ADDENDUM

This information applies only to the OTDR modules.

SCPI Commands for Your Module

This addendum presents detailed information on the commands and queries supplied with your OTDR module.

IMPORTANT

Since the platforms can house many instruments, you must explicitly specify which instrument you want to remotely control.

You must add the following mnemonic *at the beginning of any command or query* that you send to an instrument:

LINStrument<LogicalInstrumentPos>:

where *<LogicalInstrumentPos>* corresponds to the identification number of the instrument.

Use the LINS value defined in the Remote Control Configuration tool (accessible from System Settings). For information on modifying the LINS value, refer to your platform user guide.

Quick Reference Command Tree

Command						Parameter(s)	P.
ABORt[1..n]							6
CALCulate[1..n]	ANAlysis	[UNIDirectional]				TRC1|TRC2|TRC3|TRC4	7
	ATTenuation?					TRC1|TRC2|TRC3|TRC4,<MarkerA>,<MarkerB>	8
	CLValue?					TRC1|TRC2|TRC3|TRC4,<MarkerA>	10
	EVENt?					TRC1|TRC2|TRC3|TRC4,<EventIndex>	11
	EVENt	COUNt?				TRC1|TRC2|TRC3|TRC4	13
		STATus?				TRC1|TRC2|TRC3|TRC4,<EventIndex>	14
		SUB?				TRC1|TRC2|TRC3|TRC4,<EventIndex>,<SubEventIndex>	17
		SUB	COUNt?			TRC1|TRC2|TRC3|TRC4,<SubEventIndex>	19
			STATus?			TRC1|TRC2|TRC3|TRC4,<EventIndex>,<SubEventIndex>	21
	HFACtor					TRC1|TRC2|TRC3|TRC4,<HelixFactor>	24
	HFACtor?					TRC1|TRC2|TRC3|TRC4	25
	INJection	[LEVel]?				TRC1|TRC2|TRC3|TRC4	26
	IORefraction					TRC1|TRC2|TRC3|TRC4,<IOR>	27
	IORefraction?					TRC1|TRC2|TRC3|TRC4	28
	LOSS?					TRC1|TRC2|TRC3|TRC4,<MarkerA>,<MarkerB>	29
	ORL?					TRC1|TRC2|TRC3|TRC4,<MarkerA>,<MarkerB>	31
	REFLectance?					TRC1|TRC2|TRC3|TRC4,<SubMarkerA>,<MarkerA>,<MarkerB>	33
	RBScatter					TRC1|TRC2|TRC3|TRC4,<RBS>	35
	RBScatter?					TRC1|TRC2|TRC3|TRC4	36
	SLOSs?					TRC1|TRC2|TRC3|TRC4,<SubMarkerA>,<MarkerA>,<MarkerB>,<SubMarkerB>	37
	THReshold	EOFiber				TRC1|TRC2|TRC3|TRC4,<End-of-Fiber>	39
		EOFiber?				TRC1|TRC2|TRC3|TRC4	40

Command						Parameter(s)	P.
		REFLectance				TRC1|TRC2|TRC3|TRC4,<Reflectance>	41
		REFLectance?				TRC1|TRC2|TRC3|TRC4	42
		SLOSs				TRC1|TRC2|TRC3|TRC4,<Splice Loss>	43
		SLOSs?				TRC1|TRC2|TRC3|TRC4	44
	TORL?					TRC1|TRC2|TRC3|TRC4	45
CONFigure[1..n]	ACQuisition					<Wavelength>,<Range>,<Pulse>	46
		DURation				<Duration>|MAXimum|MINimum|DEFault	47
		DURation?				[MINimum|MAXimum|DEFault]	48
		HRESolution				<HighResolution>	49
		HRESolution?					50
		MODE				ACQuisition|ASETting|CFConnector|REAltime	51
		MODE?					52
		PULSe?					53
		PULSe	LIST?			<Wavelength>,<Range>	54
		RANGe?					55
		RANGe	LIMit	HIGH?		<Wavelength>	56
				LOW?		<Wavelength>	57
			LIST?			<Wavelength>	58
		WAVelength?					59
		WAVelength	LIST?				60
	ANAlysis	HFACtor				<HelixFactor>|MAXimum|MINimum|DEFault	61
		HFACtor?				[MINimum|MAXimum|DEFault]	62
		IORefraction				<IOR>|MAXimum|MINimum|DEFault	63
		IORefraction?				[MINimum|MAXimum|DEFault]	64
		RBScatter				<RBS>|MAXimum|MINimum|DEFault	65
		RBScatter?				[MINimum|MAXimum|DEFault]	66
		THReshold	EOFiber			<End-of-Fiber>|MAXimum|MINimum|DEFault	67
			EOFiber?			[MINimum|MAXimum|DEFault]	68

Command						Parameter(s)	P.
			REFLectance			<Reflectance>|MAXimum|MINimum|DEFault	69
			REFLectance?			[MINimum|MAXimum|DEFault]	70
			SLOSs			<Splice Loss>|MAXimum|MINimum|DEFault	71
			SLOSs?			[MINimum|MAXimum|DEFault]	72
ERRor[1..n]?							73
FETCh[1..n]	ASETting	PULSe?					74
		RANGe?					75
	CFConnector?						76
	DURation?					TRC1|TRC2|TRC3|TRC4	77
	HRESolution?					TRC1|TRC2|TRC3|TRC4	78
	LFIBer?						79
	PULSe?					TRC1|TRC2|TRC3|TRC4	80
	RANGe?					TRC1|TRC2|TRC3|TRC4	81
	STEP?					TRC1|TRC2|TRC3|TRC4	82
	TRACe[1..n]	[DATA]?					83
		POINts?					84
	WAVelength?					TRC1|TRC2|TRC3|TRC4	85
INITiate[1..n]	[IMMediate]						86
	STATe?						87
MMEMory[1..n]	DATA	TYPE				BINary|BELLcore	88
		TYPE?					89
	LOAD	NAME?					90
		TRACe				<FileName>	91
	STORe	TRACe				<FileName>	92
			OVERwrite			<Overwrite>	93
			OVERwrite?				94
SOURce[1..n]	FREQuency	BURSt				<BurstFrequency>|MAXimum|MINimum|DEFault	95
		BURSt?				[MINimum|MAXimum|DEFault]	96
		BURSt	STATe			<State>	97
			STATe?				98

Command						Parameter(s)	P.
		PRF				<PulsedRepetitionFrequency>|MAXimum|MINimum|DEFault	99
		PRF?				[MINimum|MAXimum|DEFault]	100
		PRF	STATe			<State>	101
			STATe?				102
	POWer	STATe				<State>	103
		STATe?					104
		STATe	TIME			<Duration>	105
			TIME?				106
	VFLocator	AM	INTernal	FREQuency		<Frequency>|MAXimum|MINimum|DEFault	107
				FREQuency?		[MINimum|MAXimum|DEFault]	108
			STATe			<State>	109
			STATe?				110
		POWer	STATe			<State>	111
			STATe?				112
			STATe	TIME		<Duration>|MAXimum|MINimum|DEFault	113
				TIME?		[MINimum|MAXimum|DEFault]	114
	WAVelength					<Wavelength>	115
	WAVelength?						116
	WAVelength	LIST?					117
TRACe[1..n]	[DATA]?					TRC1|TRC2|TRC3|TRC4	118
	CATalog?						119
	POINts?					TRC1|TRC2|TRC3|TRC4	120

Product-Specific Commands—Description

:ABORt[1..n]

Description	This command is used to stop the scan, measurement or acquisition in progress. This command is an event and, therefore, has no associated *RST condition or query form. However, on *RST, the equivalent of an ABORt command is performed on any acquisition in progress. *RST does not affect this command.
Syntax	:ABORt[1..n]
Parameter(s)	None
Example(s)	INIT ABOR
See Also	INITiate[1..n]:STATe? ERRor[1..n]?

:CALCulate[1..n]:ANAlysis [:UNIDirectional]

Description

This command performs a unidirectional analysis. It creates or modifies the event table for the specified trace index acquisition data.

For this command to be accepted, at least one acquisition must be performed.

*RST does not affect this command.

Syntax

:CALCulate[1..n]:ANAlysis[:UNIDirectional]<wsp>TRC1|TRC2|TRC3|TRC4

Parameter(s)

Label:

The program data syntax for the first parameter is defined as a <CHARACTER PROGRAM DATA> element. The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: TRC1|TRC2|TRC3|TRC4.

Trace index of the available wavelengths.

Example(s)

CONF:ACQ:MODE ACQUISITION
INIT
INIT:STAT? Returns 0 when acquisition is complete.
CALC:ANA TRC1

See Also

CALCulate[1..n]:EVENt:COUNt?
CALCulate[1..n]:EVENt?
MMEMory[1..n]:LOAD:TRACe
TRACe[1..n]:CATalog?

:CALCulate[1..n]:ATTenuation?

Description

This query returns the value of the attenuation measured between two markers, for the trace corresponding to the specified trace index.

*RST clears this setting.

Syntax

:CALCulate[1..n]:ATTenuation?<wsp>TRC1|TRC2|TRC3|TRC4,<MarkerA>,<MarkerB>

Parameter(s)

➤ *Label:*

The program data syntax for the first parameter is defined as a <CHARACTER PROGRAM DATA> element. The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: TRC1|TRC2|TRC3|TRC4.

Trace index of the available wavelengths.

➤ *MarkerA:*

The program data syntax for <MarkerA> is defined as a <DECIMAL NUMERIC PROGRAM DATA> element.

Specifies the marker A position, in meters.

➤ *MarkerB:*

The program data syntax for <MarkerB> is defined as a <DECIMAL NUMERIC PROGRAM DATA> element.

Specifies the marker B position, in meters.

Response Syntax

<Attenuation>

:CALCulate[1..n]:ATTenuation?

Response(s)	*Attenuation:* The response data syntax for <Attenuation> is defined as a <NR3 NUMERIC RESPONSE DATA> element. Returns the attenuation value in dB/meter, between marker A and marker B.
Example(s)	CONF:ACQ:MODE ACQUISITION INIT INIT:STAT? Returns 0 when acquisition is complete. CALC:ATT? TRC1,0.0,102.6 Ex.: Returns 1.963 CALC:ATT? TRC1,0.0 M,0.1026 KM Ex.: Returns 1.963 CALC:ATT? TRC1,0.0 KM,102.6 M Ex.: Returns 1.963
See Also	MMEMory[1..n]:LOAD:TRACe TRACe[1..n]:CATalog?

:CALCulate[1..n]:CLValue?

Description

This query returns the curve level value at a specific position, for the trace corresponding to the specified trace index.

*RST clears this setting.

Syntax

:CALCulate[1..n]:CLValue?<wsp>TRC1|TRC2|TRC3|TRC4,<MarkerA>

Parameter(s)

➤ *Label:*

The program data syntax for the first parameter is defined as a <CHARACTER PROGRAM DATA> element. The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: TRC1|TRC2|TRC3|TRC4.

Trace index of the available wavelengths.

➤ *MarkerA:*

The program data syntax for <MarkerA> is defined as a <DECIMAL NUMERIC PROGRAM DATA> element.

Specifies the marker A position, in meters.

Response Syntax

<Current Level Value>

Response(s)

Current Level Value:

The response data syntax for <Current Level Value> is defined as a <NR3 NUMERIC RESPONSE DATA> element.

Returns the curve level value in dB, at the position specified by marker A.

Example(s)

CONF:ACQ:MODE ACQUISITION
INIT
INIT:STAT? Returns 0 when acquisition is complete.
CALC:CLV? TRC1,100.3 Ex.: Returns –20.371
CALC:CLV? TRC1,0.1003 KM Ex.: Returns –20.371
CALC:CLV? TRC1,100.3 M Ex.: Returns –20.371

See Also

CALCulate[1..n]:ANAlysis:[UNIDirectional]
CALCulate[1..n]:EVENt:COUNt?
CALCulate[1..n]:EVENt?
MMEMory[1..n]:LOAD:TRACe
TRACe[1..n]:CATalog?

:CALCulate[1..n]:EVENt?

Description

This query returns an event from the event table after performing an analysis on the trace corresponding to the specified trace index.
You must supply the index of the event that you want to retrieve.

*RST clears the event table.

Syntax

:CALCulate[1..n]:EVENt?<wsp>TRC1|TRC2|TRC3|TRC4,<EventIndex>

Parameter(s)

- *Label:*

The program data syntax for the first parameter is defined as a <CHARACTER PROGRAM DATA> element. The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: TRC1|TRC2|TRC3|TRC4.

Trace index of the available wavelengths.

- *EventIndex:*

The program data syntax for <EventIndex> is defined as a <DECIMAL NUMERIC PROGRAM DATA> element.

Sets the event index. This value must be between 1 and the total number of events.

Response Syntax

<Event>

Response(s)

Event:

The response data syntax for <Event> is defined as a <DEFINITE LENGTH ARBITRARY BLOCK RESPONSE DATA> element.

Returns the event from the event table corresponding to the specified trace index.

Event structure is in A, B, C, D, E format, where:
A = Location (always in meters) <NR3 NUMERIC RESPONSE DATA>
B = EventType <NR1 NUMERIC RESPONSE DATA>
C = Loss (always in dB) <NR3 NUMERIC RESPONSE DATA>

D = Reflectance (always in dB) <NR3 NUMERIC RESPONSE DATA>
E = Cumulative (always in dB) <NR3 NUMERIC RESPONSE DATA>

Here is the list of all possible event types:
1 = Positive splice
2 = Negative splice
3 = Reflection
4 = End of analysis
5 = Continuous fiber

The End of analysis event does not necessarily correspond to the last event of a fiber link. It indicates that the analysis has stopped before the end of the link because the instrument has reached the limit of its dynamic range.
In most cases, the OTDR analysis will return the type of the last event as being either reflective or non-reflective (event type 3 or 2).

Example(s)

CONF:ACQ:MODE ACQUISITION
INIT
INIT:STAT? Returns 0 when acquisition is complete.
CALC:ANA TRC1
CALC:EVEN:COUN? TRC1 Ex.: Returns 4 (corresponding to 4 events).
CALC:EVEN? TRC1,1 (where 1 is the event number. Values 1 to 4 are valid). Returns the event corresponding to the specified number.

See Also

MMEMory[1..n]:LOAD:TRACe
TRACe[1..n]:CATalog?

:CALCulate[1..n]:EVENt:COUNt?

Description

This query returns the number of events after performing an analysis on the trace corresponding to the specified trace index.

Since *RST clears the event table, the number of events will be 0.

Syntax

:CALCulate[1..n]:EVENt:COUNt?<wsp>TRC1|TRC2|TRC3|TRC4

Parameter(s)

Label:

The program data syntax for the first parameter is defined as a <CHARACTER PROGRAM DATA> element. The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: TRC1|TRC2|TRC3|TRC4.

Trace index of the available wavelengths.

Response Syntax

<EventCount>

Response(s)

EventCount:

The response data syntax for <EventCount> is defined as a <NR1 NUMERIC RESPONSE DATA> element.

Returns the number of available events for the specified trace index.

Example(s)

CONF:ACQ:MODE ACQUISITION
INIT
INIT:STAT? Returns 0 when acquisition is complete.
CALC:ANA TRC1
CALC:EVEN:COUN? TRC1 Ex.: Returns 4 (corresponding to 4 events).

:CALCulate[1..n]:EVENt:STATus?

Description

This query returns an event from the event table after performing an analysis on the trace corresponding to the specified trace index.
You must supply the index of the event that you want to retrieve.

*RST clears the event table.

Syntax

:CALCulate[1..n]:EVENt:STATus?<wsp>TRC1|TRC2|TRC3|TRC4,<EventIndex>

Parameter(s)

➤ *Label:*

The program data syntax for the first parameter is defined as a <CHARACTER PROGRAM DATA> element. The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: TRC1|TRC2|TRC3|TRC4.

Trace index of the available wavelengths.

➤ *EventIndex:*

The program data syntax for <EventIndex> is defined as a <DECIMAL NUMERIC PROGRAM DATA> element.

Sets the event index. This value must be between 1 and the total number of events.

Response Syntax

<Event>

Response(s)

Event:

The response data syntax for <Event> is defined as a <DEFINITE LENGTH ARBITRARY BLOCK RESPONSE DATA> element.

Returns the event from the event table corresponding to the specified trace index.

Event structure is in A, B, C, D, E, F format, where:
A = Location (always in meters) <NR3 NUMERIC RESPONSE DATA>
B = EventType <NR1 NUMERIC RESPONSE DATA>
C = Loss (always in dB) <NR3 NUMERIC RESPONSE DATA>

:CALCulate[1..n]:EVENt:STATus?

D = Reflectance (always in dB) <NR3 NUMERIC RESPONSE DATA>
E = Cumulative (always in dB) <NR3 NUMERIC RESPONSE DATA>
F = EventStatus <NR1 NUMERIC RESPONSE DATA>

Here is the list of all possible event types:
1 = Positive splice
2 = Negative splice

3 = Reflection
4 = End of analysis
5 = Continuous fiber

The End of analysis event does not necessarily correspond to the last event of a fiber link. It indicates that the analysis has stopped before the end of the link because the instrument has reached the limit of its dynamic range.
In most cases, the OTDR analysis will return the type of the last event as being either reflective or non-reflective (event type 3 or 2).

:CALCulate[1..n]:EVENt:STATus?

The EventStatus is an integer indicating the sum of all applicable status for the designated event:
1: Echo
2: PossibleEcho
4: EndOfFiber
8: LaunchLevel
16: Saturated
32: AddedByUser
64: SpanStart
128: SpanEnd
512: AddedForSpan
1024: AddedFromReference

For exemple, if EventStatus is 148, it is the sum of 128, 16 and 4, which corresponds to an event that is the SpanEnd, is Saturated and is the EndOfFiber.

Example(s)

CONF:ACQ:MODE ACQUISITION
INIT
INIT:STAT? Returns 0 when acquisition is complete.
CALC:ANA TRC1
CALC:EVEN:COUN? TRC1 Ex.: Returns 4 (corresponding to 4 events).
CALC:EVEN:STAT? TRC1,1 (where 1 is the event number. Values 1 to 4 are valid). Returns the event corresponding to the specified number.

See Also

MMEMory[1..n]:LOAD:TRACe
TRACe[1..n]:CATalog?

:CALCulate[1..n]:EVENt:SUB?

Description

This query returns a sub-event from the sub-event table after performing an analysis on the trace corresponding to the specified trace index.

You must supply the index of the event and the index of the sub-event that you want to retrieve.

*RST clears the event table.

Syntax

:CALCulate[1..n]:EVENt:SUB?<wsp>TRC1|TRC2|TRC3|TRC4,<EventIndex>,<SubEventIndex>

Parameter(s)

➤ *Label:*

The program data syntax for the first parameter is defined as a <CHARACTER PROGRAM DATA> element. The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: TRC1|TRC2|TRC3|TRC4.

Trace index of the available wavelengths.

➤ *EventIndex:*

The program data syntax for <EventIndex> is defined as a <DECIMAL NUMERIC PROGRAM DATA> element.

Sets the event index. This value must be between 1 and the total number of events.

➤ *SubEventIndex:*

The program data syntax for <SubEventIndex> is defined as a <DECIMAL NUMERIC PROGRAM DATA> element.

Sets the sub-event index. This value must be between 1 and the total number of sub-events for the selected event.

Response Syntax

<Event>

Response(s)

Event:

The response data syntax for <Event> is defined as a <DEFINITE LENGTH ARBITRARY BLOCK RESPONSE DATA> element.

:CALCulate[1..n]:EVENt:SUB?

Returns the sub-event from the sub-event table corresponding to the specified event and specified trace index.

Event structure is in A, B, C, D format, where:
A = Location (always in meters) <NR3 NUMERIC RESPONSE DATA>
B = EventType <NR1 NUMERIC RESPONSE DATA>

C = Loss (always in dB) <NR3 NUMERIC RESPONSE DATA>
D = Reflectance (always in dB) <NR3 NUMERIC RESPONSE DATA>

Here is the list of all possible event types:
1 = Positive splice
2 = Negative splice
3 = Reflection
4 = End of analysis
5 = Continuous fiber

The End of analysis event does not necessarily correspond to the last event of a fiber link. It indicates that the analysis has stopped before the end of the link because the instrument has reached the limit of its dynamic range.
In most cases, the OTDR analysis will return the type of the last event as being either reflective or non-reflective (event type 3 or 2).

Example(s)

CONF:ACQ:MODE ACQUISITION
INIT
INIT:STAT? Returns 0 when acquisition is complete.
CALC:ANA TRC1
CALC:EVEN:COUN? TRC1 Ex.: Returns 4 (corresponding to 4 events).

CALC:EVEN? TRC1,1 (where 1 is the event number. Values 1 to 4 are valid).
CALC:EVEN:SUB:COUN? TRC1,2 Ex.: Returns 2 (corresponding to 2 sub-events).
CALC:EVEN:SUB? TRC1,2,1 (where 2 is the event number and 1 is the sub-event number. Values 1 to 2 are valid). Returns the event corresponding to the specified number.

See Also

MMEMory[1..n]:LOAD:TRACe
TRACe[1..n]:CATalog?

:CALCulate[1..n]:EVENt:SUB:COUNt?

Description	This query returns the number of sub-events associated to an event after performing an analysis on the trace corresponding to the specified trace index. Since *RST clears the event table, the number of sub-events will be 0.
Syntax	:CALCulate[1..n]:EVENt:SUB:COUNt?<wsp>TRC1|TRC2|TRC3|TRC4,<SubEventIndex>
Parameter(s)	➤ *Label:* The program data syntax for the first parameter is defined as a <CHARACTER PROGRAM DATA> element. The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: TRC1|TRC2|TRC3|TRC4. Trace index of the available wavelengths. ➤ *SubEventIndex:* The program data syntax for <SubEventIndex> is defined as a <DECIMAL NUMERIC PROGRAM DATA> element. Sets the sub-event index. This value must be between 1 and the total number of sub-events for the selected event.
Response Syntax	<EventCount>

:CALCulate[1..n]:EVENt:SUB:COUNt?

Response(s)	*EventCount:* The response data syntax for <EventCount> is defined as a <NR1 NUMERIC RESPONSE DATA> element. Returns the number of available sub-events for the specified event and specified trace index.
Example(s)	CONF:ACQ:MODE ACQUISITION INIT INIT:STAT? Returns 0 when acquisition is complete. CALC:ANA TRC1 CALC:EVEN:COUN? TRC1 Ex.: Returns 4 (corresponding to 4 events). CALC:EVEN? TRC1,1 (where 1 is the event number. Values 1 to 4 are valid). CALC:EVEN:SUB:COUN? TRC1,2 Ex.: Returns 2 (corresponding to 2 sub-events). CALC:EVEN:SUB:STAT? TRC1,2,1 (where 2 is the event number and 1 is the sub-event number. Values 1 to 2 are valid). Returns the event corresponding to the specified number.

:CALCulate[1..n]:EVENt:SUB:STATus?

Description

This query returns a sub-event from the sub-event table after performing an analysis on the trace corresponding to the specified trace index.

You must supply the index of the event and the index of the sub-event that you want to retrieve.

*RST clears the event table.

Syntax

:CALCulate[1..n]:EVENt:SUB:STATus?<wsp>TRC1|TRC2|TRC3|TRC4,<EventIndex>,<SubEventIndex>

Parameter(s)

➤ *Label:*

The program data syntax for the first parameter is defined as a <CHARACTER PROGRAM DATA> element. The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: TRC1|TRC2|TRC3|TRC4.

Trace index of the available wavelengths.

➤ *EventIndex:*

The program data syntax for <EventIndex> is defined as a <DECIMAL NUMERIC PROGRAM DATA> element.

Sets the event index. This value must be between 1 and the total number of events.

➤ *SubEventIndex:*

The program data syntax for <SubEventIndex> is defined as a <DECIMAL NUMERIC PROGRAM DATA> element.

Sets the sub-event index. This value must be between 1 and the total number of sub-events for the selected event.

Response Syntax

<Event>

Response(s)

Event:

The response data syntax for <Event> is defined as a <DEFINITE LENGTH ARBITRARY BLOCK RESPONSE DATA> element.

:CALCulate[1..n]:EVENt:SUB:STATus?

Returns the sub-event in the sub-event table corresponding to the specified event and specified trace index.

Event structure is in A, B, C, D, E format, where:
A = Location (always in meters) <NR3 NUMERIC RESPONSE DATA>
B = EventType <NR1 NUMERIC RESPONSE DATA>

C = Loss (always in dB) <NR3 NUMERIC RESPONSE DATA>
D = Reflectance (always in dB) <NR3 NUMERIC RESPONSE DATA>
E = EventStatus <NR1 NUMERIC RESPONSE DATA>

Here is the list of all possible event types:
1 = Positive splice
2 = Negative splice

3 = Reflection
4 = End of analysis
5 = Continuous fiber

The End of analysis event does not necessarily correspond to the last event of a fiber link. It indicates that the analysis has stopped before the end of the link because the instrument has reached the limit of its dynamic range.
In most cases, the OTDR analysis will return the type of the last event as being either reflective or non-reflective (event type 3 or 2).

The EventStatus is an integer indicating the sum of all applicable status for the designated event:
1: Echo
2: PossibleEcho
4: EndOfFiber
8: LaunchLevel
16: Saturated
32: AddedByUser
64: SpanStart
128: SpanEnd
512: AddedForSpan
1024: AddedFromReference

For exemple, if EventStatus is 148, it is the sum of 128, 16 and 4, which corresponds to an event that is the SpanEnd, is Saturated and is the EndOfFiber.

Example(s)

CONF:ACQ:MODE ACQUISITION
INIT
INIT:STAT? Returns 0 when acquisition is complete.
CALC:ANA TRC1
CALC:EVEN:COUN? TRC1 Ex.: Returns 4 (corresponding to 4 events).

CALC:EVEN? TRC1,1 (where 1 is the event number. Values 1 to 4 are valid).
CALC:EVEN:SUB:COUN? TRC1,2 Ex.: Returns 2 (corresponding to 2 sub-events).
CALC:EVEN:SUB:STAT? TRC1,2,1 (where 2 is the event number and 1 is the sub-event number. Values 1 to 2 are valid). Returns the sub-event corresponding to the specified number.

See Also

MMEMory[1..n]:LOAD:TRACe
TRACe[1..n]:CATalog?

:CALCulate[1..n]:HFACtor

Description

This command sets the helix factor that will be used for the specified trace index.
Using this command will recalculate the event table automatically.

*RST clears this setting.

Syntax

:CALCulate[1..n]:HFACtor<wsp>TRC1|TRC2|TRC3|TRC4,<HelixFactor>

Parameter(s)

➤ *Label:*

The program data syntax for the first parameter is defined as a <CHARACTER PROGRAM DATA> element. The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: TRC1|TRC2|TRC3|TRC4.

Trace index of the available wavelengths.

➤ *HelixFactor:*

The program data syntax for <HelixFactor> is defined as a <DECIMAL NUMERIC PROGRAM DATA> element.

Sets the helix factor.

Example(s)

CONF:ANA:HFAC 0
CONF:ACQ:MODE ACQUISITION
INIT
INIT:STAT? Returns 0 when acquisition is complete.
CALC:HFAC? TRC1 Returns 0
CALC:HFAC TRC1,2
CALC:HFAC? TRC1 Returns 2

See Also

CALCulate[1..n]:ANAlysis:[UNIDirectional]
CALCulate[1..n]:EVENt:COUNt?
CALCulate[1..n]:EVENt?
MMEMory[1..n]:LOAD:TRACe
TRACe[1..n]:CATalog?

:CALCulate[1..n]:HFACtor?

Description

This query returns the helix factor used for the specified trace index.

Since *RST clears the helix factor value, the returned value will be 0.

Syntax

:CALCulate[1..n]:HFACtor?<wsp>TRC1|TRC2|TRC3|TRC4

Parameter(s)

Label:

The program data syntax for the first parameter is defined as a <CHARACTER PROGRAM DATA> element. The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: TRC1|TRC2|TRC3|TRC4.

Trace index of the available wavelengths.

Response Syntax

<HelixFactor>

Response(s)

HelixFactor:

The response data syntax for <HelixFactor> is defined as a <NR3 NUMERIC RESPONSE DATA> element.

Returns the helix factor used by the trace corresponding to the specified trace index.

Example(s)

CONF:ANA:HFAC 2
CONF:ACQ:MODE ACQUISITION
INIT
INIT:STAT? Returns 0 when acquisition is complete.
CALC:HFAC? TRC1 Returns 2

See Also

MMEMory[1..n]:LOAD:TRACe
TRACe[1..n]:CATalog?

:CALCulate[1..n]:INJection[:LEVel]?

Description	This query returns the injection level for the specified trace. The value is undefined if the trace is not analyzed.
Syntax	:CALCulate[1..n]:INJection[:LEVel]?<wsp>TRC1 |TRC2|TRC3|TRC4
Parameter(s)	*Label:* The program data syntax for the first parameter is defined as a <CHARACTER PROGRAM DATA> element. The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: TRC1|TRC2|TRC3|TRC4. Trace index of the available wavelengths.
Response Syntax	<InjectionLevel>
Response(s)	*InjectionLevel:* The response data syntax for <InjectionLevel> is defined as a <NR3 NUMERIC RESPONSE DATA> element. Returns the Injection level value, in dB.
Example(s)	CONF:ACQ:MODE ACQUISITION INIT INIT:STAT? Returns 0 when acquisition is complete. CALC:ANA TRC1 CALC:INJ:LEV? TRC1 Ex.: Returns 20.416
See Also	MMEMory[1..n]:LOAD:TRACe TRACe[1..n]:CATalog?

:CALCulate[1..n]:IORefraction

Description

This command sets the index of refraction that will be used for the trace corresponding to the specified trace index.
Using this command will recalculate the event table automatically.

*RST clears this setting.

Syntax

```
:CALCulate[1..n]:IORefraction<wsp>TRC1|TRC2|TRC3|TRC4,<IOR>
```

Parameter(s)

➤ *Label:*

The program data syntax for the first parameter is defined as a <CHARACTER PROGRAM DATA> element. The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: TRC1|TRC2|TRC3|TRC4.

Trace index of the available wavelengths.

➤ *IOR:*

The program data syntax for <IOR> is defined as a <DECIMAL NUMERIC PROGRAM DATA> element.

Sets the index of refraction.

Example(s)

```
CONF:ANA:IOR 1.4677
CONF:ACQ:MODE ACQUISITION
INIT
INIT:STAT? Returns 0 when acquisition is complete.
CALC:IOR? TRC1 Returns 1.4677
CALC:IOR TRC1,1.5
CALC:IOR? TRC1 Returns 1.5
```

See Also

CALCulate[1..n]:ANAlysis:[UNIDirectional]
CALCulate[1..n]:EVENt:COUNt?
CALCulate[1..n]:EVENt?
MMEMory[1..n]:LOAD:TRACe
TRACe[1..n]:CATalog?

:CALCulate[1..n]:IORefraction?

Description

This query returns the index of refraction used for the trace corresponding to the specified trace index.

Since *RST clears the index of refraction value, the returned value will be 0.

Syntax

:CALCulate[1..n]:IORefraction?<wsp>TRC1|TRC2|TRC3|TRC4

Parameter(s)

Label:

The program data syntax for the first parameter is defined as a <CHARACTER PROGRAM DATA> element. The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: TRC1|TRC2|TRC3|TRC4.

Trace index of the available wavelengths.

Response Syntax

<IOR>

Response(s)

IOR:

The response data syntax for <IOR> is defined as a <NR3 NUMERIC RESPONSE DATA> element.

Returns the index of refraction used by the trace corresponding to the specified trace index.

Example(s)

CONF:ANA:IOR 1.5
CONF:ACQ:MODE ACQUISITION
INIT
INIT:STAT? Returns 0 when acquisition is complete.
CALC:IOR? TRC1 Returns 1.5

See Also

MMEMory[1..n]:LOAD:TRACe
TRACe[1..n]:CATalog?

:CALCulate[1..n]:LOSS?

Description

This query returns the loss between two markers measured by least-square approximation, for the trace corresponding to the specified trace index.

*RST clears this value.

Syntax

:CALCulate[1..n]:LOSS?<wsp>TRC1|TRC2|TRC3|TRC4,<MarkerA>,<MarkerB>

Parameter(s)

➤ *Label:*

The program data syntax for the first parameter is defined as a <CHARACTER PROGRAM DATA> element. The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: TRC1|TRC2|TRC3|TRC4.

Trace index of the available wavelengths.

➤ *MarkerA:*

The program data syntax for <MarkerA> is defined as a <DECIMAL NUMERIC PROGRAM DATA> element.

Specifies the marker A position, in meters.

➤ *MarkerB:*

The program data syntax for <MarkerB> is defined as a <DECIMAL NUMERIC PROGRAM DATA> element.

Specifies the marker B position, in meters.

Response Syntax

<Loss>

Response(s)	*Loss:* The response data syntax for <Loss> is defined as a <NR3 NUMERIC RESPONSE DATA> element. Returns the loss value in dB, between marker A and marker B.
Example(s)	CONF:ACQ:MODE ACQUISITION INIT INIT:STAT? Returns 0 when acquisition is complete. CALC:LOSS? TRC1,10,104 Ex.: Returns 0.458 CALC:LOSS? TRC1,10 M,0.104 KM Ex.: Returns 0.458 CALC:LOSS? TRC1,0.01 KM,104 M Ex.: Returns 0.458
See Also	MMEMory[1..n]:LOAD:TRACe TRACe[1..n]:CATalog?

:CALCulate[1..n]:ORL?

Description

This query returns the value of the Optical Return Loss measured between two markers, for the trace corresponding to the specified trace index.

*RST clears this value.

Syntax

:CALCulate[1..n]:ORL?<wsp>TRC1|TRC2|TRC3|TRC4,<MarkerA>,<MarkerB>

Parameter(s)

➤ *Label:*

The program data syntax for the first parameter is defined as a <CHARACTER PROGRAM DATA> element. The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: TRC1|TRC2|TRC3|TRC4.

Trace index of the available wavelengths.

➤ *MarkerA:*

The program data syntax for <MarkerA> is defined as a <DECIMAL NUMERIC PROGRAM DATA> element.

Specifies the marker A position, in meters.

➤ *MarkerB:*

The program data syntax for <MarkerB> is defined as a <DECIMAL NUMERIC PROGRAM DATA> element.

Specifies the marker B position, in meters.

Response Syntax

<ORL>

Response(s)	*ORL:* The response data syntax for <ORL> is defined as a <NR3 NUMERIC RESPONSE DATA> element. Returns the Optical Return Loss value in dB, between marker A and marker B.
Example(s)	CONF:ACQ:MODE ACQUISITION INIT INIT:STAT? Returns 0 when acquisition is complete. CALC:ORL? TRC1,10,100 Ex.: Returns 30.305 CALC:ORL? TRC1,10 M, 0.100 KM Ex.: Returns 30.305 CALC:ORL? TRC1,0.01 KM,100 M Ex.: Returns 30.305
See Also	MMEMory[1..n]:LOAD:TRACe TRACe[1..n]:CATalog?

:CALCulate[1..n]:REFLectance?

Description

This query returns the reflectance value measured between two markers, for the trace corresponding to the specified trace index.

*RST clears this value.

Syntax

:CALCulate[1..n]:REFLectance?<wsp>TRC1|TRC2|TRC3|TRC4,<SubMarkerA>,<MarkerA>,<MarkerB>

Parameter(s)

➤ *Label:*

The program data syntax for the first parameter is defined as a <CHARACTER PROGRAM DATA> element. The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: TRC1|TRC2|TRC3|TRC4.

Trace index of the available wavelengths.

➤ *SubMarkerA:*

The program data syntax for <SubMarkerA> is defined as a <DECIMAL NUMERIC PROGRAM DATA> element.

Specifies the submarker A position, in meters.

➤ *MarkerA:*

The program data syntax for <MarkerA> is defined as a <DECIMAL NUMERIC PROGRAM DATA> element.

Specifies the marker A position, in meters.

➤ *MarkerB:*

The program data syntax for <MarkerB> is defined as a <DECIMAL NUMERIC PROGRAM DATA> element.

Specifies the marker B position, in meters.

Response Syntax

<Reflectance>

Response(s)

Reflectance:

The response data syntax for <Reflectance> is defined as a <NR3 NUMERIC RESPONSE DATA> element.

Returns the reflectance value in dB, calculated using all three markers.

:CALCulate[1..n]:REFLectance?

Example(s)	CONF:ACQ:MODE ACQUISITION INIT INIT:STAT? Returns 0 when acquisition is complete. CALC:REFL? TRC1,0,0.1 KM,200 Ex.: Returns –24.549 CALC:REFL? TRC1,0 M,100,200 M Ex.: Returns –24.549 CALC:REFL? TRC1,0 KM,100 M, 0.2 KM Ex.: Returns –24.549
Notes	See the section on reflectance measurement in the OTDR user guide.
See Also	CALCulate[1..n]:ANAlysis:[UNIDirectional] CALCulate[1..n]:EVENt:COUNt? CALCulate[1..n]:EVENt? MMEMory[1..n]:LOAD:TRACe TRACe[1..n]:CATalog?

:CALCulate[1..n]:RBScatter

Description

This command sets the Rayleigh backscatter that will be used for the trace corresponding to the specified trace index.
Using this command requires to perform a new analysis manually.

*RST clears this setting.

Syntax

```
:CALCulate[1..n]:RBScatter<wsp>TRC1|TRC2|T
RC3|TRC4,<RBS>
```

Parameter(s)

➤ *Label:*

The program data syntax for the first parameter is defined as a <CHARACTER PROGRAM DATA> element. The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: TRC1|TRC2|TRC3|TRC4.

Trace index of the available wavelengths.

➤ *RBS:*

The program data syntax for <RBS> is defined as a <DECIMAL NUMERIC PROGRAM DATA> element.

Sets the Rayleigh backscatter.

Example(s)

```
CONF:ANA:RBS –79.5
CONF:ACQ:MODE ACQUISITION
INIT
INIT:STAT? Returns 0 when acquisition is
complete.
CALC:RBS? TRC1 Returns –79.5
CALC:RBS TRC1,–80
CALC:ANA TRC1
CALC:RBS? TRC1 Returns –80
```

See Also

CALCulate[1..n]:ANAlysis:[UNIDirectional]
CALCulate[1..n]:EVENt:COUNt?
CALCulate[1..n]:EVENt?
MMEMory[1..n]:LOAD:TRACe
TRACe[1..n]:CATalog?

:CALCulate[1..n]:RBScatter?

Description

This query returns the Rayleigh backscatter used for the trace corresponding to the specified trace index.

Since *RST clears the RBS value, the returned value will be 0.

Syntax

:CALCulate[1..n]:RBScatter?<wsp>TRC1|TRC2|TRC3|TRC4

Parameter(s)

Label:

The program data syntax for the first parameter is defined as a <CHARACTER PROGRAM DATA> element. The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: TRC1|TRC2|TRC3|TRC4.

Trace index of the available wavelengths.

Response Syntax

<RBS>

Response(s)

RBS:

The response data syntax for <RBS> is defined as a <NR3 NUMERIC RESPONSE DATA> element.

Returns the Rayleigh backscatter used by the trace corresponding to the specified trace index.

Example(s)

CONF:ANA:RBS –80
CONF:ACQ:MODE ACQUISITION
INIT
INIT:STAT? Returns 0 when acquisition is complete.
CALC:RBS? TRC1 Returns –80

Notes

Resets to a new default value when wavelength and range change.

See Also

MMEMory[1..n]:LOAD:TRACe
TRACe[1..n]:CATalog?

:CALCulate[1..n]:SLOSs?

Description

This query returns the value of the measured loss for a given splice identified using four markers, for the trace corresponding to the specified trace index.

*RST clears this value.

Syntax

:CALCulate[1..n]:SLOSs?<wsp>TRC1|TRC2|TRC3|TRC4,<SubMarkerA>,<MarkerA>,<MarkerB>,<SubMarkerB>

Parameter(s)

➤ *Label:*

The program data syntax for the first parameter is defined as a <CHARACTER PROGRAM DATA> element. The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: TRC1|TRC2|TRC3|TRC4.

Trace index of the available wavelengths.

➤ *SubMarkerA:*

The program data syntax for <SubMarkerA> is defined as a <DECIMAL NUMERIC PROGRAM DATA> element.

Specifies the submarker A position, in meters.

➤ *MarkerA:*

The program data syntax for <MarkerA> is defined as a <DECIMAL NUMERIC PROGRAM DATA> element.

Specifies the marker A position, in meters.

➤ *MarkerB:*

The program data syntax for <MarkerB> is defined as a <DECIMAL NUMERIC PROGRAM DATA> element.

Specifies the marker B position, in meters.

➤ *SubMarkerB:*

The program data syntax for <SubMarkerB> is defined as a <DECIMAL NUMERIC PROGRAM DATA> element.

Specifies the submarker B position, in meters.

Returns the splice loss value, calculated using all four markers.

Response Syntax

<Splice Loss>

:CALCulate[1..n]:SLOSs?

Response(s)

Splice Loss:

The response data syntax for <Splice Loss> is defined as a <NR3 NUMERIC RESPONSE DATA> element.

Returns the splice loss value, calculated using all four markers.

Example(s)

CONF:ACQ:MODE ACQUISITION
INIT
INIT:STAT? Returns 0 when acquisition is complete.
CALC:SLOS? TRC1,10,100,200,300 Ex.: Returns 0.058
CALC:SLOS? TRC1,0.01 KM,100 M, 0.2 KM,300 Ex.: Returns 0.058

CALC:SLOS? TRC1,10 M,100 M,200 M,300 M Ex.: Returns 0.058
CALC:SLOS? TRC1,0.01 KM, 0.1 KM, 0.2 KM,0.3 KM Ex.: Returns 0.058

Notes

See the section on loss measurement in the OTDR user guide.

See Also

CALCulate[1..n]:ANAlysis:[UNIDirectional]
CALCulate[1..n]:EVENt:COUNt?
CALCulate[1..n]:EVENt?
MMEMory[1..n]:LOAD:TRACe
TRACe[1..n]:CATalog?

:CALCulate[1..n]:THReshold:EOFiber

Description

This command sets the end-of-fiber threshold that will be used for the specified trace index. Using this command requires to perform a new analysis manually.

*RST clears this setting.

Syntax

```
:CALCulate[1..n]:THReshold:EOFiber<wsp>TRC1|TRC2|TRC3|TRC4,<End-of-Fiber>
```

Parameter(s)

➤ *Label:*

The program data syntax for the first parameter is defined as a <CHARACTER PROGRAM DATA> element. The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: TRC1|TRC2|TRC3|TRC4.

Trace index of the available wavelengths.

➤ *End-of-Fiber:*

The program data syntax for <End-of-Fiber> is defined as a <DECIMAL NUMERIC PROGRAM DATA> element.

Sets the end-of-fiber threshold.

Example(s)

```
CONF:ANA:THR:EOF 5.1
CONF:ACQ:MODE ACQ
INIT
INIT:STAT? Returns 0 when acquisition is complete.
CALC:THR:EOF? TRC1 Returns 5.1
CALC:THR:EOF TRC1,5.2
CALC:ANA TRC1
CALC:THR:EOF? TRC1 Returns 5.2
```

See Also

CALCulate[1..n]:ANAlysis:[UNIDirectional]
CALCulate[1..n]:EVENt:COUNt?
CALCulate[1..n]:EVENt?
MMEMory[1..n]:LOAD:TRACe
TRACe[1..n]:CATalog?

:CALCulate[1..n]:THReshold:EOFiber?

Description

This query returns the end-of-fiber threshold used for the specified trace index.

*RST clears this value.

Syntax

:CALCulate[1..n]:THReshold:EOFiber?<wsp>TRC1|TRC2|TRC3|TRC4

Parameter(s)

Label:

The program data syntax for the first parameter is defined as a <CHARACTER PROGRAM DATA> element. The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: TRC1|TRC2|TRC3|TRC4.

Trace index of the available wavelengths.

Response Syntax

<End-of-Fiber>

Response(s)

End-of-Fiber:

The response data syntax for <End-of-Fiber> is defined as a <NR3 NUMERIC RESPONSE DATA> element.

Returns the end-of-fiber threshold used by the trace corresponding to the specified trace index.

Example(s)

CONF:ANA:THR:EOF 5.1
CONF:ACQ:MODE ACQ
INIT
INIT:STAT? Returns 0 when acquisition is complete.
CALC:THR:EOF? TRC1 Returns 5.1

See Also

MMEMory[1..n]:LOAD:TRACe
TRACe[1..n]:CATalog?

:CALCulate[1..n]:THReshold:REFLectance

Description

This command sets the reflectance threshold that will be used for the specified trace index. Using this command requires to perform a new analysis manually.

*RST clears this setting.

Syntax

:CALCulate[1..n]:THReshold:REFLectance<wsp>TRC1|TRC2|TRC3|TRC4,<Reflectance>

Parameter(s)

➤ *Label:*

The program data syntax for the first parameter is defined as a <CHARACTER PROGRAM DATA> element. The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: TRC1|TRC2|TRC3|TRC4.

Trace index of the available wavelengths.

➤ *Reflectance:*

The program data syntax for <Reflectance> is defined as a <DECIMAL NUMERIC PROGRAM DATA> element.

Sets the reflectance threshold.

Example(s)

CONF:ANA:THR:REFL -72.1
CONF:ACQ:MODE ACQ
INIT
INIT:STAT? Returns 0 when acquisition is complete.
CALC:THR:REFL? TRC1 Returns -72.1
CALC:THR:REFL TRC1,-72.2
CALC:ANA TRC1
CALC:THR:REFL? TRC1 Returns -72.2

See Also

CALCulate[1..n]:ANAlysis:[UNIDirectional]
CALCulate[1..n]:EVENt:COUNt?
CALCulate[1..n]:EVENt?
MMEMory[1..n]:LOAD:TRACe
TRACe[1..n]:CATalog?

:CALCulate[1..n]:THReshold:REFLectance?

Description

This query returns the reflectance threshold used for the specified trace index.

*RST clears this value.

Syntax

:CALCulate[1..n]:THReshold:REFLectance?<wsp>TRC1|TRC2|TRC3|TRC4

Parameter(s)

Label:

The program data syntax for the first parameter is defined as a <CHARACTER PROGRAM DATA> element. The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: TRC1|TRC2|TRC3|TRC4.

Trace index of the available wavelengths.

Response Syntax

<Reflectance>

Response(s)

Reflectance:

The response data syntax for <Reflectance> is defined as a <NR3 NUMERIC RESPONSE DATA> element.

Returns the reflectance threshold used by the trace corresponding to the specified trace index.

Example(s)

CONF:ANA:THR:REFL -72.1
CONF:ACQ:MODE ACQ
INIT
INIT:STAT? Returns 0 when acquisition is complete.
CALC:THR:REFL? TRC1 Returns -72.1

See Also

MMEMory[1..n]:LOAD:TRACe
TRACe[1..n]:CATalog?

:CALCulate[1..n]:THReshold:SLOSs

Description

This command sets the splice loss threshold that will be used for the specified trace index.
Using this command requires to perform a new analysis manually.

*RST clears this setting.

Syntax

:CALCulate[1..n]:THReshold:SLOSs<wsp>TRC1|TRC2|TRC3|TRC4,<Splice Loss>

Parameter(s)

➤ *Label:*

The program data syntax for the first parameter is defined as a <CHARACTER PROGRAM DATA> element. The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: TRC1|TRC2|TRC3|TRC4.

Trace index of the available wavelengths.

➤ *Splice Loss:*

The program data syntax for <Splice Loss> is defined as a <DECIMAL NUMERIC PROGRAM DATA> element.

Sets the splice loss threshold.

Example(s)

CONF:ANA:THR:SLOS 0.03
CONF:ACQ:MODE ACQ
INIT
INIT:STAT? Returns 0 when acquisition is complete.
CALC:THR:SLOS? TRC1 Returns 0.03
CALC:THR:SLOS TRC1,0.04
CALC:ANA TRC1
CALC:THR:SLOS? TRC1 Returns 0.04

See Also

CALCulate[1..n]:ANAlysis:[UNIDirectional]
CALCulate[1..n]:EVENt:COUNt?
CALCulate[1..n]:EVENt?
MMEMory[1..n]:LOAD:TRACe
TRACe[1..n]:CATalog?

:CALCulate[1..n]:THReshold:SLOSs?

Description

This query returns the splice loss threshold used for the specified trace index.

*RST clears this value.

Syntax

:CALCulate[1..n]:THReshold:SLOSs?<wsp>TRC1|TRC2|TRC3|TRC4

Parameter(s)

Label:

The program data syntax for the first parameter is defined as a <CHARACTER PROGRAM DATA> element. The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: TRC1|TRC2|TRC3|TRC4.

Trace index of the available wavelengths.

Response Syntax

<Splice Loss>

Response(s)

Splice Loss:

The response data syntax for <Splice Loss> is defined as a <NR3 NUMERIC RESPONSE DATA> element.

Returns the splice loss threshold used by the trace corresponding to the specified trace index.

Example(s)

CONF:ANA:THR:SLOS 0.03
CONF:ACQ:MODE ACQ
INIT
INIT:STAT? Returns 0 when acquisition is complete.
CALC:THR:SLOS? TRC1 Returns 0.03

See Also

MMEMory[1..n]:LOAD:TRACe
TRACe[1..n]:CATalog?

:CALCulate[1..n]:TORL?

Description

This query returns the sum of all optical return loss (ORL) values measured on the total fiber length, for the trace corresponding to the specified trace index. This total ORL value does not include the launch reflection. A negative total value indicates that the real value is smaller.

*RST clears this value.

Syntax

:CALCulate[1..n]:TORL?<wsp>TRC1|TRC2|TRC3|TRC4

Parameter(s)

Label:

The program data syntax for the first parameter is defined as a <CHARACTER PROGRAM DATA> element. The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: TRC1|TRC2|TRC3|TRC4.

Trace index of the available wavelengths.

Response Syntax

<TotalOrl>

Response(s)

TotalOrl:

The response data syntax for <TotalOrl> is defined as a <NR3 NUMERIC RESPONSE DATA> element.

Returns the total ORL value, in dB.

Example(s)

CONF:ACQ:MODE ACQUISITION
INIT
INIT:STAT? Returns 0 when acquisition is complete.
CALC:ANA TRC1
CALC:TORL? TRC1 Ex.: Returns 20.416

See Also

MMEMory[1..n]:LOAD:TRACe
TRACe[1..n]:CATalog?

:CONFigure[1..n]:ACQuisition

Description

This command specifies the wavelength, range and pulse that will be used for the next acquisition.

*RST does not affect this command.

Syntax

:CONFigure[1..n]:ACQuisition<wsp><Wavelength>,<Range>,<Pulse>

Parameter(s)

➤ *Wavelength:*

The program data syntax for <Wavelength> is defined as a <DECIMAL NUMERIC PROGRAM DATA> element.

Sets the wavelength, in meters.

➤ *Range:*

The program data syntax for <Range> is defined as a <DECIMAL NUMERIC PROGRAM DATA> element.

Sets the range, in meters.
Range value depends on the wavelength parameter.

➤ *Pulse:*

The program data syntax for <Pulse> is defined as a <DECIMAL NUMERIC PROGRAM DATA> element.

Sets the pulse, in seconds.
Pulse value depends on the range parameter.

Example(s)

CONF:ACQ:WAV:LIST? Returns the available wavelength list
CONF:ACQ:RANG:LIST? 1310 NM Returns the available range list (where 1310 is an item of CONF:ACQ:WAV:LIST?)

CONF:ACQ:PULS:LIST? 1310 NM,1250 M Returns the available pulse list (where 1250 is an item of CONF:ACQ:RANG:LIST?)
CONF:ACQ 1310 NM,1250 M,10 NS (where 10 is an item of CONF:ACQ:PULS:LIST?)

See Also

CONFigure[1..n]:ACQuisition:WAVelength?
CONFigure[1..n]:ACQuisition:RANGe?
CONFigure[1..n]:ACQuisition:PULSe?

:CONFigure[1..n]:ACQuisition:DURation

Description	This command specifies the duration that will be used for the next acquisition. *RST reverts this setting to default value.
Syntax	:CONFigure[1..n]:ACQuisition:DURation<wsp> <Duration>|MAXimum|MINimum|DEFault
Parameter(s)	*Duration:* The program data syntax for <Duration> is defined as a <numeric_value> element. The <Duration> special forms MINimum, MAXimum and DEFault are accepted on input. MINimum allows to set the instrument to the smallest supported value. MAXimum allows to set the instrument to the greatest supported value. DEFault allows the instrument to select a value for the <Duration> parameter. Sets the acquisition duration, in seconds.
Example(s)	CONF:ACQ:DUR? Ex.: Returns 15 CONF:ACQ:DUR 10 CONF:ACQ:DUR? Returns 10
See Also	FETCh[1..n]:DURation? FETCh[1..n]:ASETting:DURation?

:CONFigure[1..n]:ACQuisition:DURation?

Description

This query returns the current duration setting.

*RST reverts this setting to default value.

Syntax

:CONFigure[1..n]:ACQuisition:DURation?[<wsp>MINimum|MAXimum|DEFault]

Parameter(s)

Parameter 1:

The program data syntax for the first parameter is defined as a <CHARACTER PROGRAM DATA> element. The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: MINimum|MAXimum|DEFault.

MINimum is used to retrieve the instrument's smallest supported value.
MAXimum is used to retrieve the instrument's greatest supported value.
DEFault is used to retrieve the instrument's default value.

Response Syntax

<Duration>

Response(s)

Duration:

The response data syntax for <Duration> is defined as a <NR1 NUMERIC RESPONSE DATA> element.

Returns the duration, in seconds.

Example(s)

CONF:ACQ:DUR 10
CONF:ACQ:DUR? Returns 10

See Also

FETCh[1..n]:DURation?
FETCh[1..n]:ASETting:DURation?

:CONFigure[1..n]:ACQuisition: HRESolution

Description

This command enables the high-resolution feature that allows you to obtain more data points per acquisition (greater distance resolution for the trace).

*RST reverts this setting to default value.

Syntax

:CONFigure[1..n]:ACQuisition:HRESolution<wsp><HighResolution>

Parameter(s)

HighResolution:

The program data syntax for <HighResolution> is defined as a <Boolean Program Data> element. The <HighResolution> special forms ON and OFF are accepted on input for increased readability. ON corresponds to 1 and OFF corresponds to 0.

Enables or disables the high-resolution feature.

Example(s)

CONF:ACQ:HRES 1 The acquisition will be performed using high resolution.

See Also

CONFigure[1..n]:ACQuisition:HRESolution?
FETCh[1..n]:HRESolution?

:CONFigure[1..n]:ACQuisition:HRESolution?

Description	This query returns a value indicating if the high-resolution feature is enabled for the next acquisition. *RST reverts this setting to default value.
Syntax	:CONFigure[1..n]:ACQuisition:HRESolution?
Parameter(s)	None
Response Syntax	<HighResolution>
Response(s)	*HighResolution:* The response data syntax for <HighResolution> is defined as a <NR1 NUMERIC RESPONSE DATA> element. Indicates if the high-resolution feature is enabled or not for the next acquisition.
Example(s)	CONF:ACQ:HRES? Returns 1 if the high resolution is enabled.
See Also	CONFigure[1..n]:ACQuisition:HRESolution FETCh[1..n]:HRESolution?

:CONFigure[1..n]:ACQuisition:MODE

Description

This command specifies the mode that will be used for the next acquisition.

Acquisition: Allows the OTDR to perform a standard acquisition.

Auto Setting: Lets the OTDR evaluate the length of the fiber and find the appropriate range and pulse width.
Check First Connector: Used to detect a low injection level.
Real Time: Used to view sudden changes in the fiber under test. In this mode, measurements are not allowed.

*RST sets the current acquisition mode to ACQUISITION.

Syntax

:CONFigure[1..n]:ACQuisition:MODE<wsp>ACQuisition|ASETting|CFConnector|REAltime

Parameter(s)

Mode:

The program data syntax for the first parameter is defined as a <CHARACTER PROGRAM DATA> element. The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: ACQuisition|ASETting|CFConnector|REAltime.

Sets the acquisition mode.

Example(s)

CONF:ACQ:MODE? Ex.: Returns ASETTING
CONF:ACQ:MODE ACQ
CONF:ACQ:MODE? Returns ACQUISITION

See Also

INITiate[1..n][:IMMediate]
ABORt[1..n]

:CONFigure[1..n]:ACQuisition:MODE?

Description

This query returns the current acquisition mode.

*RST sets the current acquisition mode to ACQUISITION.

Syntax

:CONFigure[1..n]:ACQuisition:MODE?

Parameter(s)

None

Response Syntax

<Mode>

Response(s)

Mode:

The response data syntax for <Mode> is defined as a <CHARACTER RESPONSE DATA> element.

Returns the current acquisition mode.

Example(s)

CONF:ACQ:MODE ACQ
CONF:ACQ:MODE? Returns ACQUISITION

:CONFigure[1..n]:ACQuisition:PULSe?

Description	This query returns the current pulse setting. *RST reverts this setting to default value.
Syntax	:CONFigure[1..n]:ACQuisition:PULSe?
Parameter(s)	None
Response Syntax	<Pulse>
Response(s)	*Pulse:* The response data syntax for <Pulse> is defined as a <NR3 NUMERIC RESPONSE DATA> element. Returns the pulse, in seconds.
Example(s)	CONF:ACQ 1310 NM,1250 M,10 NS CONF:ACQ:PULS? Returns 1E–8
See Also	CONFigure[1..n]:ACQuisition:WAVelength:LIST? CONFigure[1..n]:ACQuisition:RANGe:LIST? CONFigure[1..n]:ACQuisition:PULSe:LIST?

:CONFigure[1..n]:ACQuisition:PULSe:LIST?

Description

This query returns the list of available pulses for the specified wavelength and range.

*RST does not affect this command.

Syntax

:CONFigure[1..n]:ACQuisition:PULSe:LIST?<wsp><Wavelength>,<Range>

Parameter(s)

➤ *Wavelength:*

The program data syntax for <Wavelength> is defined as a <DECIMAL NUMERIC PROGRAM DATA> element.

Specifies the wavelength, in meters, that filters out invalid pulses from all pulses.

➤ *Range:*

The program data syntax for <Range> is defined as a <DECIMAL NUMERIC PROGRAM DATA> element.

Specifies the range, in meters, related to the wavelength, in meters, that filters out invalid pulses from all pulses.

Response Syntax

<PulseList>

Response(s)

PulseList:

The response data syntax for <PulseList> is defined as a <DEFINITE LENGTH ARBITRARY BLOCK RESPONSE DATA> element.

Returns the list of valid pulses, in seconds.

Example(s)

CONF:ACQ:WAV:LIST? Returns a wavelength list.
CONF:ACQ:RANG:LIST? 1310 NM Returns a range list (where 1310 is an item of CONF:ACQ:WAV:LIST?)
CONF:ACQ:PULS:LIST? 1310 NM,1250 M Returns a pulse list (where 1250 is an item of CONF:ACQ:RANG:LIST?)

See Also

CONFigure[1..n]:ACQuisition:PULSe?
CONFigure[1..n]:ACQuisition

:CONFigure[1..n]:ACQuisition:RANGe?

Description

This query returns the current range setting.

*RST reverts this setting to default value.

Syntax

:CONFigure[1..n]:ACQuisition:RANGe?

Parameter(s)

None

Response Syntax

<Range>

Response(s)

Range:

The response data syntax for <Range> is defined as a <NR3 NUMERIC RESPONSE DATA> element.

Returns the range, in meters.

Example(s)

CONF:ACQ 1310 NM,1250 M,10 NS
CONF:ACQ:RANG? Returns 1.25E+3

See Also

CONFigure[1..n]:ACQuisition:WAVelength:LIST?
CONFigure[1..n]:ACQuisition:RANGe:LIST?
CONFigure[1..n]:ACQuisition:PULSe:LIST?

:CONFigure[1..n]:ACQuisition:RANGe:LIMit:HIGH?

Description	This query returns the highest possible value for the acquisition range, at the specified wavelength. *RST does not affect this command.
Syntax	:CONFigure[1..n]:ACQuisition:RANGe:LIMit:HIGH?<wsp><Wavelength>
Parameter(s)	*Wavelength:* The program data syntax for <Wavelength> is defined as a <DECIMAL NUMERIC PROGRAM DATA> element. Wavelength for which you want to know the maximum value allowed for the acquisition range.
Response Syntax	<Range>
Response(s)	*Range:* The response data syntax for <Range> is defined as a <NR3 NUMERIC RESPONSE DATA> element. Maximum value allowed for the acquisition range at the specified wavelength, in meters.
Example(s)	CONF:ACQ:RANG:LIM:HIGH? 1310 NM Returns 1.25E+3
See Also	CONFigure[1..n]:ACQuisition:RANGe:LIMit:LOW?

:CONFigure[1..n]:ACQuisition:RANGe:LIMit:LOW?

Description

This query returns the lowest possible value for the acquisition range, at the specified wavelength.

*RST does not affect this command.

Syntax

:CONFigure[1..n]:ACQuisition:RANGe:LIMit:LOW?<wsp><Wavelength>

Parameter(s)

Wavelength:

The program data syntax for <Wavelength> is defined as a <DECIMAL NUMERIC PROGRAM DATA> element.

Wavelength for which you want to know the minimum value allowed for the acquisition range.

Response Syntax

<Range>

Response(s)

Range:

The response data syntax for <Range> is defined as a <NR3 NUMERIC RESPONSE DATA> element.

Minimum value allowed for the acquisition range at the specified wavelength, in meters.

Example(s)

CONF:ACQ:RANG:LIM:LOW? 1310 NM Returns 2.5+2

See Also

CONFigure[1..n]:ACQuisition:RANGe:LIMit:HIGH?

:CONFigure[1..n]:ACQuisition:RANGe:LIST?

Description	This query returns the list of available ranges for the specified wavelength. *RST does not affect this command.
Syntax	:CONFigure[1..n]:ACQuisition:RANGe:LIST?<wsp><Wavelength>
Parameter(s)	*Wavelength:* The program data syntax for <Wavelength> is defined as a <DECIMAL NUMERIC PROGRAM DATA> element. Specifies the wavelength, in meters, that filters out invalid ranges from all ranges.
Response Syntax	<RangeList>
Response(s)	*RangeList:* The response data syntax for <RangeList> is defined as a <DEFINITE LENGTH ARBITRARY BLOCK RESPONSE DATA> element. Returns the list of valid ranges, in meters.
Example(s)	CONF:ACQ:WAV:LIST? Returns a wavelength list. CONF:ACQ:RANG:LIST? 1310 NM Returns a range list (where 1310 is an item of CONF:ACQ:WAV:LIST?)
See Also	CONFigure[1..n]:ACQuisition:RANGe? CONFigure[1..n]:ACQuisition

:CONFigure[1..n]:ACQuisition: WAVelength?

Description

This query returns the current wavelength setting.

*RST reverts this setting to default value.

Syntax

:CONFigure[1..n]:ACQuisition:WAVelength?

Parameter(s)

None

Response Syntax

<Wavelength>

Response(s)

Wavelength:

The response data syntax for <Wavelength> is defined as a <NR3 NUMERIC RESPONSE DATA> element.

Returns the wavelength, in meters.

Example(s)

CONF:ACQ 1310 NM,1250 M,10 NS
CONF:ACQ:WAV? Returns 1.31E–6

See Also

CONFigure[1..n]:ACQuisition:WAVelength:LIST?
CONFigure[1..n]:ACQuisition:RANGe:LIST?
CONFigure[1..n]:ACQuisition:PULSe:LIST?

:CONFigure[1..n]:ACQuisition:WAVelength:LIST?

Description

This query returns the list of all available wavelengths.

*RST does not affect this command.

Syntax

:CONFigure[1..n]:ACQuisition:WAVelength:LIST?

Parameter(s)

None

Response Syntax

<WavelengthList>

Response(s)

WavelengthList:

The response data syntax for <WavelengthList> is defined as a <DEFINITE LENGTH ARBITRARY BLOCK RESPONSE DATA> element.

Returns the list of all available wavelengths, in meters.

Example(s)

CONF:ACQ:WAV:LIST? Returns a wavelength list.

See Also

CONFigure[1..n]:ACQuisition:WAVelength?
CONFigure[1..n]:ACQuisition

:CONFigure[1..n]:ANAlysis:HFACtor

Description

This command sets the helix factor that will be used for the next acquisition.

*RST returns this setting to default value.

Syntax

```
:CONFigure[1..n]:ANAlysis:HFACtor<wsp><HelixFactor>|MAXimum|MINimum|DEFault
```

Parameter(s)

HelixFactor:

The program data syntax for <HelixFactor> is defined as a <numeric_value> element. The <HelixFactor> special forms MINimum, MAXimum and DEFault are accepted on input.

MINimum allows to set the instrument to the smallest supported value.
MAXimum allows to set the instrument to the greatest supported value.
DEFault allows the instrument to select a value for the <HelixFactor> parameter.

Sets the helix factor.

Example(s)

```
CONF:ANA:HFAC? Ex.: Returns 0
CONF:ANA:HFAC 2
CONF:ANA:HFAC? Returns 2
```

:CONFigure[1..n]:ANAlysis:HFACtor?

Description	This query returns the helix factor that will be used for the next acquisition. *RST reverts this setting to default value.
Syntax	:CONFigure[1..n]:ANAlysis:HFACtor?[<wsp>MINimum|MAXimum|DEFault]
Parameter(s)	*Parameter 1:* The program data syntax for the first parameter is defined as a <CHARACTER PROGRAM DATA> element. The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: MINimum|MAXimum|DEFault. MINimum is used to retrieve the instrument's smallest supported value. MAXimum is used to retrieve the instrument's greatest supported value. DEFault is used to retrieve the instrument's default value.
Response Syntax	<HelixFactor>
Response(s)	*HelixFactor:* The response data syntax for <HelixFactor> is defined as a <NR3 NUMERIC RESPONSE DATA> element. Returns the helix factor.
Example(s)	CONF:ANA:HFAC 2 CONF:ANA:HFAC? Returns 2

:CONFigure[1..n]:ANAlysis: IORefraction

Description

This command sets the index of refraction that will be used for the next acquisition.

*RST reverts this setting to default value.

Syntax

```
:CONFigure[1..n]:ANAlysis:IORefraction<wsp>
<IOR>|MAXimum|MINimum|DEFault
```

Parameter(s)

IOR:

The program data syntax for <IOR> is defined as a <numeric_value> element. The <IOR> special forms MINimum, MAXimum and DEFault are accepted on input.

MINimum allows to set the instrument to the smallest supported value.
MAXimum allows to set the instrument to the greatest supported value.
DEFault allows the instrument to select a value for the <IOR> parameter.

Sets the index of refraction.

Example(s)

```
CONF:ANA:IOR? Ex.: Returns 1.4677
CONF:ANA:IOR 1.5
CONF:ANA:IOR? Returns 1.5
```

:CONFigure[1..n]:ANAlysis: IORefraction?

Description

This query returns the index of refraction that will be used for the next acquisition.

*RST reverts this setting to default value.

Syntax

:CONFigure[1..n]:ANAlysis:IORefraction?[<wsp>MINimum|MAXimum|DEFault]

Parameter(s)

Parameter 1:

The program data syntax for the first parameter is defined as a <CHARACTER PROGRAM DATA> element. The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: MINimum|MAXimum|DEFault.

MINimum is used to retrieve the instrument's smallest supported value.
MAXimum is used to retrieve the instrument's greatest supported value.
DEFault is used to retrieve the instrument's default value.

Response Syntax

<IOR>

Response(s)

IOR:

The response data syntax for <IOR> is defined as a <NR3 NUMERIC RESPONSE DATA> element.

Returns the index of refraction.

Example(s)

CONF:ANA:IOR 1.5
CONF:ANA:IOR? Returns 1.5

:CONFigure[1..n]:ANAlysis:RBScatter

Description

This command sets the Rayleigh backscatter that will be used for the next acquisition.

*RST reverts this setting to default value.

Syntax

:CONFigure[1..n]:ANAlysis:RBScatter<wsp><RBS>|MAXimum|MINimum|DEFault

Parameter(s)

RBS:

The program data syntax for <RBS> is defined as a <numeric_value> element. The <RBS> special forms MINimum, MAXimum and DEFault are accepted on input.

MINimum allows to set the instrument to the smallest supported value.
MAXimum allows to set the instrument to the greatest supported value.
DEFault allows the instrument to select a value for the <RBS> parameter.

Sets the Rayleigh backscatter.

Example(s)

CONF:ANA:RBS? Ex.: Returns –79.5
CONF:ANA:RBS –80
CONF:ANA:RBS? Returns –80

:CONFigure[1..n]:ANAlysis: RBScatter?

Description

This query returns the Rayleigh backscatter that will be used for the next acquisition.

*RST reverts this setting to default value.

Syntax

:CONFigure[1..n]:ANAlysis:RBScatter?[<wsp>MINimum|MAXimum|DEFault]

Parameter(s)

Parameter 1:

The program data syntax for the first parameter is defined as a <CHARACTER PROGRAM DATA> element. The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: MINimum|MAXimum|DEFault.

MINimum is used to retrieve the instrument's smallest supported value.
MAXimum is used to retrieve the instrument's greatest supported value.
DEFault is used to retrieve the instrument's default value.

Response Syntax

<RBS>

Response(s)

RBS:

The response data syntax for <RBS> is defined as a <NR3 NUMERIC RESPONSE DATA> element.

Returns the Rayleigh backscatter.

Example(s)

CONF:ANA:RBS –80
CONF:ANA:RBS? Returns –80

:CONFigure[1..n]:ANAlysis:THReshold:EOFiber

Description

This command sets the end-of-fiber threshold that will be used for the next acquisition.

*RST returns this setting to default value.

Syntax

:CONFigure[1..n]:ANAlysis:THReshold:EOFiber<wsp><End-of-Fiber>|MAXimum|MINimum|DEFault

Parameter(s)

End-of-Fiber:

The program data syntax for <End-of-Fiber> is defined as a <numeric_value> element. The <End-of-Fiber> special forms MINimum, MAXimum and DEFault are accepted on input.

MINimum allows to set the instrument to the smallest supported value.
MAXimum allows to set the instrument to the greatest supported value.
DEFault allows the instrument to select a value for the <End-of-Fiber> parameter.

Sets the end-of-fiber threshold.

Example(s)

```
CONF:ANA:THR:EOF? Ex.: Returns 5.0
CONF:ANA:THR:EOF  5.5
CONF:ANA:THR:EOF? Returns 5.5
```

:CONFigure[1..n]:ANAlysis:THReshold:EOFiber?

Description

This query returns the end-of-fiber threshold that will be used for the next acquisition.

*RST reverts this setting to default value.

Syntax

:CONFigure[1..n]:ANAlysis:THReshold:EOFiber?[<wsp>MINimum|MAXimum|DEFault]

Parameter(s)

Parameter 1:

The program data syntax for the first parameter is defined as a <CHARACTER PROGRAM DATA> element. The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: MINimum|MAXimum|DEFault.

MINimum is used to retrieve the instrument's smallest supported value.
MAXimum is used to retrieve the instrument's greatest supported value.
DEFault is used to retrieve the instrument's default value.

Response Syntax

<End-of-Fiber>

Response(s)

End-of-Fiber:

The response data syntax for <End-of-Fiber> is defined as a <NR3 NUMERIC RESPONSE DATA> element.

Returns the end-of-fiber threshold.

Example(s)

CONF:ANA:THR:EOF 5.5
CONF:ANA:THR:EOF? Returns 5.5

:CONFigure[1..n]:ANAlysis:THReshold:REFLectance

Description

This command sets the reflectance threshold that will be used for the next acquisition.

*RST returns this setting to default value.

Syntax

:CONFigure[1..n]:ANAlysis:THReshold:REFLectance<wsp><Reflectance>|MAXimum|MINimum|DEFault

Parameter(s)

Reflectance:

The program data syntax for <Reflectance> is defined as a <numeric_value> element. The <Reflectance> special forms MINimum, MAXimum and DEFault are accepted on input.

MINimum allows to set the instrument to the smallest supported value.
MAXimum allows to set the instrument to the greatest supported value.
DEFault allows the instrument to select a value for the <Reflectance> parameter.

Sets the reflectance threshold.

Example(s)

CONF:ANA:THR:REFL? Ex.: Returns -72.0
CONF:ANA:THR:REFL -72.5
CONF:ANA:THR:REFL? Returns -72.5

:CONFigure[1..n]:ANAlysis:THReshold:REFLectance?

Description

This query returns the reflectance threshold that will be used for the next acquisition.

*RST reverts this setting to default value.

Syntax

:CONFigure[1..n]:ANAlysis:THReshold:REFLectance?[<wsp>MINimum|MAXimum|DEFault]

Parameter(s)

Parameter 1:

The program data syntax for the first parameter is defined as a <CHARACTER PROGRAM DATA> element. The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: MINimum|MAXimum|DEFault.

MINimum is used to retrieve the instrument's smallest supported value.
MAXimum is used to retrieve the instrument's greatest supported value.
DEFault is used to retrieve the instrument's default value.

Response Syntax

<Reflectance>

Response(s)

Reflectance:

The response data syntax for <Reflectance> is defined as a <NR3 NUMERIC RESPONSE DATA> element.

Returns the reflectance threshold.

Example(s)

CONF:ANA:THR:REFL -72.5
CONF:ANA:THR:REFL? Returns -72.5

:CONFigure[1..n]:ANAlysis:THReshold:SLOSs

Description

This command sets the splice loss threshold that will be used for the next acquisition.

*RST returns this setting to default value.

Syntax

:CONFigure[1..n]:ANAlysis:THReshold:SLOSs<wsp><Splice Loss>|MAXimum|MINimum|DEFault

Parameter(s)

Splice Loss:

The program data syntax for <Splice Loss> is defined as a <numeric_value> element. The <Splice Loss> special forms MINimum, MAXimum and DEFault are accepted on input.

MINimum allows to set the instrument to the smallest supported value.
MAXimum allows to set the instrument to the greatest supported value.
DEFault allows the instrument to select a value for the <Splice Loss> parameter.

Sets the splice loss threshold.

Example(s)

CONF:ANA:THR:SLOS? Ex.: Returns 0.02
CONF:ANA:THR:SLOS 0.03
CONF:ANA:THR:SLOS? Returns 0.03

:CONFigure[1..n]:ANAlysis:THReshold:SLOSs?

Description

This query returns the splice loss threshold that will be used for the next acquisition.

*RST reverts this setting to default value.

Syntax

:CONFigure[1..n]:ANAlysis:THReshold:SLOSs?[<wsp>MINimum|MAXimum|DEFault]

Parameter(s)

Parameter 1:

The program data syntax for the first parameter is defined as a <CHARACTER PROGRAM DATA> element. The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: MINimum|MAXimum|DEFault.

MINimum is used to retrieve the instrument's smallest supported value.
MAXimum is used to retrieve the instrument's greatest supported value.
DEFault is used to retrieve the instrument's default value.

Response Syntax

<Splice Loss>

Response(s)

Splice Loss:

The response data syntax for <Splice Loss> is defined as a <NR3 NUMERIC RESPONSE DATA> element.

Returns the splice loss threshold.

Example(s)

CONF:ANA:THR:SLOS 0.03
CONF:ANA:THR:SLOS? Returns 0.03

:ERRor[1..n]?

Description

This query returns the last error or event.

*RST does not affect this query.

Syntax

:ERRor[1..n]?

Parameter(s)

None

Response Syntax

<Error>

Response(s)

Error:

The response data syntax for <Error> is defined as a <DEFINITE LENGTH ARBITRARY BLOCK RESPONSE DATA> element.

Returns the specified error. A zero value in the number field indicates that no error or event has occurred.

Error structure is in A, B, C, D, E, F, G format, where:
A = Source <STRING RESPONSE DATA>
B = Number <NR1 NUMERIC RESPONSE DATA>
C = Description <STRING RESPONSE DATA>

D = HelpFile <STRING RESPONSE DATA>
E = HelpContext <NR1 NUMERIC RESPONSE DATA>
F = Interface <STRING RESPONSE DATA>
G = AdditionalInfo <STRING RESPONSE DATA>

Example(s)

ERR? Ex.: Returns: "#10", if no error
ERR? Ex.: Returns: #3126Exfo.Instrument7000.Instrument7000.1,-1073471488,"An offset error occured in the module.",,,"{...}","Instrument7000:Initialize"

Notes

{...} means GUID

:FETCh[1..n]:ASETting:PULSe?

Description	This query returns the pulse found after an initiate (INIT) command. Note that the acquisition mode (CONF:ACQ:MODE) must be set to ASETting. Since *RST clears the pulse value, the returned value will be 0.
Syntax	:FETCh[1..n]:ASETting:PULSe?
Parameter(s)	None
Response Syntax	<Pulse>
Response(s)	*Pulse:* The response data syntax for <Pulse> is defined as a <NR3 NUMERIC RESPONSE DATA> element. Returns the pulse, in meters.
Example(s)	CONF:ACQ:MODE ASET INIT INIT:STAT? Returns 0 when acquisition is complete. FETC:ASET:PULS? Ex.: Returns 1E–8
See Also	CONFigure[1..n]:ACQuisition:RANGe? CONFigure[1..n]:ACQuisition:PULSe? CONFigure[1..n]:ACQuisition:DURation? CONFigure[1..n]:ACQuisition:WAVelength:LIST? CONFigure[1..n]:ACQuisition:RANGe:LIST? CONFigure[1..n]:ACQuisition:PULSe:LIST? CONFigure[1..n]:ACQuisition

:FETCh[1..n]:ASETting:RANGe?

Description

This query returns the range found after an initiate (INIT) command. Note that the acquisition mode (CONF:ACQ:MODE) must be set to ASETting.

Since *RST clears the range value, the returned value will be 0.

Syntax

:FETCh[1..n]:ASETting:RANGe?

Parameter(s)

None

Response Syntax

<Range>

Response(s)

Range:

The response data syntax for <Range> is defined as a <NR3 NUMERIC RESPONSE DATA> element.

Returns the range, in meters.

Example(s)

CONF:ACQ:MODE ASET
INIT
INIT:STAT? Returns 0 when acquisition is complete.
FETC:ASET:RANG? Ex.: Returns 1.25E+3

See Also

CONFigure[1..n]:ACQuisition:RANGe?
CONFigure[1..n]:ACQuisition:PULSe?
CONFigure[1..n]:ACQuisition:DURation?
CONFigure[1..n]:ACQuisition:WAVelength:LIST?
CONFigure[1..n]:ACQuisition:RANGe:LIST?

CONFigure[1..n]:ACQuisition:PULSe:LIST?
CONFigure[1..n]:ACQuisition

:FETCh[1..n]:CFConnector?

Description

This query returns a state indicating whether the first connector has been found or not, after an initiate (INIT) command. Note that the acquisition mode (CONF:ACQ:MODE) must be set to CFConnector.

*RST clears this setting.

Syntax

:FETCh[1..n]:CFConnector?

Parameter(s)

None

Response Syntax

<CheckFirstConnectorState>

Response(s)

CheckFirstConnectorState:

The response data syntax for <CheckFirstConnectorState> is defined as a <NR1 NUMERIC RESPONSE DATA> element.

The current <CheckFirstConnectorState>, where:
1 - (TRUE) connector was found.
0 - (FALSE) connector was not found.

Example(s)

CONF:ACQ:MODE CFC
INIT
INIT:STAT? Returns 0 when acquisition is complete.
FETC:CFC? Returns 1 if the connector has been found.

:FETCh[1..n]:DURation?

Description

This query returns the duration for the trace corresponding to the specified trace index.

*RST clears this setting.

Syntax

:FETCh[1..n]:DURation?<wsp>TRC1|TRC2|TRC3|TRC4

Parameter(s)

Label:

The program data syntax for the first parameter is defined as a <CHARACTER PROGRAM DATA> element. The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: TRC1|TRC2|TRC3|TRC4.

Trace index of the available wavelengths.

Response Syntax

<Duration>

Response(s)

Duration:

The response data syntax for <Duration> is defined as a <NR1 NUMERIC RESPONSE DATA> element.

Returns the duration.

Example(s)

CONF:ACQ:DUR 15
CONF:ACQ:MODE ACQ
INIT
INIT:STAT? Returns 0 when acquisition is complete.
FETC:DUR? TRC1 Returns 15

See Also

FETCh[1..n]:ASETting:DURation?
MMEMory[1..n]:LOAD:TRACe
TRACe[1..n]:CATalog?

:FETCh[1..n]:HRESolution?

Description

This query returns a value indicating if the high-resolution feature was enabled for the current trace.

*RST clears this setting.

Syntax

:FETCh[1..n]:HRESolution?<wsp>TRC1|TRC2|TRC3|TRC4

Parameter(s)

Label:

The program data syntax for the first parameter is defined as a <CHARACTER PROGRAM DATA> element. The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: TRC1|TRC2|TRC3|TRC4.

Trace index of the available wavelengths.

Response Syntax

<HighResolution>

Response(s)

HighResolution:

The response data syntax for <HighResolution> is defined as a <NR1 NUMERIC RESPONSE DATA> element.

Indicates if the high-resolution feature was enabled or not for the current trace.

Example(s)

FETC:HRES? TRC1 Returns 1 if the high-resolution feature was enabled for the current trace.

See Also

CONFigure[1..n]:ACQuisition:HRESolution

:FETCh[1..n]:LFIBer?

Description	This query returns a state indicating whether live activity has been found on the fiber, after an initiate (INIT) command. This is valid for all acquisition modes. *RST clears this setting.
Syntax	:FETCh[1..n]:LFIBer?
Parameter(s)	None
Response Syntax	<LiveFiberState>
Response(s)	*LiveFiberState:* The response data syntax for <LiveFiberState> is defined as a <NR1 NUMERIC RESPONSE DATA> element. The current <LiveFiberState>, where: 1 - (TRUE) a live activity was found on fiber. 0 - (FALSE) no live activity found on fiber.
Example(s)	INIT INIT:STAT? Returns 0 when acquisition is complete. FETC:LFIB? Returns 1 if a live activity was found on fiber.

:FETCh[1..n]:PULSe?

Description

This query returns the pulse for the specified trace index.

*RST clears this setting.

Syntax

:FETCh[1..n]:PULSe?<wsp>TRC1|TRC2|TRC3|TRC4

Parameter(s)

Label:

The program data syntax for the first parameter is defined as a <CHARACTER PROGRAM DATA> element. The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: TRC1|TRC2|TRC3|TRC4.

Trace index of the available wavelengths.

Response Syntax

<Pulse>

Response(s)

Pulse:

The response data syntax for <Pulse> is defined as a <NR3 NUMERIC RESPONSE DATA> element.

Returns the pulse, in seconds.

Example(s)

CONF:ACQ 1310,NM1250,M10 NS
CONF:ACQ:MODE ACQ
INIT
INIT:STAT? Returns 0 when acquisition is complete.
FETC:PULS? TRC1 Returns 1E–8

See Also

FETCh[1..n]:ASETting:PULSe?
MMEMory[1..n]:LOAD:TRACe
TRACe[1..n]:CATalog?

:FETCh[1..n]:RANGe?

Description	This query returns the range for the trace corresponding to the specified trace index. *RST clears this setting.
Syntax	:FETCh[1..n]:RANGe?<wsp>TRC1|TRC2|TRC3|TRC4
Parameter(s)	*Label:* The program data syntax for the first parameter is defined as a <CHARACTER PROGRAM DATA> element. The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: TRC1|TRC2|TRC3|TRC4. Trace index of the available wavelengths.
Response Syntax	<Range>
Response(s)	*Range:* The response data syntax for <Range> is defined as a <NR3 NUMERIC RESPONSE DATA> element. Returns the range, in meters.
Example(s)	CONF:ACQ 1310,NM1250,M10 NS CONF:ACQ:MODE ACQ INIT INIT:STAT? Returns 0 when acquisition is complete. FETC:RANG? TRC1 Returns 1.25E+3
See Also	FETCh[1..n]:ASETting:RANGe? MMEMory[1..n]:LOAD:TRACe TRACe[1..n]:CATalog?

:FETCh[1..n]:STEP?

Description	This query returns the step between each point of the trace corresponding to the specified trace index. *RST clears this setting.
Syntax	:FETCh[1..n]:STEP?<wsp>TRC1|TRC2|TRC3|TRC4
Parameter(s)	*Label:* The program data syntax for the first parameter is defined as a <CHARACTER PROGRAM DATA> element. The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: TRC1|TRC2|TRC3|TRC4. Trace index of the available wavelengths.
Response Syntax	<Step>
Response(s)	*Step:* The response data syntax for <Step> is defined as a <NR3 NUMERIC RESPONSE DATA> element. Returns the step value, in meters.
Example(s)	CONF:ACQ:MODE ACQ INIT INIT:STAT? Returns 0 when acquisition is complete. FETC:STEP? TRC1 Ex.: Returns 0.07979
See Also	MMEMory[1..n]:LOAD:TRACe TRACe[1..n]:CATalog?

:FETCh[1..n]:TRACe[1..n][:DATA]?

Description

This query returns all the points of a trace. It can be used with already-completed acquisitions or acquisitions in progress.

*RST clears this setting.

Syntax

:FETCh[1..n]:TRACe[1..n][:DATA]?

Parameter(s)

None

Response Syntax

<Data>

Response(s)

Data:

The response data syntax for <Data> is defined as a <DEFINITE LENGTH ARBITRARY BLOCK RESPONSE DATA> element.

Returns a list of power values representing the trace.
Each power value represents a point of the trace and is always returned in dB as a <NR3 NUMERIC RESPONSE DATA> type.

Example(s)

CONF:ACQ:MODE ACQ
INIT
INIT:STAT? Returns 1 when acquisition is in progress
FETC:TRAC? Returns a trace, while acquisition is in progress or complete

See Also

FETCh[1..n]:TRACe[1..n]:POIN?
TRACe[1..n]:CATalog?

:FETCh[1..n]:TRACe[1..n]:POINts?

Description	This query returns the number of points of the trace. It can be used with already-completed acquisitions or acquisitions in progress. *RST clears this setting.
Syntax	:FETCh[1..n]:TRACe[1..n]:POINts?
Parameter(s)	None
Response Syntax	<PointsCount>
Response(s)	*PointsCount:* The response data syntax for <PointsCount> is defined as a <NR1 NUMERIC RESPONSE DATA> element. Returns the number of points.
Example(s)	CONF:ACQ:MODE ACQ INIT INIT:STAT? Returns 1 when acquisition is in progress FETC:TRAC:POIN? Returns the number of points of the current FETC:TRAC?
See Also	FETCh[1..n]:TRACe[1..n][:DATA]?

:FETCh[1..n]:WAVelength?

Description

This query returns the wavelength for the trace corresponding to the specified trace index.

*RST clears this setting.

Syntax

:FETCh[1..n]:WAVelength?<wsp>TRC1|TRC2|TRC3|TRC4

Parameter(s)

Label:

The program data syntax for the first parameter is defined as a <CHARACTER PROGRAM DATA> element. The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: TRC1|TRC2|TRC3|TRC4.

Trace index of the available wavelengths.

Response Syntax

<Wavelength>

Response(s)

Wavelength:

The response data syntax for <Wavelength> is defined as a <NR3 NUMERIC RESPONSE DATA> element.

Returns the wavelength, in meters.

Example(s)

CONF:ACQ 1310,NM1250,M10 NS
CONF:ACQ:MODE ACQ
INIT
INIT:STAT? Returns 0 when acquisition is complete.
FETC:WAV? TRC1 Returns 1.31E–6

See Also

MMEMory[1..n]:LOAD:TRACe
TRACe[1..n]:CATalog?

:INITiate[1..n][:IMMediate]

Description

This command starts the acquisition according to the active acquisition mode.

Acquisition mode:
ACQuisition: Acquisition stops after the duration value has elapsed.

REALtime: Acquisition is in progress until an abort event is sent.
CFConnector: Acquisition stops after determining the injection level at the first connector.
ASETting: Acquisition stops after determining the adequate range and pulse values.

This command is asynchronous.

This command is an event and, therefore, has no associated *RST condition or query form. However, on *RST, the equivalent of an ABORt command is performed on any acquisition in progress.

Syntax

:INITiate[1..n][:IMMediate]

Parameter(s)

None

Example(s)

INIT

See Also

CONFigure[1..n]:ACQuisition:MODE
INITiate[1..n]:STATe?
ABORt[1..n]

:INITiate[1..n]:STATe?

Description

This query returns a state indicating whether an acquisition is in progress or stopped (ABORt).

*RST sets state to OFF (all acquisitions are stopped).

Syntax

:INITiate[1..n]:STATe?

Parameter(s)

None

Response Syntax

<AcquisitionState>

Response(s)

AcquisitionState:

The response data syntax for <AcquisitionState> is defined as a <NR1 NUMERIC RESPONSE DATA> element.

The current acquisition <AcquisitionState>, where:
1 - (TRUE) acquisition is in progress.
0 - (FALSE) acquisition is complete.

Example(s)

INIT
INIT:STAT? Returns 0 or 1

See Also

CONFigure[1..n]:ACQuisition:MODE
ABORt[1..n]

:MMEMory[1..n]:DATA:TYPE

Description

This command sets file format for a trace to be saved to a file.

*RST sets type to BINARY.

Syntax

:MMEMory[1..n]:DATA:TYPE<wsp>BINary|BELLcore

Parameter(s)

FileType:

The program data syntax for the first parameter is defined as a <CHARACTER PROGRAM DATA> element. The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: BINary|BELLcore.

Sets the file format.

Example(s)

MMEM:DATA:TYPE? Ex.: Returns BINARY

See Also

CONFigure[1..n]:ACQuisition:MODE
INITiate[1..n][:IMMediate]
MMEMory[1..n]:STORe:TRACe
MMEMory[1..n]:LOAD:TRACe

:MMEMory[1..n]:DATA:TYPE?

Description	This query returns the current file format. *RST sets type to BINARY.
Syntax	:MMEMory[1..n]:DATA:TYPE?
Parameter(s)	None
Response Syntax	<FileType>
Response(s)	*FileType:* The response data syntax for <FileType> is defined as a <CHARACTER RESPONSE DATA> element. Returns the file format.
Example(s)	MMEM:DATA:TYPE BIN MMEM:DATA:TYPE? Returns BINARY
Notes	Will not change if a different file type is loaded.
See Also	MMEMory[1..n]:LOAD:TRACe

:MMEMory[1..n]:LOAD:NAME?

Description

This query returns the name of the current loaded file.

*RST clears this setting.

Syntax

:MMEMory[1..n]:LOAD:NAME?

Parameter(s)

None

Response Syntax

<FileName>

Response(s)

FileName:

The response data syntax for <FileName> is defined as a <STRING RESPONSE DATA> element.

Returns the loaded file name.

Example(s)

MMEM:LOAD:TRAC "Trace1.trc"
MMEM:LOAD:NAME? Returns "Trace1.trc"

See Also

MMEMory[1..n]:LOAD:TRACe
MMEMory[1..n]:STORe:TRACe

:MMEMory[1..n]:LOAD:TRACe

Description	This command is used to load traces from a file. *RST does not affect this command.
Syntax	:MMEMory[1..n]:LOAD:TRACe<wsp><FileName>
Parameter(s)	*FileName:* The program data syntax for <FileName> is defined as a <STRING PROGRAM DATA> element. The <FileName> parameter can either be only the filename or the filename and its path. If no path is specified, the default path is used. The default path name depends on the location of the installation directory.
Example(s)	MMEM:LOAD:TRAC "Trace1.trc"
Notes	No effect on MMEM:DATA:TYPE?
See Also	MMEMory[1..n]:DATA:TYPE? CONFigure[1..n]:ACQuisition:MODE NITiate[1..n][:IMMediate] MMEMory[1..n]:STORe:TRACe

:MMEMory[1..n]:STORe:TRACe

Description

This command is used to store traces to a file.

*RST does not affect this command.

Syntax

```
:MMEMory[1..n]:STORe:TRACe<wsp><FileNam
e>
```

Parameter(s)

FileName:

The program data syntax for <FileName> is defined as a <STRING PROGRAM DATA> element.

The <FileName> parameter can either be only the filename or the filename and its path.

If no path is specified, the default path is used. The default path name depends on the location of the installation directory.

Example(s)

CONF:ACQ:MODE ACQ
INIT
INIT:STAT? Returns 0 when acquisition is complete.
MMEM:STOR:TRAC "Trace2.trc"

See Also

MMEMory[1..n]:LOAD:TRACe
MMEMory[1..n]:DATA:TYPE
MMEM:STORe:TRACe:OVERwrite

:MMEMory[1..n]:STORe:TRACe:OVERwrite

Description

This command specifies if an existing file can be overwritten without generating an error when the MMEMory:STORe:TRACe command is used. Attempting to save a new file under the name of an existing file will generate an error if the value is set to OFF.

*RST sets overwrite to OFF.

Syntax

```
:MMEMory[1..n]:STORe:TRACe:OVERwrite<wsp><Overwrite>
```

Parameter(s)

Overwrite:

The program data syntax for <Overwrite> is defined as a <Boolean Program Data> element. The <Overwrite> special forms ON and OFF are accepted on input for increased readability. ON corresponds to 1 and OFF corresponds to 0.

Enables or disables the right to overwrite an existing file.

Example(s)

```
CONF:ACQ:MODE ACQ
INIT
INIT:STAT? Returns 0 when acquisition is complete.
MMEM:STOR:TRAC:OVER? Ex.: Returns 0
MMEM:STOR:TRAC "Trace3.trc" If file already exists, an error occurs.

MMEM:STOR:TRAC:OVER 1
MMEM:STOR:TRAC "Trace3.trc" File will be saved without generating errors.
```

:MMEMory[1..n]:STORe:TRACe: OVERwrite?

Description

This query indicates if an existing file can be overwritten.

*RST sets overwrite to OFF.

Syntax

:MMEMory[1..n]:STORe:TRACe:OVERwrite?

Parameter(s)

None

Response Syntax

<Overwrite>

Response(s)

Overwrite:

The response data syntax for <Overwrite> is defined as a <NR1 NUMERIC RESPONSE DATA> element.

Overwrite state.
1 - (TRUE) Always overwrites file.
0 - (FALSE) Does not overwrite file if it already exists.

Example(s)

MMEM:STOR:TRAC:OVER 1
MMEM:STOR:TRAC:OVER? Returns 1

:SOURce[1..n]:FREQuency:BURSt

Description	This command sets the frequency of the ON-OFF modulated signal of the source during its ON period (modulation for fiber identification). This signal is referred to as "burst signal" . *RST reverts this setting to its default value.
Syntax	:SOURce[1..n]:FREQuency:BURSt<wsp><Burst Frequency>|MAXimum|MINimum|DEFault
Parameter(s)	*BurstFrequency:* The program data syntax for <BurstFrequency> is defined as a <numeric_value> element. The <BurstFrequency> special forms MINimum, MAXimum and DEFault are accepted on input. MINimum allows to set the instrument to the smallest supported value. MAXimum allows to set the instrument to the greatest supported value. DEFault allows the instrument to select a value for the <BurstFrequency> parameter. Frequency of the burst signal of the source, in hertz.
Example(s)	SOUR:FREQ:BURS 1000 SOUR:FREQ:BURS:STAT ON SOUR:POW:STAT:TIME 60 SOUR:POW:STAT ON
See Also	SOURce[1..n]:FREQuency:BURSt? SOURce[1..n]:FREQuency:BURSt:STATe SOURce[1..n]:FREQuency:PRF SOURce[1..n]:FREQuency:PRF:STATe SOURce[1..n]:POWer:STATe SOURce[1..n]:POWer:STATe:TIME

:SOURce[1..n]:FREQuency:BURSt?

Description

This query returns the frequency of the ON-OFF modulated signal of the source during its ON period (modulation for fiber identification). This signal is referred to as "burst signal" .

*RST reverts this setting to its default value.

Syntax

:SOURce[1..n]:FREQuency:BURSt?[<wsp>MINimum|MAXimum|DEFault]

Parameter(s)

Parameter 1:

The program data syntax for the first parameter is defined as a <CHARACTER PROGRAM DATA> element. The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: MINimum|MAXimum|DEFault.

MINimum is used to retrieve the instrument's smallest supported value.
MAXimum is used to retrieve the instrument's greatest supported value.
DEFault is used to retrieve the instrument's default value.

Response Syntax

<BurstFrequency>

Response(s)

BurstFrequency:

The response data syntax for <BurstFrequency> is defined as a <NR3 NUMERIC RESPONSE DATA> element.

Frequency of the burst signal of the source, in hertz.

Example(s)

SOUR:FREQ:BURS 1000
SOUR:FREQ:BURS? Returns 1.000000e+3

See Also

SOURce[1..n]:FREQuency:BURSt
SOURce[1..n]:FREQuency:BURSt:STATe
SOURce[1..n]:FREQuency:PRF
SOURce[1..n]:FREQuency:PRF:STATe
SOURce[1..n]:POWer:STATe
SOURce[1..n]:POWer:STATe:TIME

:SOURce[1..n]:FREQuency:BURSt:STATe

Description

This command turns on or off the burst signal of the source (modulation for fiber identification).

At *RST, the burst signal state of the source is set to OFF (source emits in continuous output- CW).

Syntax

:SOURce[1..n]:FREQuency:BURSt:STATe<wsp><State>

Parameter(s)

State:

The program data syntax for <State> is defined as a <Boolean Program Data> element. The <State> special forms ON and OFF are accepted on input for increased readability. ON corresponds to 1 and OFF corresponds to 0.

Burst signal state of the source (on or off).
ON: Modulation for fiber identification
OFF: CW (continuous output)

Example(s)

```
SOUR:FREQ:BURS 1000
SOUR:FREQ:BURS:STAT ON
SOUR:POW:STAT:TIME 60
SOUR:POW:STAT ON
```

See Also

SOURce[1..n]:FREQuency:BURSt
SOURce[1..n]:FREQuency:BURSt:STATe?
SOURce[1..n]:FREQuency:PRF
SOURce[1..n]:FREQuency:PRF:STATe
SOURce[1..n]:POWer:STATe
SOURce[1..n]:POWer:STATe:TIME

:SOURce[1..n]:FREQuency:BURSt: STATe?

Description	This query returns a value indicating the current state of the burst signal of the source. At *RST, the burst signal state of the source is set to OFF (source emits in continuous output- CW).
Syntax	:SOURce[1..n]:FREQuency:BURSt:STATe?
Parameter(s)	None
Response Syntax	<State>
Response(s)	*State:* The response data syntax for <State> is defined as a <NR1 NUMERIC RESPONSE DATA> element. Burst signal state of the source (on or off). ON: Modulation for fiber identification OFF: CW (continuous output)
Example(s)	SOUR:FREQ:BURS:STAT ON SOUR:FREQ:BURS:STAT? Returns 1
See Also	SOURce[1..n]:FREQuency:BURSt SOURce[1..n]:FREQuency:BURSt:STATe SOURce[1..n]:FREQuency:PRF SOURce[1..n]:FREQuency:PRF:STATe SOURce[1..n]:POWer:STATe SOURce[1..n]:POWer:STATe:TIME

:SOURce[1..n]:FREQuency:PRF

Description

This command sets the repetition frequency of the on-off modulation of the source signal that is periodically switched on and off (flashing pattern). This characteristic is referred to as "Pulsed Repetition Frequency" (PRF).

*RST reverts this setting to its default value.

Syntax

:SOURce[1..n]:FREQuency:PRF<wsp><PulsedRepetitionFrequency>|MAXimum|MINimum|DEFault

Parameter(s)

PulsedRepetitionFrequency:

The program data syntax for <PulsedRepetitionFrequency> is defined as a <numeric_value> element. The <PulsedRepetitionFrequency> special forms MINimum, MAXimum and DEFault are accepted on input.

MINimum allows to set the instrument to the smallest supported value.
MAXimum allows to set the instrument to the greatest supported value.
DEFault allows the instrument to select a value for the <PulsedRepetitionFrequency> parameter.

Pulsed Repetition Frequency (PRF) of the signal of the source.

Example(s)

```
SOUR:FREQ:PRF 10
SOUR:FREQ:PRF:STAT ON
SOUR:POW:STAT:TIME 60
SOUR:POW:STAT ON
```

Notes

Using a flashing pattern makes fiber identification easier. In a flashing pattern, the modulated signal will be sent for 1 second, then will be off for the next second, then will be sent again for 1 second, and so on.

See Also

SOURce[1..n]:FREQuency:PRF?
SOURce[1..n]:FREQuency:PRF:STATe
SOURce[1..n]:FREQuency:BURSt
SOURce[1..n]:FREQuency:BURSt:STATe
SOURce[1..n]:POWer:STATe
SOURce[1..n]:POWer:STATe:TIME

:SOURce[1..n]:FREQuency:PRF?

Description

This query returns the repetition frequency of the on-off modulation of the source signal that is periodically switched on and off (flashing pattern). This characteristic is referred to as "Pulsed Repetition Frequency" (PRF).

*RST reverts this setting to its default value.

Syntax

:SOURce[1..n]:FREQuency:PRF?[<wsp>MINimum|MAXimum|DEFault]

Parameter(s)

Parameter 1:

The program data syntax for the first parameter is defined as a <CHARACTER PROGRAM DATA> element. The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: MINimum|MAXimum|DEFault.

MINimum is used to retrieve the instrument's smallest supported value.
MAXimum is used to retrieve the instrument's greatest supported value.
DEFault is used to retrieve the instrument's default value.

Response Syntax

<PulsedRepetitionFrequency>

Response(s)

PulsedRepetitionFrequency:

The response data syntax for <PulsedRepetitionFrequency> is defined as a <NR3 NUMERIC RESPONSE DATA> element.

Pulsed Repetition Frequency (PRF) of the signal of the source.

Example(s)

SOUR:FREQ:PRF 10
SOUR:FREQ:PRF? Returns 1.000000e+1

See Also

SOURce[1..n]:FREQuency:PRF
SOURce[1..n]:FREQuency:PRF:STATe
SOURce[1..n]:FREQuency:BURSt
SOURce[1..n]:FREQuency:BURSt:STATe
SOURce[1..n]:POWer:STATe
SOURce[1..n]:POWer:STATe:TIME

:SOURce[1..n]:FREQuency:PRF:STATe

Description

This command is used to turn on or off the pulsed repetition frequency (PRF) of the source (enable or disable the flashing pattern).

At *RST, the PRF signal state is set to OFF.

Syntax

```
:SOURce[1..n]:FREQuency:PRF:STATe<wsp><State>
```

Parameter(s)

State:

The program data syntax for <State> is defined as a <Boolean Program Data> element. The <State> special forms ON and OFF are accepted on input for increased readability. ON corresponds to 1 and OFF corresponds to 0.

State of the PRF signal of the source.

Example(s)

```
SOUR:FREQ:PRF 10
SOUR:FREQ:PRF:STAT ON
SOUR:POW:STAT:TIME 60
SOUR:POW:STAT ON
```

See Also

SOURce[1..n]:FREQuency:PRF:STATe?
SOURce[1..n]:FREQuency:PRF
SOURce[1..n]:FREQuency:BURSt
SOURce[1..n]:FREQuency:BURSt:STATe
SOURce[1..n]:POWer:STATe
SOURce[1..n]:POWer:STATe:TIME

:SOURce[1..n]:FREQuency:PRF:STATe?

Description

This query returns a value indicating the current state of the pulsed repetition frequency (PRF) signal (flashing pattern enabled or disabled) of the source.

At *RST, the PRF signal state is set to OFF.

Syntax

:SOURce[1..n]:FREQuency:PRF:STATe?

Parameter(s)

None

Response Syntax

<State>

Response(s)

State:

The response data syntax for <State> is defined as a <NR1 NUMERIC RESPONSE DATA> element.

State of the PRF signal of the source.

Example(s)

```
SOUR:FREQ:PRF:STAT ON
SOUR:FREQ:PRF:STAT?  Returns 1
```

See Also

SOURce[1..n]:FREQuency:PRF
SOURce[1..n]:FREQuency:PRF:STATe
SOURce[1..n]:FREQuency:BURSt
SOURce[1..n]:FREQuency:BURSt:STATe
SOURce[1..n]:POWer:STATe
SOURce[1..n]:POWer:STATe:TIME

:SOURce[1..n]:POWer:STATe

Description

This command turns the source on or off.

*RST sets the source to OFF.

Syntax

:SOURce[1..n]:POWer:STATe<wsp><State>

Parameter(s)

State:

The program data syntax for <State> is defined as a <Boolean Program Data> element. The <State> special forms ON and OFF are accepted on input for increased readability. ON corresponds to 1 and OFF corresponds to 0.

New power state of the source.
1 or ON, turns the source on.
0 or OFF, turns the source off.

Example(s)

```
SOUR:POW:STAT:TIME 60
SOUR:POW:STAT ON
```

See Also

SOURce[1..n]:POWer:STATe?
SOURce[1..n]:POWer:STATe:TIME
SOURce[1..n]:FREQuency:PRF
SOURce[1..n]:FREQuency:PRF:STATe
SOURce[1..n]:FREQuency:BURSt
SOURce[1..n]:FREQuency:BURSt:STATe

:SOURce[1..n]:POWer:STATe?

Description

This query returns a value indicating the state of the source (on or off).

*RST sets the source to OFF.

Syntax

:SOURce[1..n]:POWer:STATe?

Parameter(s)

None

Response Syntax

<State>

Response(s)

State:

The response data syntax for <State> is defined as a <NR1 NUMERIC RESPONSE DATA> element.

State of the source power.
0: Source is off.
1: Source is on.

Example(s)

SOUR:POW:STAT ON
SOUR:POW:STAT? Returns 1

See Also

SOURce[1..n]:POWer:STATe
SOURce[1..n]:POWer:STATe:TIME
SOURce[1..n]:FREQuency:PRF
SOURce[1..n]:FREQuency:PRF:STATe
SOURce[1..n]:FREQuency:BURSt
SOURce[1..n]:FREQuency:BURSt:STATe

:SOURce[1..n]:POWer:STATe:TIME

Description

This command sets the duration after which the source will stop emitting light automatically (auto-off feature).
Note that this command does not turn the source on.

*RST sets this value to 600 seconds.

Syntax

:SOURce[1..n]:POWer:STATe:TIME<wsp><Duration>

Parameter(s)

Duration:

The program data syntax for <Duration> is defined as a <DECIMAL NUMERIC PROGRAM DATA> element.

Duration after which the source will stop emitting light automatically, in seconds.

Example(s)

SOUR:POW:STAT:TIME 60
SOUR:POW:STAT ON

See Also

SOURce[1..n]:POWer:STATe:TIME?
SOURce[1..n]:POWer:STATe
SOURce[1..n]:FREQuency:PRF
SOURce[1..n]:FREQuency:PRF:STATe
SOURce[1..n]:FREQuency:BURSt
SOURce[1..n]:FREQuency:BURSt:STATe

:SOURce[1..n]:POWer:STATe:TIME?

Description

This query returns a value indicating the duration after which the source will stop emitting light automatically (auto-off feature).

*RST sets this value to 600 seconds.

Syntax

:SOURce[1..n]:POWer:STATe:TIME?

Parameter(s)

None

Response Syntax

<Duration>

Response(s)

Duration:

The response data syntax for <Duration> is defined as a <NR3 NUMERIC RESPONSE DATA> element.

Duration after which the source will stop emitting light automatically, in seconds.

Example(s)

SOUR:POW:STAT:TIME 60
SOUR:POW:STAT:TIME? Returns 60

See Also

SOURce[1..n]:POWer:STATe:TIME
SOURce[1..n]:POWer:STATe
SOURce[1..n]:FREQuency:PRF
SOURce[1..n]:FREQuency:PRF:STATe
SOURce[1..n]:FREQuency:BURSt
SOURce[1..n]:FREQuency:BURSt:STATe

:SOURce[1..n]:VFLocator:AM:INTernal:FREQuency

Description

This command selects the internal modulation frequency of the visual fault locator (VFL). The internal modulation corresponds to 50 % of the duty cycle at the selected frequency.

*RST sets the modulation frequency to 1 Hz.

Syntax

:SOURce[1..n]:VFLocator:AM:INTernal:FREQuency<wsp><Frequency>|MAXimum|MINimum|DEFault

Parameter(s)

Frequency:

The program data syntax for <Frequency> is defined as a <numeric_value> element. The <Frequency> special forms MINimum, MAXimum and DEFault are accepted on input.

MINimum allows to set the instrument to the smallest supported value.
MAXimum allows to set the instrument to the greatest supported value.
DEFault allows the instrument to select a value for the <Frequency> parameter.

New modulation frequency: 1 to 1000.

Example(s)

```
SOUR:VFL:AM:INT:FREQ 1
SOUR:VFL:AM:STAT ON
SOUR:VFL:POW:STAT ON
```

See Also

SOURce[1..n]:VFLocator:AM:INTernal:FREQuency?
SOURce[1..n]:VFLocator:AM:STATe
SOURce[1..n]:VFLocator:POWer:STATe
SOURce[1..n]:VFLocator:POWer:STATe:TIME

:SOURce[1..n]:VFLocator:AM:INTernal:FREQuency?

Description

This query returns a value indicating the current internal modulation frequency.

*RST sets the modulation frequency to 1 Hz.

Syntax

:SOURce[1..n]:VFLocator:AM:INTernal:FREQuency?[<wsp>MINimum|MAXimum|DEFault]

Parameter(s)

Parameter 1:

The program data syntax for the first parameter is defined as a <CHARACTER PROGRAM DATA> element. The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: MINimum|MAXimum|DEFault.

MINimum is used to retrieve the instrument's smallest supported value.
MAXimum is used to retrieve the instrument's greatest supported value.
DEFault is used to retrieve the instrument's default value.

Response Syntax

<Frequency>

Response(s)

Frequency:

The response data syntax for <Frequency> is defined as a <NR3 NUMERIC RESPONSE DATA> element.

The <Frequency> response corresponds to the internal modulation frequency of the VFL, in Hz. If the VFL is in CW mode, the returned value is 0.

Example(s)

SOUR:VFL:AM:INT:FREQ 1
SOUR:VFL:AM:INT:FREQ? Returns 1

See Also

SOURce[1..n]:VFLocator:AM:INTernal:FREQuency
SOURce[1..n]:VFLocator:AM:STATe
SOURce[1..n]:VFLocator:POWer:STATe
SOURce[1..n]:VFLocator:POWer:STATe:TIME

:SOURce[1..n]:VFLocator:AM:STATe

Description

This command turns ON or OFF the amplitude modulation of the visual fault locator (VFL).

At *RST, this value is set to OFF.

Syntax

```
:SOURce[1..n]:VFLocator:AM:STATe<wsp><Sta
te>
```

Parameter(s)

State:

The program data syntax for <State> is defined as a <Boolean Program Data> element. The <State> special forms ON and OFF are accepted on input for increased readability. ON corresponds to 1 and OFF corresponds to 0.

The <State> parameter corresponds to the amplitude modulation state of the VFL.

Example(s)

```
SOUR:VFL:AM:INT:FREQ 1
SOUR:VFL:POW:STAT:TIME 60
SOUR:VFL:AM:STAT ON
SOUR:VFL:POW:STAT ON
```

See Also

SOURce[1..n]:VFLocator:AM:STATe?
SOURce[1..n]:VFLocator:AM:INTernal:FREQuenc
y
SOURce[1..n]:VFLocator:POWer:STATe
SOURce[1..n]:VFLocator:POWer:STATe:TIME

:SOURce[1..n]:VFLocator:AM:STATe?

Description

This query returns a value indicating the current state of the amplitude modulation (on or off) of the visual fault locator (VFL).

At *RST, the amplitude modulation state is set to OFF.

Syntax

:SOURce[1..n]:VFLocator:AM:STATe?

Parameter(s)

None

Response Syntax

<State>

Response(s)

State:

The response data syntax for <State> is defined as a <NR1 NUMERIC RESPONSE DATA> element.

Amplitude modulation state of the VFL.
ON: Signal is modulated.
OFF: Signal is continuous (CW).

Example(s)

SOUR:VFL:AM:STAT ON
SOUR:VFL:AM:STAT? Returns 1

See Also

SOURce[1..n]:VFLocator:AM:STATe
SOURce[1..n]:VFLocator:AM:INTernal:FREQuency
SOURce[1..n]:VFLocator:POWer:STATe
SOURce[1..n]:VFLocator:POWer:STATe:TIME

:SOURce[1..n]:VFLocator:POWer:STATe

Description

This command turns the visual fault locator (VFL) on or off.

*RST sets the visual fault locator to OFF.

Syntax

```
:SOURce[1..n]:VFLocator:POWer:STATe<wsp>
<State>
```

Parameter(s)

State:

The program data syntax for <State> is defined as a <Boolean Program Data> element. The <State> special forms ON and OFF are accepted on input for increased readability. ON corresponds to 1 and OFF corresponds to 0.

New power state of the VFL.

Example(s)

```
SOUR:VFL:AM:INT:FREQ 1
SOUR:VFL:POW:STAT:TIME 60
SOUR:VFL:AM:STAT ON
SOUR:VFL:POW:STAT ON
```

See Also

SOURce[1..n]:VFLocator:POWer:STATe?
SOURce[1..n]:VFLocator:POWer:STATe:TIME
SOURce[1..n]:VFLocator:AM:STATe?
SOURce[1..n]:VFLocator:AM:INTernal:FREQuenc
y

:SOURce[1..n]:VFLocator:POWer:STATe?

Description

This query returns a value indicating if the visual fault locator (VFL) is on or off.

*RST sets the VFL to OFF.

Syntax

:SOURce[1..n]:VFLocator:POWer:STATe?

Parameter(s)

None

Response Syntax

<State>

Response(s)

State:

The response data syntax for <State> is defined as a <NR1 NUMERIC RESPONSE DATA> element.

Power state of the VFL (on or off).

Example(s)

SOUR:VFL:POW:STAT ON
SOUR:VFL:POW:STAT? Returns 1

See Also

SOURce[1..n]:VFLocator:POWer:STATe
SOURce[1..n]:VFLocator:POWer:STATe:TIME
SOURce[1..n]:VFLocator:AM:STATe?
SOURce[1..n]:VFLocator:AM:INTernal:FREQuency

:SOURce[1..n]:VFLocator:POWer:STATe:TIME

Description

This command sets the duration after which the visual fault locator (VFL) will stop emitting light automatically (auto-off feature).
Note that this command does not turn the VFL on.

*RST sets this value to 600 seconds.

Syntax

:SOURce[1..n]:VFLocator:POWer:STATe:TIME<wsp><Duration>|MAXimum|MINimum|DEFault

Parameter(s)

Duration:

The program data syntax for <Duration> is defined as a <numeric_value> element. The <Duration> special forms MINimum, MAXimum and DEFault are accepted on input.

MINimum allows to set the instrument to the smallest supported value.
MAXimum allows to set the instrument to the greatest supported value.
DEFault allows the instrument to select a value for the <Duration> parameter.

Duration after which the laser will stop emitting light automatically, in seconds.

Example(s)

SOUR:VFL:AM:INT:FREQ 1
SOUR:VFL:POW:STAT:TIME 60
SOUR:VFL:AM:STAT ON
SOUR:VFL:POW:STAT ON

See Also

SOURce[1..n]:VFLocator:POWer:STATe:TIME?
SOURce[1..n]:VFLocator:POWer:STATe
SOURce[1..n]:VFLocator:AM:STATe?
SOURce[1..n]:VFLocator:AM:INTernal:FREQuency

:SOURce[1..n]:VFLocator:POWer:STATe: TIME?

Description

This query returns a value indicating the duration after which the visual fault locator (VFL) will stop emitting light automatically (auto-off feature).

*RST sets this value to 600 seconds.

Syntax

:SOURce[1..n]:VFLocator:POWer:STATe:TIME?[<wsp>MINimum|MAXimum|DEFault]

Parameter(s)

Parameter 1:

The program data syntax for the first parameter is defined as a <CHARACTER PROGRAM DATA> element. The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: MINimum|MAXimum|DEFault.

MINimum is used to retrieve the instrument's smallest supported value.
MAXimum is used to retrieve the instrument's greatest supported value.
DEFault is used to retrieve the instrument's default value.

Response Syntax

<Duration>

Response(s)

Duration:

The response data syntax for <Duration> is defined as a <NR3 NUMERIC RESPONSE DATA> element.

Duration after which the laser will stop emitting light automatically, in seconds.

Example(s)

SOUR:VFL:POW:STAT:TIME 60
SOUR:VFL:POW:STAT:TIME? Returns 60

See Also

SOURce[1..n]:VFLocator:POWer:STATe:TIME
SOURce[1..n]:VFLocator:POWer:STATe
SOURce[1..n]:VFLocator:AM:STATe?
SOURce[1..n]:VFLocator:AM:INTernal:FREQuency

:SOURce[1..n]:WAVelength

Description	This command selects the wavelength of the source, in meters. At *RST, the wavelength that will be selected depends on the instrument you have.
Syntax	:SOURce[1..n]:WAVelength<wsp><Wavelength>
Parameter(s)	*Wavelength:* The program data syntax for <Wavelength> is defined as a <DECIMAL NUMERIC PROGRAM DATA> element. Spectrum value in meters or in hertz.
Example(s)	SOUR:WAV 1550 NM SOUR:POW:STAT:TIME 60 SOUR:POW:STAT ON
See Also	SOURce[1..n]:WAVelength? SOURce[1..n]:WAVelength:LIST?

:SOURce[1..n]:WAVelength?

Description

This query returns the output wavelength of the currently selected source, in meters.

At *RST, the wavelength that will be selected depends on the instrument you have.

Syntax

:SOURce[1..n]:WAVelength?

Parameter(s)

None

Response Syntax

<Wavelength>

Response(s)

Wavelength:

The response data syntax for <Wavelength> is defined as a <NR3 NUMERIC RESPONSE DATA> element.

Current wavelength, in meters.

Example(s)

SOUR:WAV 1550 NM
SOUR:WAV? Returns 1550.0E-9

See Also

SOURce[1..n]:WAVelength
SOURce[1..n]:WAVelength:LIST?

:SOURce[1..n]:WAVelength:LIST?

Description

This query returns the list of all available wavelengths.

*RST does not affect this command.

Syntax

:SOURce[1..n]:WAVelength:LIST?

Parameter(s)

None

Response Syntax

<WavelengthList>

Response(s)

WavelengthList:

The response data syntax for <WavelengthList> is defined as a <DEFINITE LENGTH ARBITRARY BLOCK RESPONSE DATA> element.

Returns the list of all available wavelengths, in meters.

Example(s)

SOUR:WAV:LIST? Returns a wavelength list.

See Also

SOURce[1..n]:WAVelength

:TRACe[1..n][:DATA]?

Description

This query returns all points of the trace corresponding to the specified trace index. The trace is the result of a complete acquisition cycle or a loaded file.

*RST clears this setting.

Syntax

:TRACe[1..n][:DATA]?<wsp>TRC1|TRC2|TRC3|TRC4

Parameter(s)

Label:

The program data syntax for the first parameter is defined as a <CHARACTER PROGRAM DATA> element. The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: TRC1|TRC2|TRC3|TRC4.

Trace index of the available wavelengths.

Response Syntax

<Data>

Response(s)

Data:

The response data syntax for <Data> is defined as a <DEFINITE LENGTH ARBITRARY BLOCK RESPONSE DATA> element.

Returns a list of power values representing the trace.
Each power value represents a point in the trace and is always returned in dB as a <NR3 NUMERIC RESPONSE DATA> type.

Example(s)

CONF:ACQ:MODE ACQ
INIT
INIT:STAT? Returns 0 when acquisition is complete.
TRAC? TRC1 Returns a trace

See Also

MMEMory[1..n]:LOAD:TRACe
TRACe[1..n]:POINts?
MMEMory[1..n]:LOAD:TRACe

:TRACe[1..n]:CATalog?

Description

This query returns all the available labels associated to a trace, at a given wavelength.

*RST clears this setting.

Syntax

:TRACe[1..n]:CATalog?

Parameter(s)

None

Response Syntax

<Catalog>

Response(s)

Catalog:

The response data syntax for <Catalog> is defined as a <DEFINITE LENGTH ARBITRARY BLOCK RESPONSE DATA> element.

Returns a list of labels corresponding to the acquired or loaded wavelengths.

Example(s)

MMEM:LOAD:TRAC "Trace1.trc" (Where "Trace1.trc" is an existing file)
TRAC:CAT? Returns "TRC1,TRC2,TRC3,TRC4" if 4 acquisitions at different wavelength values are in the loaded file.

:TRACe[1..n]:POINts?

Description

This query returns the number of points of the trace corresponding to the specified trace index. The trace is the result of a complete acquisition cycle or a loaded file.

*RST clears this setting.

Syntax

:TRACe[1..n]:POINts?<wsp>TRC1|TRC2|TRC3|TRC4

Parameter(s)

Label:

The program data syntax for the first parameter is defined as a <CHARACTER PROGRAM DATA> element. The allowed <CHARACTER PROGRAM DATA> elements for this parameter are: TRC1|TRC2|TRC3|TRC4.

Trace index of the available wavelengths.

Response Syntax

<PointsCount>

Response(s)

PointsCount:

The response data syntax for <PointsCount> is defined as a <NR1 NUMERIC RESPONSE DATA> element.

Returns the number of points.

Example(s)

CONF:ACQ:MODE ACQ
INIT
INIT:STAT? Returns 0 when acquisition is complete.
TRAC:POIN? TRC1 Returns the number of points.

See Also

MMEMory[1..n]:LOAD:TRACe
TRACe[1..n][:DATA]?
MMEMory[1..n]:LOAD:TRACe