SCPI Commands

CMA5000(a)
Standard OTDR Application

Third Edition

For safety and warning information, please read this manual before attempting to use the equipment.
Keep this manual with the equipment.

ANRITSU CORPORATION

Part No.: 42469

SCPI Commands – CMA5000(a) Standard OTDR Application
10 October 2009 (Third Edition)

Copyright © 2009, ANRITSU CORPORATION.
All rights reserved. No part of this manual may be reproduced without the prior written permission of the publisher.
The contents of this manual may be changed without prior notice.

Composed in the USA

Table of Contents

This section details the SCPI commands for the CMA5000 Standard OTDR application. The Command Summary section presents a brief summary of each command, while each command is detailed in the subsequent sections.

Table of Contents iii
1 Command Summary 7
2 Root Level Commands 12
2.1 ABORt Command 12
2.2 INITiate Command 12
2.3 INITiate AUTo Command 13
2.4 INITiate? Command 13
3 Sensing Commands 14
3.1 SENSe:AVERages:COMPleted? Command 14
3.2 SENSe:FIBer:IOR Command 14
3.3 SENSe:FIBer:IOR? Command 14
3.4 SENSe:FIBer:BSC Command 15
3.5 SENSe:FIBer:BSC? Command 15
3.6 SENSe:TRACE:READY? Command 15
4 Signal Generation – The SOURce Subsystem 16
4.1 SOURce:HOFFset Command 16
4.2 SOURce:HOFFset? Command 16
4.3 SOURce:VOFFset Command 16
4.4 SOURce:VOFFset? Command 17
4.5 SOURce:WAVelength Command 17
4.6 SOURce:WAVelength? Command 17
4.7 SOURce:ACURsor:POINt Command 17
4.8 SOURce:ACURsor:POINt? Command 18
4.9 SOURce:BCURsor:POINt Command 18
4.10 SOURce:BCURsor:POINt? Command 18
4.11 SOURce:PULSe:WIDTh Command 19
4.12 SOURce:PULSe:WIDTh? Command 19
4.13 SOURce:RANge:RESo Command 20
4.14 SOURce:RANge:RESo? Command 20
4.15 SOURce:RANge:RESo:ALL? Command 21
4.16 SOURce:WAVelength:AVAilable? Command 21
4.17 SOURce:LSALeft Command 22
4.18 SOURce:LSALeft? Command 22
4.19 SOURce:LSARight Command 22
4.20 SOURce:LSARight? Command 23
4.21 SOURce:Loss:Mode Command 23
4.22 SOURce:Loss:Mode? Command 23
4.23 SOURce:ANALyze:ON Command 24

4.24 SOURce:ANALyze:ON? Command 24
4.25 SOURce:CONTinuous:Laser:Fire Command 24
4.26 SOURce:CONTinuous:Laser:Fire? Command 25
4.27 SOURce:PARameters:CURRent:TRACE? Command 25
5 Data Calculations – CALCulate Subsystem 26
5.1 CALCulate:MATH:EXPRession:Loss? Command 26
5.2 CALCulate:MATH:EXPRession:EELoss? Command 26
6 File Operations – The MMEMory Subsystem 27
6.1 MMEMory:LOAD:T6? Command 27
6.2 MMEMory:LOAD:T5? Command 27
6.3 MMEMory:LOAD:SOR? Command 28
6.4 MMEMory:LOAD:MODule? Command 28
6.5 MMEMory:SAVE:File Command 29
6.6 MMEMory:LOAD:T6Text? Command 29
7 Display Operations 32
7.1 DISPLay:TAB Command 32
7.2 DISPLay:TAB? Command 32
7.3 DISPLay:DISTance:UNits Command 33
7.4 DISPLay:DISTance:Units? Command 33
7.5 DISPLay:Format Command 34
7.6 DISPLay:Format? Command 34
7.7 DISPLay:Full:Zoom Command (Deprecated) 34
7.8 DISPLay:Zoom:Full Command 35
7.9 DISPLay:Zoom:Horizontal Command 35
7.10 DISPLay:Zoom:Horizontal? Command 35
7.11 DISPLay:Zoom:Vertical Command 36
7.12 DISPLay:Zoom:Vertical? Command 36
7.13 DISPLay:Module:INFO? Command 36

1 Command Summary

The Command Summary section provides a list of the SCPI commands for the CMA5000 Standard OTDR application, along with a brief definition for each.

ABORt
Stop test.

INITiate
Start test. This command requires average count or time as parameter.

INITiate:AUTo
Start auto-test.

INITiate?
Queries if test is active.

SENSe:AVERages:COMPLeted?
Queries number of averages that have been completed since the start of the test.

SENSe:FIBer:IOR
Sets index of refraction. This value will be used for the next test.

SENSe:FIBer:IOR?
Queries index of refraction.

SENSe:FIBer:BSC
Sets backscatter coefficient. This value will be used for the next test.

SENSe:FIBer:BSC?
Queries backscatter coefficient.

SENSe:TRACE:READY?
Queries if trace data is ready.

SOURce:HOFFset
Set horizontal offset for the displayed trace(s).

SOURce:HOFFset?
Query horizontal offset for the displayed trace(s).

SOURce:VOFFset
Set vertical offset for the displayed trace(s).

SOURce:VOFFset?
Query vertical offset for the displayed trace(s).

SOURce:WAVelength
Set current wavelength.

SOURce:WAVelength?
Query current wavelength.

SOURce:AMARKer:POINt
Set A marker position.

SOURce:AMARKer:POINt?
Query current A marker position.

SOURce:BMARKer:POINt
Set B marker position.

SOURce:BMARKer:POINt?
Query current B marker position.

SOURce:PULSe:WIDTh
Set current pulse width.

SOURce:PULSe:WIDTh?
Query current pulse width.

SOURce:RANGEReso
Set current range and resolution.

SOURce:RANGEReso?
Query current range and resolution.

SOURce:RANge:RESo:ALL?
Query the entire range and resolution table.

SOURce:WAVelength:AVAilable?
Query the list of available wavelengths.

SOURce:LSALeft
Set left LSA value.

SOURce:LSALeft?

Query left LSA value.

SOURce:LSARight
Set right LSA value.

SOURce:LSARight?
Query right LSA value.

SOURce:Loss:Mode
Set current Loss Mode.

SOURce:Loss:Mode?
Query current Loss Mode.

SOURce:ANALyze:ON
Set if analysis to be performed after the test.

SOURce:ANALyze:ON?
Query if analysis is set on.

SOURce:CONTinuous:Laser:Fire
Set continuous laser fire.

SOURce:CONTinuous:Laser:Fire?
Check if continuous laser fire.

SOURce:PARameters:CURRent:TRACE?
Query current trace parameters.

CALCulate:MATH:EXPRession:Loss?
Calculate loss.

CALCulate:MATH:EXPRession:EELoss?
Calculate end-to-end loss.

MMEMory:LOAD:T5?
Get T5 trace object.

MMEMory:LOAD:T6?
Get T6 trace object.

MMEMory:LOAD:SOR?
Get SOR trace object.

MMEMory:LOAD:T6Text?
Load T6 trace text data.

MMEMory:LOAD:MODule?
Get module data.

MMEMory:SAVE:File
Save current file.

DISPLay:TAB
Set instrument GUI tab.
DISPLay:TAB?
Queries current instrument GUI tab.

DISPLay:DISTance:Units
Set current distance units.

DISPLay:DISTance:Units?
Query current distance units.

DISPLay:Format
Set display format state.

DISPLay:Format?
Query display format

DISPLay:Full:Zoom (Deprecated)
Render full zoom on the trace screen.

DISPLayl:Zoom:Full
Render full zoom on the trace screen.

DISPLay:Zoom:Horizontal
Set current horizontal zoom level.

DISPLay:Zoom:Horizontal?
Query current horizontal zoom level.

DISPLay:Zoom:Vertical
Set current vertical zoom level.

DISPLay:Zoom:Vertical?
Query current vertical zoom level.

DISPLay:Module:INFO?

Display model name and serial number of the OTDR module.

2 Root Level Commands

2.1 ABORt Command

Syntax:	ABORt
Parameters:	None
Description:	Aborts the active test.
Example:	`abor`
Response:	None
Errors:	-200, “std_execGen, State is already IDLE!”

2.2 INITiate Command

Syntax:	INITiate<wsp><number of averages>,<is timed averaging>
Parameters:	<number of averages> Integer format Range: If <is timed averaging> parameter is 0, then <number of averages> parameter should be 8 – 21. These values are power of 2 exponents. If <is timed averaging> parameter is 1, then <number of averages> parameter should be 5 – 5995 seconds. If <number of averages> is 0 (zero), then the real-time rest will be initiated. In this case, second parameter is still required but will not be parsed. <is timed averaging> Integer format Range: 0|1 0 = <number of averages> parameter is used as power of 2 exponent in this case. 1 = <number of averages> parameter is used as number of seconds for the timed scan.
Description:	Starts test. This command requires average count or time as parameter. This command is an overlapping command, except if the test if real-time (<number of averages> parameter is equal to zero).
Example:	`init 20,1`
Response:	None
Error:	-200, “std_execGen, Test is already active!” -224, “std_illegalParmValue, Parameters are out of range!”

2.3 INITiate AUTo Command

Syntax:	INITiate:AUTo
Parameters:	None
Description:	Stars auto-test. All test parameters are automatically obtained during the pre-scan. This command is an overlapping command.
Example:	`init:aut`
Response:	None
Errors:	-200, “std_execGen, Test is already active!”

2.4 INITiate? Command

Syntax:	INITiate?
Parameters:	None
Description:	Queries if test is active.
Example:	`Init? --- 0<END>`
Response:	Possible response range: 0|1 0 = Test is NOT active. 1 = Test is currently active.
Errors:	None

3 Sensing Commands

The SENSe subsystem lets you control measurement parameters.

3.1 SENSe:AVERages:COMPleted? Command

Syntax:	SENSe:AVERages:COMPleted?
Parameters:	None
Description:	Queries number of averages that have been completed since the start of the test. Number of averages is equal to 128 during the Real Time test.
Example:	`sens:aver:comp? --- 4096<END>`
Response:	Number of averages
Errors:	-200, “std_execGen, No primary trace!”

3.2 SENSe:FIBer:IOR Command

Syntax:	SENSe:FIBer:IOR<wsp><value>
Parameters:	<value> Floating point format Range: 1.300000 - 1.700000
Description:	Sets index of refraction. This value will be used for the next test.
Example:	`sens:fib:ior 1.45`
Response:	None
Errors:	-224, “std_illegalParmValue, Parameter is out of range!”

3.3 SENSe:FIBer:IOR? Command

Syntax:	SENSe:FIBer:IOR?
Parameters:	None
Description:	Queries index of refraction.
Example:	`sens:fib:ior? --- 1.45<END>`
Response:	Possible response is value in range: 1.300000 - 1.700000
Errors:	None

3.4 SENSe:FIBer:BSC Command

Syntax:	SENSe:FIBer:BSC<wsp><value>
Parameters:	<value> Floating point format Range: -90.0 - -40.0
Description:	Sets backscatter coefficient. This value will be used for the next test.
Example:	`sens:fib:bsc -83.0`
Response:	None
Errors:	-224, “std_illegalParmValue, Parameter is out of range!”

3.5 SENSe:FIBer:BSC? Command

Syntax:	SENSe:FIBer:BSC?
Parameters:	None
Description:	Queries backscatter coefficient.
Example:	`sens:fib:bsc? --- -83.0<END>`
Response:	Possible response is value in range: -90.0 - -40.0
Errors:	None

3.6 SENSe:TRACE:READY? Command

Syntax:	SENSe:TRACE:READY?
Parameters:	None
Description:	Queries if trace data is ready.
Example:	`sens:trace:ready? --- true<END>`
Response:	Possible response values: true = trace data is ready and can be transferred or saved. false = no trace data available in the memory.
Errors:	None

4 Signal Generation – The SOURce Subsystem

The SOURce subsystem allows controlling of the OTDR's optical source. It also controls of the cursors.

4.1 SOURce:HOFFset Command

Syntax:	SOURce:HOFFset<wsp><value>
Parameters:	<value> Floating point format Range: Offset value can be set plus/minus current distance range.
Description:	Set horizontal offset for the displayed trace(s). Offset value units – KM.
Example:	`sour:hoff 10.0`
Response:	None
Errors:	None

4.2 SOURce:HOFFset? Command

Syntax:	SOURce:HOFFset?
Parameters:	None
Description:	Queries horizontal offset for the displayed trace(s).
Example:	`sour:hoff? --- 0<END>`
Response:	Current horizontal offset value.
Errors:	None

4.3 SOURce:VOFFset Command

Syntax:	SOURce:VOFFset<wsp><value>
Parameters:	<value> Floating point format Range: Offset value can be set plus/minus current distance range.
Description:	Set vertical offset for the displayed trace(s). Offset value units – DB..
Example:	`sour:voff -5.0`
Response:	None
Errors:	None

4.4 SOURce:VOFFset? Command

Syntax:	SOURce:VOFFset?
Parameters:	None
Description:	Queries vertical offset for the displayed trace(s).
Example:	`sour:voff? --- 0<END>`
Response:	Current vertical offset value.
Errors:	None

4.5 SOURce:WAVelength Command

Syntax:	SOURce:WAVelength<wsp><value>
Parameters:	<value> Floating point format Range: Offset value can be set plus/minus current dynamic range.
Description:	Set current wavelength. Wavelength value units – NM. Wavelength will be set only if the new value matches the one in the list of available once.
Example:	`sour:wav 1310`
Response:	None
Errors:	-224, “std_illegalParmValue, Invalid parameter value!”

4.6 SOURce:WAVelength? Command

Syntax:	SOURce:WAVelength?
Parameters:	None
Description:	Queries current wavelength.
Example:	`sour:wav? --- 1310<wsp>nm<END>`
Response:	Current wavelength value.
Errors:	None

4.7 SOURce:ACURsor:POINt Command

Syntax:	SOURce: ACURsor:POINt<wsp><value>
Parameters:	<value> Floating point format Range: 0.0 - 273.8043.
Description:	Set A cursor position. Cursor position units – KM.
Example:	`sour:acur:poin 20.5`
Response:	None
Errors:	-224, “std_illegalParmValue, Parameter is out of range!”

4.8 SOURce:ACURsor:POINt? Command

Syntax:	SOURce: ACURsor:POINt?
Parameters:	None.
Description:	Queries current A cursor position.
Example:	`sour:acur:poin? --- 20.5<END>`
Response:	Current A cursor position value.
Errors:	None

4.9 SOURce:BCURsor:POINt Command

Syntax:	SOURce: BCURsor:POINt<wsp><value>
Parameters:	<value> Floating point format Range: 0.0 - 273.8043.
Description:	Set B cursor position. Cursor position units – KM.
Example:	`sour:bcur:poin 20.5`
Response:	None
Errors:	-224, “std_illegalParmValue, Parameter is out of range!”

4.10 SOURce:BCURsor:POINt? Command

Syntax:	SOURce: BCURsor:POINt?
Parameters:	None.
Description:	Queries current B cursor position.
Example:	`sour:bcur:poin? --- 20.5<END>`
Response:	Current B cursor position value.
Errors:	None

4.11 SOURce:PULSe:WIDTh Command

Syntax:	SOURce:PULSe:WIDTh<wsp>< pulse width >,<mode>
Parameters:	<pulse width> Integer format Range: 5 – 30000 nanoseconds (NS). Available pulse width values are dependant on current range/resolution. Pulse width value will be ignored if it doesn't mach any value in the list of available once. <mode> Integer format Range: 0 - 7 1 = Sets Long Haul to be on 2 = Sets Gain Splice to be on 4 = Sets Box Car Filter to be on To enable more than one of these options, use sum of their corresponding values. Setting Long Haul and Gain Splice for example would require <mode>=3 (1 + 2). Setting Box Car Filter and Long Haul would require <mode>=5 (4 + 1) etc..
Description:	Set current pulse width.
Example:	`sour:puls:widt 1000,4`
Response:	None
Errors:	-224, "std_illegalParmValue, Parameter is out of range!"

4.12 SOURce:PULSe:WIDTh? Command

Syntax:	SOURce:PULSe:WIDTh?
Parameters:	None
Description:	Query current pulse width.
Example:	`sour:puls:widt? --- 1000,4<END>`
Response:	First value is current pulse width in nanoseconds (NS), the second value is pulse width mode 0 – 7. 1 = Sets Long Haul to be on 2 = Sets Gain Splice to be on 4 = Sets Box Car Filter to be on For example 3 would mean Gain Splice and Long Haul (2 + 1).
Errors:	None

4.13 SOURce:RANge:RESo Command

Syntax:	SOURce:RANge:RESo<wsp><range>,<resolution>
Parameters:	<range> Integer format Range: 5 – 300 kilometers (KM). <resolution> Floating point format Range: 0.125 – 16.0 meters (M).
Description:	Set current range and resolution.
Example:	`sour:ran:res 1000,4`
Response:	None
Errors:	-224, “std_illegalParmValue, Parameter is out of range!”

4.14 SOURce:RANge:RESo? Command

Syntax:	SOURce:RANge:RESo?
Parameters:	None
Description:	Query current range and resolution.
Example:	`sour:ran:res? --- 50,4.0 <END>`
Response:	First value is current range in kilometers (KM). Second value is current resolution in meters (M).
Errors:	None

4.15 SOURce:RANge:RESo:ALL? Command

Syntax:	SOURce:RANge:RESo:ALL?
Parameters:	None
Description:	Query the entire range and resolution table for every wavelength.
Example:	`sour:ran:res:all? ---` `1310.0,5.0,0.125,1310.0,5.0,0.5,1310.0,5.0,2.0,1310.0,20.0,0.125,1310.0,20.0,1.0,1310.0,20.0,4.0,1310.0,50.0,0.25,1310.0,50.0,1.0,1310.0,50.0,4.0,1310.0,75.0,0.5,1310.0,75.0,2.0,1310.0,75.0,8.0,1310.0,125.0,0.5,1310.0,125.0,2.0,1310.0,125.0,8.0,1310.0,250.0,1.0,1310.0,250.0,4.0,1310.0,250.0,16.0,1310.0,300.0,2.0,1310.0,300.0,4.0,1310.0,300.0,16.0,1550.0,5.0,0.125,1550.0,5.0,0.5,1550.0,5.0,2.0,1550.0,20.0,0.125,1550.0,20.0,1.0,1550.0,20.0,4.0,1550.0,50.0,0.25,1550.0,50.0,1.0,1550.0,50.0,4.0,1550.0,75.0,0.5,1550.0,75.0,2.0,1550.0,75.0,8.0,1550.0,125.0,0.5,1550.0,125.0,2.0,1550.0,125.0,8.0,1550.0,250.0,1.0,1550.0,250.0,4.0,1550.0,250.0,16.0,1550.0,300.0,2.0,1550.0,300.0,4.0,1550.0,300.0,16.0,1625.0,5.0,0.125,1625.0,5.0,0.5,1625.0,5.0,2.0,1625.0,20.0,0.125,1625.0,20.0,1.0,1625.0,20.0,4.0,1625.0,50.0,0.25,1625.0,50.0,1.0,1625.0,50.0,4.0,1625.0,75.0,0.5,1625.0,75.0,2.0,1625.0,75.0,8.0,1625.0,125.0,0.5,1625.0,125.0,2.0,1625.0,125.0,8.0,1625.0,250.0,1.0,1625.0,250.0,4.0,1625.0,250.0,16.0,1625.0,300.0,2.0,1625.0,300.0,4.0,1625.0,300.0,16.0 <END>`
Response:	Comma separated floating point values. Wavelength in nanometers (NM) followed by range in kilometers (KM), followed by resolution in meters (M).
Errors:	None

4.16 SOURce:WAVelength:AVAilable? Command

Syntax:	SOURce:WAVelength:AVAilable?
Parameters:	None
Description:	Queries the list of available wavelengths.
Example:	`sour:wav:ava? --- 1310,1550,<END>`
Response:	List of available wavelength in nanometers (NM).
Errors:	None

4.17 SOURce:LSALeft Command

Syntax:	SOURce:LSALeft<wsp><value>
Parameters:	<left> Floating point format Range: -273.8043 - 273.8043. <right> Floating point format Range: -273.8043 - 273.8043.
Description:	Set start and stop for left LSA marker. Start and stop units – kilometers (KM).
Example:	`sour:lsal 0.0,0.5`
Response:	None
Errors:	-224, “std_illegalParmValue, Parameters are out of range!”

4.18 SOURce:LSALeft? Command

Syntax:	SOURceLSALeft?
Parameters:	None
Description:	Query left LSA value.
Example:	`sour:lsal? --- 0.0,0.5<END>`
Response:	Start and stop for left LSA marker. Start and stop units – kilometers (KM).
Errors:	None

4.19 SOURce:LSARight Command

Syntax:	SOURce:LSARight<wsp><value>
Parameters:	<left> Floating point format Range: -273.8043 - 273.8043. <right> Floating point format Range: -273.8043 - 273.8043.
Description:	Set start and stop for right LSA marker. Start and stop units – kilometers (KM).
Example:	`sour:lsar 0.0,0.5`
Response:	None
Errors:	-224, “std_illegalParmValue, Parameters are out of range!”

4.20 SOURce:LSARight? Command

Syntax:	SOURceLSARight?
Parameters:	None
Description:	Query right LSA value.
Example:	`sour:lsar? --- 0.0,0.5<END>`
Response:	Start and stop for right LSA marker. Start and stop units – kilometers (KM).
Errors:	None

4.21 SOURce:Loss:Mode Command

Syntax:	SOURce:Loss:Mode<wsp><mode>
Parameters:	<mode> Integer format Range: 0|1|2|3|4|5|6 0 = Splice Loss Mode 1 = Two-point Loss Mode 2 = Two-point LSA 3 = dB/km Loss Mode 4 = dB/km LSA Loss Mode 5 = ORL Loss Mode 6 = Two-point Attenuation Corrected Loss Mode
Description:	Set current Loss Mode.
Example:	`sour:l:m 0`
Response:	None
Errors:	-224, “std_illegalParmValue, Parameter is out of range!

4.22 SOURce:Loss:Mode? Command

Syntax:	SOURce:Loss:Mode?
Parameters:	none
Description:	Query current Loss Mode.
Example:	`sour:l:m? --- 0<END>`
Response:	Possible responses are: 0 = Splice Loss Mode 1 = Two-point Loss Mode 2 = Two-point LSA 3 = dB/km Loss Mode 4 = dB/km LSA Loss Mode 5 = ORL Loss Mode 6 = Two-point Attenuation Corrected Loss Mode
Errors:	None

4.23 SOURce:ANALyze:ON Command

Syntax:	SOURce:ANALyze:ON<wsp><value>
Parameters:	<value> Integer format Range: 0|1 0 = Auto-analysis is OFF. 1 = Auto-analysis is ON.
Description:	Set if analysis to be performed automatically after the test.
Example:	`sour:anal:on 0`
Response:	None
Errors:	None

4.24 SOURce:ANALyze:ON? Command

Syntax:	SOURce:ANALyze:ON?
Parameters:	None
Description:	Query if auto-analysis is set on. Trace is automatically analyzed after test completion.
Example:	`sour:anal:on? --- 0<END>`
Response:	Possible responses are: 0 = Auto-analysis is OFF. 1 = Auto-analysis is ON.
Errors:	None

4.25 SOURce:CONTinuous:Laser:Fire Command

Syntax:	SOURce:CONTinuous:Laser:Fire<wsp><value>
Parameters:	<value> Integer format Range: 0|1 0 = Laser does not fire continuously. 1 = Laser fires continuously.
Description:	Set continuous laser fire.
Example:	`sour:cont:l:f 0`
Response:	None
Errors:	None

4.26 SOURce:CONTinuous:Laser:Fire? Command

Syntax:	SOURce:CONTinuous:Laser:Fire?
Parameters:	None
Description:	Check if continuous laser fire.
Example:	`sour:cont:l:f? --- 0<END>`
Response:	Possible responses are: 0 = Laser does not fire continuously. 1 = Laser fires continuously.
Errors:	None

4.27 SOURce:PARameters:CURRent:TRACE? Command

Syntax:	SOURce:PARameters:CURRent:TRACE?
Parameters:	None
Description:	Query current trace parameters.
Example:	`sour:par:curr:trace?---50.0,4.0,200,true,1550,5120,1.5,-82.111111,0.0,0.0,CMA5000,,2001-1343,Other,T6,,0.00,0.00,0.00<END>`
Response:	<range> = Integer format. Distance range in kilometers. <reso> = Floating point format. Resolution in meters. <pw> = Integer format. Pulse width in nanoseconds. <hiRes> = Boolean format. Is pulse high resolution. Boolean value. “true” – high resolution, “false” – long haul. <wl> = Integer format. Nominal wavelength in nanometers. <avg> = Integer format. Number of averages completed. <ior> = Floating point format. Index of refraction. <bsc> = Floating point format. Backscatter coefficient. <hShift> = Floating point format. Horizontal shift in kilometers. <vShift> = Floating point format. Vertical shift in decibels. <prodType> = String format. Product type. <optMod> = String format. Optical module type. <optModSN> = String format. Optical module serial number. <fibType> = String format. Fiber type. <trcType> = String format. Trace type. <trcFlags> = String format. Trace flags. <fasLossThresh> = Floating point format. FAS loss threshold in decibels. <fasReflThresh> = Floating point format. FAS reflectance threshold in decibels. <fasFibBreakThresh> = Floating point format. FAS fiber break threshold in decibels.
Errors:	-200, “std_execGen, No primary trace!”

5 Data Calculations – CALCulate Subsystem

The CALCulate subsystem allows the executing of loss calculation expressions.

5.1 CALCulate:MATH:EXPRession:Loss? Command

Syntax:	CALCulate:MATH:EXPRession:Loss?
Parameters:	None
Description:	Calculate loss. The calculated value is depends on current loss mode.
Example:	`calc:math:expr:l? --- -4.610<END>`
Response:	Calculated loss value.
Errors:	None

5.2 CALCulate:MATH:EXPRession:EELoss? Command

Syntax:	CALCulate:MATH:EXPRession:EELoss?
Parameters:	None
Description:	Calculate end-to-end loss. The calculated value is depends on current loss mode.
Example:	`calc:math:expr:eel? --- -4.610<END>`
Response:	Calculated end-to-end loss value.
Errors:	None

6 File Operations – The MMEMory Subsystem

The MMEMory subsystem provides access to the currently loaded file.

6.1 MMEMory:LOAD:T6? Command

Syntax:	MMEMory:LOAD:T6?
Parameters:	None
Description:	Get T6 trace object.
Example:	`mmem:load:t6? --- "BINARY ARRAY"<END>`
Response:	T6 as array of bytes. Following is an example of first few bytes of the "BINARY ARRAY" response: #43255… # – First byte to signify that the binary data is coming 4 – Second byte is a digit and the digit value indicates the number of size bytes that follow. 3255 – In this example, the next 4 bytes make number 3255 which is the number of data bytes to be received. … – Actual data bytes that need to be received and stores in a file.
Errors:	-200, "std_execGen, Test is active!" -200, "std_execGen, No primary trace!"

6.2 MMEMory:LOAD:T5? Command

Syntax:	MMEMory:LOAD:T5?
Parameters:	None
Description:	Get T5 trace object.
Example:	`mmem:load:t5? --- "BINARY ARRAY"<END>`
Response:	T5 as array of bytes. Following is an example of first few bytes of the "BINARY ARRAY" response: #43255… # – First byte to signify that the binary data is coming 4 – Second byte is a digit and the digit value indicates the number of size bytes that follow. 3255 – In this example, the next 4 bytes make number 3255 which is the number of data bytes to be received. … – Actual data bytes that need to be received and stores in a file.
Errors:	-200, "std_execGen, Test is active!" -200, "std_execGen, No primary trace!"

6.3 MMEMory:LOAD:SOR? Command

Syntax:	MMEMory:LOAD:SOR?
Parameters:	None
Description:	Get SOR trace object.
Example:	`mmem:load:sor? --- “BINARY ARRAY”<END>`
Response:	SOR as array of bytes. Following is an example of first few bytes of the “BINARY ARRAY” response: #43255… # – First byte to signify that the binary data is coming 4 – Second byte is a digit and the digit value indicates the number of size bytes that follow. 3255 – In this example, the next 4 bytes make number 3255 which is the number of data bytes to be received. … – Actual data bytes that need to be received and stores in a file.
Errors:	-200, “std_execGen, Test is active!” -200, “std_execGen, No primary trace!”

6.4 MMEMory:LOAD:MODule? Command

Syntax:	MMEMory:LOAD:MODule?
Parameters:	None
Description:	Get module data.
Example:	`mmem:load:mod? --- “BINARY ARRAY”<END>`
Response:	Module data as array of bytes. Following is an example of first few bytes of the “BINARY ARRAY” response: #43255… # – First byte to signify that the binary data is coming 4 – Second byte is a digit and the digit value indicates the number of size bytes that follow. 3255 – In this example, the next 4 bytes make number 3255 which is the number of data bytes to be received. … – Actual data bytes that need to be received and stores in a file.
Errors:	None

6.5 MMEMory:SAVE:File Command

Syntax:	MMEMory:SAVE:File<wsp><path><wsp><name>
Parameters:	<path> String format The parameter <path> is a legal, full Windows path to the directory which is to be set as the directory to write the file to. <path> consists of a legal Windows drive designator followed by an optional sequence of Windows directories separated by backslashes (i.e. '\'). For example a <path> consisting only of drive designator would have the form: d:\ - This <path> points to the root directory of a system's d: drive. More extensive examples of <path>'s are: d:\traces d:\traces\data\january <name> String format Filename must be in Windows compatible format.
Description:	Save current file. Both parameters have to be quoted strings.
Example:	`Mmem:save:f "d:\\mypath" "myname.trc"`
Response:	None
Errors:	-200, "std_execGen, Path does not exist!" -200, "std_execGen, Filename does already exist!" -200, "std_execGen, Error while saving file!" -200, "std_execGen, No primary trace or test is active!"

6.6 MMEMory:LOAD:T6Text? Command

Syntax:	MMEMory:LOAD:T6Text?
Parameters:	None
Description:	Get T6 trace text data.

Example:	`mmem:load:t6t? --- "BINARY ARRAY"<END>`
	
	`Following is an example of first few bytes of the "BINARY ARRAY" response: #43255…`
	
	`# – First byte to signify that the binary data is coming`
	`4 – Second byte is a digit and the digit value indicates the number of size bytes that follow.`
	`3255 – In this example, the next 4 bytes make number 3255 which is the number of data bytes to be received.`
	`… – Actual data bytes that need to be received and stores in a file.`
	
	`Here is also the example of the data bytes content when saved as the text file:`

```
   20              "// Number of header lines, including this line."
"T6TrcText - Version 04/12/02"
FN   = "AUTO13100000.SOR"
PN   = "AUTO13100000.SOR"
TYPE = "Anr SR4731"
INST = "MT9083B-053"
OPTC = ""
WL   = 1310 nm
PW   = 200 ns
HRLH = "[H]"
FBR  = "Singlemode"
AVG  = 1024
IOR  = 1.467700
BSC  = -78.50 dB
DATE = "11/06/08"
TIME = "10:23 AM"
MXDB = 64 dB
RESO = 1.001 m
DX   = 1.02200707368672 m
PTS  = 25001
 1000            "// Scale Factor."
 5783
46324
46563
46876
...
29365
29365
29365
Events  = 1
Type                = E
Location            = 4.4519 km
Loss                = >3.00 dB
Event-Event Loss    = 1.57 dB
Event-Event Loss/km = 0.353 dB
Reflectance         = -38.43  dB

Created text file contains the following lines :

   {Number of lines in the header}             "// Number of header
lines, including this line."
"T6TrcText - Version {Trace text utility version}"
FN   = "{Trace filename}"
```

	PN = "{Trace file pathname+filename}" TYPE = "{Trace type e.g. T5, T6, SR4731}" INST = "{Instrument}" OPTC = "{Optics module}" WL = {Wavelength in nm} nm PW = {PulseWidth in nsec} ns HRLH = "{HiRes/Long Haul preamp, [H]/[L]}" FBR = "{Fiber Type e.g. Singlemode or Multimode}" AVG = {Number of Averages} IOR = {IOR value} BSC = {Backscatter Coefficient in dB} dB DATE = "{Date}" TIME = "{Time}" MXDB = {Maximum display range in dB} dB RESO = {Nominal Resolution in meters} m DX = {Distance between data points in meters} m PTS = {Number of Data Points} {Scale Factor} "// Scale Factor." {Data Point Value as Integer Number. The formula for calculating the value as displayed on the screen is ((Data Point Value / Scale Factor) - MXDB)} ... {Data Point Value as an Integer Number} Events = {Number of Events} Type = {Event Type e.g. R = Reflective, N = Non-reflective, G = Grouped, E = End, ? = Questionable End} Location = {Event Location in current distance units xx} xx Loss = {Loss Value} dB Event-Event Loss = {Event to Event Loss Value} dB Event-Event Loss/km = {Event to Event Loss Value per kilometer} dB Reflectance = {Reflectance Value} dB
Response:	T6 trace text data as array of bytes.
Errors:	-200, “std_execGen, Test is active!” -200, “std_execGen, No primary trace!” -200, “std_execGen, Cannot get text data!”

7 Display Operations

The DISPlay subsystem allows you to control what you see on the OTDR's display.

7.1 DISPLay:TAB Command

Syntax:	DISPLay:TAB<wsp><value>
Parameters:	<value> Integer format Range: 0|1|2|3 0 = Display Trace tab 1 = Display Analysis tab 2 = Display File Manager tab 3 = Display Help tab
Description:	Switches particular instrument tab pane to be the foreground one.
Example:	`disp:tab 0`
Response:	None
Errors:	-224, "std_illegalParmValue, Parameter is out of range!"

7.2 DISPLay:TAB? Command

Syntax:	DISPLay:TAB?
Parameters:	None
Description:	Queries current foreground tab.
Example:	`displ:tab? --- 0<END>`
Response:	Possible responses are: 0 = Display Trace tab 1 = Display Analysis tab 2 = Display File Manager tab 3 = Display Help tab
Errors:	None

7.3 DISPLay:DISTance:UNits Command

Syntax:	DISPLay:DISTance:Units<wsp><value>
Parameters:	<value> Integer format Range: 0|1|2|3|4 0 = Sets units to miles (mi) 1 = Sets units to feet (ft) 2 = Sets units to kilofeet (kft) 3 = Sets units to miters (m) 4 = Sets units to kilometers (km)
Description:	Set the units used by instrument. All length measurements will be given in the units set by this command.
Example:	`dipl:dist:un 0`
Response:	None
Errors:	-224, "std_illegalParmValue, Parameter is out of range!"

7.4 DISPLay:DISTance:Units? Command

Syntax:	DISPLay:DISTance:UNits?
Parameters:	None
Description:	Queries current distance units.
Example:	`displ:dist:un? --- 0<END>`
Response:	Possible responses are: 0 = Units in miles (mi) 1 = Units in feet (ft) 2 = Units in kilofeet (kft) 3 = Units in miters (m) 4 = Units in kilometers (km)
Errors:	None

7.5 DISPLay:Format Command

Syntax:	DISPLay:Format<wsp><value>
Parameters:	<value> Integer format Range: 0|1|2|3 0 = display from Origin 1 = display from Cursor A 2 = display from cursor B 3 = display from Anywhere
Description:	Allows set up of the display format.
Example:	`displ:f 0`
Response:	None
Errors:	-224, "std_illegalParmValue, Parameter is out of range!"

7.6 DISPLay:Format? Command

Syntax:	DISPLay:Format?
Parameters:	None
Description:	Queries current display format.
Example:	`displ:f? --- 0<END>`
Response:	Possible responses are: 0 = display from Origin 1 = display from Cursor A 2 = display from cursor B 3 = display from Anywhere
Errors:	None

7.7 DISPLay:Full:Zoom Command (Deprecated)

Syntax:	DISPLay:Full:Zoom
Parameters:	None
Description:	Renders full zoom at display from anywhere OTDR trace screen. This command is deprecated do to its name. Please use DISPLay:Zoom:Full instead.
Example:	`displ:f:z`
Response:	None
Errors:	None

7.8 DISPLay:Zoom:Full Command

Syntax:	DISPLay:Zoom:Full
Parameters:	None
Description:	Renders full zoom at display from anywhere OTDR trace screen. For every combination of view (A, B, Origin, Anywhere) / tab (Trace, Analysis), there is separate zoom level.
Example:	`displ:z:f`
Response:	None
Errors:	None

7.9 DISPLay:Zoom:Horizontal Command

Syntax:	DISPLay:Zoom:Horizontal<wsp><value>
Parameters:	<value> Integer format Range: 0 – 13 The parameter range is variable and depends on current range/resolution settings with maximum value from 9 to 13.
Description:	Sets horizontal zoom level as number of expansions (number of right key presses from full zoom). For every combination of view (A, B, Origin, Anywhere) / tab (Trace, Analysis), there is separate zoom level.
Example:	`displ:z:h 0`
Response:	None
Errors:	-224, "std_illegalParmValue, Parameter is out of range!" -200, "std_execGen, Invalid Tab Selected!"

7.10 DISPLay:Zoom:Horizontal? Command

Syntax:	DISPLay: Zoom:Horizontal?
Parameters:	None
Description:	Queries current horizontal zoom level. For every combination of view (A, B, Origin, Anywhere) / tab (Trace, Analysis), there is separate zoom level.
Example:	`displ:z:h? --- 0<END>`
Response:	Value form 0 to 13.
Errors:	None

7.11 DISPLay:Zoom:Vertical Command

Syntax:	DISPLay:Zoom:Vertical<wsp><value>
Parameters:	<value> Integer format Range: 0 – 13 The parameter range is variable and depends on current range/resolution settings with maximum value from 9 to 13.
Description:	Sets vertical zoom level as number of expansions (number of right key presses from full zoom). For every combination of view (A, B, Origin, Anywhere) / tab (Trace, Analysis), there is separate zoom level.
Example:	`displ:z:v 0`
Response:	None
Errors:	-224, "std_illegalParmValue, Parameter is out of range!" -200, "std_execGen, Invalid Tab Selected!"

7.12 DISPLay:Zoom:Vertical? Command

Syntax:	DISPLay: Zoom:Vertical?
Parameters:	None
Description:	Queries current vertical zoom level. For every combination of view (A, B, Origin, Anywhere) / tab (Trace, Analysis), there is separate zoom level.
Example:	`displ:z:v? --- 0<END>`
Response:	Value form 0 to 13.
Errors:	None

7.13 DISPLay:Module:INFO? Command

Syntax:	DISPLay:MODule:INFO?
Parameters:	None
Description:	Displays model name and serial number of the OTDR module.
Example:	`displ:mod:info? --- 5236-111-OTDR-1,N/A`
Response:	Two comma separated strings. If either of those are invalid, N/A will be displayed.
Errors:	None