWERUNITS or PWRUNITS TXOPMODE or TXMODE TXLASERON or TXON TXNOMFREQ or XRFREQ TXISI TXFILTER TXPOWER TXEXTINCTIONRATIO or TXER RXLASERON or RXON RXNOMFREQ or RXFREQ RXWAVELENGTH or RXWL RXFILTER

**Write-only operations**    SCONFIGDISK or SDISK  
                                 RCONFIGDISK or RDISK

## LTS Remote Control Commands

### **LTS:AttachDevCount?**

Retrieves the number of unique BERTScope LTSs (unique serial numbers) that are currently responding, whether or not the remote computer has opened a session to any of them.

Returns: < >

### **LTS:attachdevNAMES?**

Retrieves a comma-separated list of all BERTScope LTSs that are currently communicating with the Clock Recovery Service, whether or not the remote computer has opened a session to any of them.

Each BERTScope LTS is identified by its device name.

Returns:            <"string">                            Comma-separated list of BERTScope LTSs found, or "" if none found

### **LTS:BUSY?**

Retrieves the busy status of the currently selected BERTScope LTS.

Returns:            < 0 >                            Device is for available for communication  
                         < 1 >                            Device is temporarily unavailable for communication

### **LTS:CLOSE**

Close the communication connection to the current BERTScope LTS. Closing the current BERTScope LTS requires that a new current BERTScope LTS be selected (or opened) prior to issuing any more device-specific commands.



**LTS:CONNEcted?**

Retrieves the connection status of the currently selected BERTScope LTS.

Returns:	< 0 >	Device is opened, but not found by the host computer
	< 1 >	Device is opened and available for communication

**LTS:DeviceName?**

Retrieves the device name of the currently selected BERTScope LTS. A device name can be from 1 to 11 characters in length.

Returns:	<"string">	Device name of current device
----------	------------	-------------------------------

**LTS:EXPrev?**

Retrieves the expansion board revision string of the currently selected BERTScope LTS. If no expansion board is installed in the BERTScope LTS, then "XXXX" is returned.

Returns:	<"string">	Expansion board revision string
----------	------------	---------------------------------

**LTS:FPGArev?**

Retrieves the FPGA revision string of the currently selected BERTScope LTS.

Returns:	<"string">	FPGA revision string
----------	------------	----------------------

**LTS:HWrev?**

Retrieves the hardware revision string of the currently selected BERTScope LTS.

Returns:	<"string">	Hardware revision string
----------	------------	--------------------------

**LTS:invalidVALUE?**

Retrieves the "magic number" returned if a BERTScope LTS measurement cannot be made.

Returns:	<"string">	Invalid value
----------	------------	---------------

**LTS:MODEL?**

Retrieves the model type of the currently selected BERTScope LTS.

Returns:	<8500A_TXRX>	LTS contains both transmitter and receiver
	<8500A_TX>	LTS contains only transmitter
	<8500A_RX>	LTS contains only receiver

**LTS:OPEN <“string”>**

**LTS:OPEN?**

Selects or returns the device name of the current BERTScope LTS. The user-assigned device name or serial number string might be passed to select a particular BERTScope LTS. An empty string might be passed to select the first (usually only) BERTScope LTS found. A BERTScope LTS can be opened by multiple clients.

Params:	<“string”>	Serial number, device name, or “”
Returns:	<“string”>	Current device name, or “NONE”

**LTS:opticalPoWeRUNITS <enum>**

**LTS:opticalPoWeRUNITS?**

Sets and retrieves the units to use for the optical power settings and measurements

Params:	<dBm>	dB per milliWatt
	<mW>	milliWatt
	<uW>	microWatt
Returns:	< dBm   mW   uW >	

**LTS:RconfigDISK<“string”>**

Restores the BERTScope LTS settings from a file. The complete path must be specified in the parameter string. The path must be accessible to the host computer or BERTScope running the remote control software. The file extension should be .LTS. Action only.

Example:

LTS :RCONFIGDISK “D:\BitAlyzer\Configurations\mycfg.LTS”

Params:	<“string”>	Path name of the configuration file
---------	------------	-------------------------------------

**LTS:RXCRLOCKED?**

Retrieves the receiver clock recovery hardware state in the currently selected BERTScope LTS. Query only.

Returns:        < 0 >    Path name of the configuration file  
                      < 1 >

**LTS:RXFILTER <enum>****LTS:RXFILTER?**

Sets or retrieves the receiver Low Pass Filter setting.

Param:          <Bypass>    Bypasses this control  
                      <10GbE>    Use internal filter, suitable for 10 Gbit Ethernet  
                      <8XFC>    Use internal filter, suitable for 8x Fiber Channel  
                      <4XFC>    Use internal filter, suitable for 4x Fiber Channel  
                      <2XFC>    Use internal filter, suitable for 2x Fiber Channel

Returns:        < Bypass | 10GbE | 8XFC | 4XFC | 2XFC >

**LTS:RXlaserON <boolean>****LTS:RXlaserON?**

Enables or disables the receiver optical to electrical transducer.

Param:          < 0 >    Laser transducer Disabled  
                      < 1 >    Laser transducer Enabled

Returns:        < 0 | 1 >

**LTS:RXNOMFREQ**  
**LTS:RXNOMFREQ?**

Sets or retrieves the receiver input data rate, in Hz. The LTS needs to know the receiver input data rate to select the proper calibration values.

NOTE: See LTS:TXNOMFREQ.

Param:	<numeric>	Set the current data rate in Herz. Range from 150 MHz to 12.5 GHz
Returns:	<numeric>	Measured, or previously entered data rate, in Hertz

**LTS:RXPOWER?**

Retrieves the measured average power of the receiver optical data input.

NOTE: See LTS:opticalPowerUNITS.

Returns:	<numeric>	In dBm, mW, or uW, depending upon the current optical power units setting.
----------	-----------	--

**LTS:RXWaveLength <numeric>**  
**LTS:RXWaveLength?**

Sets or Retrieves the wavelength of the receiver optical input, in nm.

Param:	<numeric>	850, 1310, or 1550 nm
Returns:	<numeric>	Current setting, in nm

**LTS:SconfigDISK <"string">**

Saves the current BERTScope LTS settings to a file. The complete path must be specified in the parameter string. The path must be accessible to the host computer or BERTScope running the remote control software. The file extension should be .LTS.

Example:

LTS:SCONFIGDISK "D:\BitAlyzer\Configurations\mycfg.LTS"

Param:	<"string">	Path name of the configuration file
--------	------------	-------------------------------------

**LTS:SerialNum?**

Retrieves the serial number string of the currently selected BERTScope LTS.

Param:           <"string">                               Serial number string of current device

**LTS:SWrev?**

Retrieves the firmware revision string of the currently selected BERTScope LTS.

Returns:         <"string">                               Firmware revision string

**LTS:TotalDevCount?**

Retrieves the total number of unique BERTScope LTSs (unique serial numbers) found. Query only.

---

**NOTE.** *This number might include devices that are no longer connected to the host computer, or are turned off.*

---

Returns:         <numeric>

**LTS:TXExtinctionRatio <numeric>****LTS:TXExtinctionRatio?**

Sets or retrieves the extinction ratio of the transmitter optical output, in dB.

Param:           <numeric>                               2.0 dB to 6.0 dB.

Returns:         <numeric>                               Current ER setting.

**LTS:TXFILTER <enum>****LTS:TXFILTER?**

Sets or retrieves the transmitter Low Pass Filter setting.

Param:           <Bypass>                                Bypasses this control

                  <10GbE>                                Use internal filter, suitable for 10 Gbit Ethernet

                  <External>                            Use external filter connected to the front of the LTS unit.

Returns:         < BYPASS | 10GBE | EXTERNAL >

**LTS:TXISI <enum>**  
**LTS:TXISI?**

Sets or retrieves the transmitter ISI setting.

Param:     <Bypass>                    Bypasses this control  
            <8XFC>                     Set suitable for 8x Fiber Channel

Returns:   < BYPASS | 8XFC >

**LTS:TXlaserON <boolean>**  
**LTS:TXlaserON?**

Enables or disables the transmitter output laser.

Param:     < 0 >                        Laser Disabled  
            < 1 >                        Laser Enabled

Returns:   < 0 | 1 >

**LTS:TXNOMFREQ**  
**LTS:TXNOMFREQ?**

Sets or retrieves the transmitter input data rate, in Hertz. The LTS needs to know the transmitter input data rate to select the proper calibration values.

NOTE: See LTS :RXNOMFREQ.

Param:     <numeric>                    Range [150 MHz to 12.5 GHz]

Returns:   <numeric>                    Measured, or previously entered data rate, in Hertz

**LTS:TXopMODE <enum>**  
**LTS:TXopMODE?**

Sets or Rretrieves the operational mode of the currently selected BERTScope LTS. When an operational mode is set, the transmitter ISI, Filter, average Power, and Extinction Ratio are forced to settings appropriate for the mode. If one of these settings is subsequently manually changed, then the operation mode will return to Manual.

NOTE: See LTS:TXISI, LTS:TXFILTER, LTS:TXPOWER, and LTS:TXEXTINCTIONRATIO.

Param:	<Manual>	Any setting allowed
	<Reference>	ISI = Bypass Filter = Bypass Power = 1.0 mW Extinction Ratio = 5.0 dB
	<OMA_Cal>	ISI = Bypass Filter = Bypass Power = 1.0 mW Extinction Ratio = 3.0 dB
	<Filter>	ISI = Bypass Filter = 10 Gbit Ethernet Power = 1.0 mW Extinction Ratio = 3.0 dB
Returns:	< MANUAL   REFERENCE   OMA_CAL   FILTER >	

**LTS:TXPOWER <numeric>**  
**LTS:TXPOWER?**

Sets or retrieves the average power of the transmitter optical output, (in dBm, mW, or  $\mu$ W, depending upon the current optical power units setting).

NOTE: See LTS:opticalPowerUNITS.

Param:	<numeric>	0.1 mW to 1.2 mW, or equiv. in other units.
Returns:	<numeric>	In dBm, mW, or $\mu$ W, depending upon the current optical power units setting.





# Analysis Engine

Complete command listing (See page 7, *Command listing*.)

The following commands apply to bit error rate testing and analysis.

Error Free Interval (See page 158, *Error Free Interval*.)

Basic BER(See page 149, *Basic BER*.)

Block Errors (See page 150, *Block Errors*.)

Burst Length (See page 153, *Burst Length*.)

Correlation (See page 155, *Correlation*.)

2D Error Map (See page 160, *Error Map*.)

FEC Emulation (See page 163, *FEC Emulation*.)

Pattern Sensitivity (See page 169, *Pattern Sensitivity*.)

Strip Chart (See page 172, *Strip Chart*.)

## **EANalysis:BEFThreshold <long>**

### **EANalysis:BEFThreshold?**

Set or retrieve Burst Error Free Threshold.

Params:	<long>	Range [1 to 1,000,000]
---------	--------	------------------------

Returns:	< long >	Burst Error Free Threshold
----------	----------	----------------------------

## **EANalysis:BMLength <long>**

### **EANalysis:BMLength?**

Set or retrieve Burst Minimum Length.

Params:	<long>	Burst Minimum Length. Range [2 to 1,000,000]
---------	--------	--

Returns:	< long >	Burst Minimum Length
----------	----------	----------------------

## **EANalysis:BRANalysis?**

Returns Analysis Bit Rate. Query only.

## **EANalysis:BSBLanking <bool>**

### **EANalysis:BSBLanking?**

Set or retrieve whether Bursts should span Blanking.

Params:	<bool>	On = 1, Off = 0
---------	--------	-----------------

Returns:	< 1 >	Turn on Bursts Span Blanking
----------	-------	------------------------------

	< 0 >	Turn off Bursts Span Blanking
--	-------	-------------------------------

**EANalysis:BSIPeriod <bool>**  
**EANalysis:BSIPeriod?**

Set or retrieve whether Bursts should span Integration Periods.

Params:	<bool>	On = 1, Off = 0
Returns:	< 1 >	Turn on Bursts Span Integration Period
	< 0 >	Turn off Bursts Span Integration Period

**EANalysis:BSMarker <bool>**  
**EANalysis:BSMarker?**

Set or retrieve whether Bursts should span Markers.

Params:	<bool>	On = 1, Off = 0
Returns:	< 1 >	Turn on Bursts Span Markers
	< 0 >	Turn off Bursts Span Markers

**EANalysis:BRACquisition?**

Returns Acquisition Bit Rate. Query only.

**EANalysis:CAResults**

Clear all Error Analysis results. Action only.

**EANalysis:CDBLanking <bool>**  
**EANalysis:CDBLanking?**

Set or retrieve whether to Count During Blanking.

Params:	<bool>	On = 1, Off = 0
Returns:	< 1 >	Count During Blanking
	< 0 >	Do not Count During Blanking

**EANalysis:CPUFile**

Cancel Playback of UER File. Action only.

**EANalysis:EPSecond?**

Returns Events Per Second. Query only.

**EANalysis:IBLanking <bool>**  
**EANalysis:IBLanking?**

Set or retrieve whether to Invert Blanking.

Params:	<bool>	On = 1, Off = 0
Returns:	< 1 >	Invert Blanking
	< 0 >	Do not Invert Blanking

**EANalysis:IMODE <BLANK | DATA | MARKer>**  
**EANalysis:IMODE?**

Set or retrieve Integration Mode to define the Integration Period boundary.

Params:	<BLANK>	Select Blank signal as the boundary
	<DATA>	Set the number of bits as the boundary
	<MARKer>	Select Marker signal as the boundary
Returns:	< BLANK   DATA   MARKER >	

**EANalysis:IPERiod <double>**  
**EANalysis:IPERiod?**

Set or retrieve the number of bits in an Integration Period.

Params:	<long>	Integration Period in bits. Range [10,000,000 to 1.00E+20]
Returns:	< long >	Integration Period

**EANalysis:MPSecond?**

Returns Markers Per Second. Query only.

**EANalysis:OUFile <bool>**  
**EANalysis:OUFile?**

Set or retrieve whether it is OK To Overwrite UER File.

Params:	<bool>	On = 1, Off = 0
Returns:	< 1 >	OK to Overwrite UER File
	< 0 >	Do not Overwrite UER File

**EANalysis:PUPRogress?**

Returns Playback UER Progress. Query only.

**EANalysis:RAPRoperties<"filename">**

Restore All Properties. Action only.

Params:	<"filename">	Use the absolute path name and enclose the string in double quotes.
---------	--------------	---

**EANalysis:RFPathname <"string">**  
**EANalysis:RFPathname?**

Set or retrieve Record File Path name.

Params:	<"string">	Use the absolute path name and enclose the string in double quotes.
Returns:	<"string">	The path name to the Record File

**EANalysis:RFSLimit <long>**  
**EANalysis:RFSLimit?**

Set or retrieve Record File Size Limit

Params:	<long>	Range [100 kB to 500 MB]
Returns:	< long >	Record file size limit

**EANalysis:RPATtern?**

Returns Recognized Pattern. Query only.

**EANalysis:SADefaults**

Set All Defaults. Action only.

**EANalysis:SANalyzer?**

Returns Analyzer State. Query only.

**EANalysis:SAPRoperties<“string”>**

Save All Properties. Action only.

Params:	<“string”>	Use absolute path name, enclose string in double quotes.
---------	------------	--

**EANalysis:SECount?**

Returns Squelch Event Count. Query only.

**EANalysis:SMODE <BLANK | MARKer | NONE | PATtern>**  
**EANalysis:SMODE?**

Set or retrieve Synchronization Mode.

Params:	<BLANK>	Select Blank signal as the cue for resync
	<MARKer>	Select Marker signal as the cue for resync
	<NONE>	No auto-resync
	<PATtern>	Select Pattern Cycle as the cue for resync
Returns:	< BLANK   MARKER   NONE   PATTERN >	

**EANalysis:SPUFile**

Start Playback of UER File. Action only.

**EANalysis:SState?**

Returns Squelch State. Query only.

**EANalysis:SSYNchronization?**

Returns Synchronization State. Query only.

**EANalysis:UFPathname <"string">****EANalysis:UFPathname?**

Set or retrieve UER File path name

Params:	<"string">	Use absolute path name, enclose string in double quotes.
---------	------------	--

Returns:	<"string">	UER File path name
----------	------------	--------------------

**EANalysis:UFSize?**

Returns UER File Size. Query only.

## Basic BER

Complete command listing (See page 7, *Command listing*.)

**EANalysis:BBER:BERate?**

Retrieve Burst Error Rate. Query only.

**EANalysis:BBER:BERrors?**

Retrieve Burst Errors. Query only.

**EANalysis:BBER:BEVents?**

Retrieve Burst Events. Query only.

**EANalysis:BBER:IPCount?**

Retrieve Integration Period Count. Query only.

**EANalysis:BBER:MCOunt?**

Retrieve Marker Count. Query only.

**EANalysis:BBER:NAValue?**

Retrieve "Not Available" value. Query only.

**EANalysis:BBER:NBERate?**

Retrieve Non-Burst Error Rate. Query only.

**EANalysis:BBER:SCOunt?**

Retrieve Squelched Count. Query only.

**EANalysis:BBER:STATistics?**

Retrieve Total Bit Count, Total Error Count, and Total Squelched Count. Query only.

**EANalysis:BBER:TBERate?**

Retrieve Total Error Rate. Query only.

**EANalysis:BBER:TCOunt?**

Retrieve Total Bit Count. Query only.

**EANalysis:BBER:TERRors?**

Retrieve Total Errors. Query only.

**EANalysis:BBER:TSCOunt?**

Retrieve Total Squelched Count. Query only.

## Block Errors

Complete command listing (See page 7, *Command listing*.)

**EANalysis:BLOCK:ACENTER**

Auto Center Block Errors view. Action only.

**EANalysis:BLOCK:BCOunt?**

Retrieve Block Errors chart Bin Count. Query only.

**EANalysis:BLOCK:BDATA?**

Retrieve Block Errors binary Bin Data. Query only. The binary query should return #1nnbbb...b where:

- 1 is a number from 1 to 9 (tells number of digits for block length)
- nn is the block length (number of bytes to follow)
- b is a byte of binary data

**EANalysis:BLOCK:BMAPPING <int, int>**

**EANalysis:BLOCK:BMAPPING?**

Set or retrieve Block Errors chart Bin Mapping (start, end).

Params:	<int>	Bin Map Start value. Range [0 to 998]
	<int>	Bin Map End value. Range [1 to 32,767]
Returns:	< int, int >	

**EANalysis:BLOCK:BMODE <BLANking | MARKer | PATtern | USERquantity>  
EANalysis:BLOCK:BMODE?**

Set or retrieve mode used to define the Block Errors view Block boundary.

Params:	<BLANking>	Select Blank signal as the boundary
	<MARKer>	Select Marker signal as the boundary
	<PATtern>	Select Pattern Cycle as the boundary
	<USERquantity>	Select a user-specified quantity of bits as the boundary
Returns:	< BLANKING   MARKER   PATTERN   USERQUANTITY >	

**EANalysis:BLOCK:BRResolution?**

Retrieve Block Errors view Bin Resolution. Query only.

**EANalysis:BLOCK:BQUANTITY?**

Retrieve quantity of bits set as the Block boundary. Query only.

**EANalysis:BLOCK:CEXTents <Xmin, Xmax, Ymin, Ymax>  
EANalysis:BLOCK:CEXTents?**

Set or retrieve Block Errors view Chart Extents

Params:	<Xmin>	X-Axis minimum, double
	<Xmax>	X-Axis maximum, double
	<Ymin>	Y-Axis minimum, double
	<Ymax>	Y-Axis maximum, double
Returns:	< Xmin, Xmax, Ymin, Ymax >	

**EANalysis:BLOCK:ECOunt?**

Retrieve Block Errors Element Count. Query only.

**EANalysis:BLOCK:FCURSorsEAN:BLOC:FCUR**

Fit Block Errors Cursors. Action only.

**EANalysis:BLOCK:HCURSors <bool>  
EANalysis:BLOCK:HCURSors?**

Enable/disable Block Errors Horizontal Cursors.

Params:	<bool>	On = 1, Off = 0
Returns:	< 1 >	Enable horizontal cursors
	< 0 >	Disable horizontal cursors

**EANalysis:BLOCK:LSCale <bool>****EANalysis:BLOCK:LSCale?**

Enable/disable Block Errors Log Scale.

Params:	<bool>	On = 1, Off = 0
Returns:	< 1 >	Enable Log scale on chart
	< 0 >	Disable Log scale on chart

**EANalysis:BLOCK:UQUantity <double>****EANalysis:BLOCK:UQUantity?**

Set or retrieve Block Errors view Block Mode User Quantity, if BLOCK:BMODE (above) is set to USERquantity.

Params:	<UQUantity>	Range [1 to 1,073,741,823] (Upper limit is 2E30-1)
Returns:	< double >	Quantity of bits defining a Block

**EANalysis:BLOCK:VCURsors <bool>****EANalysis:BLOCK:VCURsors?**

Enable/disable Block Errors Vertical Cursors.

Params:	<bool>	On = 1, Off = 0
Returns:	< 1 >	Enable Block Errors vertical cursors
	< 0 >	Disable Block Errors vertical cursors



# Burst Length

Complete command listing (See page 7, *Command listing*.)

## **EAnalysis:BLENGTH:ACENTER**

Auto Center the Burst Length view. Action only.

## **EAnalysis:BLENGTH:BCOUNT?**

Retrieve Burst Length view Bin Count. Query only.

## **EAnalysis:BLENGTH:BDATA?**

Retrieve Burst Length binary Bin Data. Query only. The binary query should return #1nnbbb...b where:

1 is a number from 1 to 9 (tells number of digits for block length)

nn is the block length (number of bytes to follow)

b is a byte of binary data

## **EAnalysis:BLENGTH:BMAPPING <int, int>**

## **EAnalysis:BLENGTH:BMAPPING?**

Set or retrieve Burst Length Bin Mapping (start, end)

Params:	<int>	Bin Map Start value. Range [0 to 998]
	<int>	Bin Map End value. Range [1 to 32,767]

Returns: < int, int >

## **EAnalysis:BLENGTH:BRRESOLUTION?**

Retrieve Burst Length bin resolution. Query only.

## **EAnalysis:BLENGTH:CEXTENTS <Xmin, Xmax, Ymin, Ymax>**

## **EAnalysis:BLENGTH:CEXTENTS?**

Set or retrieve Burst Length Chart Extents.

Params:	<Xmin>	X-Axis minimum, double
	<Xmax>	X-Axis maximum, double
	<Ymin>	Y-Axis minimum, double
	<Ymax>	Y-Axis maximum, double

Returns: < Xmin, Xmax, Ymin, Ymax >

## **EAnalysis:BLENGTH:ECOUNT?**

Retrieve Block Errors Element Count. Query only.

**EANalysis:BLENght:FCURsors**

Fit Burst Length Cursors. Action only.

**EANalysis:BLENght:HCURsors <bool>**

**EANalysis:BLENght:HCURsors?**

Enable/disable horizontal cursors on Burst Length view.

Params:	<bool>	On = 1, Off = 0
Returns:	< 1 >	Enable horizontal cursors
	< 0 >	Disable horizontal cursors

**EANalysis:BLENght:LSCale <bool>**

**EANalysis:BLENght:LSCale?**

Enable/disable Log scale on Burst Length View.

Params:	<bool>	On = 1, Off = 0
Returns:	< 1 >	Enable Log scale on chart
	< 0 >	Disable Log scale on chart

**EANalysis:BLENght:VCURsors <bool>**

**EANalysis:BLENght:VCURsors?**

Enable/disable vertical cursors on Burst Length view.

Params:	<bool>	On = 1, Off = 0
Returns:	< 1 >	Enable vertical cursors
	< 0 >	Disable vertical cursors

# Correlation

Complete command listing (See page 7, *Command listing*.)

## **EANalysis:CORRelation:ACENter**

Auto Center Correlation View. Action only.

## **EANalysis:CORRelation:BCOunt?**

Retrieve Correlation Bin Count. Query only.

## **EANalysis:CORRelation:BDATA?**

Retrieve Correlation binary Bin Data. Query only. The binary query should return #1nnbbb . . . b, where:

- 1 is a number from 1 to 9 (tells number of digits for block length)
- nn is the block length (number of bytes to follow)
- b is a byte of binary data

## **EANalysis:CORRelation:BMApping <int, int>**

## **EANalysis:CORRelation:BMApping?**

Set or retrieve Bin Mapping (start, end) values.

Params:	<int>	Bin Map Start value. Range [0 to 998]
	<int>	Bin Map End value. Range [1 to 37,767]
Returns:	< int, int >	

## **EANalysis:CORRelation:BREResolution?**

Retrieve Correlation Bin Resolution. Query only.

## **EANalysis:CORRelation:CMODE <BLANKing | MARKer | PATtern | USERquantity>**

**EANalysis:CORRelation:CMODE?**

Set or retrieve Correlation Mode to define the Correlation Cycle boundary.

Params:	<BLANKing>	Select Blank signal as the boundary
	<MARKer>	Select Marker signal as the boundary
	<PATTern>	Select Pattern Cycle as the boundary
	<USERquantity>	Select a user-specified quantity of bits as the boundary
Returns:	< BLANKING   MARKER   PATTERN   USERQUANTITY >	

**EANalysis:CORRelation:CEXTents <Xmin, Xmax, Ymin, Ymax>**  
**EANalysis:CORRelation:CEXTents?**

Set or retrieve Correlation Chart Extents.

Params:	<Xmin>	X-Axis minimum, double
	<Xmax>	X-Axis maximum, double
	<Ymin>	Y-Axis minimum, double
	<Ymax>	Y-Axis maximum, double
Returns:	< Xmin, Xmax, Ymin, Ymax >	

**EANalysis:CORRelation:CQUANTITY?**

Retrieve Correlation Quantity. Query only.

**EANalysis:CORRelation:ECOunt?**

Retrieve Correlation Element Count. Query only.

**EANalysis:CORRelation:FCURsors**

Fit Correlation Cursors. Action only.

**EANalysis:CORRelation:HCURsors <bool>**  
**EANalysis:CORRelation:HCURsors?**

Enable/disable horizontal cursors on Correlation view.

Params:	<bool>	On = 1, Off = 0
Returns:	< 1 >	Enable horizontal cursors
	< 0 >	Disable horizontal cursors

**EANalysis:CORRelation:LSCale <bool>**  
**EANalysis:CORRelation:LSCale?**

Enable/disable Log scale on Correlation chart.

Params:	<bool>	On = 1, Off = 0
Returns:	< 1 >	Enable Log scale on chart
	< 0 >	Disable Log scale on chart

**EANalysis:CORRelation:UQUANTITY <double>**  
**EANalysis:CORRelation:UQUANTITY?**

Set or retrieve User Quantity used to define the Correlation Cycle boundary.

Params:	<USERquantity>	Range [1 to 1,073,741,823] (Upper limit is 2E30-1)
Returns:	< double >	Quantity of bits defining a Correlation Cycle

**EANalysis:CORRelation:VCURsors <bool>**  
**EANalysis:CORRelation:VCURsors?**

Enable/disable vertical cursors on Correlation view.

Params:	<bool>	On = 1, Off = 0
Returns:	< 1 >	Enable Correlation vertical cursors
	< 0 >	Disable Correlation vertical cursors

## Error Free Interval

Complete command listing (See page 7, *Command listing*.)

### **EANalysis:EFINterval:ACENter**

Auto Center the Error Free Interval view. Action only.

### **EANalysis:EFINterval:BCOunt?**

Retrieve Bin Count on EFI Analysis view. Query only.

### **EANalysis:EFINterval:BDATA?**

Retrieve binary Bin Data on EFI Analysis view. Query only. The binary query should return #1nnbbb . . . b, where:

- 1 is a number from 1 to 9 (tells number of digits for block length)
- nn is the block length (number of bytes to follow)
- b is a byte of binary data

### **EANalysis:EFINterval:BMAPPing <int, int>**

### **EANalysis:EFINterval:BMAPPing?**

Set or retrieve Bin Mapping (start, end) values on EFI Analysis view.

Params:	<int>	Bin Map Start value. Range [1 to 999]
	<int>	Bin Map End value. Range [2 to 32,767]
Returns:	< int, int >	

### **EANalysis:EFINterval:BRESolution?**

Retrieve Bin Resolution on EFI Analysis view. Query only.

### **EANalysis:EFINterval:CEXTents <Xmin, Xmax, Ymin, Ymax>**

### **EANalysis:EFINterval:CEXTents?**

Set or retrieve Chart Extents on EFI Analysis view.

Params:	<Xmin>	X-Axis minimum, double
	<Xmax>	X-Axis maximum, double
	<Ymin>	Y-Axis minimum, double
	<Ymax>	Y-Axis maximum, double
Returns:	< Xmin, Xmax, Ymin, Ymax >	

### **EANalysis:EFINterval:ECOunt?**

Retrieve Element Count. Query only.

**EANalysis:EFINterval:FCURsors**

Fit Cursors. Action only.

**EANalysis:EFINterval:HCURsors <bool>****EANalysis:EFINterval:HCURsors?**

Enable/disable horizontal cursors on EFI Analysis view.

Params:	<bool>	On = 1, Off = 0
Returns:	< 1 >	Enable horizontal cursors
	< 0 >	Disable horizontal cursors

**EANalysis:EFINterval:LSCale <bool>****EANalysis:EFINterval:LSCale?**

Enable/disable Log scale on EFI Analysis view.

Params:	<bool>	On = 1, Off = 0
Returns:	< 1 >	Enable Log scale on chart
	< 0 >	Disable Log scale on chart

**EANalysis:EFINterval:VCURsors <bool>****EANalysis:EFINterval:VCURsors?**

Enable/disable vertical cursors on EFI Analysis view.

Params:	<bool>	On = 1, Off = 0
Returns:	< 1 >	Enable EFI Analysis vertical cursors
	< 0 >	Disable EFI Analysis vertical cursors

## Error Map

Complete command listing (See page 7, *Command listing*.)

### **EANalysis:EMAP:ACENter**

Auto Center 2-D Error Map View. Action only.

### **EANalysis:EMAP:CEXTents <Xmin, Xmax, Ymin, Ymax>**

#### **EANalysis:EMAP:CEXTents?**

Set or retrieve 2-D Error Map Chart Extents.

Params:	<Xmin>	X-Axis minimum, double
	<Xmax>	X-Axis maximum, double
	<Ymin>	Y-Axis minimum, double
	<Ymax>	Y-Axis maximum, double
Returns:	< Xmin, Xmax, Ymin, Ymax >	

### **EANalysis:EMAP:CMODE <NONE | SEGMENT | SLENGTH | SINGLE>**

#### **EANalysis:EMAP:CMODE?**

Set or retrieve the 2-D Error Map View Cursor Mode.

Params:	<NONE>	No cursors are displayed
	<SEGMENT>	Segment axis cursor displayed
	<SLENGTH>	Length axis cursor displayed
	<SINGLE>	Single cursor is displayed
Returns:	< NONE   SEGMENT   SLENGTH   SINGLE >	

### **EANalysis:EMAP:DATA?**

Error Map Data. Query only.

### **EANalysis:EMAP:DBCCount?**

Retrieve 2-D Error Map Database Burst Count. Query only.

### **EANalysis:EMAP:DMCCount?**

Retrieve 2-D Error Map Database Marker Count. Query only.

### **EANalysis:EMAP:DSCCount?**

Retrieve 2-D Error Map Database Squelch Count. Query only.

### **EANalysis:EMAP:FCURSors**

Fit 2-D Error Map Cursors. Action only.



**EANalysis:EMAP:FGRaph**

Fit 2-D Error Map Graph. Action only.

**EANalysis:EMAP:HCURsors <bool>****EANalysis:EMAP:HCURsors?**

Enable/disable horizontal cursors on 2-D Error Map.

Params:	<bool>	On = 1, Off = 0
Returns:	< 1 >	Enable horizontal cursors
	< 0 >	Disable horizontal cursors

**EANalysis:EMAP:LSCale <bool>****EANalysis:EMAP:LSCale?**

Enable/disable Log scale on 2-D Error Map.

Params:	<bool>	On = 1, Off = 0
Returns:	< 1 >	Enable Log scale on chart
	< 0 >	Disable Log scale on chart

**EANalysis:EMAP:RPRogress?**

Retrieve 2-D Error Map Rendering Progress. Query only.

**EANalysis:EMAP:SBURsts <bool>****EANalysis:EMAP:SBURsts?**

Choose whether to show bursts on 2-D Error Map.

Params:	<bool>	On = 1, Off = 0
Returns:	< 1 >	Enable Show Bursts
	< 0 >	Disable Show Bursts

**EANalysis:EMAP:SCOut?**

Retrieve 2-D Error Map Segment Count. Query only.

**EANalysis:EMAP:SERRors <bool>****EANalysis:EMAP:SERRors?**

Choose whether to show errors on 2-D Error Map.

Params:	<bool>	On = 1, Off = 0
Returns:	< 1 >	Enable Show Errors
	< 0 >	Disable Show Errors

**EANalysis:EMAP:SLENgth?**

Retrieve 2-D Error Map Segment Length. Query only.

**EANalysis:EMAP:SMODE <MARKers | QUANtity | SEConds>****EANalysis:EMAP:SMODE?**

Set or retrieve mode used to define Error Map Segment.

Params:	<MARKers>	Use marker signal to define Segment
	<QUANtity>	Use quantity of bits to define Segment
	<SEConds>	Use quantity of seconds to define Segment
Returns:	< MARKERS   QUANTITY   SECONDS >	

**EANalysis:EMAP:SSEConds <double>****EANalysis:EMAP:SSEConds?**

Set or retrieve the number of seconds used to define a 2-D Error Map Segment if mode is set to SEConds.

Params:	<SEConds>	Number of seconds to define a Segment. Range [1 to 3,600].
Returns:	< double >	Number of seconds defining a Segment

**EANalysis:EMAP:SQUANtity <double>****EANalysis:EMAP:SQUANtity?**

Set or retrieve the number of bits used to define a 2-D Error Map Segment if mode is set to QUANtity.

Params:	<QUANtity>	Range [1 to 1.00E+100].
Returns:	< double >	Number of bits defining a Segment.

**EANalysis:EMAP:SSQuelches <bool>****EANalysis:EMAP:SSQuelches?**

Choose whether to show squelches on 2-D Error Map.

Params:	<bool>	On = 1, Off = 0
Returns:	< 1 >	Enable Show Squelches
	< 0 >	Disable Show Squelches

**EANalysis:EMAP:VCURSors <bool>**  
**EANalysis:EMAP:VCURSors?**

Enable or disable vertical cursors on 2-D Error Map.

Params:	<bool>	On = 1, Off = 0
Returns:	< 1 >	Enable 2-D Error Map vertical cursors
	< 0 >	Disable 2-D Error Map vertical cursors

## FEC Emulation

Complete command listing (See page 7, *Command listing*.)

**EANalysis:FEC:CNAME?**

Retrieve Forward Error Correction Analysis Configuration filename. Query only.

See the *EANalysis:FEC:SCONfiguration* command.

Returns:	<"filename">	"filename" is the name of the FEC configuration.
----------	--------------	--

**EANalysis:FEC:DTAB <CINTERval | PROCessing | TACCumulation>**  
**EANalysis:FEC:DTAB?**

Set or retrieve FEC Display Tab.

Params:	<CINTERval>	Current Interval
	<PROCessing>	Processing
	<TACCumulation>	Total Accumulation
Returns:	< CINTERVAL   PROCESSING   TACCUMULATION >	

**EANalysis:FEC:EMODE <DISabled | FULL | IAOuter | IONLy>  
EANalysis:FEC:EMODE?**

Set or retrieve FEC Emulation Mode.

Params:	<DISabled>	Disable FEC Emulation
	<FULL>	Emulation Mode to Full: all stages engaged.
	<IAOuter>	Emulation Mode to Inner and Outer (2-D code), erasure deactivated
	<IONLy>	Emulation Mode to Inner Only (one-dimensional). Outer code and erasure are disabled.
Returns:	< DISABLED   FULL   IAOUTER   IONLY >	

**EANalysis:FEC:IMODE <BIT | SYMBol>  
EANalysis:FEC:IMODE?**

Set or retrieve FEC Interleave Mode.

Params:	<BIT>	On = 1, Off = 0
	<SYMBol>	On = 1, Off = 0
Returns:	< BIT   SYMBol >	

**EANalysis:FEC:KINNeR <long>  
EANalysis:FEC:KINNeR?**

Set or retrieve Inner (or one-dimensional code) k value (output block size).

Params:	<long>	Range [1 to 255]
Returns:	< long >	

**EANalysis:FEC:KOUTer <long>  
EANalysis:FEC:KOUTer?**

Set or retrieve a two-dimensional code Outer k value (output block size).

Params:	<long>	Range [1 to 1.00E+00]
Returns:	< long >	

**EANalysis:FEC:NINNeR <long>**  
**EANalysis:FEC:NINNeR?**

Set or retrieve Inner (or one-dimensional code) n value (input block size).

Params:	<long>	Range [1 to 65,536]
Returns:	< long >	

**EANalysis:FEC:NOUteR <long>**  
**EANalysis:FEC:NOUteR?**

Set or retrieve a two-dimensional code Outer n value (input block size).

Params:	<long>	Range [1 to 65,536]
Returns:	< long >	

**EANalysis:FEC:RIINteRval <double>**  
**EANalysis:FEC:RIINteRval?**

Set or retrieve Report Interval in seconds.

Params:	<double>	Range [1 to 300]
Returns:	< double >	

**EANalysis:FEC:RIINteRval?**

Retrieve the number of FEC Emulation Tables during the Report Interval. Query only.

**EANalysis:FEC:SFEC <bool>**  
**EANalysis:FEC:SFEC?**

Enable/disable Strip FEC Overhead.

Params:	<bool>	On = 1, Off = 0
Returns:	< 1 >	Enable Strip FEC Overhead
	< 0 >	Disable Strip FEC Overhead

**EANalysis:FEC:SPINteRval?**

Retrieve FEC Emulation Symbols Per Interval. Query only.

**EANalysis:FEC:SSIZe <long>**  
**EANalysis:FEC:SSIZe?**

Set or retrieve FEC (Forward Error Correction) symbol size.

Params:	<long>	Range [1 to 32]
Returns:	< long >	

**EANalysis:FEC:TDFLag <bool>**  
**EANalysis:FEC:TDFLag?**

Enable/disable Two Dimensional Flag.

Params:	<bool>	On = 1, Off = 0
Returns:	< 1 >	Enable Two-Dimensional flag
	< 0 >	Disable Two-Dimensional flag

**EANalysis:FEC:TERasure <long>**  
**EANalysis:FEC:TERasure?**

Set or retrieve a two-dimensional code Erasure t value (erasure strength).

Params:	<long>	Range [0 to 1.00]
Returns:	< long >	

**EANalysis:FEC:TINNER <long>**  
**EANalysis:FEC:TINNER?**

Set or retrieve Inner (or one-dimensional code) t value (correction strength).

Params:	<long>	Range [0 to 255]
Returns:	< long >	

**EANalysis:FEC:TOUTer <long>**  
**EANalysis:FEC:TOUTer?**

Set or retrieve two-dimensional code Outer t value (correction strength).

Params:	<long>	Range [0 to 1.00]
Returns:	< long >	

**EANalysis:FEC:AABCount?**

Retrieve total accumulated Bit Count after FEC Emulation. Query only.

**EANalysis:FEC:AAECount?**

Retrieve total accumulated Error Count after FEC Emulation. Query only.

**EANalysis:FEC:AAERate?**

Retrieve total accumulated Error Rate after FEC Emulation. Query only.

**EANalysis:FEC:ABBCount?**

Retrieve total accumulated Bit Count before FEC Emulation. Query only.

**EANalysis:FEC:ABECount?**

Retrieve total accumulated Error Count before FEC Emulation. Query only.

**EANalysis:FEC:ABERate?**

Retrieve total accumulated Error Rate before FEC Emulation. Query only.

**EANalysis:FEC:AICorrections?**

Retrieve total accumulated Inner Corrections. Query only.

**EANalysis:FEC:AIFailures?**

Retrieve total accumulated Inner Failures. Query only.

**EANalysis:FEC:AOCorrections?**

Retrieve total accumulated Outer Corrections. Query only.

**EANalysis:FEC:AOFailures?**

Retrieve total accumulated Outer Failures. Query only.

**EANalysis:FEC:AECorrections?**

Retrieve total accumulated Erasure Corrections. Query only.

**EANalysis:FEC:AEFailures?**

Retrieve total accumulated Erasure Failures. Query only.

**EANalysis:FEC:ADRate?**

Retrieve Data Rate after FEC Emulation. Query only.

**EANalysis:FEC:BDRate?**

Retrieve Data Rate before FEC Emulation. Query only.

**EANalysis:FEC:IABCount?**

Retrieve current interval Bit Count after FEC Emulation. Query only.

**EANalysis:FEC:IAECount?**

Retrieve current interval Error Count after FEC Emulation. Query only.

**EANalysis:FEC:IAERate?**

Retrieve current interval Error Rate after FEC Emulation. Query only.

**EANalysis:FEC:IBBCount?**

Retrieve current interval Bit Count before FEC Emulation. Query only.

**EANalysis:FEC:IBECount?**

Retrieve current interval Error Count before FEC Emulation. Query only.

**EANalysis:FEC:IBERate?**

Retrieve current interval Error Rate before FEC Emulation. Query only.

**EANalysis:FEC:IECorrections?**

Retrieve interval Erasure Corrections. Query only.

**EANalysis:FEC:IEFailures?**

Retrieve current interval Erasure Failures. Query only.

**EANalysis:FEC:IICorrections?**

Retrieve current interval Inner Corrections. Query only.

**EANalysis:FEC:IIFailures?**

Retrieve current interval Inner Failures. Query only.

**EANalysis:FEC:IOCorrections?**

Retrieve current interval Outer Corrections. Query only.

**EANalysis:FEC:IOFailures?**

Retrieve current interval Outer Failures. Query only.

**EANalysis:FEC:MINPut?**

Retrieve FEC Emulation Markers Input. Query only.

**EANalysis:FEC:MOUOutput?**

Retrieve FEC Emulation Markers output. Query only.

**EANalysis:FEC:OVERhead?**

Retrieve FEC Emulation FEC Overhead. Query only.

**EANalysis:FEC:RCONfiguration<“filename”>**

Restore FEC Configuration. Action only.

Params:	<“filename”>	Configuration filename used for restore the configuration (see SCONfiguration). Enclose the filename in double quotes.
---------	--------------	--

---

**EANalysis:FEC:SCONfiguration<“filename”>**

Save FEC Configuration. Action only.

Params:	<“filename”>	Configuration filename used to save the configuration (see RCONfiguration). Enclose the filename in double quotes.
---------	--------------	--

---



**EANalysis:FEC:TBLanked?**

Retrieve FEC Emulation Tables Blanked. Query only.

**EANalysis:FEC:TOVerrun?**

Retrieve FEC Emulation Tables Overrun. Query only.

**EANalysis:FEC:TPRocessed?**

Retrieve FEC Emulation Tables Processed. Query only.

**EANalysis:FEC:TSQuelched?**

Retrieve FEC Emulation Tables Squelched. Query only.

## Pattern Sensitivity

Complete command listing (See page 7, *Command listing*.)

**EANalysis:PSENSitivity:ACENter**

Auto Center Pattern Sensitivity View. Action only

**EANalysis:PSENSitivity:BCOunt?**

Retrieve Pattern Sensitivity Bin Count. Query only.

**EANalysis:PSENSitivity:BDATA?**

Retrieve binary Pattern Sensitivity Bin Data. Query only. The binary query should return #1nnbbb...b where:

- 1 is a number from 1 to 9 (tells number of digits for block length)
- nn is the block length (number of bytes to follow)
- b is a byte of binary data

**EANalysis:PSENSitivity:BMAPPING <int, int>**

**EANalysis:PSENSitivity:BMAPPING?**

Set or retrieve Pattern Sensitivity Bin Mapping (start, end) values.

Params:	<int>	Bin Map Start value. Range [1 to 999]
	<int>	Bin Map End value. Range [1 to 32,767]
Returns:	< int, int >	

**EANalysis:PSENSitivity:BRResolution?**

Retrieve Bin Resolution. Query only.

**EANalysis:PSENSitivity:CEXTents <Xmin, Xmax, Ymin, Ymax>  
EANalysis:PSENSitivity:CEXTents?**

Set or retrieve Pattern Sensitivity Chart Extents.

Params:	<Xmin>	X-Axis minimum, double
	<Xmax>	X-Axis maximum, double
	<Ymin>	Y-Axis minimum, double
	<Ymax>	Y-Axis maximum, double
Returns:	< Xmin, Xmax, Ymin, Ymax >	

**EANalysis:PSENSitivity:ECOut?**

Retrieve Element Count. Query only

**EANalysis:PSENSitivity:FCURsors**

Fit Cursors. Action only.

**EANalysis:PSENSitivity:HCURsors <bool>  
EANalysis:PSENSitivity:HCURsors?**

Enable/disable horizontal cursors on Pattern Sensitivity view.

Params:	<bool>	On = 1, Off = 0
Returns:	< 1 >	Enable horizontal cursors
	< 0 >	Disable horizontal cursors

**EANalysis:PSENSitivity:LSCale <bool>  
EANalysis:PSENSitivity:LSCale?**

Enable/disable Log scale on Pattern Sensitivity view.

Params:	<bool>	On = 1, Off = 0
Returns:	< 1 >	Enable Log scale on chart
	< 0 >	Disable Log scale on chart

**EANalysis:PSENSitivity:PFOut?**

Pattern found. Read-only result; query only.

**EANalysis:PSENSitivity:PMODE <PRBS | FILE>**

**EANalysis:PSENSitivity:PMODE?**

Set or retrieve the Pattern Sensitivity Pattern Mode; a pre-defined PRBS pattern, or a User Pattern contained in the named file.

Params:	<PRBS>	Pseudo-Random Pattern used in Pattern Sensitivity analysis.
	<"filename">	The filename specified by the EAN:PSEN:UPF command will be used.
Returns:	< PRBS   filename >	

**EANalysis:PSENSitivity:UPFile <"filename">**

**EANalysis:PSENSitivity:UPFile?**

Set or retrieve Pattern Sensitivity User Pattern filename

Params:	<"filename">	Filename of the User Pattern File, enclosed in double quotes.
Returns:	< filename >	

**EANalysis:PSENSitivity:VCURSors <bool>**

**EANalysis:PSENSitivity:VCURSors?**

Enable/disable vertical cursors on Pattern Sensitivity view.

Params:	<bool>	On = 1, Off = 0
Returns:	< 1 >	Enable Pattern Sensitivity vertical cursors
	< 0 >	Disable Pattern Sensitivity vertical cursors

# Strip Chart

Complete command listing (See page 7, *Command listing*.)

## **EANalysis:SHart:ACENTER**

Auto Center Strip Chart View. Action only.

## **EANalysis:SHart:ASCROLL <bool>**

### **EANalysis:SHart:ASCROLL?**

Enable/disable Strip Chart auto scroll.

Params:	<bool>	On = 1, Off = 0
Returns:	< 1 >	Enable Strip Chart auto scroll
	< 0 >	Disable Strip Chart auto scroll

## **EANalysis:SHart:CEXTENTS <Xmin, Xmax, Ymin, Ymax>**

### **EANalysis:SHart:CEXTENTS?**

Set or retrieve Strip Chart chart extents.

Params:	<Xmin>	X-Axis minimum, double
	<Xmax>	X-Axis maximum, double
	<Ymin>	Y-Axis minimum, double
	<Ymax>	Y-Axis maximum, double
Returns:	< Xmin, Xmax, Ymin, Ymax >	

## **EANalysis:SHart:FCURSORS**

Fit Cursors. Action only.

## **EANalysis:SHart:HCURSORS <bool>**

### **EANalysis:SHart:HCURSORS?**

Enable/disable horizontal cursors on Strip Chart.

Params:	<bool>	On = 1, Off = 0
Returns:	< 1 >	Enable horizontal cursors
	< 0 >	Disable horizontal cursors

**EANalysis:SCHart:VCURsors <bool>**  
**EANalysis:SCHart:VCURsors?**

Enabl/disable vertical cursors on Strip Chart.

Params:	<bool>	On = 1, Off = 0
Returns:	< 1 >	Enable Strip Chart vertical cursors
	< 0 >	Disable Strip Chart vertical cursors



---

# Physical Layer Test

Complete command listing (See page 7, *Command listing*.)

The following commands apply to the physical layer testing aspects of the BERTScope/BitAnalyzer Analyzers. Eye Diagram and Mask Test are available as a part of the standard testing suite. The Physical Layer Test for BERTScope Analyzer may include the optional BER Contour, Jitter Peak, Jitter Tolerance, and Q-Factor.

Eye Diagram (See page 176, *Eye Diagram*.)

CleanEye (See page 187, *Eye: CleanEye*.)

Single Value Waveform (See page 207, *Eye: Single Value Waveform*.)

Data Collecting (See page 206, *Eye: Data Collecting*.)

Optical Mode (See page 208, *Eye: Optical Mode*.)

Read Eye Measurement (See page 212, *Eye: Read Eye Measurement*.)

Mask Test (See page 247, *Mask Test*.)

BER Contour (See page 215, *BER Contour*.)

Contour Optical Mode (See page 221, *BER Contour: Optical Mode*.)

Jitter Map (See page 222, *Jitter Map*.)

Jitter Peak (See page 234, *Jitter Peak*.)

Jitter Tolerance (See page 242, *Jitter Tolerance*.)

Q-Factor (See page 253, *Q-Factor*.)

# Eye Diagram

Complete command listing (See page 7, *Command listing*.)

**EYE:AMPPf <UW | DBM>**  
**EYE:AMPPf?**

BERTScope Analyzer only, Optical mode. Set or retrieve the Auto Center optical power format.

Params:	<UW>	Value displayed in $\mu$ W
	<DBM>	Value displayed in dBm
Returns:	< MW   DBM >	

**EYE:ASMode <EOPening | TRANSition>**  
**EYE:ASMode?**

Set or retrieve the Auto Center mode of the Eye view.

Params:	<EOPening>	Auto center at the eye opening
	<TRANSition>	Auto center at the transition
Returns:	< EYEOPENING   TRANSITION >	

**EYE:AVGPwrPf <UW | DBM>**  
**EYE:AVGPwrPf?**

Set or retrieve Optical Mode Average Power format. BERTScope Analyzer only.

Params:	<UW>	Value displayed in $\mu$ W
	<DBM>	Value displayed in dBm
Returns:	< UW   DBM >	



**EYE:CMODE <NONE | TIME | VOLT | TVOL | SINGLE>**  
**EYE:CMODE?**

Set or retrieve the cursor mode of the Eye view.

Params:	<NONE>	No cursor is displayed
	<TIME>	Only two time cursors are displayed
	<VOLT>	Only two voltage cursors are displayed
	<TVOL>	Two time cursors and two voltage cursors are displayed
	<SINGLE>	Single cursor is displayed
Returns:	< NONE   TIME   VOLTAGE   TIME&VOLTAGE   SINGLE >	

**EYE:CN0Pf <UW | DBM>**  
**EYE:CN0Pf?**

BERTScope Analyzer only, Optical mode. Set or retrieve Cross Noise 0 optical power format.

Params:	<UW>	Value displayed in $\mu$ W
	<DBM>	Value displayed in dBm
Returns:	< UW   DBM >	

**EYE:CN1Pf <UW | DBM>**  
**EYE:CN1Pf?**

BERTScope Analyzer only, Optical mode. Set or retrieve Cross Noise 1 optical power format.

Params:	<UW>	Value displayed in $\mu$ W
	<DBM>	Value displayed in dBm
Returns:	< UW   DBM >	

**EYE:CRAPf <UW | DBM>**  
**EYE:CRAPf?**

BERTScope Analyzer only, Optical mode. Set or retrieve Cross Amplitude optical power format.

Params:	<UW>	Value displayed in $\mu$ W
	<DBM>	Value displayed in dBm
Returns:	< MW   DBM >	

**EYE:CVFOrmat <MV | %>**  
**EYE:CVFOrmat?**

Set or retrieve Crossing Voltage power format.

Params:	<MV>	Value displayed in mV
	<%>	Value displayed in %UI
Returns:	< MV   % >	

**EYE:CVPF <UW | DBM>**  
**EYE:CVPF?**

BERTScope Analyzer only, Optical mode. Set or retrieve Crossing Voltage optical power format.

Params:	<UW>	Value displayed in $\mu$ W
	<DBM>	Value displayed in dBm
Returns:	< UW   DBM >	

**EYE:EHFOrmat <MV | %>**  
**EYE:EHFOrmat?**

Set or retrieve the Eye Height measurement value format.

Params:	<MV>	Value displayed in mV
	<%>	Value displayed in %UI
Returns:	< MV   % >	

**EYE:EHPF <UW | DBM>**  
**EYE:EHPF?**

BERTScope Analyzer only, Optical mode. Set or retrieve Eye Height optical power format.

Params:	<UW>	Value displayed in $\mu$ W
	<DBM>	Value displayed in dBm
Returns:	< UW   DBM >	

**EYE:EMODE <8020 | 9010>**  
**EYE:EMODE?**

Set or retrieve the edge mode of the Eye view.

Params:	<8020>	Use 80%–20% edge
	<9010>	Use 90%–10% edge
Returns:	< 8020   9010 >	

**EYE:ERFO <dB | %>**  
**EYE:ERFO?**

Set or retrieve the Extinction Ratio measurement value format.

Params:	<dB>	Value displayed in dB
	<%>	Value displayed in %
Returns:	< dB   % >	

**EYE:ESAV <"filename">**

Save Eye diagram matrix data to a CSV file. Full path name should be given. Action only.

Params:	<"filename">	Filename in double quotes, include full path
---------	--------------	--

**EYE:EWFO <TIME | %UI>**  
**EYE:EWFO?**

Set or retrieve the Eye Width measurement value format.

Params:	<TIME>	Value displayed in ps
	<%UI>	Value displayed in %UI
Returns:	< TIME   %UI >	

**EYE:EXPFileopt**  
**EYE:EXPFileopt?**

Set or retrieve the file format for exporting single-value waveform data to a .csv file.

Params:	<YVAL>	Export Y values
	<YVALVERB>	Export Y values with header information
	<XYVAL>	Export X and Y values
	<XYVALVERB>	Export X and Y values with header information
Returns:	< YVAL   YVALVERB   XYVAL   XYVALVERB >	

**EYE:EYEOper <CLASSIC | CLEAN>**  
**EYE:EYEOper?**

Set or retrieve the Eye diagram operating mode.

Params:	<CLASSIC>	Classic Eye mode
	<CLEAN>	CleanEye mode
Returns:	< CLASSIC   CLEAN >	

**EYE:GENable <bool>**  
**EYE:GENable?**

Enable/disable Eye display graticule (grid).

Params:	<bool>	On = 1, Off = 0
Returns:	< 1 >	Eye graticule is on
	< 0 >	Eye graticule is off

**EYE:JPFOrmat <TIME | %UI>**  
**EYE:JPFOrmat?**

Set or retrieve the Eye Jitter P-P measurement value format.

Params:	<TIME>	Value displayed in ps
	<%UI>	Value displayed in %UI
Returns:	< TIME   %UI >	

**EYE:JRFOrmat <TIME | %UI>**  
**EYE:JRFOrmat?**

Set or retrieve the Eye Jitter RMS measurement value format.

Params:	<TIME>	Value displayed in ps
	<%UI>	Value displayed in %UI
Returns:	< TIME   %UI >	

**EYE:L0PF <UW | DBM>**  
**EYE:L0PF?**

BERTScope Analyzer only, Optical mode. Set or retrieve Level 0 optical power format.

Params:	<UW>	Value displayed in $\mu$ W
	<DBM>	Value displayed in dBm
Returns:	< UW   DBM >	

**EYE:L1PF <UW | DBM>**  
**EYE:L1PF?**

BERTScope Analyzer only, Optical mode. Set or retrieve Level 1 optical power format.

Params:	<UW>	Value displayed in $\mu$ W
	<DBM>	Value displayed in dBm
Returns:	< UW   DBM >	

**EYE:LCRTHresh <numeric>**  
**EYE:LCRTHresh?**

Set or retrieve the threshold level (in mV or  $\mu$ W, depending on the mode) for the Rising or Falling Level Crossing measurement.

Params:	<numeric>	The threshold value in mV or $\mu$ W that defines the level at which the 80/20 or 90/10 rising or falling edge crosses to set the Rising or Falling Level Crossing in ps. Limits vary depending on the signal, and must be between the 80/20 or 90/10 points on the rising or falling edges, depending on the edge mode configuration.
---------	-----------	--

---

Returns: <numeric>

---

**EYE:MBARsel <OFF | MEASUREMENTS | STRESS>**  
**EYE:MBARsel?**

Set or retrieve the Sidebar selection in the Eye Diagram.

Params:	<OFF>	No Sidebar
	<MEASUREMENTS>	Show Measurement Sidebar
	<STRESS>	Show Stress Sidebar (if Stress Option is enabled)

---

Returns: < OFF | MEASUREMENTS | STRESS >

---

**EYE:MDENable <bool>**  
**EYE:MDENable?**

Enable/disable display of Eye measurements.

Params:	<bool>	On = 1, Off = 0
Returns:	< 1 >	Eye measurements display enabled
	< 0 >	Eye measurements display disabled

---

**EYE:NL0Pf <UW | DBM>**  
**EYE:NL0Pf?**

BERTScope Analyzer only, Optical mode. Set or retrieve Noise Level 0 optical power format.

Params:	<UW>	Value displayed in $\mu$ W
	<DBM>	Value displayed in dBm
Returns:	< UW   DBM >	

**EYE:NL1Pf <UW | DBM>**  
**EYE:NL1Pf?**

BERTScope Analyzer only, Optical mode. Set or retrieve Noise Level 1 optical power format.

Params:	<UW>	Value displayed in $\mu$ W
	<DBM>	Value displayed in dBm
Returns:	< UW   DBM >	

**EYE:NLPF <UW | DBM>**  
**EYE:NLPF?**

BERTScope Analyzer only, Optical mode. Set or retrieve Noise Level optical power format.

Params:	<UW>	Value displayed in $\mu$ W
	<DBM>	Value displayed in dBm
Returns:	< UW   DBM >	

**EYE:OMAPf <UW | DBM>**  
**EYE:OMAPf?**

BERTScope Analyzer only, Optical mode. Set or retrieve OMA optical power format.

Params:	<UW>	Value displayed in $\mu$ W
	<DBM>	Value displayed in dBm
Returns:	< UW   DBM >	

**EYE:OverSHoot1FOrmat <MV | %>**  
**EYE:OverSHoot1FOrmat?**

Set or retrieve the Eye Overshoot1 measurement format.

Params:	<MV>	Value displayed in mV
	<%>	Value displayed in %UA
Returns:	< MV   % >	

**EYE:OverSHoot0Format <MV | %>**  
**EYE:OverSHoot0Format?**

Set or retrieve the Eye Overshoot0 measurement format.

Params:	<MV>	Value displayed in mV
	<%>	Value displayed in %UA
Returns:	< MV   % >	

**EYE:PERsistence <numeric>**  
**EYE:PERsistence?**

Set or retrieve the persistence of the Eye view.

Params:	<numeric>	Eye persistence as a number of images being persisted. Range [0 to 5]. Input out of range will be clipped and recorded in the status queue.  A value of -1 runs the Eye diagram with infinite persistence.
Returns:	<numeric>	

**EYE:SDEPth <numeric>**  
**EYE:SDEPth?**

Set or retrieve the Eye Sample Depth in bits.

Params:	<numeric>	Range [2,000 to 10,000,000]. Input out of range will be clipped and recorded in the status queue.  A value of -1 runs the Eye diagram in the 'Auto' mode with a sample depth of 10,000 bits.
Returns:	<numeric>	

**EYE:SSAV <"filename">**

Save Single Value Eye diagram data to a CSV file. Full path name should be given. Action only.

Params:	<"filename">	Filename in double quotes, include full path
---------	--------------	--

**EYE:TCOffset <numeric>  
EYE:TCOffset?**

Set or retrieve the Eye center time offset of the Eye view.

Params:	<numeric>	Eye center time offset in ps. The range of the input is the same as described in the numeric keypad if you click the volt Center button in the Eye view. Range [0 to 33,000]
---------	-----------	--

---

Returns:	<numeric>
----------	-----------

---

**EYE:TEXTent <numeric>  
EYE:TEXTent?**

Set or retrieve the time extent of the Eye view.

Params:	<numeric>	Eye time extent in ps. Range [200 to 33,000]. Input out of range will be clipped and recorded in the status queue.
---------	-----------	--

---

Returns:	<numeric>
----------	-----------

---

**EYE:TOFFset <numeric>  
EYE:TOFFset?**

Set or retrieve the time offset of the Eye view.

Params:	<numeric>	Eye time offset in ps. Range [-16,500 to +16,500]. Input out of range will be clipped and recorded in the status queue.
---------	-----------	---

---

Returns:	<numeric>	An integer. Range [-16,500 to +16,500]
----------	-----------	--

---



**EYE:VCOffset <numeric>**  
**EYE:VCOffset?**

Set or retrieve the Eye center voltage offset of the Eye view.

Params:	<numeric>	Eye center voltage offset in mV. The range of the input is the same as described in the numeric keypad if you click the time Center button in the Eye view. Range [-2000 to +4000]
---------	-----------	--

---

Returns:	<numeric>
----------	-----------

---

**EYE:VEXTent <numeric>**  
**EYE:VEXTent?**

Set or retrieve the voltage extent of the Eye view.

Params:	<numeric>	Eye voltage extent in mV. Range [160 to 6,000]. Input out of range will be clipped and recorded in the status queue.
---------	-----------	--

---

Returns:	<numeric>
----------	-----------

---

**EYE:VMAPf <UW | DBM>**  
**EYE:VMAPf?**

BERTScope Analyzer only, Optical mode. Set or retrieve Maximum voltage power format.

Params:	<UW>	Value displayed in $\mu$ W
	<DBM>	Value displayed in dBm

---

Returns:	< UW   DBM >
----------	--------------

---

**EYE:VMIPf <UW | DBM>**  
**EYE:VMIPf?**

BERTScope Analyzer only, Optical mode. Set or retrieve Minimum voltage power format.

Params:	<UW>	Value displayed in $\mu$ W
	<DBM>	Value displayed in dBm

---

Returns:	< UW   DBM >
----------	--------------

---

**EYE:VOFFset <numeric>**  
**EYE:VOFFset?**

Set or retrieve the voltage offset of the Eye view.

Params:	<numeric>	Eye voltage offset in mV. Range [-2000 to 4000]. Input out of range will be clipped and recorded in the status queue.
Returns:	<numeric>	

**EYE:VOFPf <UW | DBM>**  
**EYE:VOFPf?**

BERTScope Analyzer only, Optical mode. Set or retrieve Voltage Offset power format.

Params:	<UW>	Value displayed in $\mu$ W
	<DBM>	Value displayed in dBm
Returns:	< UW   DBM >	

**EYE:VPPPf <UW | DBM>**  
**EYE:VPPPf?**

BERTScope Analyzer only, Optical mode. Set or retrieve Peak-to-Peak power format.

Params:	<UW>	Value displayed in $\mu$ W
	<DBM>	Value displayed in dBm
Returns:	< UW   DBM >	

## Eye: CleanEye

Complete command listing (See page 7, *Command listing*.)

### **EYE:AMPLitude:CleanEye:CONFIG?**

Retrieve configuration data of the CleanEye Amplitude measurement. Command available for BSA7500, BSA125C, BSA175C and BSA286C models only. Query only. The data is returned in the following format (in one line, shown here as multiline for clarity):

Max point: <from>%UI-<to>%UI(<method>),  
Min point: <from>%UI-<to>%UI(<method>)

Example:

Max point: 30%UI-70%UI(Average), Min point:  
40%UI-55%UI(Min)

In this example, the “from”, “to”, and “method” values correspond to the current settings. “Method” could be either “Average”, “Mode” or “Max” for the Max point and “Average”, “Mode” or “Min” for the Min point. These parameters are set using the EYE:AMPL:CEYE:POINT command.

### **EYE:AMPLitude:CleanEye:ENABLE <bool>**

### **EYE:AMPLitude:CleanEye:ENABLE?**

Set or retrieve whether CleanEye amplitude measurement is enabled. Command available BSA7500, BSA125C, BSA175C and BSA286C models only.

Params:	<bool>	On = 1, Off = 0
Returns:	< 1 >	CleanEye amplitude measurement is enabled
	< 0 >	CleanEye amplitude measurement is not enabled

**EYE:AMPLitude:CleanEye:POINT <0 | 1>, <numeric>, <numeric>, <AVERAGE | MODE | MIN | MAX>**

Set an individual point for CleanEye Amplitude measurement. Command available for BSA7500, BSA125C, BSA175C and BSA286C models only. Action only.

Params:	<0   1>	Point Number (0 for Max point, 1 for Min point)
	<numeric>	From: (integer value in %UI)
	<numeric>	To: (integer value in %UI)
	<method>	AVERAGE, MODE, or MAX for Max point AVERAGE, MODE, or MIN for Min point
Returns:	< 0   1 >, <numeric>, <numeric>, < AVERAGE   MODE   MIN   MAX >	

**EYE:AMPLitude:CleanEye:PROTOCOL <protocol>  
EYE:AMPLitude:CleanEye:PROTOCOL?**

Set a protocol preset for CleanEye Amplitude measurement. The preset affects “from”, “to”, and “method” values for both measurement points. If no protocol was set, returns “None”. Command available for BSA7500, BSA125C, BSA175C and BSA286C models only.

Params:	<protocol>	PCIE SATA_minamp SATA_maxamp SAS DPORT10 DPORT11 USB3 None
Returns:	< protocol >	

**EYE:CLRSamdep <long>  
EYE:CLRSamdep?**

Set or retrieve CleanEye sample depth.

Params:	<numeric>	Range [2,000 to 1,000,000].
Returns:	<numeric>	

**EYE:CLRPtlen <numeric>**  
**EYE:CLRPtlen?**

Set or retrieve CleanEye Pattern length in bits.

Params:	<numeric>	Range [64 to 1,048,576]. A value of -1 calculates pattern length automatically based on input detector pattern.
Returns:	<numeric>	

**EYE:DEEMphasisRatio:CleanEye:CONFIG?**

Retrieve configuration data of the CleanEye Deemphasis Ratio measurement. Command available for BSA7500, BSA125C, BSA175C and BSA286C models only. Query only. The data is returned in the following format (in one line, shown here as multiline for clarity):

Full 1 point: <from>%UI-<to>%UI(<method>),  
 Deemp 1 point: <from>%UI-<to>%UI(<method>),  
 Full 0 point: <from>%UI-<to>%UI(<method>),  
 Deemp 0 point: <from>%UI-<to>%UI(<method>)

Example:

Full 1 point: 30%UI-70%UI(Average), Deemp 1 point:  
 120%UI-350%UI(Mode),  
 Full 0 point: 120%UI-450%UI(Min), Deemp 0 point:  
 40%UI-55%UI(Mode)

In this example, the “from”, “to”, and “method” values correspond to the current settings. “Method” could be either “Average”, “Mode” or “Max” for the “1” points and “Average”, “Mode” or “Min” for the “0” points. These parameters are set using the EYE:DEEM:CEYE:POINT command.

**EYE:DEEMphasisRatio:CleanEye:ENABLE <bool>**  
**EYE:DEEMphasisRatio:CleanEye:ENABLE?**

Set or retrieve whether CleanEye de-emphasis ratio measurement is enabled. Command available for BSA7500, BSA125C, BSA175C and BSA286C models only.

Params:	<bool>	On = 1, Off = 0
Returns:	< 1 >	CleanEye de-emphasis ratio measurement is enabled
	< 0 >	CleanEye de-emphasis ratio measurement is not enabled

**EYE:DEEMphasisRatio:CleanEye:POINT <0 | 1 | 2 | 3>, <numeric>, <numeric>, <AVERAGE | MODE | MIN | MAX>**

Set an individual point for CleanEye Deemphasis Ratio measurement. Command available for BSA7500, BSA125C, BSA175C and BSA286C models only.

Params:	<0   1   2   3>	Point Number 0: Full 1 point 1: De-emphasized 1 point 2: Full 0 point 3: De-emphasized 0 point)
	<from>	From: (integer value in %UI)
	<to>	To: (integer value in %UI)
	<method>	AVERAGE MODE, or MAX for Max point AVERAGE, MODE, or MIN for Min point
Returns:	< 0   1   2   3 >, <numeric>, <numeric>, < AVERAGE   MODE   MIN   MAX >	

**EYE:DEEMphasisRatio:CleanEye:PROTOCOL <protocol>  
EYE: DEEMphasisRatio:CleanEye:PROTOCOL?**

Set a protocol preset for CleanEye De-Emphasis Ratio measurements. When setting, the preset affects “from”, “to”, and “method” values for all 4 measurement points. If no protocol was set, returns “None”. Command available for BSA7500, BSA125C, BSA175C and BSA286C models only.

Params:	<protocol>	PCIE_dsfs PCIE_dsds PCIE_fsfs PCIE_fsds SAS DPORT10 DPORT11 USB3 None
Returns:	< protocol >	

**EYE:EXPFileopt**  
**EYE:EXPFileopt?**

Set or retrieve the file format for exporting Single Value Waveform data to a .csv file.

Params:	<YVAL>	Export Y values
	<YVALVERB>	Export Y values with header information
	<XYVAL>	Export X and Y values
	<XYVALVERB>	Export X and Y values with header information
Returns:	< YVAL   YVALVERB   XYVAL   XYVALVERB >	

**EYE:EYEOper <CLASSIC | CLEAN>**  
**EYE:EYEOper?**

Set or retrieve the Eye diagram operating mode.

Params:	<CLASSIC>	Classic Eye mode
	<CLEAN>	CleanEye mode
Returns:	< CLASSIC   CLEAN >	

**EYE:FIREnab <bool>**  
**EYE:FIREnab?**

Enable/disable FIR filter.

Params:	<bool>	On = 1, Off = 0
Returns:	< 1 >	FIR filter is enabled
	< 0 >	FIR filter is disabled

**EYE:FIRTpspac <double>**  
**EYE:FIRTpspac?**

Set or retrieve FIR filter spacing in UI.

Params:	<double>	Range [10 to 100]
Returns:	< double >	

**EYE:FIRLoad <"filename">**

Load the FIR filter parameters specified in the file. Full path name should be given. Action only.

Params:	<"filename">	Filename in double quotes, include full path.
---------	--------------	---

**EYE:FIRSave <"filename">**

Save the current FIR filter parameters in the file specified. Full path name should be given. Action only.

Params:	<"filename">	Filename in double quotes, include full path.
---------	--------------	---

---

**EYE:FIRDetails?**

Retrieve the current FIR filter parameters. Query only.

Returns:	< TapCount(TapCoeff) >	Example: 2(0.1) (0.2)
----------	------------------------	-----------------------

---

**EYE:FTIME:CleanEye:CONFIG?**

Retrieve configuration data of the CleanEye Fall Time measurement. Command available for BSA7500, BSA125C, BSA175C and BSA286C models only. Query only. The data is returned in the following format (in one line, shown here as multiline for clarity):

Edge Mode: <mode>,  
 Max point: <from>%UI-<to>%UI(<method>),  
 Min point: <from>%UI-<to>%UI(<method>)

Example:

Edge Mode: 8020, Max point: 30%UI-70%UI(Average), Min point: 40%UI-55%UI(Min)

In this example, the “edge mode”, “from”, “to”, and “method” values correspond to the current settings. “Edge Mode” could be either “8020” or “9010”. “Method” could be either “Average”, “Mode” or “Max” for the Max point and “Average”, “Mode” or “Min” for the Min point. These parameters are set using the EYE:FTIME:CEYE:POINT command.

**EYE:FTIME:CleanEye:ENABLE <bool>**

**EYE:FTIME:CleanEye:ENABLE?**

Set or retrieve whether CleanEye Fall Time measurement is enabled. Command available for BSA7500, BSA125C, BSA175C and BSA286C models only.

Params:	<bool>	On = 1, Off = 0
Returns:	< 1 >	CleanEye fall time measurement is enabled
	< 0 >	CleanEye fall time measurement is not enabled

---



**EYE:FTIME:CleanEye:POINT <0 | 1>, <numeric>, <numeric>, <AVERAGE | MODE | MIN | MAX>**

Set an individual point for CleanEye Fall Time measurement. Command available for BSA7500, BSA125C, BSA175C and BSA286C models only.

Params:	<0   1>	Point Number 0: Max point 1: Min point
	<from>	From: (integer value in %UI)
	<to>	To: (integer value in %UI)
	<method>	AVERAGE MODE, or MAX for Max point AVERAGE, MODE, or MIN for Min point
Returns:	< 0   1 >, <numeric>, <numeric>, < AVERAGE   MODE   MIN   MAX >	

**EYE:FTIME:CleanEye:PROTOCOL <protocol>  
EYE:FTIME:CleanEye:PROTOCOL?**

Retrieve a protocol preset for CleanEye Fall Time measurement. If no protocol was set, returns "None". Command available for BSA7500, BSA125C, BSA175C and BSA286C models only.

Params:	<protocol>	PCIE SATA SAS DPORT10 None
Returns:	< protocol >	

**EYE:LRMbits <numeric>  
EYE:LRMbits?**

Set or retrieve number of bits to Single Value Waveform export.

Params:	<numeric>	Range [1 to length of input pattern]
Returns:	<numeric>	

**EYE:LRMSAmperbit <numeric>  
EYE:LRMSAmperbit?**

Set or retrieve samples per bit for Single Value Waveform export.

Params:	<numeric>	Samples per bit to export, a numeric value. Range [1 to 1,000]
Returns:	<numeric>	

**EYE:LRMStartbit <numeric>  
EYE:LRMStartbit?**

Set or retrieve Single Value Waveform start bit.

Params:	<numeric>	Range [0 to pattern length]
Returns:	<numeric>	

**EYE:LRMExport <"filename">**

Export Single Value Waveform data to a CSV file. Full path name should be given. Action only.

Params:	<"filename">	Filename in double quotes, include full path.
---------	--------------	---

**EYE:LRMCancel**

Cancel the LRM export. Action only.

**EYE:LRMPRPcnt?**

Retrieve the progress of the LRM export in percentage. Query only.

**EYE:LRMPRCnt?**

Retrieve the progress of the LRM export in terms of sample count. Query only.

**EYE:RTIME:CleanEye:CONFIG?**

Retrieve configuration data of the CleanEye Rise Time measurement. Command available for BSA7500, BSA125C, BSA175C and BSA286C models only. Query only. The data is returned in the following format (in one line, shown here as multiline for clarity):

Edge Mode: <mode>,  
Max point: <from>%UI-<to>%UI(<method>),  
Min point: <from>%UI-<to>%UI(<method>)

Example:

Edge Mode: 8020, Max point: 30%UI-70%UI(Average), Min point: 40%UI-55%UI(Min)

In this example, the “edge mode”, “from”, “to”, and “method” values correspond to the current settings. “Edge Mode” could be either “8020” or “9010”. “Method” could be either “Average”, “Mode” or “Max” for the Max point and “Average”, “Mode” or “Min” for the Min point. These parameters are set using the EYE:RTIM:CEYE:POINT command.

**EYE:RTIME:CleanEye:ENABLE <bool>**  
**EYE:RTIME:CleanEye:ENABLE?**

Set or retrieve whether CleanEye Rise Time measurement is enabled. Command available for BSA7500, BSA125C, BSA175C and BSA286C models only.

Params:	<bool>	On = 1, Off = 0
Returns:	< 1 >	CleanEye rise time measurement is enabled
	< 0 >	CleanEye rise time measurement is not enabled

**EYE:RTIME:CleanEye:POINT <0 | 1>, <numeric>, <numeric>, <AVERAGE | MODE | MIN | MAX>**

Set an individual point for CleanEye Rise Time measurement. Command available for BSA7500, BSA125C, BSA175C and BSA286C models only.

Params:	<0   1>	Point Number 0: Max point 1: Min point
	<from>	From: (integer value in %UI)
	<to>	To: (integer value in %UI)
	<method>	AVERAGE, MODE, or MAX for Max point AVERAGE, MODE, or MIN for Min point
Returns:	< 0   1 >, <numeric>, <numeric>, < AVERAGE   MODE   MIN   MAX >	

**EYE:RTIME:CleanEye:PROTOCOL <protocol>**  
**EYE:RTIME:CleanEye:PROTOCOL?**

Retrieve a protocol preset for CleanEye Rise Time measurement. If no protocol was set, returns “None”. Command available for BSA7500, BSA125C, BSA175C and BSA286C models only.

Params:	<protocol>	PCIE SATA SAS DPORT10 None
Returns:	< protocol >	

**EYE:SSAV <"filename">**

Save Single Value Eye diagram data to a CSV file. Full path name should be given. Action only.

Params:	<"filename">	Filename in double quotes, include full path.
---------	--------------	---

---

## Eye: Enable Eye Measurement Overlay

Complete command listing (See page 7, *Command listing*.)

**EYE:MCONfig:ADD<0LEV | 1LEV | AMPL | AVGV | CAMP | CAN0 | CAN1 | CVOLt | DCAL | DEEMR | DSNR | EHE | EWID | EXTR | FLC | FTIM | JITT | JRMS | NLEV | NLV0 | NLV1 | OMA | OVSH1 | OVSH2 | RLC | RTIM | SCNT | UINT | VECP | VMAX | VMIN | VOFS | VPP>**

**EYE:MCON:ADD**

Add the specified measurement to the Eye measurement list. Action only.

Params:	<0LEV or 0LEVel>	0 Level
	<1LEV or 1LEVel>	1 Level
	<AMPL or AMPLitude>	Amplitude
	<AVGVP or AVGVoltsPwr>	Average Volts/Power
	<CAMP or CAMPitude>	Cross Amplitude
	<CAN0>	Cross Amplitude Noise 0
	<CAN1>	Cross Amplitude Noise 1
	<CVOL or CVOLT>	Crossing Voltage
	<DCAL or DCALibration>	Dark Calibration
	<DEEMR or DEEMphasisRatio>	De-Emphasis Ratio
	<DSNR>	DSNR
	<EHE or EHEight>	Eye Height
	<EWID or EWIDth>	Eye Width
	<EXTR or EXTRatio>	Extinction Ratio
	<FLC or FLCross>	Falling Level Crossing
	<FTIM or FTIme>	Fall Time
	<JITT or JITTer>	Jitter Peak-to-Peak

---

<JRMS>	Jitter RMS
<NLEV or NLEVel>	Noise Level
<NLV0>	Noise Level 0
<NLV1>	Noise Level 1
<OMA>	OMA
<OVSH1 or OVerSHoot1>	Overshoot 1
<OVSH2 or OVerSHoot2>	Overshoot 2
<RLC or RLCross>	Rising Level Crossing
<RTIM or RTIME>	Rise Time
<SCNT>	Sample Count
<UNIT or UINterval>	Unit Interval
<VECP>	VECP
<VMAX>	Voltage Maximum
<VMIN>	Voltage Minimum
<VOFS or VOFSet>	Voltage Offset
<VPP>	Voltage Peak-to-Peak

**EYE:MConfig:REM<0LEV | 1LEV | AMPL | AVGV | CAMP | CAN0 | CAN1 | CVOLt | DCAL | DEEMR | DSNR | EHE | EWID | EXTR | FLC | FTIM | JITT | JRMS | NLEV | NLV0 | NLV1 | OMA | OVSH1 | OVSH2 | RLC | RTIM | SCNT | UINT | VECP | VMAX | VMIN | VOFS | VPP>**

#### **EYE:MCON:REM**

Remove the specified measurement from the Eye measurement list. Action only.

Params:	<0LEV or 0LEVel>	0 Level
	<1LEV or 1LEVel>	1 Level
	<AMPL or AMPLitude>	Amplitude
	<AVGVP or AVGVoltsPwr>	Average Volts/Power
	<CAMP or CAMPlitude>	Cross Amplitude
	<CAN0>	Cross Amplitude Noise 0
	<CAN1>	Cross Amplitude Noise 1
	<CVOL or CVOLt>	Crossing Voltage
	<DCAL or DCALibration>	Dark Calibration

<DEEMR or DEEMphasisRatio>	De-Emphasis Ratio
<DSNR>	DSNR
<EHE or EHEight>	Eye Height
<EWID or EWIDth>	Eye Width
<EXTR or EXTRatio>	Extinction Ratio
<FLC or FLCross>	Falling Level Crossing
<FTIM or FTIME>	Fall Time
<JITT or JITTer>	Jitter Peak-to-Peak
<JRMS>	Jitter RMS
<NLEV or NLEVel>	Noise Level
<NLV0>	Noise Level 0
<NLV1>	Noise Level 1
<OMA>	OMA
<OVSH1 or OverSHoot1>	Overshoot 1
<OVSH2 or OverSHoot2>	Overshoot 2
<RLC or RLCross>	Rising Level Crossing
<RTIM or RTIME>	Rise Time
<SCNT>	Sample Count
<UNIT or UINterval>	Unit Interval
<VECP>	VECP
<VMAX>	Voltage Maximum
<VMIN>	Voltage Minimum
<VOFS or VOFSet>	Voltage Offset
<VPP>	Voltage Peak-to-Peak

**EYE:MCONfig:RMAIl**

Remove all measurements from the Eye measurement list. Action only.

---

**NOTE.** The command is uppercase RMA, optionally followed by two lowercase l's – not capital I's.

---

**EYE:MOENable:0LEVel <bool>**  
**EYE:MOENable:0LEVel?**

Set or retrieve whether Eye 0-Level measurement is overlaid.

Params:	<bool>	On = 1, Off = 0
Returns:	< 1 >	Eye Level 0 measurement is overlaid
	< 0 >	Eye Level 0 measurement is not overlaid

**EYE:MOENable:1LEVel <bool>**  
**EYE:MOENable:1LEVel?**

Set or retrieve whether Eye Level1 measurement is overlaid.

Params:	<bool>	On = 1, Off = 0
Returns:	< 1 >	Level 1 measurement is overlaid
	< 0 >	Level 1 measurement is not overlaid

**EYE:MOENable:AMPLitude <bool>**  
**EYE:MOENable:AMPLitude?**

Set or retrieve whether Eye Amplitude measurement is overlaid.

Params:	<bool>	On = 1, Off = 0
Returns:	< 1 >	Amplitude measurement is overlaid
	< 0 >	Amplitude measurement is not overlaid

**EYE:MOENable:AVGVoltsPwr <bool>**  
**EYE:MOENable:AVGVoltsPwr?**

Set or retrieve whether Avg Volts/Pwr measurement is overlaid

Params:	<bool>	On = 1, Off = 0
Returns:	< 1 >	Avg Volts/Pwr measurement is overlaid
	< 0 >	Avg Volts/Pwr measurement is not overlaid

**EYE:MOENable:CAMPLitude <bool>**  
**EYE:MOENable:CAMPLitude?**

Set or retrieve whether Cross Amplitude measurement is overlaid.

Params:	<bool>	On = 1, Off = 0
Returns:	< 1 >	Cross Amplitude measurement is overlaid
	< 0 >	Cross Amplitude measurement is not overlaid

**EYE:MOENable:CAN0 <bool>**  
**EYE:MOENable:CAN0?**

Set or retrieve whether Cross Amplitude Noise Level0 measurement is overlaid.

Params:	<bool>	On = 1, Off = 0
Returns:	< 1 >	Cross Amplitude Noise Level0 measurement is overlaid
	< 0 >	Cross Amplitude Noise Level0 measurement is not overlaid

**EYE:MOENable:CAN1 <bool>**  
**EYE:MOENable:CAN1?**

Set or retrieve whether Cross Amplitude Noise Level1 measurement is overlaid.

Params:	<bool>	On = 1, Off = 0
Returns:	< 1 >	Cross Amplitude Noise Level1 measurement is overlaid
	< 0 >	Cross Amplitude Noise Level1 measurement is not overlaid

**EYE:MOENable:CVOLt <bool>**  
**EYE:MOENable:CVOLt?**

Enable/disable Crossing Voltage measurement is overlaid.

Params:	<bool>	On = 1, Off = 0
Returns:	< 1 >	Crossing Voltage measurement is overlaid
	< 0 >	Crossing Voltage measurement is not overlaid

**EYE:MOENable:DCALibration <bool>**  
**EYE:MOENable:DCALibration?**

Enable/disable Dark Calibration measurement is overlaid.

Params:	<bool>	On = 1, Off = 0
Returns:	< 1 >	Dark Calibration measurement is overlaid
	< 0 >	Dark Calibration measurement is not overlaid

**EYE:MOENable:DEEMphasisRatio <bool>**  
**EYE:MOENable:DEEMphasisRatio?**

Set or retrieve whether De-Emphasis Ratio measurement is overlaid.

Params:	<bool>	On = 1, Off = 0
Returns:	< 1 >	De-Emphasis Ratio measurement is overlaid
	< 0 >	De-Emphasis Ratio measurement is not overlaid

**EYE:MOENable:DSNR <bool>**  
**EYE:MOENable:DSNR?**

Enable/disable Eye DSNR measurement is overlaid.

Params:	<bool>	On = 1, Off = 0
Returns:	< 1 >	Eye DSNR measurement is overlaid
	< 0 >	Eye DSNR measurement is not overlaid



**EYE:MOENable:EHEight <bool>**  
**EYE:MOENable:EHEight?**

Set or retrieve whether Eye height measurement is overlaid.

Params:	<bool>	On = 1, Off = 0
Returns:	< 1 >	Eye height measurement is overlaid
	< 0 >	Eye height measurement is not overlaid

**EYE:MOENable:EWIDth <bool>**  
**EYE:MOENable:EWIDth?**

Set or retrieve whether Eye Width measurement is overlaid.

Params:	<bool>	On = 1, Off = 0
Returns:	< 1 >	Eye's eye width measurement is overlaid
	< 0 >	Eye's eye width measurement is not overlaid

**EYE:MOENable:EXTRatio <bool>**  
**EYE:MOENable:EXTRatio?**

Set or retrieve whether Eye Extinction Ratio measurement is overlaid.

Params:	<bool>	On = 1, Off = 0
Returns:	< 1 >	Eye's Extinction Ratio measurement is overlaid
	< 0 >	Eye's Extinction Ratio measurement is not overlaid

**EYE:MOENable:FLCRoss <bool>**  
**EYE:MOENable:FLCRoss?**

Set or retrieve whether Falling Level Crossing measurement is overlaid.

Params:	<bool>	On = 1, Off = 0
Returns:	< 1 >	Falling Level Crossing measurement is overlaid
	< 0 >	Falling Level Crossing measurement is not overlaid

**EYE:MOENable:FTIME <bool>**  
**EYE:MOENable:FTIME?**

Set or retrieve whether Eye Fall Time measurement is overlaid.

Params:	<bool>	On = 1, Off = 0
Returns:	< 1 >	Eye fall time measurement is overlaid
	< 0 >	Eye fall time measurement is not overlaid

**EYE:MOENable:JITTer <bool>**  
**EYE:MOENable:JITTer?**

Set or retrieve whether Eye Jitter P-P measurement is overlaid.

Params:	<bool>	On = 1, Off = 0
Returns:	< 1 >	Eye jitter P-P measurement is overlaid
	< 0 >	Eye jitter P-P measurement is not overlaid

**EYE:MOENable:JRMS <bool>**  
**EYE:MOENable:JRMS?**

Set or retrieve whether Eye Jitter RMS measurement is overlaid.

Params:	<bool>	On = 1, Off = 0
Returns:	< 1 >	Eye jitter RMS measurement is overlaid
	< 0 >	Eye jitter RMS measurement is not overlaid

**EYE:MOENable:NLEVel <bool>**  
**EYE:MOENable:NLEVel?**

Set or retrieve whether Eye Noise Level measurement is overlaid.

Params:	<bool>	On = 1, Off = 0
Returns:	< 1 >	Eye Noise Level measurement is overlaid
	< 0 >	Eye Noise Level measurement is not overlaid

**EYE:MOENable:NLV0 <bool>**  
**EYE:MOENable:NLV0?**

Set or retrieve whether Eye Noise Level0 measurement is overlaid.

Params:	<bool>	On = 1, Off = 0
Returns:	< 1 >	Eye Noise Level0 measurement is overlaid
	< 0 >	Eye Noise Level0 measurement is not overlaid

**EYE:MOENable:NLV1 <bool>**  
**EYE:MOENable:NLV1?**

Set or retrieve whether Eye Noise Level1 measurement is overlaid.

Params:	<bool>	On = 1, Off = 0
Returns:	< 1 >	Eye Noise Level1 measurement is overlaid
	< 0 >	Eye Noise Level1 measurement is not overlaid

**EYE:MOENable:OMA <bool>**  
**EYE:MOENable:OMA?**

Set or retrieve whether Eye OMA measurement is overlaid.

Params:	<bool>	On = 1, Off = 0
Returns:	< 1 >	Eye OMA measurement is overlaid
	< 0 >	Eye OMA measurement is not overlaid

**EYE:MOENable:OverSHoot0 <bool>**  
**EYE:MOENable:OverSHoot0?**

Set or retrieve whether Overshoot0 measurement is overlaid.

Params:	<bool>	On = 1, Off = 0
Returns:	< 1 >	Eye Overshoot0 measurement is overlaid
	< 0 >	Eye Overshoot0 measurement is not overlaid

**EYE:MOENable:OverSHoot0 <bool>**  
**EYE:MOENable:OverSHoot0?**

Set or retrieve whether Overshoot0 measurement is overlaid.

Params:	<bool>	On = 1, Off = 0
Returns:	< 1 >	Eye Overshoot0 measurement is overlaid
	< 0 >	Eye Overshoot0 measurement is not overlaid

**EYE:MOENable:OverSHoot1 <bool>**  
**EYE:MOENable:OverSHoot1?**

Set or retrieve whether Overshoot1 measurement is overlaid.

Params:	<bool>	On = 1, Off = 0
Returns:	< 1 >	Eye Overshoot1 measurement is overlaid
	< 0 >	Eye Overshoot1 measurement is not overlaid

**EYE:MOENable:RLCRoss <bool>**  
**EYE:MOENable:RLCRoss?**

Set or retrieve whether Rising Level Crossing measurement is overlaid.

Params:	<bool>	On = 1, Off = 0
Returns:	< 1 >	Rising Level Crossing measurement is overlaid
	< 0 >	Rising Level Crossing measurement is not overlaid

**EYE:MOENable:RTIME <bool>**  
**EYE:MOENable:RTIME?**

Set or retrieve whether Eye Rise Time measurement is overlaid.

Params:	<bool>	On = 1, Off = 0
Returns:	< 1 >	Eye rise time measurement is overlaid
	< 0 >	Eye rise time measurement is not overlaid

**EYE:MOENable:SCNT <bool>**  
**EYE:MOENable:SCNT?**

Set or retrieve whether Sample Count measurement is overlaid.

Params:	<bool>	On = 1, Off = 0
Returns:	< 1 >	Eye Sample Count measurement is overlaid
	< 0 >	Eye Sample Count measurement is not overlaid

**EYE:MOENable:UINterval <bool>**  
**EYE:MOENable:UINterval?**

Set or retrieve whether Eye Unit Interval measurement is overlaid.

Params:	<bool>	On = 1, Off = 0
Returns:	< 1 >	Eye unit interval measurement is overlaid
	< 0 >	Eye unit interval measurement is not overlaid

**EYE:MOENable:VECP <bool>**  
**EYE:MOENable:VECP?**

Set or retrieve whether Eye VECP measurement is overlaid.

Params:	<bool>	On = 1, Off = 0
Returns:	< 1 >	Eye VECP measurement is overlaid
	< 0 >	Eye VECP measurement is not overlaid

**EYE:MOENable:VMAX <bool>**  
**EYE:MOENable:VMAX?**

Set or retrieve whether Voltage Maximum measurement is overlaid.

Params:	<bool>	On = 1, Off = 0
Returns:	< 1 >	Eye V MAX measurement is overlaid
	< 0 >	Eye V MAX measurement is not overlaid

**EYE:MOENable:VMIN <bool>**  
**EYE:MOENable:VMIN?**

Set or retrieve whether Voltage Minimum measurement is overlaid.

Params:	<bool>	On = 1, Off = 0
Returns:	< 1 >	Eye V MIN measurement is overlaid
	< 0 >	Eye V MIN measurement is not overlaid

**EYE:MOENable:VOFS <bool>**  
**EYE:MOENable:VOFS?**

Set or retrieve whether Eye V offset measurement is overlaid.

Params:	<bool>	On = 1, Off = 0
Returns:	< 1 >	Eye V offset measurement is overlaid
	< 0 >	Eye V offset measurement is not overlaid

**EYE:MOENable:VPP <bool>**  
**EYE:MOENable:VPP?**

Set or retrieve whether Eye Voltage Peak-to-Peak measurement is overlaid.

Params:	<bool>	On = 1, Off = 0
Returns:	< 1 >	Eye V PP measurement is overlaid
	< 0 >	Eye V PP measurement is not overlaid

**EYE:PDARkcalibration**

Perform Dark Calibration. Action only. May require some delay to complete.

## Eye: Data Collecting

Complete command listing (See page 7, *Command listing*.)

### **EYE:ASSuccess?**

Retrieve report of whether the auto center is successful or not. Query only.

Returns:	<1>	Eye auto center is successful
	<0>	Eye auto center is not successful

---

### **EYE:EPProgress?**

Retrieve the progress of the Eye data collection. Query only.

### **EYE:ISNumber?**

Retrieve the serial number of the latest Eye image.

### **EYE:MSNumber?**

Retrieve the serial number of the latest Eye measurements.

### **EYE:NAVailable?**

Retrieve the not-available value of the Eye measurement. Query only.

### **EYE:STATus?**

Retrieve the status of the Eye diagram. Query only.

Returns:	NOTRUNNING	Status codes
	OK	
	NOCLOCK	
	LOWCLOCK	
	DROPCLOCK	
	FREQCHANGE	

---

## Eye: Single Value Waveform

Complete command listing (See page 7, *Command listing*.)

**EYE:LRMbits <numeric>**

**EYE:LRMbits?**

Set or retrieve number of bits to Single Value Waveform export.

Params:	<numeric>	Range [1 to length of input pattern]
---------	-----------	--------------------------------------

Returns:	<numeric>
----------	-----------

**EYE:LRMSAmperbit <numeric>**

**EYE:LRMSAmperbit?**

Set or retrieve samples per bit for Single Value Waveform export.

Params:	<numeric>	Samples per bit to export, a numeric value. Range [1 to 1,000]
---------	-----------	--

Returns:	<numeric>
----------	-----------

**EYE:LRMStartbit <numeric>**

**EYE:LRMStartbit?**

Set or retrieve Single Value Waveform start bit.

Params:	<numeric>	Range [0 to pattern length]
---------	-----------	-----------------------------

Returns:	<numeric>
----------	-----------

**EYE:LRMExport <"filename">**

Export Single Value Waveform data to a CSV file. Full path name should be given. Action only.

Params:	<"filename">	Filename in double quotes, include full path.
---------	--------------	---

**EYE:LRMCancel**

Cancel the LRM export. Action only.

**EYE:LRMPRPcnt?**

Retrieve the progress of the LRM export in percentage. Query only.

**EYE:LRMPRCnt?**

Retrieve the progress of the LRM export in terms of sample count. Query only.

**EYE:SSAV <"filename">**

Save Single Value Eye diagram data to a CSV file. Use full path name enclosed in double quotes. Action only.

Params:	<"filename">	CSV file to be saved. Filename in double quotes, include full path.
---------	--------------	---

---

## Eye: Optical Mode

Complete command listing (See page 7, *Command listing*.)

---

**NOTE.** *In Electrical mode, choose between mVolts or percent of amplitude for Eye Height, Cross Voltage, and Contour Eye Height measurements. In Optical mode, select a power format of  $\mu W$  or dBm.*

---

**EYE:AMPPf <UW | DBM>**  
**EYE:AMPPf?**

**BERTScope Analyzer only**, Optical mode. Set or retrieve Amplitude power format.

Params:	<UW>	Value displayed in $\mu W$
	<DBM>	Value displayed as dBm
Returns:	< UW   DBM >	

---

**EYE:CN0Pf <UW | DBM>**  
**EYE:CN0Pf?**

**BERTScope Analyzer only**, Optical mode. Set or retrieve Cross Noise 0 power format.

Params:	<UW>	Value displayed in $\mu W$
	<DBM>	Value displayed in dBm
Returns:	< UW   DBM >	

---



**EYE:CN1Pf <UW | DBM>**  
**EYE:CN1Pf?**

**BERTScope Analyzer only**, Optical mode. Set or retrieve Cross Noise 1 power format.

Params:	<UW>	Value displayed in $\mu$ W
	<DBM>	Value displayed in dBm
Returns:	< UW   DBM >	

**EYE:CRAPf <UW | DBM>**  
**EYE:CRAPf?**

**BERTScope Analyzer only**, Optical mode. Set or retrieve Cross Amplitude power format.

Params:	<UW>	Value displayed in $\mu$ W
	<DBM>	Value displayed in dBm
Returns:	< UW   DBM >	

**EYE:CVPF <UW | DBM>**  
**EYE:CVPF?**

**BERTScope Analyzer only**, Optical mode. Set or retrieve Cross voltage power format.

Params:	<UW>	Value displayed in $\mu$ W
	<DBM>	Value displayed in dBm
Returns:	< UW   DBM >	

**EYE:EHPF <UW | DBM>**  
**EYE:EHPF?**

**BERTScope Analyzer only**, Optical mode. Set or retrieve Eye Height power format.

Params:	<UW>	Value displayed in $\mu$ W
	<DBM>	Value displayed in dBm
Returns:	< UW   DBM >	

**EYE:L0PF <UW | DBM>**  
**EYE:L0PF?**

**BERTScope Analyzer only**, Optical mode. Set or retrieve Level 0 power format.

Params:	<UW>	Value displayed in $\mu$ W
	<DBM>	Value displayed in dBm
Returns:	< UW   DBM >	

**EYE:L1PF <UW | DBM>**  
**EYE:L1PF?**

**BERTScope Analyzer only**, Optical mode. Set or retrieve Level 1 power format.

Params:	<UW>	Value displayed in $\mu$ W
	<DBM>	Value displayed in dBm
Returns:	< UW   DBM >	

**EYE:NL0Pf <UW | DBM>**  
**EYE:NL0Pf?**

**BERTScope Analyzer only**, Optical mode. Set or retrieve Noise Level 0 power format.

Params:	<UW>	Value displayed in $\mu$ W
	<DBM>	Value displayed in dBm
Returns:	< UW   DBM >	

**EYE:NL1Pf <UW | DBM>**  
**EYE:NL1Pf?**

**BERTScope Analyzer only**, Optical mode. Set or retrieve Noise Level 1 power format.

Params:	<UW>	Value displayed in $\mu$ W
	<DBM>	Value displayed in dBm
Returns:	< UW   DBM >	

**EYE:NLPF <UW | DBM>**  
**EYE:NLPF?**

**BERTScope Analyzer only**, Optical mode. Set or retrieve Noise Level power format.

Params:	<UW>	Value displayed in $\mu$ W
	<DBM>	Value displayed in dBm
Returns:	< UW   DBM >	

**EYE:OMAPf <UW | DBM>**  
**EYE:OMAPf?**

**BERTScope Analyzer only**, Optical mode. Set or retrieve OMA power format.

Params:	<UW>	Value displayed in $\mu$ W
	<DBM>	Value displayed in dBm
Returns:	< UW   DBM >	

**EYE:OVSH1Pf <UW | DBM>**  
**EYE:OVSH1Pf?**

**BERTScope Analyzer only**, Optical mode. Set or retrieve Overshoot1 power format.

Params:	<UW>	Value displayed in $\mu$ W
	<DBM>	Value displayed in dBm
Returns:	< UW   DBM >	

**EYE:OVSH0Pf <UW | DBM>**  
**EYE:OVSH0Pf?**

**BERTScope Analyzer only**, Optical mode. Set or retrieve Overshoot0 power format.

Params:	<UW>	Value displayed in $\mu$ W
	<DBM>	Value displayed in dBm
Returns:	< UW   DBM >	

**EYE:VMAPf <UW | DBM>**  
**EYE:VMAPf?**

**BERTScope Analyzer only**, Optical mode. Set or retrieve Maximum voltage power format.

Params:	<UW>	Value displayed in $\mu$ W
	<DBM>	Value displayed in dBm
Returns:	< UW   DBM >	

**EYE:VMIPf <UW | DBM>**  
**EYE:VMIPf?**

**BERTScope Analyzer only**, Optical mode. Set or retrieve Minimum voltage power format.

Params:	<UW>	Value displayed in $\mu$ W
	<DBM>	Value displayed in dBm
Returns:	< UW   DBM >	

**EYE:VOFPf <UW | DBM>**  
**EYE:VOFPf?**

**BERTScope Analyzer only**, Optical mode. Set or retrieve Voltage Offset power format.

Params:	<UW>	Value displayed in $\mu$ W
	<DBM>	Value displayed in dBm
Returns:	< UW   DBM >	

**EYE:VPPPf <UW | DBM>**  
**EYE:VPPPf?**

**BERTScope Analyzer only**, Optical mode. Set or retrieve Peak to Peak power format.

Params:	<UW>	Value displayed in $\mu$ W
	<DBM>	Value displayed in dBm
Returns:	< UW   DBM >	

## Eye: Read Eye Measurement

Complete command listing (See page 7, *Command listing*.)

**EYE:AScale**

Auto center the Eye. Action only. May require some delay to complete.

**EYE:CLear**

Clear the Eye image and measurements. Action only.

**EYE:MVALue:0LEVel?**

Retrieve the Eye 0-Level. Query only.

**EYE:MVALue:1LEVel?**

Retrieve the Eye 1-Level. Query only.

**EYE:MVALue:AMPLitude?**

Retrieve the Amplitude. Query only.

**EYE:MVALue:AVGVoltsPwr?**

Retrieve the Eye 0-Level. Query only.

**EYE:MVALue:CAMPLitude?**

Retrieve Cross Amplitude. Query only.

**EYE:MVALue:CAN0?**

Retrieve Cross Amplitude Noise Level 0. Query only.

**EYE:MVALue:CAN1?**

Retrieve Cross Amplitude Noise Level 1. Query only.

**EYE:MVALue:CVOLT?**

Retrieve the Cross Volt measurement value. Query only.

The return value is a number in the format that is determined by the Cross Volt measurement value format. If the format is MV, the value is in mVolts. If the format is %, the value is in percentage. The format can be queried and set by using EYE:CVFOMat command in the Eye: Enable Eye Measurement Overlay section.

**EYE:MVALue:DCALibration?**

Retrieve the Dark Calibration. Query only.

**EYE:MVALue:DEEMphasisRatio?**

Retrieve the current ratio value. Query only.

**EYE:MVALue:DSNR?**

Retrieve the rise DSNR. Query only.

**EYE:MVALue:EHEight?**

Retrieve the Eye Height. Query only.

**EYE:MVALue:EWIDth?**

Retrieve the Eye Width. Query only.

**EYE:MVALue:EXTRatio?**

Retrieve the Extinction Ratio. Query only.

**EYE:MVALue:FLCRoss?**

Queries the Falling Level Crossing in ps.

This is the point in time at which the falling edge (the same edge used for the Fall Time measurement) crosses a configurable level crossing threshold in mV or  $\mu$ W (depending on the mode). Either the 80/20 or the 90/10 points of the falling edge are used, based on the configuration.

See EYE:LCRTHresh to set the level crossing threshold and EYE:EdgeMODE <8020 | 9010> to set the amount of edge to use.

**EYE:MVALue:FTIME?**

Retrieve the Fall Time. Query only.

**EYE:MVALue:JITTer?**

Retrieve the Jitter P-P. Query only.

The return value is a number in the format that is determined by the Jitter P-P measurement value format. If the format is in TIME, the value is in picoseconds (ps). If the format is in %UI, the value is in percentage of a Unit Interval. The format can be queried and set using the EYE:JPFormat command in the Eye: Enable Eye Measurement Overlay section.

**EYE:MVALue:JRMS?**

Retrieve the Eye Jitter RMS measurement value. Query only.

The return value is a number in the format that is determined by the Jitter RMS measurement value format. If the format is in TIME, the value is in picoseconds (ps). If the format is in %UI, the value is in percentage of a Unit Interval. The format can be queried or set using the EYE:JRFormat command in the Eye: Enable Eye Measurement Overlay section.

**EYE:MVALue:NLEVel?**

Retrieve the maximum Noise Level. Query only.

**EYE:MVALue:NLV0?**

Retrieve Noise Level 0. Query only.

**EYE:MVALue:NLV1?**

Retrieve Noise Level 1. Query only.

**EYE:MVALue:OMA?**

Retrieve the OMA. Query only.

**EYE:MVALue:OverSHoot0?**

Retrieve the Overshoot0 measurement. Query only.

**EYE:MVALue:OverSHoot1?**

Retrieve the Overshoot1 measurement. Query only.

**EYE:MVALue:RLCRoss?**

Queries the Rising Level Crossing in ps.

This is the point in time at which the rising edge (the same edge used for the Rise Time measurement) crosses a configurable level crossing threshold in mV or  $\mu$ W (depending on the mode). Either the 80/20 or the 90/10 points of the rising edge are used, based on the configuration.

See EYE:LCRThresh to set the level crossing threshold and EYE:EdgeMODE <8020 | 9010> to set the amount of edge to use.

**EYE:MVALue:RTIME?**

Retrieve the Rise Time. Query only.

**EYE:MVALue:SCNT?**

Retrieve the Sample Count. Query only.

**EYE:MVALue:UINterval?**

Retrieve the Unit Interval. Query only.

**EYE:MVALue:VECP?**

Retrieve VECP. Query only.

**EYE:MVALue:VMAX?**

Retrieve Voltage Maximum. Query only.

**EYE:MVALue:VMIN?**

Retrieve Voltage Minimum. Query only.

**EYE:MVALue:VOFS?**

Retrieve the Eye Voffset value. Query only.

**EYE:MVALue:VPP?**

Retrieve Voltage Peak-to-Peak. Query only.

## BER Contour

Complete command listing (See page 7, *Command listing*.)

**CONTour:BBER?**

Retrieve BER Contour Best BER. Query only.

**CONTour:BDELay?**

Retrieve BER Contour Best Delay in ps. Query only.

**CONTour:BTHReshold?**

Retrieve BER Contour Best Threshold in mV. Query only.

**CONTour:CBPPf <UW | DBM>****CONTour:CBPPf?**

**BERTScope Analyzer only**, Optical mode. Set or retrieve Contour Best Point Threshold power format.

Params:	<UW>	Value displayed in $\mu$ W
	< DBM >	Value displayed in dBm
Returns:	< UW   DBM >	

**CONTour:CEHPf <UW | DBM>**

**CONTour:CEHPf?**

BERTScope Analyzer only, Optical mode. Set or retrieve Contour Eye Height power format.

Params:	<UW>	Value displayed in $\mu$ W
	< DBM >	Value displayed in dBm
Returns:	< UW   DBM >	

**CONTour:CERate:EM<6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16>?**

Retrieve specified Contour's Error Rate. Contour is is EM6, EM7, EM8, ..., EM16. Query only.

---

**NOTE.** Note there is no space before the contour number.

---

Params:	<6   7   8   9   10   11   12   13   14   15   16>	Contour is EM6, EM7, EM8, ..., EM16
Returns:	< EMErrRate >	Requested Contour error rate

**CONTour:CFG?**

Retrieve four configuration parameters for the current BER Contour, including: Unit Interval, Amplitude, Center Time, Center Threshold Voltage. Query only.

Returns:	< UI UA CT CV >	UI is unit interval in ps UA is amplitude in mV CT is center time in ps CV is center threshold voltage in mV.
----------	-----------------	--

Example: contour:cfg?  
93.379 357 167.335 1263

**CONTour:CLear**

Clear the BER Contour image and measurements. Action only.

**CONTour:CMFName <"filename">**

**CONTour:CMFName?**

Set or retrieve the BER Contour Mask filename.

Params:	<"filename">	MSK file to load. Filename is enclosed in double quotes.
Returns:	< filename >	



**CONTour:CMResult?**

Retrieve the Compliance Mask test result as “Pass” or “Fail”. Query only.

**CONTour:CStatus?**

Retrieve BER Contour Status. Query only.

Returns:	NOTRUNNING OK NOCLOCK LOWCLOCK NOPATTERN NOTFOUND ACQERROR	Status codes
----------	--	--------------

---

**CONTour:ECONtour:EM<6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16> <bool>  
CONTour:ECONtour:EM<6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16>?**

Enable or retrieve state of specified Contour. Note that there is *no space before* the contour number.

Params:	<6   7   8   9   10   11   12   13   14   15   16>	Contour is EM6, EM7, EM8, ..., EM16
	<bool>	On = 1, Off = 0
Returns:	< 6   7   8   9   10   11   12   13   14   15   16 > < 0   1 >	

**CONTour:ECTFile <layer> <“filename”>**

Export Contour to file. Action only.

Params:	<layer>	Layer is EM6, EM7, EM8, ..., EM16
	<“filename”>	Filename is “full path, name and extension” enclosed in double quotes.

---

**CONTour:EMContour <EM6 | EM7 | EM8 | EM9 | EM10 | EM11 | EM12 |  
EM13 | EM14 | EM15 | EM16>**
**CONTour:EMContour?**

Enable or retrieve state of Eye Measure Contour.

Params:	<EM6   EM7   EM8   EM9   EM10   EM11   EM12   EM13   EM14   EM15   EM16>	Enable contour, selecting from EM6, EM7, EM8, ..., EM16
Returns:	< EM6   EM7   EM8   EM9   EM10   EM11   EM12   EM13   EM14   EM15   EM16 >	

**CONTour:EMHeight?**

Retrieve BER Contour Eye Measure Height in mV. Query only.

**CONTour:EMWidth?**

Retrieve BER Contour Eye Measure Width in ps. Query only.

**CONTour:ERRThr <double>**

**CONTour:ERRThr?**

Set or retrieve BER Contour Jitter Error Threshold.

Params:	<double>	Range [1 to 10,000]
Returns:	< double >	

**CONTour:IMEasurement?**

Retrieve BER Contour Invalid Measurement. Query only.

**CONTour:MBARsel <OFF | MEASUREMENTS | STRESS | ADVANCED | LAYERS>**

**CONTour:MBARsel?**

Set or retrieve the BER Contour Sidebar selection.

Params:	<OFF>	No Sidebar
	<MEASUREMENTS>	Show Measurement Sidebar
	<STRESS>	Show Stress Sidebar (if Stress Option is enabled)
	<ADVANCED>	Show Advanced features Sidebar
	<LAYERS>	Show Contour layers Sidebar
Returns:	< OFF   MEASUREMENTS   STRESS   ADVANCED   LAYERS >	

**CONTour:MBDepth?**

Measure BER depth. Query only.

**CONTour:MERate <double>**

**CONTour:MERate?**

Set or retrieve BER Contour Mask Error Rate.

Params:	<double>	Range [1E-18 to 1E-02]
Returns:	< double >	

**CONTour:MFILE <“filename”>  
CONTour:MFILE?**

Saves current BER Contour to a Mask File. An error will result if BER Contour has not been run yet.

Params:	<“filename”>	MSK file. Filename is enclosed in double quotes.
Returns:	< filename >	Last saved filename of the BER Contour mask file

**CONTour:PTS <layered>?**

Returns the X,Y coordinates of a specified contour produced by the BER Contour view. The contour must be available when queried; otherwise a SERVER ERROR will be returned. Query only.

Params:	<layered>	Select contour layer from EM6, EM7, EM8, ..., EM16
Returns:	< N (ps1, mv1) (ps2, mv2), ..., (psN, mvN) >	N is the number of points in the contour; ps1 is time of the first point in ps; mv1 is the voltage of the first point in mV. The response is generated on one line of text output. The series of points can be considered a 'closed polygon,' where the first point can be connected to the last point.

Example: contour:points em12?  
32 (167.000, 1106.404) (159.430, 1104.830) (151.122, 1107.907) ...

**CONTour:REFMode <LIVEData | DPATtern>**

**CONTour:REFMode?**

Set or retrieve BER Contour Reference mode.

Params:	<LIVEData>	Live Data mode
	<DPATtern>	Detected Pattern mode
Returns:	< LIVEDATA   DPATTERN >	

**CONTour:REFPattern?**

Retrieve BER Reference Pattern. Query only.

**CONTour:RETime?**

Retrieve BER Contour Run Elapsed Time in seconds. Query only.

**CONTour:STFile<"filename">**

Save Contour results to named file. Action only.

Params:	<"filename">	CSV file where Contour analysis results will be saved. Filename is enclosed in double quotes.
---------	--------------	---

**CONTour:SUFBits <value>**

**CONTour:SUFBits?**

Set or retrieve BER Contour Sufficient Bits.

Params:	<value>	Range [1,000 to 1E+11]. A value of -1 initiates automatic mode.
---------	---------	---

Returns:	< value >	Numeric Value or Automatic
----------	-----------	----------------------------

## BER Contour: Optical Mode

Complete command listing (See page 7, *Command listing*.)

Optical Mode commands apply only to BERTScope Analyzer.

---

**NOTE.** In Electrical mode, choose between *mV* or *percent of amplitude* for *Eye Height*, *Crossing Voltage*, *Contour Eye Height* measurements. In Optical mode, choose between *μW* or *dBm*.

---

**CONTour:CBPPf <UW | DBM>**

**CONTour:CBPPf?**

Set or retrieve BER Contour Optical Mode Best Point Threshold power format.

Params:	<UW>	Value displayed in μW
	<DBM>	Value displayed in dBm
Returns:	< UW   DBM >	

**CONTour:CEHPf <UW | DBM>**

**CONTour:CEHPf?**

Set or retrieve BER Contour Optical Mode Eye Height power format.

Params:	<UW>	Value displayed in μW
	<DBM>	Value displayed in dBm
Returns:	< UW   DBM >	

# Jitter Map

Complete command listing (See page 7, *Command listing*.)

## **JMAP:AVGTT?**

Retrieve the average transition time. Query only.

Returns:	<numeric>	Average transition time in %UI, mUI, or ps
----------	-----------	--

## **JMAP:BUJ?**

Retrieve the BUJ. Measurement value. Query only.

Returns:	<numeric>	BUJ in %UI, mUI, or ps
----------	-----------	------------------------

## **JMAP:BUJLOCKED JMAP:BUJLOCKED?**

Set or retrieve the BUJ locked measurement value in units specified by JUNITS.

Returns:	<numeric>	BUJ in %UI, mUI, or ps
----------	-----------	------------------------

## **JMAP\_BUJ:MBARSEL<OFF | MEASUREMENTS | GENERATOR | STRESS | CLOCKRECOVERY> JMAP\_BUJ:MBARSEL?**

Set or retrieve the Sidebar state for the JitterMap BUJ view.

Params:	<OFF>	Hide Sidebar
	<MEASUREMENTS>	Show Measurement Sidebar
	<GENERATOR>	Show Generator Sidebar
	<STRESS>	Show Stress Sidebar (if Stress Option is enabled)
	<CLOCKRECOVERY>	Show Clock Recovery Sidebar (if Clock Recovery unit is detected)

Returns:	< OFF   MEASUREM ENTS   GENERATOR   STRESS   CLOCKRECOVERY >
----------	--

## **JMAP:CYCLES?**

Retrieve the number of measurement cycles completed during a run (RJPJ + DDJ + SRJ / cycle). Query only.

Returns:	<numeric>	Measurement cycles.
----------	-----------	---------------------



**JMAP\_EJ:MBARSEL<OFF | MEASUREMENTS | GENERATOR | STRESS | CLOCKRECOVERY>  
JMAP\_EJ:MBARSEL?**

Set or retrieve the Sidebar state for the JitterMap EJ view.

Params:	<OFF>	Hide Sidebar
	<MEASUREMENTS>	Show Measurement Sidebar
	<GENERATOR>	Show Generator Sidebar
	<STRESS>	Show Stress Sidebar (if Stress Option is enabled)
	<CLOCKRECOVERY>	Show Clock Recovery Sidebar (if Clock Recovery unit is detected)
Returns:	< OFF   MEASUREMENTS   GENERATOR   STRESS   CLOCKRECOVERY >	

**JMAP:EJTROF?**

Retrieve Emphasis Jitter Transition Offset measurement value in units specified by JUNITS. Query only.

Returns:	<numeric>	Emphasis Jitter Transition Offset in %UI, mUI, or ps.
----------	-----------	---

**JMAP:EMPRATMODE<OFF | MEASUREMENTS | GENERATOR | STRESS | CLOCKRECOVERY>  
JMAP:EMPRATMODE?**

Set or retrieve the emphasis ratio mode.

**NOTE.** *If you select MANUAL, use JMAP:EMPRAT to set or retrieve the value.*

Params:	y	Use Auto-emphasis ratio
	<MANUAL>	Allow manual ratio entry
Returns:	< AUTO   MANUAL >	



---

**JMAP\_ETB:MBARSEL<OFF | MEASUREMENTS | GENERATOR | STRESS | CLOCKRECOVERY>  
JMAP\_ETB:MBARSEL?**

Set or retrieve the Sidebar state for the JitterMap Edge Time Per Bit view.

Params:	<OFF>	Hide Sidebar
	<MEASUREMENTS>	Show Measurement Sidebar
	<GENERATOR>	Show Generator Sidebar
	<STRESS>	Show Stress Sidebar (if Stress Option is enabled)
	<CLOCKRECOVERY>	Show Clock Recovery Sidebar (if Clock Recovery unit is detected)
Returns:	< OFF   MEASUREMENTS   GENERATOR   STRESS   CLOCKRECOVERY >	

---

**JMAP\_ETH:MBARSEL <OFF | MEASUREMENTS | GENERATOR | STRESS | CLOCKRECOVERY>**

**JMAP\_ETH:MBARSEL?**

Set or retrieve the Sidebar state for the JitterMap Edge Timings histogram view.

Params:	<OFF>	Hide Sidebar
	<MEASUREMENTS>	Show Measurement Sidebar
	<GENERATOR>	Show Generator Sidebar
	<STRESS>	Show Stress Sidebar (if the option is enabled)
	<CLOCKRECOVERY>	Show Clock Recovery Sidebar (if Clock Recovery unit is detected)
Returns:	< OFF   MEASUREMENTS   GENERATOR   STRESS   CLOCKRECOVERY >	

---

**JMAP:F2J?**

Retrieve F2 Jitter measurement value in the units specified by JUNITS. Query only.

Returns:	<numeric>	F2 Jitter in %UI, mUI, or ps.
----------	-----------	-------------------------------

---

**JMAP:INVALIDVALUE?**

Retrieve the "magic number" returned if a JitterMap measurement cannot be made. Query only.

Returns:	<numeric>	Invalid value.
----------	-----------	----------------

---

**JMAP:ISI?**

Retrieve the Inter-Symbol Interference measurement value in the units specified by JUNITS. Query only.

Returns: <numeric> Inter-Symbol Interference in %UI, mUI, or ps.

---

**JMAP:JUNITS<PICOSECS | PERCENTUI | MILLIUI>  
JMAP:JUNITS?**

Set or retrieve the units in which the jitter measurements are reported and displayed.

Params:	<PICOSECS>	Returns jitter in picoseconds
	<PERCENTUI>	Returns jitter in %UI
	<MILLIUI>	Returns jitter in mUI

Returns: < PICOSECS | PERCENTUI | MILLIUI >

---

**JMAP:MANUALTHRESH  
JMAP:MANUALTHRESH?**

Set or retrieve the value used if JMAP:THRESHMODE is set to MANUAL

Params:	<numeric>	In mVolts if Detector is in Electrical mode, in uWatts if in Optical mode
---------	-----------	---

Returns: <numeric>

---

**JMAP\_MAP:MBARSEL<OFF | MEASUREMENTS | GENERATOR | STRESS | CLOCKRECOVERY>  
JMAP\_MAP:MBARSEL?**

Set or retrieve the Sidebar state for the JitterMap Map view.

Params:	<OFF>	Hide Sidebar
	<MEASUREMENTS>	Show Measurement Sidebar
	<GENERATOR>	Show Generator Sidebar
	<STRESS>	Show Stress Sidebar (if Stress Option is enabled)
	<CLOCKRECOVERY>	Show Clock Recovery Sidebar (if Clock Recovery unit is detected)

Returns: < OFF | MEASUREMENTS | GENERATOR | STRESS | CLOCKRECOVERY >

---

**JMAP:MEASURE\_EJ<bool>**  
**JMAP:MEASURE\_EJ?**

Set or retrieve the ON /OFF state of the EJ measurement category.

Params:	<bool>	On = 1, Off = 0
Returns:	< 1 >	Include EJ measurements in the run
	< 0 >	Do NOT include EJ measurements in the run

**JMAP:MEASURE\_SRJ<bool>**  
**JMAP:MEASURE\_SRJ?**

Set or retrieve the ON /OFF state of the SRJ measurement category.

Params:	<bool>	On = 1, Off = 0
Returns:	< 1 >	Include SRJ/NSR measurements in the run
	< 0 >	Do NOT include SRJ/NSR measurements in the run

**JMAP:NISI?**

Retrieve the Non-ISI (TJ minus ISI) measurement value in the units specified by JUNITS. Query only.

Returns:	<numeric>	Non-ISI, in %UI, mUI, or ps
----------	-----------	-----------------------------

**JMAP:NSR?**

Retrieve the Non-Subrate Jitter measurement value in the units specified by JUNITS. Query only.

Returns:	<numeric>	Non-Subrate Jitter, in %UI, mUI, or ps
----------	-----------	--

**JMAP:NUMEDGES?**

Retrieve the number of edges measured by RJ. Query only.

Returns:	<numeric>	Number of edges
----------	-----------	-----------------

**JMAP:PATLEN<numeric>**  
**JMAP:PATLEN?**

Set or retrieve the value used if JMAP:PATLENMODE = 'MANUAL'.

Params:	<numeric>	User pattern length, in bits
Returns:	<numeric>	

**JMAP:PATLENMODE<FROMFILE | AUTO | MANUAL>  
JMAP:PATLENMODE?**

Set or retrieve the selection mode for the user pattern length.

NOTE: If 'MANUAL' is selected, use JMAP:PATLEN (above) to set or retrieve the value.

Params:	<FROMFILE>	Use the length specified in the currently selected user pattern file.
	<AUTO>	Determine the pattern length from the signal.
	<MANUAL>	Allow manual specification of the pattern length.
Returns:	< FROMFILE   AUTO   MANUAL >	

**JMAP:PPBER <double>  
JMAP:PPBER?**

Set or retrieve the BER where Jitter Map Total Jitter measurements are made.

Params:	<double>	Range [1.00E-50 to 1.00E-02.]
Returns:	< double >	

**JMAP:RconfigDISK<“filename”>**

Restores the BERTScope Analyzer JitterMap settings from a file. The complete path must be specified in the parameter string. The path must be accessible to the host computer or BERTScope Analyzer running the remote control software. The file extension should be .CJM. Action only.

Example:

JMAP\_MAP:RCONFIGDISK “D:\BitAnalyzer\Configurations\mycfg.CJM”

Params:	<“filename”>	Enclose string in double quotes.
---------	--------------	----------------------------------

**JMAP:RETIME?**

Retrieve the elapsed run time, in seconds. Query only.

Returns:	<numeric>	Elapsed run time in seconds
----------	-----------	-----------------------------

**JMAP:RJ?**

Retrieve the Random Jitter RMS measurement value in the units specified by JUNITS. Query only.

Returns:	<numeric>	Random Jitter, in %UI-rms, mUI-rms or ps-rms
----------	-----------	--

**JMAP:RJLOCKED**  
**JMAP:RJLOCKED?**

Sets/Retrieve the RJ locked measurement value in the units specified by JUNITS.

Returns: <numeric> RJ in %UI, mUI, or ps

---

**JMAP:RJAVERLY?**

Retrieve the average RJ for the early bits in the units specified by JUNITS. Query only.

Returns: <numeric> Random Jitter avg of early bits in %UI rms, mUI rms, or ps rms

---

**JMAP:RJAVERGFALL?**

Retrieve the average RJ for the falling transitions e in the units specified by JUNITS. Query only.

Returns: <numeric> Random Jitter average of falling transitions in %UI rms, mUI rms, or ps rms

---

**JMAP:RJAVERGLATE?**

Retrieve the average RJ for the late bits v in the units specified by JUNITS. Query only.

Returns: <numeric> Random Jitter for late bits in %UI rms, mUI rms, or ps rms.

---

**JMAP:RJAVERRISE?**

Retrieve the average RJ for the rising edges in the units specified by JUNITS. Query only.

Returns: <numeric> Random Jitter for rising bits in %UI rms, mUI rms, or ps rms.

---

**JMAP:RJDD?**

Retrieve the dirac-delta computed RJ dirac-delta in the units specified by JUNITS. Query only.

Returns: <numeric> Random Jitter for dirac-delta RJ in %UI rms, mUI rms, or ps rms.

---

**JMAP:RJEVEN?**

Retrieve the RJ for even bits in the units specified by JUNITS. Query only.

Returns: <numeric> Random Jitter for even bits in %UI rms, mUI rms, or ps rms.

---



**JMAP:RUNMODE <NORMAL | LONGPATTERNLOCK>**  
**JMAP:RUNMODE?**

Set or retrieve the run mode; Normal, or Long Pattern Lock.

Params:	<NORMAL>	Run in normal mode – long pattern lock off
	<LONGPATTERNLOCK>	Run with long pattern lock on
Returns:	< NORMAL   LONGPATTERNLOCK >	

**JMAP:SconfigDISK<“string”>**  
**JMAP:SDISK <“string”>**

Saves the current BERTScope Analyzer JitterMap settings to a file. The complete path must be specified in the parameter string. The path must be accessible to the host computer or BERTScope Analyzer running the remote control software. The file extension should be .CJM. Action only.

Example:

JMAP:SCONFIGDISK “D:\BitAlyzer\Configurations\mycfg.CJM”

Params:	<“string”>	The string is enclosed in double quotes.
---------	------------	--

**JMAP:SRJ?**

Retrieve the computed subrate jitter in the units specified by JUNITS. Query only. Query only.

Returns:	<numeric>	Subrate jitter in %UI-rms, mUI-rms, or ps-rms
----------	-----------	---

**JMAP\_SRJ:MBARSEL<OFF | MEASUREM ENTS | GENERATOR | STRESS | CLOCKRECOVERY>**  
**JMAP\_SRJ:MBARSEL?**

Set or retrieve the Sidebar state for the JitterMap SRJ view.

Params:	<OFF>	Hide Sidebar
	<MEASUREMENTS>	Show Measurement Sidebar
	<GENERATOR>	Show Generator Sidebar
	<STRESS>	Show Stress Sidebar (if Stress Option is enabled)
	<CLOCKRECOVERY>	Show Clock Recovery Sidebar (if Clock Recovery unit is detected)
Returns:	< OFF   MEASUREM ENTS   GENERATOR   STRESS   CLOCKRECOVERY >	

**JMAP:SUBRATESELECTED\_#**  
**JMAP:SUBRATESELECTED\_#?**

Set or retrieve the ON /OFF state of a subrate.

---

**NOTE.** Replace the # symbol with a valid Subrate: 2, 4, 5, 8, 10, 16, 20, 32, 40, 64, or 128.

---

Params:	#	Replace with a valid subrate
	<bool>	On = 1, Off = 0
Returns:	< 1 >	Include this Subrate in the SRJ measurement
	< 0 >	Do NOT include this Subrate in the SRJ measurement

---

**JMAP:THRESHMODE<enum>**  
**JMAP:THRESHMODE?**

Set or retrieve the threshold selection mode.

NOTE: If MANUAL is selected, use JMAP:MANUALTHRESH (above to set or retrieve value).

Params:	<enum>	Threshold selection mode
Returns:	< enum >	

---

**JMAP:TJ?**

Retrieve the computed TJ in the units specified by JUNITS. Query only.

Returns:	<numeric>	Total Jitter for TJ , in %UI-rms, mUI-rms, or ps-rms
----------	-----------	--

---

**JMAP:TJBERDPTH?**

Retrieve the TJ BER Depth. Query only.

Returns:	<numeric>	TJ BER Depth
----------	-----------	--------------

---



**JMAP\_TJ:MBARSEL<OFF | MEASUREMENTS | GENERATOR | STRESS | CLOCKRECOVERY>  
JMAP\_TJ:MBARSEL?**

Set or retrieve the Sidebar state for the JitterMap TJ view.

Params:	<OFF>	Hide Sidebar
	<MEASUREMENTS>	Show Measurement Sidebar
	<GENERATOR>	Show Generator Sidebar
	<STRESS>	Show Stress Sidebar (if Stress Option is enabled)
	<CLOCKRECOVERY>	Show Clock Recovery Sidebar (if Clock Recovery unit is detected)
Returns:	< OFF   MEASUREMENTS   GENERATOR   STRESS   CLOCKRECOVERY >	

**JMAP:TJSTARTBER  
JMAP:TJSTARTBER?**

Set or retrieve the starting BER value used during the TJ measurement.

Param:	<numeric>	BER from 1E-10 to 1E-2
Returns:	<numeric>	

**JMAP:UJ?**

Retrieve the computed UJ in the units specified by JUNITS. Query only.

Returns:	<numeric>	Uncorrelated Jitter in %UI rms, mUI rms, or ps rms
----------	-----------	--

**JMAP:UJBERDPH?**

Retrieve the UJ BER Depth. Query only.

Returns:	<numeric>	UJ BER Depth
----------	-----------	--------------

**JMAP:UJRMS?**

Retrieve the RMS UJ in the units specified by JUNITS. Query only.

Returns:	<numeric>	UJ rms in %UI rms, mUI rms, or ps rms
----------	-----------	---------------------------------------

**JMAP:UJSTARTBER  
JMAP:UJSTARTBER?**

Set or retrieve the starting BER value used during the UJ measurement.

Param:	<numeric>	BER from 1E-10 to 1E-2
Returns:	<numeric>	

**OPT\_JITTERMAP?**

Check to see if the instrument has JitterMap Option enabled. Query only.

Returns:	< 0 >	Jitter Map Option is enabled
	< 1 >	Jitter Map Option is disabled

## Jitter Peak

Complete command listing (See page 7, *Command listing*.)

**JITTer:AScale <LEGE | TEDGE | CTIME>**

Set Jitter Peak Auto Scale parameters.

Params:	<LEGE>	Leading Edge
	<TEDGE>	Trailing Edge
	<CTIME>	Center Time

**JITTer:BITPos <long>**

**JITTer:BITPos?**

Set or retrieve the bit position the Jitter Peak is to operate on.

Params:	<numeric>	Bit Position, a numeric value. Range [0 to (length of the input pattern 1)]
Returns:	<numeric>	

**JITTer:CFG?**

Retrieve the auto-align results used to automatically align the Jitter Peak. Query only.

Returns:	<UI UA CT CV>	UI is the Unit Interval in ps UA is amplitude in mV CT is center time in ps CV is center threshold voltage in mV.
----------	---------------	--

**JITTer:CLear**

Clear the Jitter Peak image and measurements. Action only.

**JITTer:DJITter?**

Retrieve the Deterministic Jitter. Query only. The return value is a number in the format that is determined by the Deterministic Jitter format of either TIME or %UI. See JITTer:DJFormat command above for querying and changing the format.

**JITTer:DMODE <COUNT | GRAPh | RDATA>**  
**JITTer:DMODE?**

Set or retrieve the Jitter Peak display mode.

Params:	<COUNT>	Counters
	<GRAPh>	Graph
	<RDATA>	Raw data
Returns:	< COUNT   GRAPH   RDATA >	

**JITTer:EGRaticule <bool>**  
**JITTer:EGRaticule?**

Enable/disable the graticule (grid) on the Jitter Peak display.

Params:	<bool>	On = 1, Off = 0
Returns:	< 1 >	Enable Jitter Peak graticule
	< 0 >	Disable Jitter Peak graticule

**JITTer:EQScale <bool>**  
**JITTer:EQScale?**

Enable/disable the Q Scale on the Jitter Peak display.

Params:	<bool>	On = 1, Off = 0
Returns:	< 1 >	Enable Q Scale.
	< 0 >	Disable Q Scale.

**JITTer:ERRThr <double>**  
**JITTer:ERRThr?**

Set or retrieve Jitter Peak Error Threshold.

Params:	<double>	Range [1 to 10,000].
Returns:	< double >	

**JITTer:IVALue?**

Retrieve Jitter Peak Invalid Value. Query only.

**JITTer:L0EValue?**

Retrieve Jitter Peak Leading Edge error value. Query only.

**JITTer:L0Mean?**

Retrieve Jitter Peak Leading Edge mean. Query only.

**JITTer:L0NPoints?**

Retrieve Jitter Peak Leading Edge number of measured points. Query only.

**JITTer:L0Sigma?**

Retrieve Jitter Peak Leading Edge sigma. Query only.

**JITTer:L1EValue?**

Retrieve Jitter Peak Trailing Edge error value. Query only.

**JITTer:L1Mean?**

Retrieve Jitter Peak Trailing Edge mean. Query only.

**JITTer:L1NPoints?**

Retrieve Jitter Peak Trailing Edge number of measured points. Query only.

**JITTer:L1Sigma?**

Retrieve Jitter Peak Trailing Edge sigma. Query only.

**JITTer:LData?**

Retrieve the measured points for the left edge of the Jitter Peak display. Query only.

Returns:            < N (x1, y1) (x2, y2) ... (xN, yN) >            where N is the number of points, the x values are delay values represented in ps, and the y values are BER measurements.

---

**JITTer:LFit?**

Retrieve the extrapolated points for the left edge of the Jitter Peak display. Query only.

Returns:            < N (x1, y1) (x2, y2) ... (xN, yN) >            where N is the number of points, the x values are delay values represented in ps, and the y values are BER measurements.

---

**JITTer:MBARsel <OFF | MEASUREMENTS | STRESS | ADVANCED >  
JITTer:MBARsel?**

Set or retrieve the Jitter Peak Sidebar selection.

Params:	<OFF>	Hide Sidebar
	<MEASUREMENTS>	Show Measurement Sidebar
	<STRESS>	Show Stress Sidebar (if Stress Option is enabled)
	<ADVANCED>	Show Advanced features Sidebar

Returns:            < OFF | MEASUREMENTS | STRESS | ADVANCED >

---

**JITTer:MBDepth?**

Measure Jitter Peak BER Depth. Query only.

**JITTer:OBString?**

Retrieve Jitter Peak Optimum BER String. Query only.

**JITTer:OLBer?**

Retrieve Jitter Peak Optimum Log BER. Query only.

**JITTer:OPERmode <ALLBITS | SINGLEBIT>****JITTer:OPERmode?**

Set or retrieve the Jitter Peak operating mode.

Params:	<ALLBITS>	Jitter Peak operates on all bits
	< SINGLEBIT >	Jitter Peak operates on the chosen single bit
Returns:	< ALLBITS   SINGLEBIT >	

**JITTer:OTValue?**

Retrieve Optimum Jitter Peak threshold value in mV. Query only.

**JITTer:PPBer <double>****JITTer:PPBer?**

Set or retrieve the BER where Jitter Peak measurements are made.

Params:	<double>	Range [1.00E-50 to 1.00E-05]
Returns:	< double >	

**JITTer:RData?**

Retrieve the measured points for the right edge of the Jitter Peak display.

Returns:	< N (x1, y1) (x2, y2) ... (xN, yN) >	where N is the number of points, the x values are delay values represented in ps, and the y values are BER measurements.
----------	--------------------------------------	--

**JITTer:REFMode <LIVEdata | DPATtern>****JITTer:REFMode?**

Set or retrieve Jitter Peak Reference mode.

Params:	<LIVEdata>	Live Data mode
	<DPATtern>	Detected Pattern mode
Returns:	< LIVEDATA   DPATTERN >	

**JITTer:REFPattern?**

Retrieve Jitter Peak Reference Pattern. Query only.

**JITTer:RETime?**

Retrieve Jitter Peak Run Elapsed Time in seconds. Query only.

**JITTer:RFit?**

Retrieve the measured points for the right edge of the Jitter Peak display.

Returns: < N (x1, y1) (x2, y2) ... (xN, yN) > where N is the number of points, the x values are delay values represented in ps, and the y values are BER measurements.

---

**JITTer:RJITter?**

Retrieve the Random Jitter. Query only.

The return value is a number in the format that is determined by the Random Jitter format of either TIME or %UI. See the JITTer:RJFormat command for querying or changing the format.

**JITTer:RJFormat <TIME | %UI>**

**JITTer:RJFormat?**

Set or retrieve the Jitter Peak Random Jitter measurement value format.

Params:	<TIME>	Value is displayed in psec
	<%UI>	Value is displayed in %UI
Returns:	< TIME   %UI >	

---

**JITTer:SBER<double>**

**JITTer:SBER?**

Set or retrieve the Starting BER for Jitter Peak measurements.

Params:	<double>	Range [1.00E-50. to 1.00E-05]
Returns:	< double >	

---

**JITTer:SCURsors<X1,X2,Y1,Y2>**

Set Jitter Peak Cursor positions. Action only.

Params:	<X1>	X1 cursor position, double
	<X2>	X2 cursor position, double
	<Y1>	Y1 cursor position, double
	<Y2>	Y2 cursor position, double
Returns:	< X1,X2,Y1,Y2 >	

---

**JITTer:SDATa? <int>**

Retrieve Jitter Peak Series Data, binary. Query only.

Returns:	<integer>	Range [0 to 14]
----------	-----------	-----------------

---

**JITTer:SDEFaults**

Set Defaults. Action only.

**JITTer:STFile<"filename">**

Save Jitter Peak results to named file. Action only.

Params:	<"filename">	File where Jitter Peak results are saved. Enclose filename in double quotes.
---------	--------------	---

---

**JITTer:SUFBits <double>****JITTer:SUFBits?**

Set or retrieve Jitter Peak Sufficient Bits.

Params:	<value>	Range [1,000 to 1E+11]. A value of 1 initiates automatic mode.
---------	---------	--

---

Returns:	< value >	Numeric Value or Automatic
----------	-----------	----------------------------

---

**JITTer:SVIew<Xmin, Ymin, Xmax, Ymax>**

Set Jitter Peak view axes.

Params:	<Xmin>	X-Axis minimum, double
---------	--------	------------------------

---

	<Ymin>	Y-Axis minimum, double
--	--------	------------------------

---

	<Xmax>	X-Axis maximum, double
--	--------	------------------------

---

	<Ymax>	Y-Axis maximum, double
--	--------	------------------------

---

Returns:	< Xmin, Ymin, Xmax, Ymax >
----------	----------------------------

---

**JITTer:TJITter?**

Retrieve the Total Jitter. Query only.

The return value is a number in the format that is determined by the Total Jitter format of either TIME or %UI. See JITTer:TJFormat command for querying and changing the format.

**JITTer:TJFormat <TIME | %UI>****JITTer:TJFormat?**

Set or retrieve the Jitter Peak Total Jitter measurement value format.

Params:	<TIME>	Value is displayed in psec
---------	--------	----------------------------

---

	<%UI>	Value is displayed in %UI
--	-------	---------------------------

---

Returns:	< TIME   %UI >
----------	----------------

---

**JITTer:THRMODe <AUTOMATIC | MANUAL>**  
**JITTer:THRMODe?**

Set or retrieve the threshold mode of the Jitter Peak.

Params:	<AUTOMATIC>	Jitter threshold mode is automatic
	< MANUAL >	Jitter threshold mode is manual
Returns:	< AUTOMATIC   MANUAL >	

**JITTer:TStatus?**

Retrieve Jitter Peak Status. Query only.

Returns:	NOTRUNNING OK NOCLOCK LOWCLOCK NOPATTERN NOTFOUND ACQERROR WAITING	Status codes
----------	---	--------------

**JITTer:TVALue <double>**  
**JITTer:TVALue?**

Set or retrieve Jitter Peak Manual Threshold.

Params:	<double>	Range [-3,000 to +3,000].
Returns:	< double >	

**JITTer:X1Cursor <double>**  
**JITTer:X1Cursor?**

Set or retrieve Jitter Peak View's X1 Cursor.

Params:	<double>	Position of Jitter Peak Cursor X1
Returns:	< double >	

**JITTer:X2Cursor <double>**  
**JITTer:X2Cursor?**

Set or retrieve Jitter Peak View's X2 Cursor.

Params:	<double>	Position of Jitter Peak Cursor X2
Returns:	< double >	



**JITTer:XMAX <double>**  
**JITTer:XMAX?**

Set or retrieve Jitter Peak View's X Maximum value.

Params:	<double>	Jitter Peak X Maximum value
Returns:	< double >	

**JITTer:XMIN <double>**  
**JITTer:XMIN?**

Set or retrieve Jitter Peak View's X Minimum value.

Params:	<double>	Jitter Peak X Minimum value
Returns:	< double >	

**JITTer:Y1Cursor <double>**  
**JITTer:Y1Cursor?**

Set or retrieve position of Jitter Peak's Cursor Y1.

Params:	<double>	Position of Jitter Peak Cursor Y1
Returns:	< double >	

**JITTer:Y2Cursor <double>**  
**JITTer:Y2Cursor?**

Set or retrieve position of Jitter Peak's Cursor Y2.

Params:	<double>	Position of Jitter Peak Cursor Y2
Returns:	< double >	

**JITTer:YMAX <double>**  
**JITTer:YMAX?**

Set or retrieve Jitter Peak View's Y Maximum value.

Params:	<double>	Jitter Peak Y Maximum value
Returns:	< double >	

**JITTer:YMIN <double>**  
**JITTer:YMIN?**

Set or retrieve Jitter Peak View's Y Minimum value.

Params:	<double>	Jitter Peak Y Minimum value
Returns:	< double >	

# Jitter Tolerance

Complete command listing (See page 7, *Command listing*.)

**JTOL:BERThr <double>**  
**JTOL:BERThr?**

Set or retrieve BER Threshold.

Params:	<double>	Range [1E-20. to 1E-7.]
Returns:	<double>	

**JTOL:CHCOrd<XLOGOFS | XLOGEXT | YLOGOFS | YLOGEXT>?**

Retrieve the chart coordinates. Query Only.

Params:	<XLOGOFS>	Retrieve X Offset from the chart
	<XLOGEXT>	Retrieve X Extent from the chart
	<YLOGOFS>	Retrieve Y Offset from the chart
	<YLOGEXT>	Retrieve Y Extent from the chart
Returns:	<double>	

**JTOL:CHRX <double>**  
**JTOL:CHRX?**

Set or retrieve Jitter Tolerance Cursor X position.

Params:	<double>	Position of X Cursor in Chart
Returns:	<double>	

**JTOL:CHRY <double>**  
**JTOL:CHRY?**

Set or retrieve Jitter Tolerance Cursor Y position.

Params:	<double>	Position of Y Cursor in Chart
Returns:	<double>	

**JTOL:CONFdet?**

Retrieve the configuration parameters of the current Jitter tolerance template.  
 Query only.

Returns:	< TC(FreqMHz,TemplateUI,Attributes,LimitUI )
----------	--

Example:

2(0.1, 0.5, 0, 0) (0.2, 0.5, 0, 0)

**JTOL:DIFLg <SJLIMIT | CURSOR, bool>  
JTOL:DIFLg<SJLIMIT | CURSOR>?**

Shows or hides different cursors on the chart.

Params:	<SJLIMIT, bool>	Shows the SJ limit in the Chart.
	<CURSOR, bool>	Shows the Cursors in the Chart
	<bool>	On = 1, Off = 0
Returns:	< 1 >	SJ Limit/Cursors On
	< 0 >	SJ Limit/Cursors Off

**JTOL:DIFMt <CHART | TABLE>  
JTOL:DIFMt?**

Set or retrieve display format.

Params:	<CHART>	Displays as a Chart
	<TABLE>	Displays as a Table
Returns:	< CHART   TABLE >	

**JTOL:ELPSt?**

Retrieve elapsed time. Query only.

Returns:	<double>	Elapsed time
----------	----------	--------------

**JTOL:EXPCsv<"filename">**

Export Jitter Tolerance results to a CSV file. Full path name should be given. Action only.

Params:	<"filename">	Filename in quotes, include full path.
---------	--------------	--

**JTOL:LDTPI<"filename">**

Load a template file. Full path name should be given. Action only.

Params:	<"filename">	Filename in quotes, include full path.
---------	--------------	--

**JTOL:MARGui <double>  
JTOL:MARGui?**

Set or retrieve Margin UI in percentage.

Params:	<double>	Range [-90 to 400]
Returns:	<double>	

**JTOL:MBARsel <OFF | GENERATOR | STRESS | CRU>  
JTOL:MBARsel?**

Set or retrieve Jitter Tolerance Sidebar selection.

Params:	<OFF>	Hide Sidebar
	<GENERATOR>	Show Generator Sidebar
	<STRESS>	Show Stress Sidebar (if Stress Option is enabled)
	<CRU>	Show Clock Recovery Sidebar (If Clock Recovery unit is detected)
Returns:	< OFF   GENERATOR   STRESS   CRU >	

**JTOL:PRECui <double>  
JTOL:PRECui?**

Set or retrieve Precision UI.

Params:	<double>	Range [0.005 to 0.05]
Returns:	<double>	

**JTOL:RelaxFLIP <bool>  
JTOL:RelaxFLIP?**

Enable/disable the relax with Pattern Generator RAM page flip.

Params:	<bool>	On = 1, Off = 0
Returns:	< 1 >	Relax is enabled
	< 0 >	Relax is disabled

**JTOL:RelaxMODE<OFF | ABSWITCH | ONESHOT >  
JTOL:RelaxMODE?**

Set or retrieve the Relax Mode.

Params:	<OFF>	Turn PageSelect off
	<ABSWITCH>	Specify Relax mode to be Page A or Page B
	<ONESHOT>	Specify Relax mode to switch to the other page for a moment and then switch back
Returns:	< OFF   ABSWITCH   ONESHOT >	

**JTOL:RELSt?**

Retrieve Relax State. Query only.

Returns:	< NORELAX   RELAX   NOTRUNNING >
----------	----------------------------------

**JTOL:RELTime <double>****JTOL:RELTime?**

Set or retrieve Relax Time in seconds.

Params:	<double>	Range [0 to 100]
Returns:	<double>	

**JTOL:RESDet?**

Retrieve the Jitter Tolerance test results. Query only.

Returns:	<TC(AmpUI, Bits, Errors, BER, Status)>
	0 = TM_Blank
	1 = TM_Skipped
	2 = TM_InProgress
	3 = TM_Passed
	4 = TM_NoSync
	5 = TM_FAIL_BER
	6 = TM_FAIL_CLKERR
	7 = TM_FAIL_DATAERR
	8 = TM_LIMIT_REACHED

Example:

2(0.1, 1000, 100, 0.1, 1) (0.2, 1000, 100, 0.1, 2)

**JTOL:RUNSt?**

Retrieve Jitter Tolerance Run Status. Query only.

Returns:	0 = TM_READY
	1 = TM_SWEEPING
	2 = TM_TESTING
	3 = TM_RELAXING
	4 = TM_AUTOALIGNING
	5 = TM_TEST_PASS
	6 = TM_TEST_FAIL
	7 = TM_RSTATUS_COUNT

**JTOL:SJBAs <double>****JTOL:SJBAs?**

Set or retrieve SJ Baseline Amplitude UI in percentage.

Params:	<double>	Range [0 to 100]
Returns:	<double>	

**JTOL:SVTPI<“filename”>**

Save the current template in a file. Full path name should be given. Action only.

Params:	<“filename”>	Enclose filename in double quotes and include full path.
---------	--------------	--

---

**JTOL:TEND <DURATION | BITSERRS | CONFIDENCE>  
JTOL:TEND?**

Set or retrieve Test End Mode.

Params:	<DURATION>	Specify Test end mode in Seconds
	<BITSERRS>	Specify Test end mode in Bits and Errors.
	<CONFIDENCE>	Specify Test end mode in Confidence Percentage

---

Returns: < DURATION | BITSERRS | CONFIDENCE >

---

**JTOL:TEVAI <DURATION | BITS | ERRORS | CONFIDENCE, double>  
JTOL:TEVAI<DURATION | BITS | ERRORS | CONFIDENCE>?**

Set or retrieve Test End Value.

Params:	<DURATION, double>	Set duration in seconds. Range [0 to 5000]
	<BITS, double>	Set bits as numeric. Range [1E8 to 1E14]
	<ERRORS, double>	Set errors as numeric. Range [0 to 100]
	<CONFIDENCE, double>	Set confidence in percentage. Range [1 to 99.99]

---

Returns: < double >

---

**JTOL:TMODe <MARGIN | SEARCH | CONTOUR>  
JTOL:TMODe?**

Set or retrieve Test mode.

Params:	<MARGIN>	Margin mode
	<SEARCH>	Search mode
	<CONTOUR>	Contour mode

---

Returns: < MARGIN | SEARCH | CONTOUR >

---

**JTOL:ToIAlG <STEP | FMTOP | FMBOT | BINARY | FMBOTFINE>  
JTOL:ToIAlG?**

Set or retrieve the algorithm type used in Search and Contour modes.

Params:	<STEP>	Refine step algorithm
	<FMTOP>	From top algorithm
	<FMBOT>	From bottom algorithm
	<BINARY>	Binary search algorithm
	<FMBOTFINE>	From bottom fine algorithm
Returns:	< STEP   FMTOP   FMBOT   BINARY   FMBOTFINE >	

## Mask Test

Complete command listing (See page 7, *Command listing*.)

**MASK:ATOffset <double>  
MASK:ATOffset?**

Adjust Mask Test time offset in ps.

Params:	<double>	Mask Test time offset in ps
Returns:	< double >	

**MASK:AVOffset <double>  
MASK:AVOffset?**

Adjust Mask Test voltage offset in millivolts.

Params:	<double>	Mask Test voltage offset in mV
Returns:	< double >	

**MASK:CLear**

Clear Mask. Action only.

**MASK:CPERrors?**

Retrieve Mask Test Center Polygon Errors. Query only.

**MASK:EMLayer:COORDs <bool>  
MASK:EMLayer:COORDs?**

Enable/disable the Mask Test Coordinates layer.

Params:	<bool>	On = 1, Off = 0
Returns:	< 1 >	Mask Test coordinates layer is enabled
	< 0 >	Mask Test coordinates layer is disabled

**MASK:EMLayer:LOAD <bool>**  
**MASK:EMLayer:LOAD?**

Enable/disable Mask Test Layer.

Params:	<bool>	On = 1, Off = 0
Returns:	< 1 >	Mask Test layer is enabled
	< 0 >	Mask Test layer is disabled

**MASK:EMLayer:PROGress <bool>**  
**MASK:EMLayer:PROGress?**

Enable/disable the Mask Test Progress layer.

Params:	<bool>	On = 1, Off = 0
Returns:	< 1 >	Mask Test progress layer is enabled
	< 0 >	Mask Test progress layer is disabled

**MASK:EMLayer:RESults <bool>**  
**MASK:EMLayer:RESults?**

Enable/disable the Mask Test Results layer.

Params:	<bool>	On = 1, Off = 0
Returns:	< 1 >	Mask Test results layer is enabled
	< 0 >	Mask Test results layer is disabled

**MASK:IVALue?**

Retrieve Mask Test Invalid Value. Query only.

**MASK:LMASK**

Load Mask. Action only. To specify a filename, see the MASK:MFName command.

**MASK:LPERrors?**

Retrieve Mask Test Lower Polygon Errors. Query only.

**MASK:MBARsel <OFF | MEASUREMENTS | STRESS | ADVANCED >**  
**MASK:MBARsel?**

Set or retrieve the Mask Test Sidebar selection.

Params:	<OFF>	Hide Sidebar
	<MEASUREMENTS>	Show Measurement Sidebar
	<STRESS>	Show Stress Sidebar (if Stress Option is enabled)
	<ADVANCED>	Show Advanced features Sidebar
Returns:	< OFF   MEASUREMENTS   STRESS   ADVANCED >	



**MASK:MFName <"filename">**  
**MASK:MFName?**

Set or retrieve Mask Filename.

Params: <"filename"> Mask filename, enclosed in double quotes.  
 To load the mask file, see the MASK:LMASK command.

---

Returns: <"filename" >

---

**MASK:MLCount?**

Retrieve Mask Layer Count. Query only.

**MASK:MStatus?**

Retrieve Mask Test Status. Query only.

Returns: NOTRUNNING Status codes  
 OK  
 NOCLOCK  
 LOWCLOCK  
 DROPCLOCK  
 FREQCHANGE  
 NOPATTERN

---

**MASK:REFMode <LIVEdata | DPATtern>**  
**MASK:REFMode?**

Set or retrieve Mask Reference mode.

Params: <LIVEdata> Live Data mode  
 <DPATtern> Detected Pattern mode

---

Returns: < LIVEDATA | DPATTERN >

---

**MASK:REFPattern?**

Retrieve Mask Test Reference Pattern. Query only.

**MASK:RETime?**

Retrieve Mask Test Run Elapsed Time in Seconds. Query only.

**MASK:RSTate <bool>**  
**MASK:RSTate?**

Set or retrieve Mask Test Run State.

Params: <bool> On = 1, Off = 0

---

Returns: < 1 > Run  
 < 0 > Stop

---

**MASK:SCOunt?**

Retrieve Mask Test Sample Count. Query only.

**MASK:SDEPth <numeric>****MASK:SDEPth?**

Set or retrieve the Mask test Sample Depth in bits.

Params:           <numeric>                           Range [2,000 to 10,000,000]. Input out of range will be clipped and recorded in the status queue.  
A value of -1 runs the Eye diagram in the 'Auto' mode with a sample depth of 10,000 bits.  
Setting a value other than -1 runs the Mask test in the Manual mode. Once the specified depth is reached, the Mask testing automatically stops running.

---

Returns:           <numeric>

---

**MASK:SLData? <int>**

Retrieve Mask Test Sketch Layer Data (binary)

Params:           <integer>                           Range [0 to 5]

---

**MASK:SMWadjustments<“filename”>**

Save Mask with adjustments. Action only.

Params:           <“filename”>                           Mask filename, enclosed in double quotes.

---

**MASK:STMArgin <double>**  
**MASK:STMArgin?**

Set or retrieve the Mask Time margin.

Params: <double> Range [-90% to +100%UI]

Returns: < double >

**MASK:STVMArgin <double>**  
**MASK:STVMArgin?**

Set or retrieve Mask Test Time and Voltage margin.

Params: <double> Time and voltage margin as %UI. Range [-90% to +100%UI]

Returns: < double >

**MASK:SVMArgin <double>**  
**MASK:SVMArgin?**

Set or retrieve Mask Test Voltage margin.

Params: <double> Range [-90% to +100%UI]

Returns: < double >

**MASK:TMARgin <double>**  
**MASK:TMARgin?**

Set or retrieve Mask Test Time margin

Params: <double> Range [-90 to +100]

Returns: < double >

**MASK:TVMARgin <double>**  
**MASK:TVMARgin?**

Set or retrieve Mask Test Time and Voltage margin

Params: <double> Range [-90 to +100], default = 0

Returns: < double >

**MASK:UAMPlitude <double>**  
**MASK:UAMPlitude?**

Set or retrieve Mask Test Unit Amplitude in millivolts.

Params: <double> Mask Test Unit Amplitude in mV

Returns: < double >

**MASK:UINterval <double>**  
**MASK:UINterval?**

Set or retrieve Mask Test Unit Interval in ps.

Params:	<double>	Mask Test Unit Interval in ps
Returns:	< double >	

---

**MASK:UPERrors?**

Retrieve Mask Test Upper Polygon errors. Query only.

**MASK:VMARgin <double>**  
**MASK:VMARgin?**

Set or retrieve Mask Test voltage margin a percentage of Unit Amplitude

Params:	<double>	Range [-90 to +100]
Returns:	< double >	

---

**MASK:WFORms?**

Retrieve the number of waveforms tested in the Mask Test. Query only.

## Q-Factor

Complete command listing (See page 7, *Command listing*.)

### **QFACTOR:AScale** <LEV0 | LEV1 | MIDLev>

Auto Scale the Q-Factor display to fit data. Action only.

Params:	<LEV0>	0-Level
	<LEV1>	1-Level
	<MIDLev>	Middle (logic threshold) level

### **QFACTOR:CFG?**

Retrieve the auto-align results used to automatically align the Q-Factor analysis. Query only.

Returns:	< UI UA CT CV >	UI is the Unit Interval in ps UA is amplitude in mV CT is center time in ps CV is center threshold voltage in mV
----------	-----------------	---

### **QFACTOR:CLEar**

Clear the Q-Factor image and measurements. Action only.

### **QFACTOR:DLYMODE** <AUTOMatic | MANUal>

#### **QFACTOR:DLYMODE?**

Set or retrieve the Q-Factor delay mode.

Params:	< AUTOMatic >	Q-Factor delay mode is automatic
	< MANUal >	Q-Factor delay mode is manual
Returns:	< AUTOMATIC   MANUAL >	

### **QFACTOR:DMODE** <COUNT | GRAPh | RDATA>

#### **QFACTOR:DMODE?**

Set or retrieve the Q-Factor display mode.

Params:	<COUNT>	Counters
	<GRAPh>	Graph
	<RDATA>	Raw data
Returns:	< COUNT   GRAPH   RDATA >	

**QFACTOR:EGRaticule <bool>**  
**QFACTOR:EGRaticule?**

Enable/disable graticule (grid) on the Q-Factor display.

Params:	<bool>	On = 1, Off = 0
Returns:	< 1 >	Enable Q-Factor graticule
	< 0 >	Disable Q-Factor graticule

**QFACTOR:EQScale <bool>**  
**QFACTOR:EQScale?**

Enable/disable the Q Scale on the Q-Factor display.

Params:	<bool>	On = 1, Off = 0
Returns:	< 1 >	Enable Q-Factor Q Scale
	< 0 >	Disable Q-Factor Q Scale

**QFACTOR:ERRThr <double>**  
**QFACTOR:ERRThr?**

Set or retrieve Jitter Error Threshold from Q-Factor analysis.

Params:	<double>	Range [1 to 10,000]
Returns:	< double >	

**QFACTOR:IVALue?**

Retrieve Invalid Value. Query only.

**QFACTOR:L0Data?**

Retrieve the measured points for the Q-Factor display Level0. Query only.

Returns:	< N (x1, y1) (x2, y2) ... (xN, yN) >	N is the number of points, the x values are threshold values represented in mV, and the y values are BER measurements.
----------	--------------------------------------	--

**QFACTOR:L0EValue?**

Retrieve Q-Factor Level0 error value. Query only.

**QFACTOR:L0Fit?**

Retrieve the extrapolated points for the Q-Factor display Level0. Query only.

Returns:	< N (x1, y1) (x2, y2) ... (xN, yN) >	N is the number of points, the x values are threshold values represented in mV, and the y values are BER measurements.
----------	--------------------------------------	--

**QFACTOR:L0Mean?**

Retrieve Q-Factor Level0 mean. Query only.

**QFACTOR:L0NPoints?**

Retrieve Q-Factor Level0 number of measured points. Query only.

**QFACTOR:L0Sigma?**

Retrieve Q-Factor Level0 sigma. Query only.

**QFACTOR:L1Data?**

Retrieve the measured points for the Q-Factor display Level1. Query only.

Returns: < N (x1, y1) (x2, y2) ... (xN, yN) >      N is the number of points, the x values are threshold values represented in mV, and the y values are BER measurements.

---

**QFACTOR:L1EValue?**

Retrieve Q-Factor Level1 error value. Query only.

**QFACTOR:L1Fit?**

Retrieve the extrapolated points for the Q-Factor display Level1. Query only.

Returns: < N (x1, y1) (x2, y2) ... (xN, yN) >      N is the number of points, the x values are threshold values represented in mV, and the y values are BER measurements.

---

**QFACTOR:L1Mean?**

Retrieve Q-Factor Level1 mean. Query only.

**QFACTOR:L1NPoints?**

Retrieve Q-Factor Level1 number of measured points. Query only.

**QFACTOR:L1Sigma?**

Retrieve Q-Factor Level1 sigma. Query only.

**QFACTOR:MBARsel <OFF | MEASUREMENTS | STRESS | ADVANCED>****QFACTOR:MBARsel?**

Set or retrieve the Q-Factor display Sidebar selection.

Params:	<OFF>	No Sidebar
	<MEASUREMENTS>	Show Measurements
	<STRESS>	Show Stress Sidebar (if Stress Option is enabled)
	<ADVANCED>	Show Advanced features Sidebar

Returns: < OFF | MEASUREMENTS | STRESS | ADVANCED >

---

**QFACTOR:MBDepth?**

Measure Q-Factor BER depth. Query only.

**QFACTOR:OQ?**

Retrieve Q-Factor optimum Q. Query only.

**QFACTOR:OTValue?**

Retrieve optimum Q-Factor value. Query only.

**QFACTOR:OLBer?**

Retrieve Q-Factor optimum log BER. Query only.

**QFACTOR:OBSTring?**

Retrieve Q-Factor optimum BER string. Query only.

**QFACTOR:REFMode <LIVEData | DPATtern>****QFACTOR:REFMode?**

Set or retrieve Q-Factor reference mode.

Params:	<LIVEData>	Live data mode
	<DPATtern>	Detected pattern mode
Returns:	< LIVEDATA   DPATTERN >	

**QFACTOR:REFPattern?**

Retrieve Q-Factor reference pattern. Query only.

**QFACTOR:RETime?**

Retrieve Q-Factor run elapsed time in seconds. Query only.

**QFACTOR:SCURsors<X1,X2,Y1,Y2>**

Set Q-Factor cursor positions.

Params:	<X1>	X1 cursor position, double
	<X2>	X2 cursor position, double
	<Y1>	Y1 cursor position, double
	<Y2>	Y2 cursor position, double
Returns:	< X1,X2,Y1,Y2 >	

**QFACTOR:SDATA? <int>**

Retrieve Q-Factor series data, binary. Query only.

Returns:	<int>	Range [0 to 14]
----------	-------	-----------------

**QFACTOR:SDEFaults**

Set Q-Factor Defaults. Action only.



**QFACTOR:STFile<"filename" >**

Save Q-Factor results to named file. Action only.

Params:	<"filename">	File where Q-Factor results are to be saved. Enclose filename in double quotes.
---------	--------------	--

---

**QFACTOR:SUFBits<double>****QFACTOR:SUFBits?**

Set or retrieve Q-Factor sufficient bits.

Params:	<value>	A numeric value with range [1,000 to 1E11]. A value of -1 initiates automatic mode.
---------	---------	--

---

Returns:	< value >	Numeric value or automatic
----------	-----------	----------------------------

---

**QFACTOR:SVIEW<Xmin, Ymin, Xmax, Ymax>**

Set Q-Factor view axes.

Params:	<Xmin>	X-Axis minimum, double
---------	--------	------------------------

---

	<Ymin>	Y-Axis minimum, double
--	--------	------------------------

---

	<Xmax>	X-Axis maximum, double
--	--------	------------------------

---

	<Ymax>	Y-Axis maximum, double
--	--------	------------------------

---

Returns:	< Xmin, Ymin, Xmax, Ymax >	
----------	----------------------------	--

---

**QFACTOR:TSTATUS?**

Q-Factor analysis status. Query only.

Returns:	NOTRUNNING OK NOCLOCK LOWCLOCK NOPATTERN NOTFOUND ACQERROR WAITING	Status codes
----------	---	--------------

---

**QFACTOR:TVALUE <double>****QFACTOR:TVALUE?**

Set or retrieve Q-Factor value.

Params:	<double>	Range [0 to 5,000]
---------	----------	--------------------

---

Returns:	< double >	
----------	------------	--

---

**QFACTOR:X1Cursor <double>**  
**QFACTOR:X1Cursor?**

Set or retrieve Q-Factor cursor X1 position.

Params:	<double>	Position of Q-Factor X1 cursor
Returns:	< double >	

---

**QFACTOR:X2Cursor <double>**  
**QFACTOR:X2Cursor?**

Set or retrieve Q-Factor cursor X2 position.

Params:	<double>	Position of Q-Factor X2 cursor
Returns:	< double >	

---

**QFACTOR:XMAX <double>**  
**QFACTOR:XMAX?**

Set or retrieve Q-Factor view X maximum

Params:	<double>	Q-Factor X maximum
Returns:	< double >	

---

**QFACTOR:XMIN <double>**  
**QFACTOR:XMIN?**

Set or retrieve Q-Factor view X minimum

Params:	<double>	Q-Factor X minimum
Returns:	< double >	

---

**QFACTOR:Y1Cursor <double>**  
**QFACTOR:Y1Cursor?**

Set or retrieve Q-Factor cursor Y1 position.

Params:	<double>	Position of Q-Factor Y1 cursor
Returns:	< double >	

---

**QFACTOR:Y2Cursor <double>**  
**QFACTOR:Y2Cursor?**

Set or retrieve Q-Factor cursor Y2 position.

Params:	<double>	Position of Q-Factor Y2 cursor
Returns:	< double >	

---

**QFACTOR:YMAX <double>**

**QFACTOR:YMAX?**

Set or retrieve Q-Factor view Y maximum

Params: <double> Q-Factor Y maximum

---

Returns: < double >

---

**QFACTOR:YMIN <double>**

**QFACTOR:YMIN?**

Set or retrieve Q-Factor view Y minimum

Params: <double> Q-Factor Y minimum

---

Returns: < double >

---



# Mainframe

Complete command listing (See page 7, *Command listing*.)

## CardTEMP?

This command is no longer supported. See SENSor:TEMP? command.

## DELAY:DETRecal?

Retrieve the information that indicates whether Detector delay needs recalibration. Query only.

---

**NOTE.** *Because monitoring for these calibrations is suspended while Physical Layer tests are running, using this command under those circumstances might not return an accurate result.*

---

Returns:	< 1 >	Detector needs delay recalibration
	< 0 >	Detector does not need a delay recalibration

---

## DELAY:GENRecal?

Retrieve the information that indicates whether Generator delay needs recalibration. Query only.

---

**NOTE.** *Because monitoring for these calibrations is suspended while Physical Layer tests are running, using this command under those circumstances might not return an accurate result.*

---

Returns:	< 1 >	Generator needs delay recalibration
	< 0 >	Generator does not need a delay recalibration

---

## FAMILY\_1500?

Check if the instrument is a BA1500. Query only.

Returns:	< 1 >	Instrument is a BA1500
	< 0 >	Instrument is not a BA1500

---

## FAMILY\_7500?

Check if the instrument is a BSA75C. Query only.

Returns:	< 1 >	Instrument is a BSA75C
	< 0 >	Instrument is not a BSA75C

---

**FAMILY\_12500?**

Check if the instrument is a BSA125C. Query only.

Returns:	< 1 >	Instrument is a BSA125C
	< 0 >	Instrument is not a BSA125C

**PLATFORM?**

Retrieve the value of the current device platform. Query only.

Returns:	< 0   1   2   3 >	0: Unknown 1: BA1500/BA1600 2: BSA7500 or BSA125C (Models A and B) 3: BSA125C, BSA175C, BSA286C
----------	-------------------	--

**PRODNAME?**

Retrieve the string identifying the product (the "Identity" shown in the Remote Client). Query only.

Returns:	"BitAlyzer 1500" "BERTScope 7500" (BSA7500 Analyzer) "BERTScope 12500" (BSA125 Analyzer; models A, B, or C) "BERTScope 17500" (BSA175 Analyzer) "BERTScope 22000" (BSA220 Analyzer) "BERTScope 25000" (BSA250 Analyzer) "BERTScope 28600" (BSA286 Analyzer) "Unknown Product"
----------	--

**GUILockout <bool>****GUILockout?**

Set or retrieve the state of the GUI lockout selection.

Params:	<bool>	On = 1, Off = 0
Returns:	< 1 >	GUI lockout is in effect
	< 0 >	GUI lockout is not in effect

**MODEL\_A?**

Check if the instrument is a BERTScope Analyzer Model A (7500 or 12500). Query only.

Returns:	< 1 >	Instrument is a BERTScope Analyzer Model A
	< 0 >	Instrument is not a Model A

**MODEL\_B?**

Check if the instrument is a BERTScope Analyzer Model B (7500 or 12500).  
Query only.

Returns:	< 1 >	Instrument is a BERTScope Analyzer Model B
	< 0 >	Instrument is not a Model B

**OPT\_CE?**

Check if the instrument has PatternVu Option enabled. Query only.

Returns:	< 1 >	PatternVu Option is enabled
	< 0 >	PatternVu Option is disabled

**OPT\_EA\_2D\_MAP?**

Check if the instrument has 2-D Error Map Analyzer Option enabled. Query only.

Returns:	< 1 >	2-D Error Map Option is enabled
	< 0 >	2-D Error Map Option is disabled

**OPT\_EA\_ECC?**

Check if the instrument has FEC Emulation Analyzer Option enabled. Query only.

Returns:	< 1 >	FEC Emulation is enabled
	< 0 >	FEC Emulation is disabled

**OPT\_GSM?**

Check if the instrument has a Generator Stress Module installed. Query only.

Returns:	< 1 >	Generator Stress Module is installed
	< 0 >	Generator Stress Module is not installed

**OPT\_JITTERMAP?**

Check if the instrument has the JitterMAP Option enabled. Query only.

Returns:	< 1 >	JitterMAP Option is enabled
	< 0 >	JitterMAP Option is disabled

**OPT\_JT?**

Check if the instrument has Jitter Tolerance Option enabled. Query only.

Returns:	< 1 >	Jitter Tolerance Option is enabled
	< 0 >	Jitter Tolerance Option is disabled

**OPT\_LIVE\_DATA?**

Check if the instrument has Live Data Option enabled. Query only.

Returns:	< 1 >	Live Data Option is enabled
	< 0 >	Live Data Option is disabled

---

**OPT\_PCI?**

Check if the instrument has PCIE Option enabled. Query only.

Returns:	< 1 >	PCIE Option is enabled
	< 0 >	PCIE Option is disabled

---

**OPT\_PHY\_TEST?**

Check if the instrument has Physical Layer Test Option enabled. Query only.

Returns:	< 1 >	Physical Layer Test Option is enabled
	< 0 >	Physical Layer Test Option is disabled

---

**OPT\_SSCPLUS?**

Retrieve the flag signifying whether SSC PLUS (XSSC) option is enabled. Query only.

Returns:	< 1 >	XSSC Option is enabled
	< 0 >	XSSC Option is disabled

---

**PTFile <"filename">**

Print the current view into a file.

Params:	<"filename">	Filename is enclosed in double quotes.
---------	--------------	--

---

**RCStatus?**

Returns the version information of Remote Control. Query only.

**RDURation <long>****RDURation?**

Set or retrieve the run duration of the BERTScope Analyzer. Setting to '0' is the equivalent of a GUI 'Clear,' which allows the run (see RState command) to go on 'forever.'

Params:	<long>	Range [1 to 36E6]. Zero (0) sets an infinite duration.
---------	--------	--

---

Returns:	< long >
----------	----------

---



**RState <bool>  
RState?**

Set or retrieve the run state of the BERTScope Analyzer.

Params:	<bool>	On = 1, Off = 0
Returns:	< 1 >	System is running
	< 0 >	System is not running

**SENSor:TEMP?**

Retrieve internal temperature in degrees Celsius . Query only. This measurement is not calibrated, and should only be used as a relative indication of temperature.

Returns:	< double >	Positive integer, internal temperature (°C)
----------	------------	---

**SENSor:V12?**

Retrieve reading of the internal +12 volt register. Query only.

Returns:	< double >	Interval 12 V register
----------	------------	------------------------

**SENSor:V1P5?**

Retrieve reading of the internal +1.5 volt register. Query only.

Returns:	< double >	Internal +1.5 V register
----------	------------	--------------------------

**SENSor:V2P5?**

Retrieve reading of the internal +2.5 volt register. Query only.

Returns:	< double >	Internal +2.5 V register
----------	------------	--------------------------

**SENSor:V3P3?**

Retrieve reading of the internal +3.3 volt register. Query only.

Returns:	< double >	Internal 3.3 V register
----------	------------	-------------------------

**SENSor:V5?**

Retrieve reading of the internal +5 volt register. Query only.

Returns:	< double >	Internal 5 V register
----------	------------	-----------------------

**SENSor:VM12?**

Retrieve reading of the internal –12 volt register. Query only.

Returns:	< double >	Internal –12 V register
----------	------------	-------------------------

**SENSor:VM19?**

Retrieve reading of the internal -19 volt register. Query only.

Returns: < double > Internal -19 V register

---

**SENSor:VM2?**

BitAlyzer BA1500 only. Retrieve reading of the internal -2 volt register. Query only.

Returns: < double > Internal -2 V register

---

**SENSor:VM5?**

Retrieve reading of the internal -5 volt register. Query only.

Returns: < double > Internal -5 V register

---

**SENSor:VM5P2?**

BitAlyzer BA1500 only. Retrieve reading of the internal -5.2 volt register. Query only.

Returns: < double > Internal -5.2 V register

---

**VIEW <BBER | BER | BLEN | CONT | CORR | CRC | CRLR | CRS |  
DET | DPP | EDIT | EFIN | EMAP | EYE | FEC | GEN | HOME | JITT |  
JMAP\_MAP | JS | JTOL | LOG | LTS | MASK | PSEN | QFAC | SSCW |SCH  
| STRESS | SYST>**

### VIEW?

Set or retrieve the current view of the BERTScope Analyzer.

Params:	<BBER>	Basic BER
	<BER or BERror>	Block Error
	<BLEN or BLENgth>	Burst Length
	<CONTour>	BER Contour
	<CORR or CORRelation>	Correlation
	<CRC or CRControl>	Clock Recovery Control
	<CRS or CRService>	Clock Recovery Service
	<CRLR or CRLoopResponse>	Clock Recovery Loop Response
	<DET or DETector>	Detector
	<DPP>	DPP Interface
	<EDIT or EDITor>	Editor
	<EFIN or EFINterval>	Error Free Interval
	<EMAP>	2D Error Map
	<EYE>	Eye Diagram
	<FEC>	FEC Emulation
	<GEN or GENerator>	Generator
	<HOME>	Home
	<JITTer>	Jitter Peak
	<JMAP_MAP>	Jitter Map
	<JS or crJitterSpectrum>	Clock Recovery Jitter Spectrum

---

<JTOL or JTOLerance>	Jitter Tolerance Test
<LOG>	System Event Log
<LTS>	Lightwave Test Set Interface
<MASK>	Mask Test
<PSEN or PSENSitivity>	Pattern Sensitivity
<QFAC or QFACtor>	Q-Factor
<SCH or SCHart>	Strip Chart
<SSCW or crSSCWaveform>	Clock Recovery SSC Waveform
<STRESS or STRESSedeye>	Stressed Eye
<SYST or SYSTem>	System

---

Returns: < BBER | BER | BLEN | CONT | CORR | CRC | CRLR | CRS | DET | DPP |  
EDIT | EFIN | EMAP | EYE | FEC | GEN | HOME | JITT | JMAP\_MAP | JS |  
JTOL | LOG | LTS | MASK | PSEN | QFAC | SSCW | SCH | SSCW | STRESS |  
SYST >

---

# System View

Complete command listing (See page 7, *Command listing*.)

## **DIALog:CDRam <bool>**

### **DIALog:CDRam?**

Enable/disable display of a warning if capturing Detector RAM.

Params:	<bool>	On = 1, Off = 0
Returns:	< 1 >	Display warning if capturing Detector RAM
	< 0 >	Do not display warning if capturing Detector RAM

## **DIALog:RDCalibration <bool>**

### **DIALog:RDCalibration?**

Enable/disable display of “Recommend delay calibration” message.

Params:	<bool>	On = 1, Off = 0.
Returns:	< 1 >	Display “Recommend delay calibration” message
	< 0 >	Do not display “Recommend delay calibration” message

## **DIALog:SRFCapture <bool>**

### **DIALog:SRFCapture?**

Enable/disable display of a warning if stop running for capture.

Params:	<bool>	On = 1, Off = 0
Returns:	< 1 >	Display warning if stop running for capture
	< 0 >	Do not display warning if stop running for capture

## **DIALog:SRView <bool>**

### **DIALog:SRView?**

Enable/disable display of a warning if switching running view.

Params:	<bool>	On = 1, Off = 0
Returns:	< 1 >	Display warning if switching running view
	< 0 >	Do not display warning if switching running view

**STAB <TOOLs | REGistry | SETTings | ABOut | LOG | SelfTESt>  
STAB?**

Set or retrieve which tab is being displayed on the System view.

Params:	<TOOLs>	System Tools tab
	<REGistry>	System Registry tab
	<SETTings>	System Settings tab
	<ABOut>	System About tab
	<LOG>	System Log tab
	<SelfTESt>	System Self-Test tab
Returns:	< TOOLS   REGISTRY   SETTINGS  ABOUT  LOG   SELFTEST >	

# RAM Capture

Complete command listing (See page 7, *Command listing*.)

## **CBLength <numeric> , <“filename”>**

Capture the Detector RAM into a file by the input length.

The captured file is in the D:\<BERTScope | BitAlyzer>\UserPatterns directory. Action only.

Params:	<numeric>	Number of words to capture into RAM Use a comma to separate this parameter and the next.
	<“filename”>	Filename is enclosed in double quotes, should not contain any path, and must have .RAM as extension. For example: “MyPattern.RAM”.

## **CBTrigger <“filename”>**

Capture the Detector RAM into a file by trigger.

The captured file is in the D:\BitAlyzer\UserPatterns or D:\BERTScope\UserPatterns directory. Action only.

Params:	<“filename”>	Filename is enclosed in double quotes, should not contain any path, and must have .RAM as extension. For example: “MyPattern.RAM”.
---------	--------------	--

## **CBTriggerlength <numeric> , <“filename”>**

BERTScope Analyzer only. Capture the Detector RAM into a file by trigger.

The captured file is in the D:\BERTScope\UserPatterns directory. Action only.

Params:	<numeric>	Number of words to capture into RAM There must be a comma between the first and second parameters.
	<“filename”>	Filename is enclosed in double quotes, should not contain any path, and must have .RAM as extension. For example: “MyPattern.RAM”.

## **RCCancel**

Cancel a running Detector RAM capture operation, if capturing was initiated by either CBLength or CBTRigger command. Action only.

**RCProgress?**

Retrieve the percentage completion of the running Detector RAM capture operation. This command should be used only after either the CBLength or CBTRigger command. Query only.

Returns:        <numeric>                      Percentage completion value of the capture operation

---



# System Event Log

Complete command listing (See page 7, *Command listing*.)

## **SLOG:CFILter:ALL <bool>**

Enable/disable all system event log filters. Action only.

Params:	<bool>	On = 1, Off = 0.
Returns:	< 1 >	All log filters are On
	< 0 >	All log filters are Off

## **SLOG:CFILter:PCHanges <bool>**

### **SLOG:CFILter:PCHanges?**

Set or retrieve whether Parameter Changes filter is enabled.

Params:	<bool>	On = 1, Off = 0
Returns:	< 1 >	Parameter Changes filter is enabled
	< 0 >	Parameter Changes filter is disabled

## **SLOG:CFILter:PRECognition <bool>**

### **SLOG:CFILter:PRECognition?**

Set or retrieve whether Pattern Recognition filter is enabled.

Params:	<bool>	On = 1, Off = 0
Returns:	< 1 >	Pattern Recognition filter is enabled
	< 0 >	Pattern Recognition filter is disabled

## **SLOG:CFILter:SEvents <bool>**

### **SLOG:CFILter:SEvents?**

Set or retrieve whether System Events filter is enabled.

Params:	<bool>	On = 1, Off = 0
Returns:	< 1 >	System Events filter is enabled
	< 0 >	System Events filter is disabled

## **SLOG:CFILter:VCHanges <bool>**

### **SLOG:CFILter:VCHanges?**

Set or retrieve whether View Changes filter is enabled.

Params:	<bool>	On = 1, Off = 0
Returns:	< 1 >	View Changes log filter is enabled
	< 0 >	View Changes log filter is disabled

**SLOG:CFILter:BATHreshold <bool>**  
**SLOG:CFILter:BATHreshold?**

Set or retrieve whether BER Above Threshold filter is enabled.

Params:	<bool>	On = 1, Off = 0
Returns:	< 1 >	BER Above Threshold filter is enabled
	< 0 >	BER Above Threshold filter is disabled

**SLOG:CLOG**

Clear the system event log. Action only.

**SLOG:ENABLE <bool>**  
**SLOG:ENABLE?**

Set or retrieve whether system event logging is enabled.

Params:	<bool>	On = 1, Off = 0
Returns:	< 1 >	System event logging is enabled
	< 0 >	System event logging is disabled

**SLOG:LBTHreshold <double>**  
**SLOG:LBTHreshold?**

Set or retrieve the Log BER Threshold, when BER Above Threshold log filter is enabled.

Params:	<double>	BER above threshold value, double
---------	----------	-----------------------------------

**SLOG:SLTFile <"filename">**

Save log to a file. Action only.

Params:	<"filename">	The location where the log file will be saved as a " " (vertical bar) separated-text fie. Filename is enclosed in double quotes.
---------	--------------	--





---

# Status Queries

Complete command listing (See page 7, *Command listing*.)

The following commands retrieve information for each component, such as driver version, H/W availability, and whether a driver is loaded.

## CLKStatus?

Retrieve the CLK driver's information. Query only.

This command is supported for backwards compatibility, but has been superseded by the NDRIvers? and DVERSion? commands.

## DELAY:DETRecal?

Retrieve the information that indicates whether Detector delay needs recalibration. Query only.

---

**NOTE.** *Because monitoring for these calibrations is suspended while Physical Layer tests are running, using this command under those circumstances might not return an accurate result.*

---

Returns:	< 1 >	Yes, Detector needs delay recalibration
	< 0 >	No, Detector does not need a delay recalibration

---

## DELAY:GENRecal?

Retrieve the information that indicates whether Generator delay needs recalibration. Query only.

---

**NOTE.** *Because monitoring for these calibrations is suspended while Physical Layer tests are running, using this command under those circumstances might not return an accurate result.*

---

Returns:	< 1 >	Yes, Generator needs delay recalibration
	< 0 >	No, Generator does not need a delay recalibration

---

## DIOStatus?

Retrieve the DIO driver's information. Query only.

This command is supported for backwards compatibility, but has been superseded by the NDRIvers? and DVERSion? commands.

**DLYMstatus?**

Retrieve the DLYM driver’s information. Query only. This command is supported for backwards compatibility, but has been superseded by the NDRivers? and DVERsion? commands.

**DVERsion? <integer>**

Retrieve the <Integer>-th driver version. Query only.

Params:           <Integer>   The number is of the range [0 to (N-1)], where N is the value returned from the NDRivers? command, above.

---

Returns:           < Driver name and version number >

---

**EYEstatus?**

Retrieve the EYE driver’s information. Query only.

This command is supported for backwards compatibility, but has been superseded by the NDRivers? and DVERsion? commands.

**GIOstatus?**

Retrieve the GIO driver’s information. Query only.

This command is supported for backwards compatibility, but has been superseded by the NDRivers? and DVERsion? commands.

**HMStatus? <CLK | GIO | DIO | DLYM | SRX | STX>**

Retrieve the hardware module information. Query only.

Params:           <CLK | GIO | DIO | DLYM | SRX | STX>           Only one parameter is accepted per query.

---

Returns:           < Hardware module information >

If the command is given a non-applicable parameter, the query returns “Requested hardware module is not supported on the platform”.

---

**ISStatus?**

Retrieve the Instrument Server’s version info. Query only.

**NDRivers?**

Retrieve the number sequence of the software drivers. Query only.

**SNUMber?**

Retrieve the instrument serial number. Query only.

---

Returns:           < Instrument serial number >

---

### SRXStatus?

Retrieve the SRX driver's information. Query only.

This command is supported for backwards compatibility, but has been superseded by the `NDRIvers?` and `DVERsion?` commands.

### STXStatus?

Retrieve the STX driver's information. Query only.

This command is supported for backwards compatibility, but has been superseded by the `NDRIvers?` and `DVERsion?` commands.

### TREAding? <CAB | GENDELAY | DETDELAY>

Retrieve the internal temperature readings in degrees Celsius. Query only

Params:	<CAB>	Only one parameter is accepted per query.
	<GENDELAY>	
	<DETDELAY>	
Returns:	< Internal temperature reading in °Celsius >	
	If the command is given a non-applicable parameter, the query returns "Requested temperature reading is not supported on the platform".	

### VREAding? < M5P2 | M2 | M12 | 3P3 | 5 | 12 >

Retrieve the internal voltage register reading. Query only.

Params:	<M5P2   M2   3P3_STBY   3P3   5   12   M12   2P5   1P5   M5   M19   PGM_3P3   PGM_M5   PGM_5   PGM_M8   PGM_8   PGM_M11   PGM_11   PGM_M17P5   PDM_3P3   PDM_5   PDM_11   PDM_M3P3   PDM_M5   PDM_M11>	Only one parameter is accepted per query.
Returns:	< Internal voltage register reading >	
	If the command is given a non-applicable parameter, the query returns "Requested hardware module is not supported on the platform".	





---

## Common Commands

Complete command listing (See page 7, *Command listing*.)

### **\*CLS**

Clear the status queue. Action only. This command will not show in the list of Trace messages displayed on the remote control client user interface.

### **\*IDN?**

Retrieve the identification of the device. Query only. This command will not show in the list of Trace messages displayed on the remote control client user interface.

### **SYSTem:ERRor?**

Retrieve items listed in the error status queue. Query only. This command will not show in the list of Trace messages displayed on the remote control client user interface.

The status queue is first in, first out. It can contain up to 100 error messages. If the queue overflows, the last error/event in the queue is replaced with Error -350, Queue overflow. When all errors/events have been read from the queue, further error/event queries will return 0, NO ERROR.

Returns: < n, "event/string" >

---



---

# File Transfer

Complete command listing (See page 7, *Command listing*.)

Transfer files between the analyzer and a remote PC.

The file transfer protocol is shown in the **Send** and **Get** diagrams. File Transfer commands will not show up in the list of Trace messages displayed on the remote control client user interface.

## **ABORT**

Cancels a SEND or GET file operation.

This command will not show in the list of Trace messages displayed on the remote control client user interface.

An ABORT reply indicates that the file transfer operation is being discontinued. This command will not show in the list of Trace messages displayed on the remote control client user interface.

## **BUFFER <Bytes> <CRC>**

During “Get” or “Send” file transfer, indicates size of next buffer to be transferred. Once file transfer is initiated, this command precedes a block transfer of the exact number of Bytes. During transfer of these Bytes, no command processing occurs. These Bytes are transferred in binary form.

Params:	<Bytes>	The number of Bytes in the next block of binary data transfer (unsigned short)
	<CRC>	BitAlyzer: The 32-bit checksum of the buffer BERTScope Analyzer: The 128-bit checksum of the buffer
Returns:	Continue or Abort	

## **CONTINUE**

Acknowledges that a file transfer operation can proceed.

This command will not show in the list of Trace messages displayed on the remote control client user interface.

After each buffer has been transmitted, the receiver replies with this CONTINUE message if the transfer may proceed; otherwise, it replies with an ABORT message.

Returns: binary data or Abort

---

**DONE**

Acknowledgement: Tells the file receiver that all the buffers have been sent out.

File transfer is completed. This command will not show in the list of Trace messages displayed on the remote control client user interface.

**FILE:DELETE "full-path-name"**

Use to delete a file or a folder.

Enclose the path name in double quotes. Search is not case-sensitive. Spaces are allowed within the string. A file is deleted only if it is not read-only. A folder is deleted only if it is empty. Action only.

**FILE:EXISTS? "full-path-name"**

Search is not case-sensitive. Spaces are allowed within the string. Enclose in double quotes. Query only.

Returns:	0	File does not exist
	1	File exists

---

**FILE:GET <"filename">**

Get a file from BitAlyzer/BERTScope Analyzer. Action only.

This command will not show in the list of Trace messages displayed on the remote control client user interface.

Params:	<"filename">	Name of the BitAlyzer/BERTScope Analyzer file to be sent to the PC.
---------	--------------	---

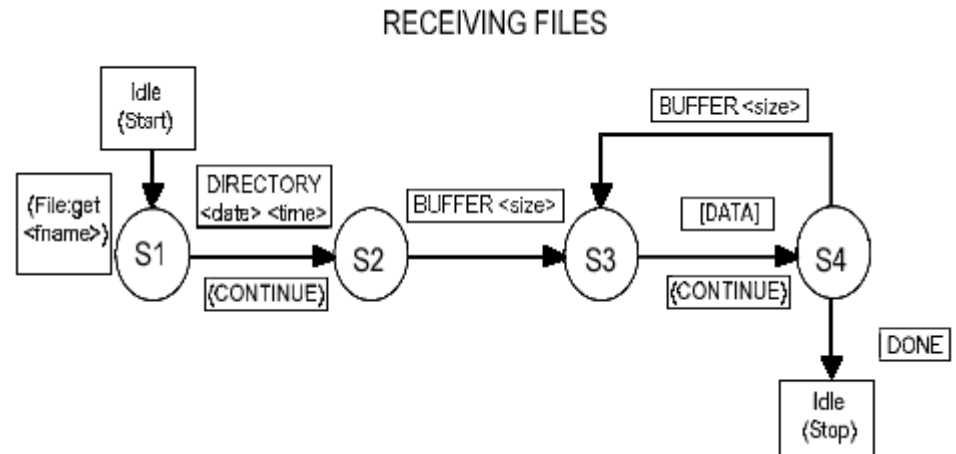
---

These commands start the file transfer. The Remote Control program needs to follow a specific protocol in order to complete the transfer (see the GET diagram, below).

The file is separated into packets of up to 4,090 Bytes. For each packet, a BUFFER <Bytes> <CRC> command is sent first. The sender should calculate the CRC value for a packet and send it in the buffer acknowledge. The receiver will recalculate and compare the CRC. If the CRC is incorrect, the receiver will request a retry. Remote Service supports retry up to three times, after which it will assume the session of file transfer is impossible to complete correctly and will issue an ABORT. When it reaches the EOF, the sender should let the receiver know by issuing a DONE.

The acknowledgement of file transfer should be all uppercase: CONTINUE, ABORT, RETRY, DONE, and BUFFER.

From analyzer to remote PC:



### **FILE:MKDIR "full-folder-name"**

Create a folder. Action only.

Enclose the full name in double quotes. The string can include upper and lower case letters, and spaces are allowed.

### **FILE:SEND <"filename">**

Send a file from the remote PC to BitAlyzer/BERTScope Analyzer. Action only.

Command will not show in the list of Trace messages displayed on the remote control client user interface.

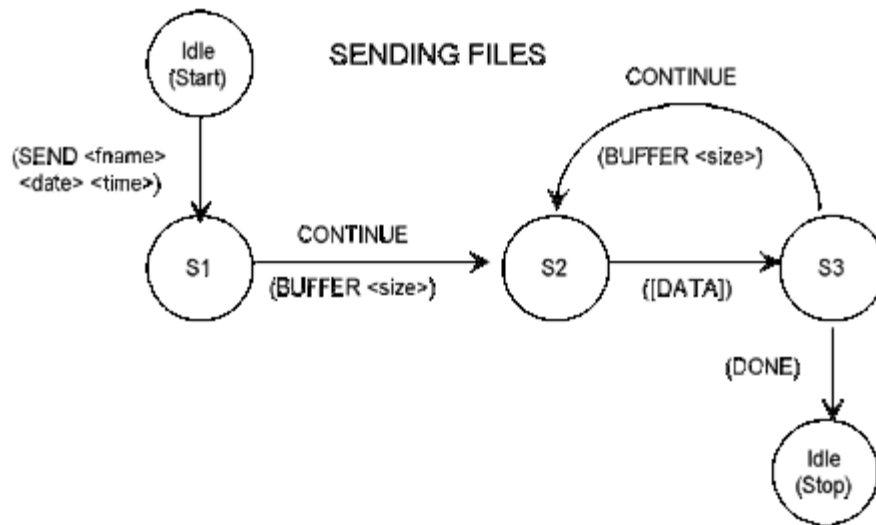
Params:	<"filename">	Name of the PC file to be sent to the BitAlyzer/BERTScope Analyzer.
---------	--------------	---

This command starts a file transfer from the remote PC to the BitAlyzer/BERTScope Analyzer. The Remote Control program needs to follow a specific protocol in order to complete the transfer (see the SEND diagram, below).

The file is separated into packets of up to 4,090 Bytes. For each packet, a BUFFER <Bytes> <CRC> command is sent first. The sender should calculate the CRC value for a packet and send it in the buffer acknowledge. The receiver will recalculate and compare the CRC. If the CRC is incorrect, the receiver will request a retry. Remote Service supports retry up to three times, after which it will assume the session of file transfer is impossible to complete correctly and will issue an ABORT. When it reaches the EOF, the sender should let the receiver know by issuing a DONE.

The acknowledgement of file transfer should be all uppercase: CONTINUE, ABORT, RETRY, DONE, and BUFFER.

From remote PC to analyzer:



### RETRY

**Acknowledgement:** Requests the file sender to resend the previous buffer. This command will not show in the list of Trace messages displayed on the remote control client user interface.

The RETRY command is used when the buffer was not received correctly. Retry is supported for up to three attempts.

# Appendix A: Sample Remote Programming

This appendix provides an example of running the remote control commands with the BERTScope analyzer.

## Connect the cables to the instrument

The hardware setup uses 50  $\Omega$  cables to match the 50  $\Omega$  output impedance of the data output connectors. All data cables must be the same length.

---

**NOTE.** Assume that the data outputs are already set for 50  $\Omega$  output impedance. If not change the instrument setup as needed to get the proper output impedance.

---

1. Connect the Generator Clock+ Output to the Dectector Clock Input.
2. Connect the Generator Data+ Output to the Detector Data+ Input.
3. Connect the Generator Data- Output to the Detector Data- Input.

## Send commands to the instrument

The following section shows the high-level steps to send the commands to the instrument followed by a list of actual commands.

1. View the Generator.

```
VIEW Generator
```

2. Turn off the Generator Amps.

```
GEN:DOP:ENAB 0      ; Turn off DataP
GEN:DON:ENAB 0      ; Turn off DataN
GEN:COP:ENAB 0      ; Turn off ClockP
GEN:CON:ENAB 0      ; Turn off ClockN
```

3. Send PN7 pattern to both the Generator and Detector.

```
GEN:PATT PN7        ; Use a HW 2^7-1 pattern. Load in
DET:PATT PN7        ; Generator
                    ; Same pattern loaded into Detector
```

4. View the Detector.

```
VIEW Detector
```

5. Turn the Detector Auto Resync mode off.

```
DET:AREN 0 ; Auto Resync mode off
```

6. Set the Detector to custom levels followed by the Clock, and then the Data.

```
DET:CINP:TAC ; voltage which Gen ClkP will be
DET:DIMP:IMOD ; set to AC coupled after the DC
DIFF ; termination voltage setting
DET:DINP:TVOL 0 ; Detector is expecting differential
; data (both inputs)
; Detector is going to terminate
; inputs to 0 mV DC
; Don't care about threshold voltage
; Auto align will find and set
```

7. Finish setting up the Generator; set the frequency, and set the output amplifier conditions.

```
VIEW Generator ; Set internal synthesizer to 9.953 GHz
GEN:ICL 9953000000 ; Gate the internal synthesizer to
GEN:CSEL INT ; the Generator (not an ext syn)

GEN:COP:TVOL 0 ; 0 mV Termination for ClockP
GEN:COP:SLOF 0 ; 0 mV Offset for ClockP
GEN:COP:SLAM 1000 ; 1000 mV (1 V) Amplitude for ClockP

GEN:DOUT:LPNS 1 ; Link DataP & DataN. Only set
GEN:DOP:TVOL 0 ; conditions for one, other follows
GEN:DOP:SLOF 0 ; 0 mV Termination for DataP
GEN:DOP:SLAM 1000 ; 0 mV Offset for DataP
; 1000 mV (1 V) Amplitude for DataP
```



8. Turn the outputs on and query their “on” status; they can take up to three seconds to turn on.

```

GEN:COP:ENAB 1      ; Enable ClockP output
GEN:DOP:ENAB 1      ; Enable DataP output
GEN:DON:ENAB 1      ; Enable DataN output

Loop                ; Is Generator ClockP Enabled yet?
Until = 1
GEN:COP:ENAB?
wait 0.25
Repeat

Loop                ; Is Detector DataP Enabled yet?
Until = 1
GEN:DOP:ENAB?
wait 0.25
Repeat

Loop                ; Is Detector DataN Enabled yet?
Until = 1
GEN:DON:ENAB?
wait 0.25
Repeat

```

9. Calibrate the Generator and Detector delay paths for this frequency at this time and temperature.

```

GEN:PCAL            ; Initiate Generator Delay Calibration
wait                ; Takes awhile (10 seconds?)
Until reply
RSTATE?

Repeat

DET:PCAL            ; Initiate Detector Delay Calibration
wait                ; Takes awhile (10 seconds?)
Until reply
RSTATE?

Repeat

```

**10. Auto align the Detector (Perform Data Centering).**

```

DET:PDC                ; Initiate an Auto Align
wait Until reply      ; Takes awhile
RSTATE?

Repeat
    
```

**11. Auto Align has completed and the Detector is ready to run (error-free if cabled back-to-back).**

```

DET:DCS?                ; Was Auto Align successful?
                        ; Result = 1 if successful

DET:ISYNC?              ; Should be sync'd
                        ; Result = 1 if sync'd
    
```

**12. First query some results from the Auto Align.**

```

DET:DCAM?                ; what is the data amplitude in mV?
                        ; Should be about 2 V pp (differential
                        ;
                        ; adds the amplitudes of the two
                        ; two out-of-phase inputs)

DET:DCHM?                : what is the data offset voltage
                        ; in mV? should be about 0 V DC

DET:DCW?                 : what is the width of the eye?
                        ; Should be inverse of frequency
    
```

**13. Run bits through the Detector for 20 seconds.**

```

DET:RUIN 20             ; Sets the run interval for 20 seconds
RSTATE 1                 ; Runs the detector for the set time
                        ; above (20 seconds)

wait Until Reply

RSTATE?                 ; Are we done with the 20-second run
                        ; yet?

Repeat
    
```

**14. Running has stopped. Did errors accumulate along the way? (If your want errors to accumulate, slightly unscrew the data cables during the run time and watch the GUI in the Detector view.)**

```

DET:BITS?           ; How many bits passed during the
                    ; 20-second run? Should be about 20
                    ; seconds x 9.953 Gb/s

DET:ERR?            ; And how many errors in those 20
                    ; seconds?

```

## Run a test and read back the measurements

This example runs the Jitter Peak analysis and obtains the total jitter result.

---

**NOTE.** *This example assumes that the Detector is in sync.*

---

```

VIEW JITTER         ; Switch to Jitter Peak View
RSTATE 1            ; Toggle Run state

wait Until Reply    ; Toggling Run state to 1 does an auto
                    ; align, which will take approximately
                    ; 5 seconds

JITT:TST?           ; Query Jitter Peak Status

If OK Proceed
Else Take
necessary action

JITT:IVALUE?        ; Query Invalid value
JITT:TJIT?          ; Query to retrieve Total Jitter

                    ; Run test until Total Jitter value is not the invalid
                    ; value returned earlier

RSTATE 0            ; Stop test once we have valid value

```



# Index

## A

- analysis engine
  - basic BER Commands, 149
  - block error commands, 150
  - burst length commands, 153
  - correlation commands, 155
  - error free interval
    - commands, 158
  - error map commands, 160
  - FEC emulation
    - commands, 163
  - pattern sensitivity
    - commands, 169
  - remote control
    - commands, 145
  - strip chart commands, 172
- ASCII text strings, 3

## B

- basic BER commands, 149
- BER contour, 215, 221
- block error commands, 150
- BSAITS125
  - remote control command
    - summary, 72
- BSAITS125 commands, 74
- burst length commands, 153

## C

- case sensitivity, 3
- CleanEye, 187
- clipped Inputs (remote control), 2

- clock recovery, 79
  - control view: remote
    - control, 85
  - jitter spectrum commands, 99
  - loop response commands, 94
  - miscellaneous
    - commands, 103
  - recovery service
    - commands, 84
  - remote control command
    - summary, 81
  - sample program, 106
  - SSC waveform
    - commands, 96
- command description, 2
- command listing, 7
- command status, 5
- command summary
  - BSAITS125, 72
  - DPP, 114
  - stress combiner, 72
- command syntax, 3
- commands
  - long form, 3
  - short form, 3
- common commands, 281
- communications timeouts, 6
- configuration commands, 275
- correlation commands, 155
- CRJ, 85

## D

- Detector, 51
- detector auto align results
  - commands, 68
- detector blank input
  - commands, 63
- detector clock input
  - commands, 59
- detector convenience
  - commands, 68
- detector data input commands, 61

- detector optical mode
  - commands, 69
- detector results commands, 65
- detector symbol filtering
  - commands, 67
- detector trigger commands, 64
- double quotes, 4
- DPP
  - remote control command
    - summary, 114
- DPP Ccommands, 116
- DPP commands, 113
- driver information
  - commands, 277

## E

- error codes, 6
- error detector, 54
  - start detect commands, 58
- error free interval commands, 158
- error map commands, 160
- error messages, 5
- eye data collecting, 206
- eye diagram, 176
  - CleanEye commands, 187
  - eye measurement overlay
    - commands, 196
  - optical mode, 208
  - read eye measurements, 212
- eye measurement overlay, 196
- eye optical mode, 208
- eye single value waveform, 207

## F

- feature groups, 3
- FEC emulation commands, 163
- file names, 4
- file transfer commands, 283

## G

- Generator Clock Input
  - Commands, 22

- generator clock output
  - commands, 24, 27, 30
- Generator commands, 11
- generator data commands, 17
- generator data output
  - commands, 31, 34, 36
- generator error inject
  - commands, 38
- generator pattern start
  - commands, 19
- generator spread spectrum clock
  - commands, 20
- generator stress module, 41
- generator trigger commands, 37
- GSM, 41
  - remote control
    - configure/enable, 41
  - remote control setup, 42

## J

- jitter map, 222
- jitter peak, 234
- jitter spectrum commands, 99
- jitter tolerance, 242

## L

- LTS
  - remote control setup, 132
- LTS command overview, 131
- LTS commands, 135, 136

## M

- mainframe commands, 261
- mask test, 247

## O

- optical mode, 221

## P

- parameters, 4
- pathname, 4
- pattern sensitivity
  - commands, 169
- physical layer testing, 175

## Q

- Q-factor, 253
- question mark, 4
- question marks, 4
- quotes, 4

## R

- RAM capture, 271
- read eye measurements, 212
- read-only operation, 4
- read-write operation, 4
- Remote Control
  - Generator Clock Input
    - Commands, 22
- remote control setup, 2

## S

- sample clock recovery
  - program, 106
- SSC waveform commands, 96
- Status Queries, 277
- status queue, 5
- status variable, 5
- stress combiner, 71
  - remote control command
    - summary, 72
- stress combiner commands, 74
- stressed eye, 41
  - configure/enable, 41
  - setup, 42
- strip chart commands, 172
- syntax, 3
- system
  - event log commands, 273
- system commands, 261
- system view, 269

## T

- text strings, 3
- theory of operation, v
- timeouts, 6

## W

- write-only operation, 4