User's Manual

AQ7292A, AQ7293A, AQ7294A, AQ7293F, AQ7293H, AQ7294H OTDR Communication Interface

YOKOGAWA
Yokogawa Test & Measurement Corporation

IM AQ7290-17EN
1st Edition

Thank you for purchasing the AQ7290 OTDR series AQ7292A, AQ7293A, AQ7294A, AQ7293F, AQ7293H, AQ7294H OTDR (Optical Time Domain Reflectometer).
This communication interface user's manual explains the following information.

- USB interface (USB communication)
- Ethernet Interface (option)
- USB interface (WLAN/LTE dongle)
- Communication commands

To ensure correct use, please read this manual thoroughly before beginning operation. After reading the manual, keep it in a convenient location for quick reference whenever a question arises during operation. The manuals for this instrument are listed on the iii page. Please read all manuals.

Contact information of Yokogawa offices worldwide is provided on the following sheet.

Document No.	Description
PIM 113-01Z2	List of worldwide contacts

Notes

- The contents of this manual are subject to change without prior notice as a result of improvements to the product's performance and functionality. Refer to our website to view our latest manuals.
- The figures given in this manual may differ from those that actually appear on your screen.
- Every effort has been made in the preparation of this manual to ensure the accuracy of its contents. However, should you have any questions or find any errors, please contact your nearest YOKOGAWA representative.
- Copying or reproduction of all or any part of the contents of this manual without YOKOGAWA's permission is strictly prohibited.
- The TCP/IP software of this product and the documents concerning it have been developed/created by YOKOGAWA based on the BSD Networking Software, Release 1 that has been licensed from the Regents of the University of California.

Trademarks

- Microsoft, Windows, Windows 10, and Windows 11 are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries.
- Adobe and Acrobat are trademarks of Adobe Systems Incorporated.
- LTE is a trademark or registered trademark of European Telecommunications Standards Institute (ETSI).
- In this manual, the ® and TM symbols do not accompany their respective registered trademark or trademark names.
- Other company and product names are registered trademarks or trademarks of their respective holders.

1st Edition: April 2025 (YMI)

USB interface and Ethernet interface

To use the USB communication features, your personal computer (PC) must have the following:

- The communication libraly (TMCTL)
- USB driver

To use the Ethernet communication features, your PC must have the following:

- The communication libraly (TMCTL)

To download the libraries and drivers listed above, go to the following website, and then browse to the download page.

https://tmi.yokogawa.com/

To use a wireless network, you need to connect a USB Type-A communication dongle (WLAN or LTE dongle) to the instrument. Please use your own communication dongle.
For the recommended communication dongles that can be used with this instrument, contact your nearest YOKOGAWA dealer.

Revisions

1st Edition: April 2025

Manuals

The following manuals, including this one, are provided as manuals for this instrument. Please read all manuals.

Manuals included with the product

Manual Title	Manual No.	Description
AQ7292A, AQ7293A, AQ7294A, AQ7293F, AQ7293H, AQ7294H OTDR Getting Started Guide	IM AQ7290-02EN	This guide focuses on the handling precautions, basic operations, and specifications of the instrument.
AQ7292A, AQ7293A, AQ7294A, AQ7293F, AQ7293H, AQ7294H OTDR	IM AQ7290-92Z1	Document for China
Model 739883 Battery Pack Handling Precautions	IM 739883-01EN	Explains the handlling precautions for the battery pack.
739883 Battery Pack	IM 739883-92Z1	Document for China
Safety Instruction Manual	IM 00C01C01-01Z1	Safety manual (European languages)

Manuals included in the instrument's internal memory

The following manuals are included in the internal memory of the instrument.
Download them to your PC for use. For how to download them, see "How to Download the User's Manual" in the Getting Started Guide (IM AQ7290-02EN). You can also download them from the YOKOGAWA website.

Manual Title	Manual No.	Description
AQ7292A, AQ7293A, AQ7294A, AQ7293F, AQ7293H, AQ7294H OTDR User's Manual	IM AQ7290-01EN	Explains all the instrument's features, except for the communication features, and how to use them.
AQ7292A, AQ7293A, AQ7294A, AQ7293F, AQ7293H, AQ7294H OTDR Communication Interface User's Manual	IM AQ7290-17EN	This manual. Explains the features related to using communication commands to control the instrument.

* The "EN" and "Z1" in the manual numbers are the language codes.

How to Use this Manual

Structure of the manual

This User's Manual consists of the following sections:

Chapter 1 USB Interface (USB communication)

Describes the functions and specifications of the USB interface used to control the instrument OTDR from a PC.

Chapter 2 Ethernet Interface (Option)

Describes the functions and specifications of the Ethernet interface.

Chapter 3 USB Interface (using a WLAN/LTE dongle)

Describes the function of connecting a WLAN- or LTE-compatible USB dongle to this instrument and controlling this instrument from a terminal via WLAN or LTE.

Chapter 4 Before Programming

Describes the syntax used to transmit commands.

Chapter 5 Commands

Describes each command that is available.

Chapter 6 Condition Register/Output Queue and Error Queue

Describes the register and queues.

Appendix

Explains error codes, compatibility information of commands for older models (AQ7280 and AQ1210 series), and other information.

Index

Conventions Used in This Manual

Notations used in the procedural explanations

On pages that describe the operating procedures in each chapter, the following notations are used to distinguish the procedure from their explanations.

Procedure This subsection contains the operating procedure used to carry out the function described in the current section. The procedures are written with inexperienced users in mind; experienced users may not need to carry out all the steps.

Explanation This subsection describes the setup parameters and the limitations on the procedures.

Note Calls attention to information that is important for proper operation of the instrument.

Terms used in explanations of procedures

Panel keys and soft keys
Bold characters used in the procedural explanations indicate characters that are marked on the panel keys or the characters of the soft keys displayed on the screen menu.

Units

k: Denotes “1000.” Example: 400 km
K: Denotes “1024.” Example: 459 KB (file data size)

Symbols used in syntax descriptions

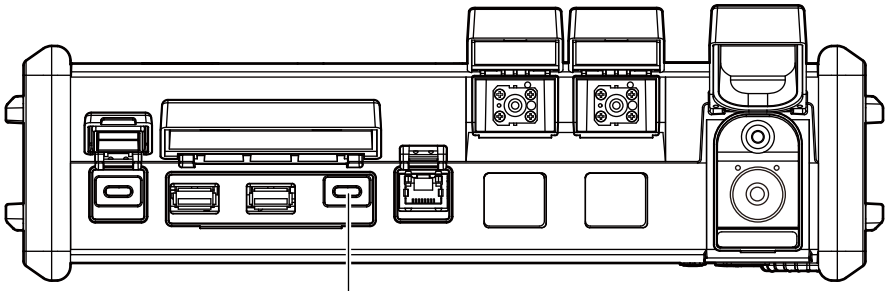
Symbols which are used in the syntax descriptions in chapter 5 are shown below.
These symbols are referred to as BNF notation (Backus-Naur Form).
For detailed information, see section 4.4, “Data.”

Symbol	Description	Example	Example of Input
<>	Defined value	SET:M<x> <x> = 1, 2, 3	-> SET:M2
{}	One of the options in {} is selected.	LMTechnique {LSA|TPA}	-> LMTechnique TPA
|	Exclusive OR		

Contents

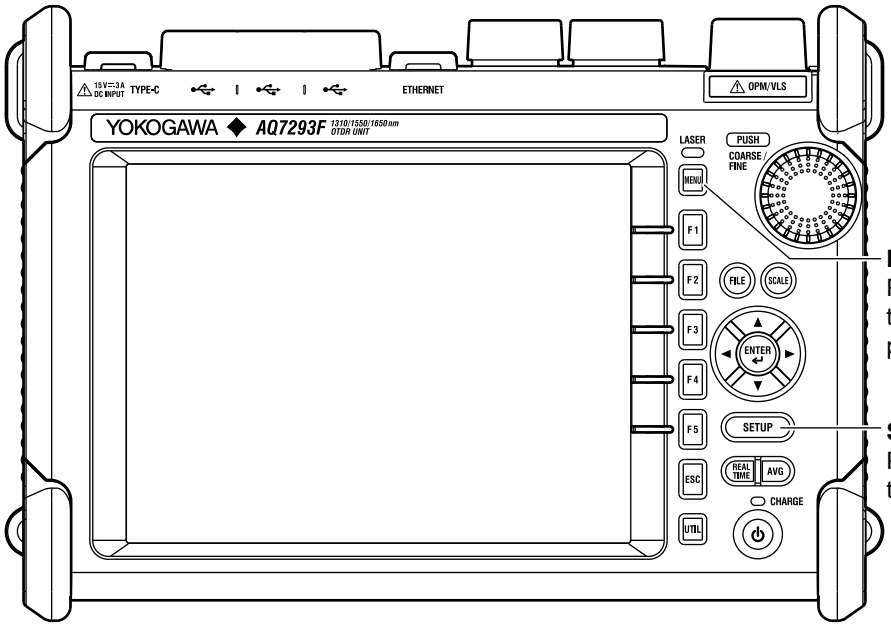
1.1 Names and Functions of Parts

Upper panel



USB type-C connector
Connector used to connect this instrument to the controller (such as a PC) using a USB cable.

Front panel



MENU key
Press this key to display the menu screen, and then press the SETUP key.

SETUP key
Press this key to select the USB interface.

1.2 USB Interface (USB communication) Functions and Specifications

USB interface functions and specifications

Reception function

You can specify the same settings as those specified by front panel key operations.
Receives output requests for measured data, setup data of the panel, and error codes.

Transmission function

Outputs measured and computed data.
Outputs panel setup data and the status byte.
Outputs error codes that have occurred.

USB interface specifications

Electrical and mechanical specifications: Conforms to USB Rev. 2.0
Connector: Type-C connector (receptacle)
Number of ports: 1
Power supply: Self-powered
Compatible PC systems: PC running Windows 10 or Windows 11 with a standard USB port (a separate device driver is needed to connect to a PC).

Switching between remote and local modes

When switching from local to remote mode

Sending a command when the instrument is in the local mode causes the instrument to switch to the remote mode.

- All keys except the **Local** button are disabled.
- Settings entered in local mode are retained even when this instrument switches to remote mode.

When switching from remote to local mode

Tapping the **Local** button in remote mode puts the instrument in local mode.

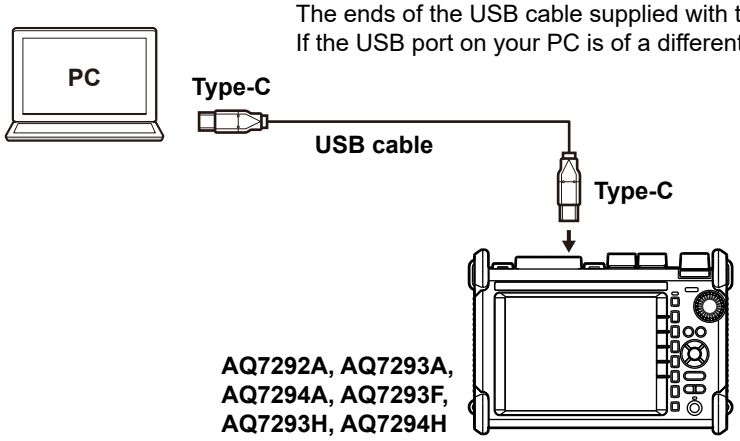
- Key operations are enabled.
- Settings entered in remote mode are retained even when this instrument switches to local mode.

1.3 Connecting via the USB Interface

Connection procedure

1. Open the cover of the upper panel.
2. Connect a USB cable to the type-C connector.

The ends of the USB cable supplied with the product are both Type-C.
If the USB port on your PC is of a different type, prepare your own USB cable.



Precautions to be taken when connecting the cable

- Connect the USB cable by inserting the connector firmly into the USB connector.
- Do not connect or disconnect the USB cable after the power is turned ON until this instrument series is ready for operation (approximately 20 s).

1.4 Setting This Instrument (USB communication)

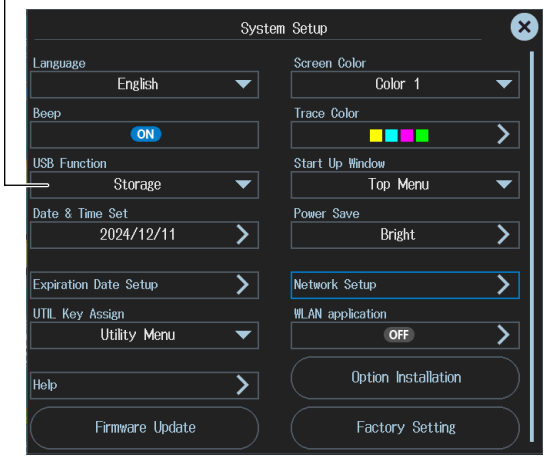
Procedure

Selecting the USB interface function

1. Press **MENU** to display the MENU screen.
2. Press **SETUP** to display the system setup screen.

   You can display the system setup screen by pressing the **SETUP** key on screens other than the MENU screen, such as those of the OTDR feature and power meter feature. Depending on the feature, you may need to tap the **System** soft key after displaying the setup screen with the **SETUP** key. The settings are applied globally regardless of which screen system setup is executed from.

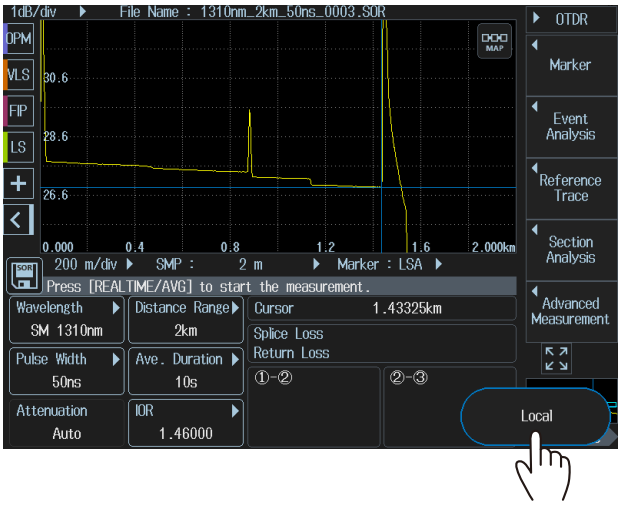
Set the USB function (Control I/O, Storage).



3. Press **USB Function** to display the item screen.
4. Select **Control**.

Releasing the remote control

Press the **Local** soft key that appears on the screen after communication starts.



Explanation

USB interface

To control this instrument remotely using communication commands through the USB port, install YOKOGAWA's USB TMC (Text and Measurement Class) driver into your PC. To obtain YOKOGAWA's USB TMC driver, contact your nearest YOKOGAWA dealer or access the following USB driver page at our Web site and download it.

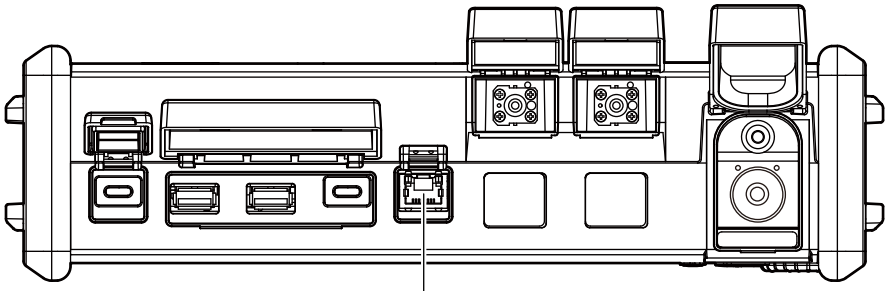
https://tmi.yokogawa.com/library/

Note

- You cannot change the display while this instrument is being remotely controlled.
- Only use the USB TMC driver (or software) provided by YOKOGAWA.

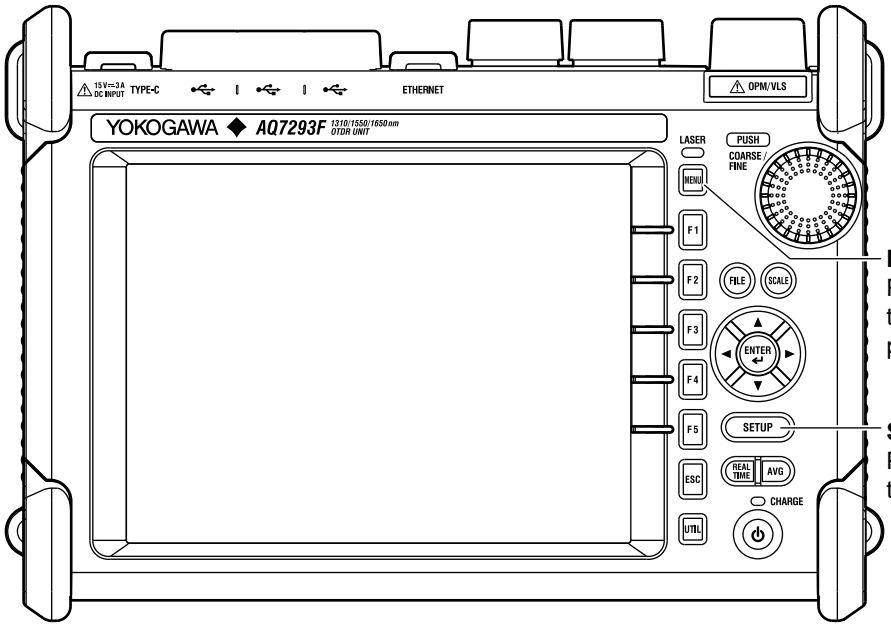
2.1 Names and Functions of Parts

Upper panel



Ethernet Port
Connector used to connect the instrument to the controller (such as a PC) using a Ethernet cable.

Front panel



MENU key
Press this key to display the menu screen, and then press the SETUP key.

SETUP key
Press this key to select the USB interface.

2.2 Ethernet Interface Functions and Specifications

Ethernet interface features

Reception feature

The instrument reception feature allows you to specify the same settings through an Ethernet connection that you can specify using the front panel keys.
The instrument can receive output requests for measured data, panel setting data, and error codes.

Transmission feature

Outputs measured and computed data.
Outputs panel setup data and the status byte.
Outputs error codes that have occurred.

Ethernet interface specifications

Electrical and mechanical specifications: Conforms to IEEE802.3
Transmission system: Ethernet (1000BASE-T)
Data rate: 250 Mbps
Number of communication ports: 1
Port number: 10001/tcp
Connector type: RJ45 connector

Switching between remote and local modes

When switching from local to remote mode

Sending a command when the instrument is in the local mode causes the instrument to switch to the remote mode.

- All keys except the **Local** button are disabled.
- Settings entered in local mode are retained even when the instrument switches to remote mode.

When switching from remote to local mode

Pressing the **Local** button in remote mode puts the instrument in local mode.

- Key operations are enabled.
- Settings entered in remote mode are retained even when the instrument to local mode.

Note

The instrument cannot be remotely controlled via the USB interface while the storage function is in operation. Remote control via the Ethernet interface is also not possible.

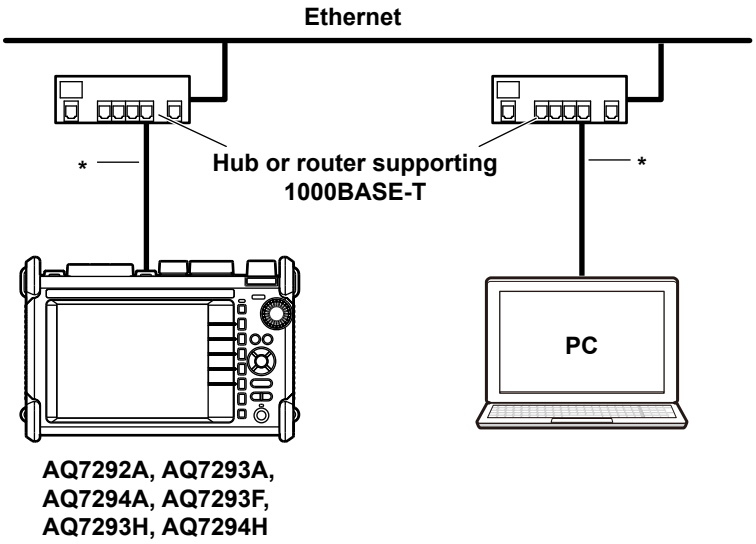
FTP function

The instrument series has an FTP function. You can transfer the data stored in the insturment's internal memory to the PC using FTP commands from the PC.

2.3 Connecting the Ethernet Interface

Connection procedure

Connect a UTP (Unshielded Twisted-Pair) cable or an STP (Shielded Twisted-Pair) cable that is connected to a hub, for example, to the 1000BASE-T port on the bottom of the instrument.



* UTP cable or STP cable (straight cable in either case)

Precautions to be taken when connecting the cable

- To connect the instrument to a PC through a hub, use a straight cable. To connect the instrument to a PC directly, use a crossover cable.
- Use a network cable that conforms to your network environment (such as the data rate).
- When using a UTP cable (straight cable), use a cable of category 5.

2.4 Setting This Instrument (Network)

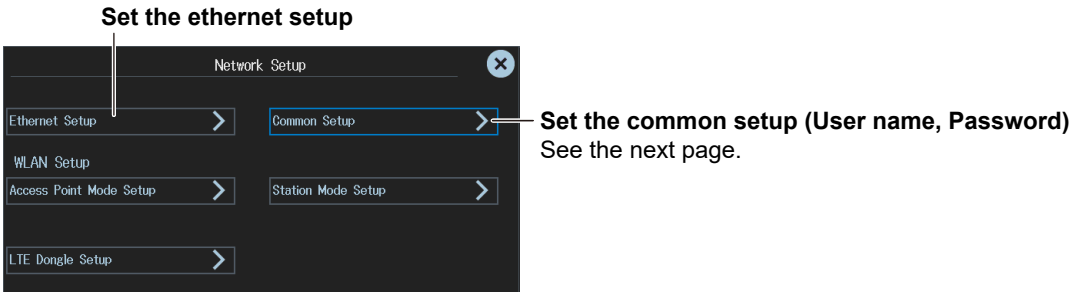
The settings for remotely controlling the instrument via the Ethernet interface are explained below.

Procedure

Selecting the ethernet interface function

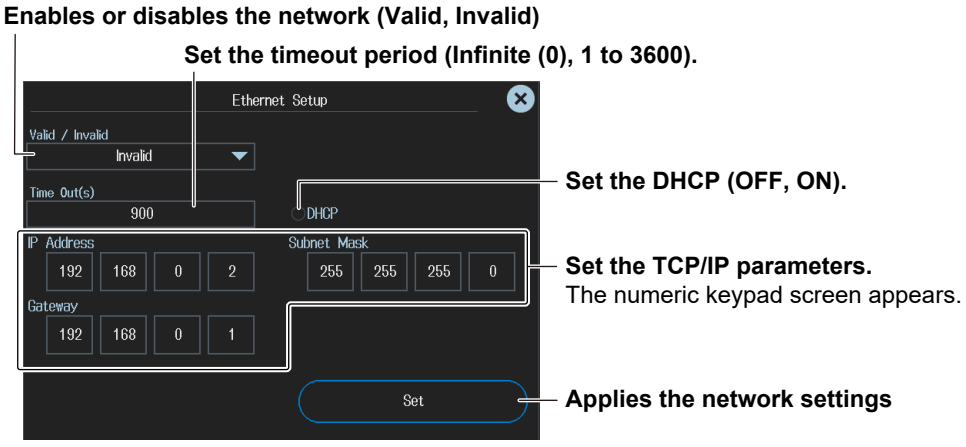
1. Press **MENU** to display the MENU screen.
2. Press **SETUP** to display the system setup screen.

   You can display the system setup screen by pressing the **SETUP** key on screens other than the MENU screen, such as those of the OTDR feature and power meter feature. Depending on the feature, you may need to tap the **System** soft key after displaying the setup screen with the **SETUP** key. The settings are applied globally regardless of which screen system setup is executed from.
3. Press **Ethernet Setup** to display the item screen.



Releasing the ethernet setup

4. Tap **Ethernet Setup** to display the Ethernet Setup screen.



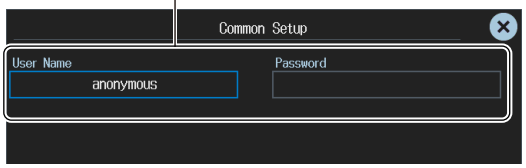
Note

For details, see Explanation in section 10.7 of the User's Manual, IM AQ7290-01EN.

Set the Common Setup

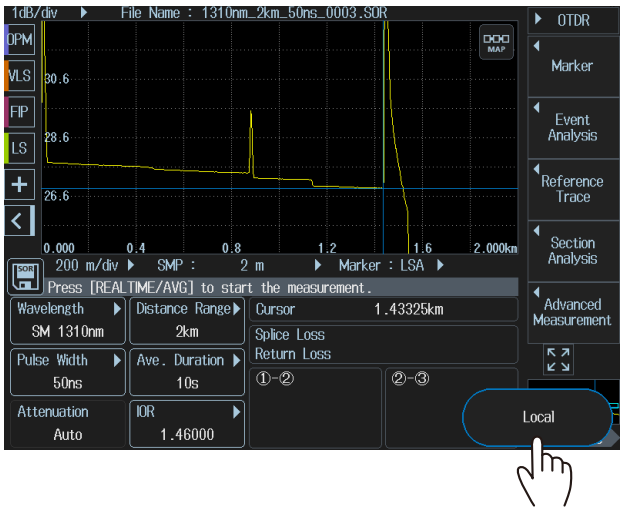
4. Tap **Common Setup** to display the Common Setup screen.

Set the user name and password.
Enter the user name and password by following the procedure provided in “Entering Character Strings” in the user’s manual, IM AQ7290-01EN.



Releasing the Remote Control

Press the **Local** button that appears on the screen after communication starts.



Explanation

Setting the ethernet interface

You must set the following parameters to use the Ethernet interface function.

- **Enabling or Disabling the Network Setup**
  After you have set the user name, password, timeout value, and TCP/IP parameters, select Valid to use the network connection.
- **User Name**
- **Password**
- **Setting the Timeout Value**
  The connection to the network is automatically disconnected if there is no access to the instrument for the specified time.
- **Setting the TCP/IP**
  IP Address
  Subnet Mask
  Gateway

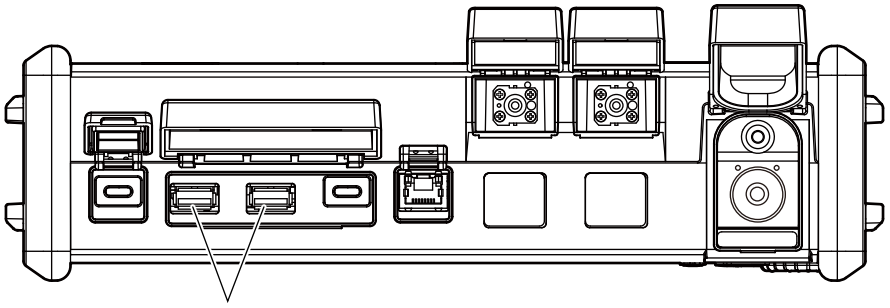
For details on how to configure the settings, see section 10.4, “Configuring Network Settings (/LAN option),” in the User’s Manual, IM AQ7290-01EN.

Note

- If a router or the like is between the instrment and the controlling PC, set a timeout period. If a timeout period is not set and the router or the like malfunctions, the instrment will no longer be able to connect to the PC.
- The instrment cannot be remotely controlled via the ethernet interface while the USB storage function is in operation.

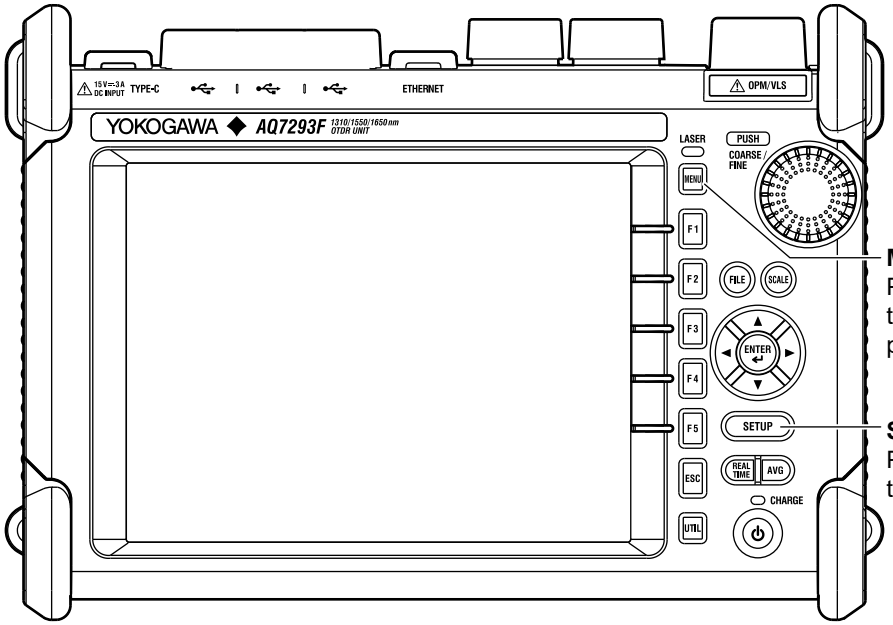
3.1 Names and Functions of Parts

Upper panel



USB Type-A connector
Connector used to connect a WLAN or LTE-enabled USB dongle.

Front panel



MENU key
Press this key to display the menu screen, and then press the SETUP key.

SETUP key
Press this key to select the USB interface.

3.2 USB Interface (using a WLAN/LTE dongle) Features and Specifications

USB interface (using a WLAN/LTE dongle) features

Reception function

You can specify the same settings as those specified by front panel key operations.
Receives output requests for measured data, setup data of the panel, and error codes.

Transmission function

Outputs measured and computed data.
Outputs panel setup data and the status byte.
Outputs error codes that have occurred.

USB interface (using a WLAN/LTE dongle) specifications

Electrical and mechanical specifications: Conforms to USB Rev. 2.0
Connector: Type-A connector (receptacle)
Number of ports: 2
Power supply: Self-powered

Switching between remote and local modes

When switching from local to remote mode

Sending a command when the instrument is in the local mode causes the instrument to switch to the remote mode.

- All keys except the **Local** button are disabled.
- Settings entered in local mode are retained even when this instrument switches to remote mode.

When switching from remote to local mode

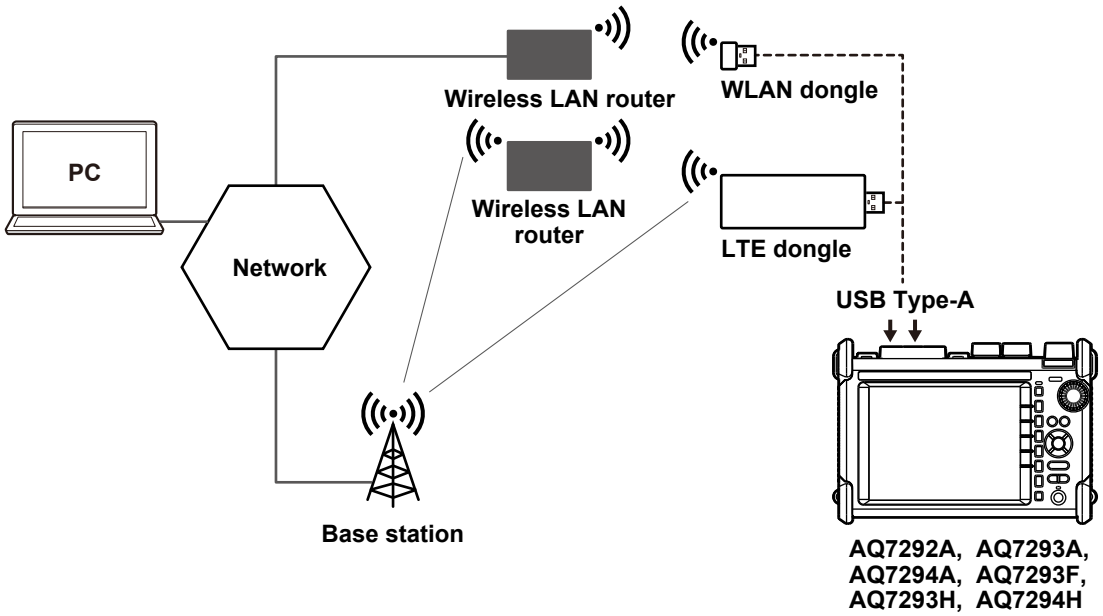
Tapping the **Local** button in remote mode puts the instrument in local mode.

- Key operations are enabled.
- Settings entered in remote mode are retained even when this instrument switches to local mode.

3.3 Connecting a WLAN/LTE Dongle

Connection procedure

1. Open the cover of the upper panel.
2. Connect a WLAN- or LTE-compatible USB dongle to a Type-A connector.



Precautions to be taken when connecting the cable

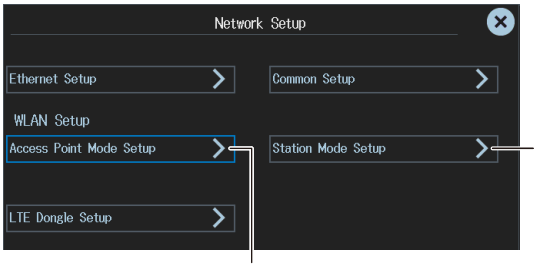
- Be sure to insert the WLAN/LTE dongle firmly into the USB port.
- Do not connect or remove the WLAN/LTE dongle from when the instrument is turned on until operation becomes available (approximately 20 seconds).

3.4 Configuring the Instrument's Settings (using a WLAN/LTE dongle)

Procedure

Network setup screen (using a WLAN dongle)

1. Press **MENU** to display the MENU screen.
2. Press **SETUP** to display the system setup screen.
3. Tap **Network Setup** to display the Network Setup screen.



Set the station mode setup
Select this option when using this instrument by connecting to an access point.
See next page.

Set the access point mode setup
Select this option when using this instrument as an access point.

Note

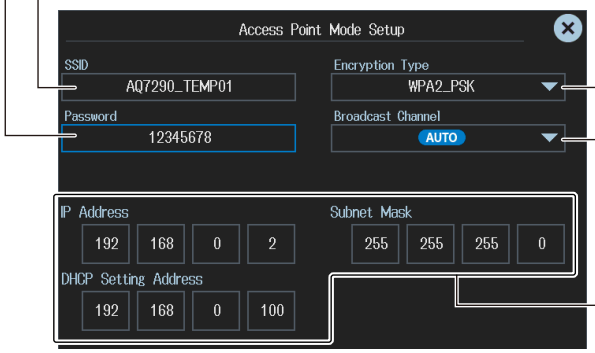
For details, see Explanation in section 10.7 of the User's Manual, IM AQ7290-01EN.

Set the access point mode

4. Tap **Access Point Mode Setup** to display the Access Point Mode Setup screen.

Set the authentication password.
A character input screen appears. For instructions on how to input characters, see the User's Manual, IM AQ7290-01EN.

Set the ID name (SSID).
A character input screen appears. For instructions on how to input characters, see the User's Manual, IM AQ7290-01EN.



Set the encryption key (None, WPA2_PSK).

Set the broadcast (beacon signal) channel to the wireless terminal (AUTO, 1ch, 2ch, 3ch, 4ch, 5ch, 6ch, 7ch, 8ch, 9ch, 10ch, 11ch).

Set the network address.
A numeric keypad appears.

Setting the station mode

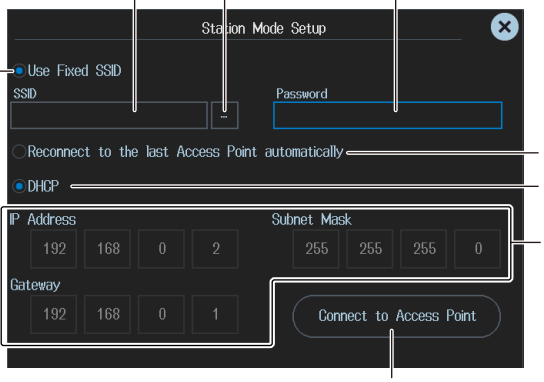
4. Tap **Station Mode Setup** to display the Station Mode Setup screen.

Use a fixed ID name (SSID).

Set the ID name (SSID).
A character input screen appears.
For instructions on how to input characters, see the User's Manual, IM AQ7290-01EN.

Select a fixed SSID.

Set the authentication password.
A character input screen appears.
For instructions on how to input characters, see the User's Manual, IM AQ7290-01EN.



Enables or disables auto reconnection

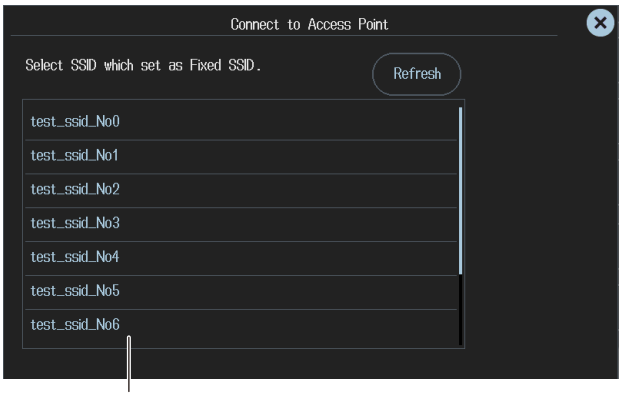
Set the DHCP (OFF, ON).

Set the network address.
A numeric keypad appears.

Connect to Access Point
Connect to an access point.

Selecting an SSID from a list of detected SSIDs

5. Tap **[...]** (select SSID from the detected list). A screen appears for selecting a fixed SSID from the detected SSIDs.

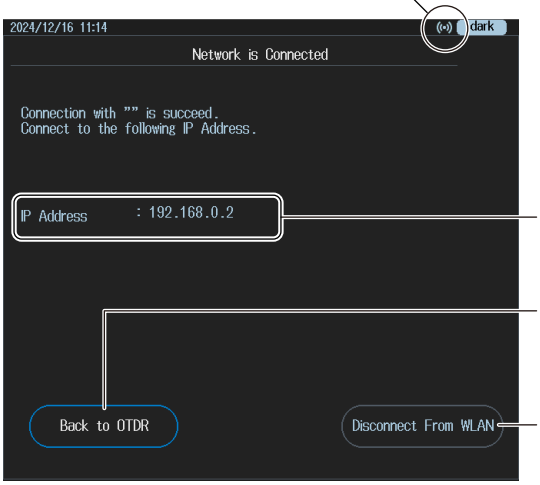


Select a fixed SSID from the list.

Connecting to an Access Point

6. Tap **Connect to Access Point**. An access point connection screen appears.

An icon indicating that the wireless LAN is in use
If you execute the Stop WLAN function, the wireless LAN operation ends, and the icon disappears. If you select Back to OTDR, the wireless operation does not end, and the icon remains displayed. In addition, even if you stop accessing from the wireless terminal, the wireless LAN operation does not end, so you can access from the wireless terminal again.



When a connection is established, the IP address assigned to this instrument is displayed.

Returns to the previous screen
You can monitor the operation through remote control or control the OTDR directly.

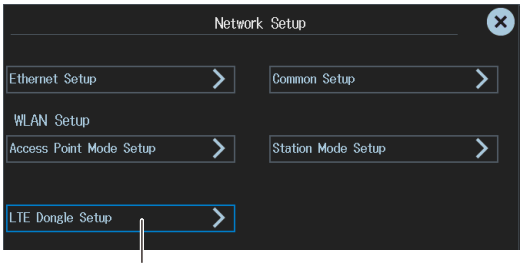
Cancels wireless LAN connection

Network setup screen (using a LTE dongle)

1. Press **MENU** to display the MENU screen.

2. Press **SETUP** to display the system setup screen.

3. Tap **Network Setup** to display the Network Setup screen.



LTE dongle setup
Set up to use an LTE dongle.
See next page.

Note

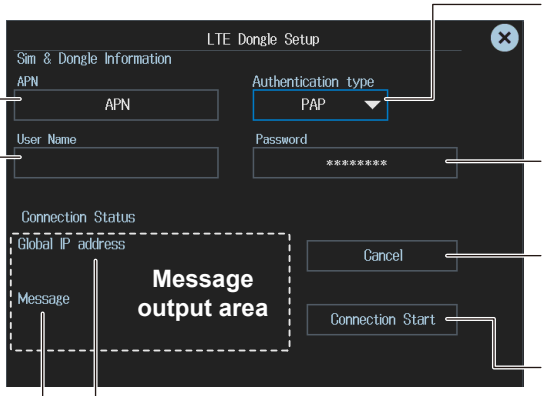
For details, see Explanation in section 10.7 of the User's Manual, IM AQ7290-01EN.

LTE Dongle Setup

4. Tap **LTE Dongle Setup**. A LTE Dongle Setup screen appears.

User setup
Enter the SIM card's unique user name.

APN setup
Enter the SIM card's unique APN information.



Authentication type setup (PAP, CHAP, PAP, CHAP)
Enter the SIM card's unique authentication type.

Password setup
Enter the SIM card's unique password.

Cancel
Disconnects communication with the LTE dongle.

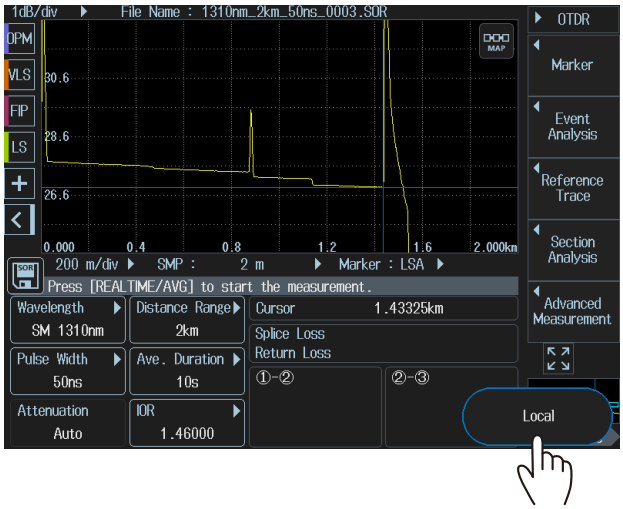
Connection Start
Starts communication with the LTE dongle.

Shows the SIM card's unique global IP once LTE communication is established. Communication with the OTDR begins by connecting to the displayed global IP.

Shows a message about the connection.

Releasing the Remote Control

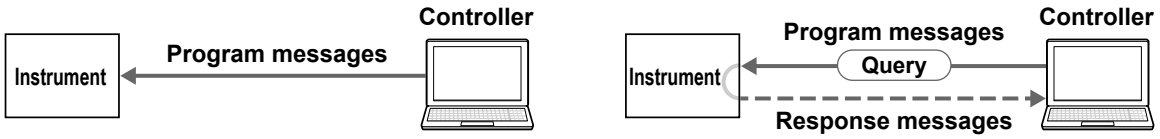
Press the **Local** soft key that appears on the screen after communication starts.



4.1 Messages

Messages

Information is exchanged between a controller and this instrument in units of messages. Messages that are sent from the controller to the instrument are called program messages, and messages that are sent from the instrument back to the controller are called response messages.
If a program message contains a command that requests a response (query), the instrument returns a response message upon receiving the program message. The instrument returns a single response message in response to a single program message.



Program messages

The program message syntax is shown below.

<Program message unit>

A program message consists of one or more program message units. Each unit corresponds to one command. The instrument executes the commands in the order that they are received.
Separate each program message unit with a semicolon.
For details on the program message syntax, see the next section.

Example `:PMETer:MODulation MOD_CW;UNIT DB<PMT>`

Unit Unit

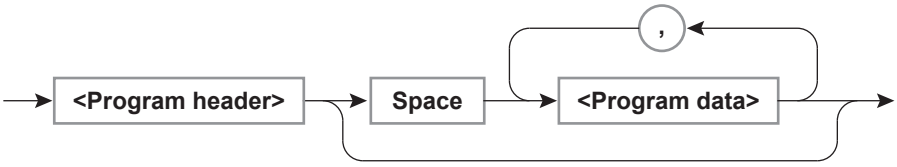
<PMT>

This is a program message terminator. The following three types are available.

NL (new line)	Same as LF (line feed). ASCII code “0AH”
^END	The END message as defined by IEEE 488.1. (EOI signal) The data byte that is sent with the END message is the last data byte of the program message.
NL^END	NL with an END message attached. NL is not included in the program message.

Program message unit syntax

The program message unit syntax is shown below.



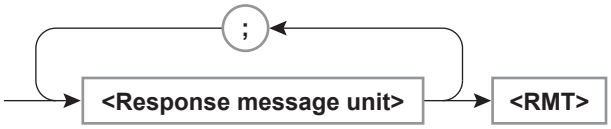
<Program header>	The program header indicates the command type. For details, see page 4-4.
<Program data>	Attach program data if there are conditions that are required to execute a command. Separate the program data from the header with a space (ASCII code 20H). If there are multiple sets of program data, separate each set with a comma. For details, see page 4-8.

Example `:PMETer:MODulation MOD_CW<PMT>`

Header **Data**

Response messages

The response message syntax is as follows:



<Response message unit>

A response message consists of one or more response message units. Each unit corresponds to one response.

Separate each response message unit with a semicolon.

For details on the response message syntax, see the next page.

Example `:PMETer:MODulation MOD_CW;UNIT DB<PMT>`

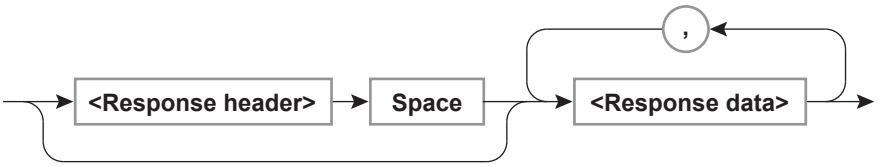
Unit **Unit**

<RMT>

RMT is a response message terminator. It is `NL^END`.

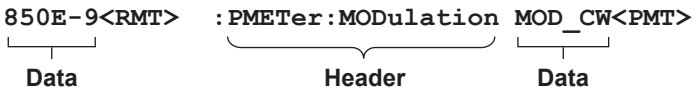
Response message unit syntax

The response message unit syntax is shown below.



<Response header>	A response header sometimes precedes the response data. Separate the data from the header with a space. For details, see page 4-7.
<Response data>	Response data contains the content of the response. If there are multiple sets of response data, separate each set with a comma. For details, see page 4-7.

Example



If there are multiple queries in a program message, responses are returned in the same order that the queries were received in. In most cases, a single query returns a single response message unit, but there are a few queries that return multiple units. The first response message unit always corresponds to the first query, but the nth response unit may not necessarily correspond to the nth query. Therefore, if you want to make sure that every response is retrieved, divide the program messages into individual messages.

Notes on sending and receiving messages

- If the controller sends a program message that does not contain a query, the controller can send the next program message at any time.
- If the controller sends a program message that contains a query, the controller must finish receiving the response message before it can send the next program message. If the controller sends the next program message before receiving the response message in its entirety, an error will occur. A response message that is not received in its entirety will be discarded.
- If the controller tries to receive a response message when there is none, an error will occur. If the controller tries to receive a response message before the transmission of the program message is complete, an error will occur.
- If the controller sends a program message containing multiple message units, but the message contains incomplete units, this instrument will try to execute the ones that are believed to be complete. However, these attempts may not always be successful. In addition, if such a message contains queries, the instrument may not necessary return responses.

Deadlock

The instrument can store at least 1024 bytes of messages in its transmit and receive buffers (the number of available bytes varies depending on the operating conditions). If both the transmit and receive buffers become full at the same time, the instrument will no longer be able to operate. This condition is called a deadlock. If this happens, you can resume operation by discarding response messages.

Deadlock will not occur if the program message (including the <PMT>) is below 1024 bytes. Program messages that do not contain queries never cause deadlocks.

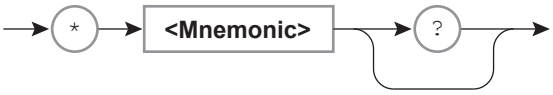
4.2 Commands

Commands

There are two types of commands (program headers) that a controller may send to the instrument. The commands differ in their program header formats.

Common command header

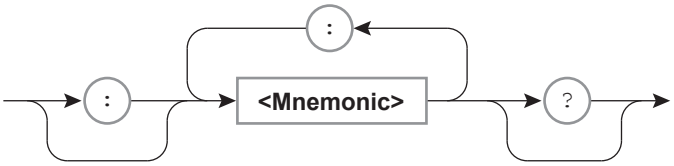
Commands that are defined in IEEE 488.2-1992 are called common commands. The common command header syntax is shown below. Be sure to include an asterisk (*) at the beginning of a common command.



Common command example: **`*CLS`**

Compound header

Other commands that are specific to the instrument are classified and arranged in a hierarchy according to their functions. The compound header syntax is shown below. Be sure to use a colon to specify a lower hierarchical level.



Compound header example: **`:PMETer:LINK:STATe`**

Note

A <mnemonic> is an alphanumeric character string.

Concatenating commands

Command groups

A command group is a group of commands that have common compound headers arranged in a hierarchy. A command group may contain sub-groups.

Example A portion of the commands from the power meter command group

- **:PMETer:DREF**
- **:PMETer:LINK:STATe**
- **:PMETer:MAXMin:STATe**
- **:PMETer:MODulation**
- **:PMETer:OFFSet**
- **:PMETer:REFerence**
- **:PMETer:WAVelength:DETail**

Concatenating commands of the same group

The instrument stores the hierarchical level of the command that is currently being executed and processes the next command on the assumption that it belongs to the same level. Therefore, the common header section can be omitted for commands that belong to the same group.

Example **:PMETer:MODulation MOD_CW;UNIT DB<PMT>**

Concatenating commands of different groups

If the subsequent command does not belong to the same group, place a colon in front of the header (this colon cannot be omitted).

Example **:PMETer:MODulation MOD_CW;:MENU:FUNCtion TOP<PMT>**

Concatenating common commands

Common commands that are defined in IEEE 488.2-1992 are independent of hierarchy. A colon is not needed before a common command.

Example **:PMETer:MODulation MOD_CW;*CLS;UNIT DB<PMT>**

Separating commands with <PMT>

If you separate two commands with a terminator, two program messages will be sent. Therefore, the common header must be specified for each command even when commands belonging to the same command group are being concatenated.

Example **:PMETer:MODulation MOD_CW<PMT>:PMETer:UNIT DB<PMT>**

Upper-level query

An upper-level query is a query that is made by appending a question mark to a command higher in the group. The controller can receive all of the settings in a group collectively by executing a highest-level query. Some upper-level queries of a group, which may be comprised of more than three hierarchical levels, can cause the instrument to transmit all the lower level settings.

Example `:NETWork:CONTrol?<PMT>`
`-> :NETW:CONT:PASS "ABC";TIM 30;USER "anonyumous"<RMT>`

The response to an upper-level query can be sent back to the instrument as a program message. This enables the settings that were present when the upper-level query was made to be reproduced later on. However, some upper-level queries do not return setup parameters that are not currently in use. Exercise caution because not all of a group's information is necessarily returned in a response.

Header interpretation rules

The instrument interprets the header that it receives according to the rules below.

- Mnemonics are not case sensitive.

  Example **`MINute`** can be written as **`minute`** or **`Minute`**.

- The lower-case characters can be omitted.

  Example **`MINute`** can be written as **`MINU`** or **`MIN`**.

- The question mark at the end of a header indicates that it is a query. You cannot omit the question mark.

  Example The shortest abbreviation for **`MINute?`** is **`MIN?`**.

- If the <x> (value) at the end of a mnemonic is omitted, it is interpreted as a 1.

  Example If you write **`M`** for **`M<x>`**, **`M1`** is specified.

4.3 Responses

Responses

When the controller sends a query with a question mark, the instrument returns a response message to the query.

Form

Response consisting of a header and data

Responses that can be used as program messages without any changes are returned with command headers attached.

Example **:PMETer:MODulation?<PMT> -> :PMETer:MODulation MOD_270HZ<RMT>**

If You Want the instrument to Return Responses without Headers

You can configure the instrument so that even responses that have both headers and data are returned without headers. Use the `:COMMunicate:HEADer` command for this purpose.

Abbreviated form

The instrument normally returns response headers with the lower-case section removed. You can configure the instrument so that full headers are returned. Use the `:COMMunicate:VERBose` command for this purpose.

4.4 Data

Data contains conditions and values that are written after the header. A space separates the data from the header. Data is classified as follows:

Group	Meaning	Page
<Decimal>	A value expressed in decimal notation Example: Western calendar year `->:MISC:DATE:YEAR 2025`	4-8
<Distance>, <Time>, <Wavelength>, <Loss>	A physical value Example: Timeout value `->:NETWork:CONTrol:TIMeout 30`	4-9
<Character data>	Predefined character string (mnemonic). Select from the available strings in braces. Example: Select the function mode `->:MENU:FUNCtion {TOP|OTDR|LIGHt|OPC|PMETER|VLS|FIP}`	4-9
<Boolean>	Indicates on and off. Specify ON, OFF, or a value. Example: Turn on the DHCP `->:NETWork:DHCP ON`	4-10
<String data>	User-defined string Example: Set the Network password `->:NETWork:CONTrol:PASSword "ABC"`	4-10
<Block data>	Data that contains 8-bit values Example: Get the file from OTDR `:FSYStem:FILE:GET? -> #220ABCDEFGHIJ1234567890`	4-10

<Decimal>

<Decimal> indicates a value expressed as a decimal number, as shown in the table below. Decimal values are written in the NR form as specified in ANSI X3.42-1975.

Symbol	Meaning	Example		
<NR1>	Integer	`125`	`-1`	`+1000`
<NR2>	Fixed-point number	`125.0`	`-.90`	`+001.`
<NR3>	Floating-point number	`125.0E+0`	`-9E-1`	`+.1E4`
<NRf>	Any form from <NR1> to <NR3>			

- The instrument can receive decimal values that are sent from the controller in any of the froms <NR1> to <NR3>. This is expressed as <NRf>.
- The instrument returns a response to the controller in one of the forms from <NR1> to <NR3> depending on the query. The same form is used regardless of the size of the value.
- For the <NR3> form, the plus sign after the “E” can be omitted. You cannot omit the minus sign.
- If a value outside the range is entered, the value is adjusted to the closest value within the range.
- If a value has more significant digits than are available, the value will be rounded.

<Distance>, <Time>, <Wavelength>, <Loss>

<Distance>, <Time>, <Wavelength>, and <Loss> indicate decimal values that have physical significance. A <Multiplier> or <Unit> can be attached to the <NRf> form that was described earlier. Use one of the following syntaxes.

Syntax	Example
`<NRf><Multiplier><Unit>`	`0.58UM`
`<NRf><Unit>`	`500m`
`<NRf><Multiplier>`	`5M`
`<NRf>`	`5E-3`

Multipliers and Units that you can use are indicated in the following table.

<Multiplier>

Symbol	Word	Multiplier
`EX`	Exa	10^{18}
`PE`	Peta	10^{15}
`T`	Tera	10^{12}
`G`	Giga	10^{9}
`MA`	Mega	10^{6}
`K`	Kilo	10^{3}
`M`	Milli	10^{-3}
`U`	Micro	10^{-6}
`N`	Nano	10^{-9}
`P`	Pico	10^{-12}
`F`	Femto	10^{-15}
`A`	Atto	10^{-18}

<Unit>

Symbol	Word	Meaning
`M`	Meter	Distance
`S`	Second	Time
`dB`	Decibel	Level
`UM`	Micro meter	Wavelength

- <Multiplier> and <Unit> are not case sensitive.
- “U” is used to indicate micro (“μ”).
- “MA” is used for Mega to distinguish it from Milli.
- If both <Multiplier> and <Unit> are omitted, the default unit is used.

<Character data>

<Character data> is a predefined character string (mnemonics). It is mainly used to indicate options and is chosen from the character strings given in { }. The data interpretation rules are the same as those described in Header interpretation rules on page 4-6.

Syntax	Example
`{AUTO|NORMal}`	`AUTO`

- As with the header, the `:COMMunicate:VERBose` command can be used to select whether response messages are returned in the full form or in the abbreviated form.
- The `:COMMunicate:HEADer` setting does not affect <Character data>.

<Boolean>

<Boolean> is data that indicates ON or OFF. The following types of expressions are possible.

Syntax	Example
{ON|OFF|<NRf>}	ON OFF 1 0

- When expressed the <NRf> form, OFF is selected if the rounded integer value is 0, and ON is selected for all other cases.
- A response message is always returned with a 1 if the value is ON and with a 0 if the value is OFF.

<String data>

<String data> is not a predefined character string like <Character data>. It can be any character string. The character string must be enclosed in single quotation marks (') or double quotation marks (").

Syntax	Example
<String data>	'ABC' "IEEE488.2-1992"

- If a character string contains a double quotation mark ("), the double quotation mark is expressed as two consecutive quotation marks (""). This rule also applies to single quotation marks.
- A response message is always enclosed in double quotation marks (").
- <String data> is any character string. Therefore, the instrument assumes that the remaining program message units are part of the character string if no closing single (') or double quotation mark (") is encountered. No error is detected if a quotation mark is omitted.

<Block data>

<Block data> is any 8-bit data. The syntax is as follows:

Syntax	Example
#N<N-digit decimal number><Data byte sequence>	#800000010ABCDEFGHIJ

- **#N**
  Indicates that the data is <Block data>. N indicates the number of succeeding data bytes (digits) in ASCII code.
- **<N-digit decimal number>**
  Indicates the number of bytes of data (example: 00000010 = 10 bytes).
- **<Data byte sequence>**
  Expresses the actual data (example: ABCDEFGHIJ).
- Data is comprised of 8-bit values (0 to 255). This means that the ASCII code “0AH,” which stands for “NL,” can also be included in the data. Hence, care must be taken when programming the controller.

5.1 List of Commands

Command	Function	Page
ACQuire Group		
:ACQuire:ADSave	Sets or queries auto data saving to files after executing averaged measurement.	5-13
:ACQuire:AESearch	Sets or queries auto event detection after executing averaged measurement.	5-13
:ACQuire:ATTenuation	Sets or queries the attenuation value.	5-13
:ACQuire:AUTO:ATTenuation?	Queries the attenuation value for auto mode.	5-13
:ACQuire:AUTO:DRANge?	Queries the distance range value for auto mode.	5-13
:ACQuire:AUTO:PWIDth?	Queries the pulse width for auto mode.	5-13
:ACQuire:AVERage:COUNt	Queries the current average count.	5-13
:ACQuire:AVERage:INDex	Sets or queries the average times.	5-13
:ACQuire:AVERage:MODE	Sets or queries the average method.	5-13
:ACQuire:AVERage:STARt	Executes averaged measurement.	5-14
:ACQuire:AVERage:STOP	Stops averaged measurement.	5-14
:ACQuire:AVERage:TIME	Sets or queries the averaged time.	5-14
:ACQuire:AVERage:TYPE	Sets or queries the average unit.	5-14
:ACQuire:DRANge	Sets or queries the distance range.	5-14
:ACQuire:OFFSet	Sets or queries the measurement start position on the horizontal axis.	5-14
:ACQuire:PLUGcheck	Sets or queries the optical plug connection check.	5-14
:ACQuire:PWIDth	Sets or queries the pulse width.	5-15
:ACQuire:REALtime:STARt	Executes real-time measurement.	5-15
:ACQuire:REALtime:STOP	Stops real-time measurement.	5-15
:ACQuire:REALtime:MODE	Sets or queries the average mode for real-time measurement.	5-15
:ACQuire:RESult:INDex?	Queries the actual averaging count after the measurement.	5-15
:ACQuire:RESult:TIME?	Queries the actual average duration after the measurement.	5-15
:ACQuire:SELection:ATTenuation?	Queries the selectable attenuation values.	5-15
:ACQuire:SELection:DRANge?	Queries the selectable distance range values.	5-16
:ACQuire:SELection:PWIDth?	Queries a list of selectable pulse widths.	5-16
:ACQuire:SELection:SMPinterval?	Queries the selectable sampling intervals.	5-16
:ACQuire:SELection:WAVelength?	Queries the selectable measurement wavelengths.	5-16
:ACQuire:SMPinterval:DATA	Sets or queries the sampling interval.	5-16
:ACQuire:SMPinterval:VALue?	Queries the sampling interval.	5-16
:ACQuire:WAVelength	Sets or queries the measurement wavelength.	5-16
ANALysis Group		
:ANALysis:ASEarch:EXECute	Executes auto event detection.	5-17
:ANALysis:ASEarch:NUMBer?	Queries the number of automatically detected events.	5-17
:ANALysis:BCOefficient	Sets or queries a backscattering light level of the current wavelength.	5-17
:ANALysis:CUMLoss	Sets or queries the cumulative loss type.	5-17

Command	Function	Page
:ANALysis:CURSor:DECibel?	Queries the cursor dB (decibels).	5-17
:ANALysis:CURSor:DELete	Deletes the cursor.	5-17
:ANALysis:CURSor:DISTance	Sets or queries the cursor position.	5-17
:ANALysis:CURSor:INDex	Sets or queries the cursor position using a data number.	5-17
:ANALysis:CURSor:LINK	Sets or queries the cursor link.	5-17
:ANALysis:DUMMy:DISTance:STARt	Sets or queries the launch fiber start position.	5-17
:ANALysis:DUMMy:DISTance:END	Sets or queries the launch fiber end position.	5-17
:ANALysis:DUMMy:END	Sets the end event of the Launch fiber event number.	5-18
:ANALysis:DUMMy:MODE	Sets or queries the launch fiber setup mode.	5-18
:ANALysis:DUMMy:STARt	Sets the start event of the Launch fiber event number.	5-18
:ANALysis:DUNit	Sets or queries the distance unit.	5-18
:ANALysis:EMARker:LMTechnique	Sets or queries the approximation method (event).	5-18
:ANALysis:EMARker:SET:M1	Sets or queries marker M1 of the current event.	5-18
:ANALysis:EMARker:SET:M2	Sets or queries marker M2 of the current event.	5-18
:ANALysis:EMARker:SET:M3	Sets or queries marker M3 of the current event.	5-18
:ANALysis:EMARker:SET:Y2	Sets or queries marker Y2 of the current event.	5-18
:ANALysis:ENABle:FEB2sploss	Sets or queries the pass/fail judgment on/off state of the two-split splitter loss threshold.	5-18
:ANALysis:ENABle:FEB4sploss	Sets or queries the pass/fail judgment on/off state of the four-split splitter loss threshold.	5-19
:ANALysis:ENABle:FEB8sploss	Sets or queries the pass/fail judgment on/off state of the eight-split splitter loss threshold.	5-19
:ANALysis:ENABle:FEB16sploss	Sets or queries the pass/fail judgment on/off state of the 16-split splitter loss threshold.	5-19
:ANALysis:ENABle:FEB32sploss	Sets or queries the pass/fail judgment on/off state of the 32-split splitter loss threshold.	5-19
:ANALysis:ENABle:FEB64sploss	Sets or queries the pass/fail judgment on/off state of the 64-split splitter loss threshold.	5-19
:ANALysis:ENABle:FECLoss	Sets or queries the pass/fail judgment on/off state of the connector loss threshold.	5-19
:ANALysis:ENABle:FERLoss	Sets or queries the pass/fail judgment on/off state of the return loss threshold.	5-19
:ANALysis:ENABle:FESLoss	Sets or queries the pass/fail judgment on/off state of the splice loss threshold.	5-19
:ANALysis:ENABle:FETLoss	Sets or queries the pass/fail judgment on/off state of the total loss threshold.	5-20
:ANALysis:ENABle:FEULoss	Sets or queries the pass/fail judgment on/off state of the dB/km threshold.	5-20
:ANALysis:ENDLoss	Sets or queries the loss indication of the END point.	5-20
:ANALysis:EVENt:CURRent:BRANch?	Gets the number of splitter splits of the current event.	5-20
:ANALysis:EVENt:CURRent:CUMLoss?	Queries the cumulate loss of the current event.	5-20
:ANALysis:EVENt:CURRent:DISTance?	Queries the distance of the current event.	5-20
:ANALysis:EVENt:CURRent:INDex	Specifies the current event.	5-20
:ANALysis:EVENt:CURRent:IOR?	Queries the section group index of the current event.	5-20
:ANALysis:EVENt:CURRent:LOSS?	Queries the splice loss of the current event.	5-20
:ANALysis:EVENt:CURRent:RETurnloss?	Queries the return loss of the current event.	5-20
:ANALysis:EVENt:CURRent:TYPE?	Queries the event type of the current event.	5-21

Command	Function	Page
:ANALysis:EVENt:CURRent:UNITloss?	Queries the dB/km of the current event.	5-21
:ANALysis:EVENt:DELete	Deletes the current event.	5-21
:ANALysis:EVENt:FIX:MODE	Sets or queries the event fix mode.	5-21
:ANALysis:EVENt:FIX:STATe	Sets or queries the event fix.	5-21
:ANALysis:EVENt:INSert	Inserts an event at the cursor position.	5-21
:ANALysis:EVENt:IOR	Sets or queries the section group index of the current event.	5-21
:ANALysis:FEDetection	Sets or queries the fault event display.	5-21
:ANALysis:FESPlitter:BRANch<x>	Sets or queries the fault event's number of splitter splits.	5-21
:ANALysis:FESPlitter:DETection	Sets or queries the fault event's network configuration on/off state.	5-21
:ANALysis:FESPlitter:STAGe	Sets or queries the fault event's number of splitter stages.	5-21
:ANALysis:FMARker:DELete	Deletes markers.	5-22
:ANALysis:FMARker:LEFT:DISTance?	Queries the distance between markers 1 and 2.	5-22
:ANALysis:FMARker:LEFT:LOSS?	Queries the loss between markers 1 and 2.	5-22
:ANALysis:FMARker:LEFT:UNITloss?	Queries the slope between markers 1 and 2.	5-22
:ANALysis:FMARker:LMTechnique	Sets or queries the marker approximation method.	5-22
:ANALysis:FMARker:LOSS?	Queries the splice loss.	5-22
:ANALysis:FMARker:REFLection:SATurated?	Queries the saturated state of the reflection level.	5-22
:ANALysis:FMARker:REFLection:VALue?	Queries the reflection level.	5-22
:ANALysis:FMARker:RETurnloss:SATurated?	Queries the saturated state of the return loss.	5-22
:ANALysis:FMARker:RETurnloss:VALue?	Queries the return loss.	5-22
:ANALysis:FMARker:RIGHt:DISTance?	Queries the distance between markers 2 and 3.	5-22
:ANALysis:FMARker:RIGHt:LOSS?	Queries the loss between markers 2 and 3.	5-22
:ANALysis:FMARker:RIGHt:UNITloss?	Queries the slope between markers 2 and 3.	5-22
:ANALysis:FMARker:SET:M<x>	Sets or queries markers.	5-23
:ANALysis:FMARker:SET:Y<x>	Sets or queries auxiliary markers.	5-23
:ANALysis:IOR	Sets or queries the index of refraction of the current wavelength.	5-23
:ANALysis:MACRobending:DISPlay	Sets or queries the macro bending display on/off state.	5-23
:ANALysis:MACRobending:THReshold	Sets or queries macro bending threshold value.	5-23
:ANALysis:REFerence:DELete	Delete the distance reference.	5-23
:ANALysis:REFerence:DISTance	Sets the distance reference or queries its current position.	5-23
:ANALysis:SECTion:BASelevel?	Queries the dB value of the interval data reference point.	5-23
:ANALysis:SECTion:DELete	Deletes the interval analysis data.	5-23
:ANALysis:SECTion:DISTance?	Queries the distance of the interval data.	5-23
:ANALysis:SECTion:END	Sets or queries the end position of the interval data.	5-23
:ANALysis:SECTion:LMTechnique	Sets or queries the interval analysis approximation method.	5-23
:ANALysis:SECTion:LOSS?	Queries the loss of the interval data.	5-24
:ANALysis:SECTion:REFerence	Sets the interval data reference point.	5-24
:ANALysis:SECTion:RETurnloss:SATurated?	Queries the optical return loss saturation of the interval data.	5-24

Command	Function	Page
:ANALysis:SECTion:RETurnloss:VALue?	Queries the optical return loss of the interval data.	5-24
:ANALysis:SECTion:STARt	Sets or queries the start position of the interval data.	5-24
:ANALysis:SNAPshot:FIX	Sets or queries the snapshot.	5-24
:ANALysis:THReshold:B2SPloss	Sets or queries the two-split splitter loss threshold.	5-24
:ANALysis:THReshold:B4SPloss	Sets or queries the four-split splitter loss threshold.	5-24
:ANALysis:THReshold:B8SPloss	Sets or queries the eight-split splitter loss threshold.	5-24
:ANALysis:THReshold:B16Sploss	Sets or queries the 16-split splitter loss threshold.	5-24
:ANALysis:THReshold:B32Sploss	Sets or queries the 32-split splitter loss threshold.	5-25
:ANALysis:THReshold:B64Sploss	Sets or queries the 64-split splitter loss threshold.	5-25
:ANALysis:THReshold:EOFiber	Sets or queries the end-of-fiber threshold.	5-25
:ANALysis:THReshold:FEB2sploss	Sets or queries the fault event's 2-split splitter loss threshold.	5-25
:ANALysis:THReshold:FEB4sploss	Sets or queries the fault event's 4-split splitter loss threshold.	5-25
:ANALysis:THReshold:FEB8sploss	Sets or queries the fault event's 8-split splitter loss threshold.	5-25
:ANALysis:THReshold:FEB16sploss	Sets or queries the fault event's 16-split splitter loss threshold.	5-25
:ANALysis:THReshold:FEB32sploss	Sets or queries the fault event's 32-split splitter loss threshold.	5-26
:ANALysis:THReshold:FEB64sploss	Sets or queries the fault event's 64-split splitter loss threshold.	5-26
:ANALysis:THReshold:FECLoss	Sets or queries the connector loss threshold of the fault event.	5-26
:ANALysis:THReshold:FERLoss	Sets or queries the return loss threshold of the fault event.	5-26
:ANALysis:THReshold:FESLoss	Sets or queries the splice loss threshold of the fault event.	5-26
:ANALysis:THReshold:FESPloss	Sets or queries the splitter loss threshold of the fault event.	5-26
:ANALysis:THReshold:FETLoss	Sets or queries the total loss threshold of the fault event.	5-26
:ANALysis:THReshold:FEULoss	Sets or queries the dB/km threshold of the fault event.	5-26
:ANALysis:THReshold:RLOSs	Sets or queries the return loss threshold.	5-27
:ANALysis:THReshold:SLOSs	Sets or queries the splice loss threshold.	5-27
:ANALysis:THReshold:SPLOss	Sets or queries the splitter loss threshold.	5-27
:ANALysis:TOTal:DISTance?	Queries the total distance of the measurement result.	5-27
:ANALysis:TOTal:LOSS?	Queries the total loss of the measurement result.	5-27
:ANALysis:TOTal:RETurnloss?	Queries the total return loss of the measurement result.	5-27
:ANALysis:TOTal:UNITloss?	Queries the dB/km value for the total distance of the measured result.	5-27
:ANALysis:TOTLoss	Sets or queries the total loss mode.	5-27
:ANALysis:TOTRloss	Sets or queries the total return loss mode.	5-27

COMMunicate Group

:COMMunicate?	Queries all communication settings.	5-28
:COMMunicate:HEADer	Sets or queries whether a header is added to the response to a query (example with header: "ACQuire:DRANge 5000"; example without header: "5000").	5-28
:COMMunicate:VERBose	Sets or queries whether the response to a query is returned fully spelled out (e.g., ACQuire:DRANge 5000) or in its abbreviated form (e.g., ACQ:DRAN 5000).	5-28

Command	Function	Page
DISPlay Group		
:DISPlay:ALINe	Sets or queries the approximated line display.	5-29
:DISPlay:COLor	Sets or queries the screen color.	5-29
:DISPlay:CURSor:DBValue	Sets or queries the cursor dB value.	5-29
:DISPlay:CURSor:SECond	Sets or queries the ghost cursor display.	5-29
:DISPlay:CURSor:TYPE	Sets or queries the cursor display format.	5-29
:DISPlay:DECibel:UPPer	Sets or queries the maximum value of the display's vertical scale.	5-29
:DISPlay:DIGit:DECibel	Sets or queries the dB decimal places.	5-29
:DISPlay:DIGit:DISTance	Sets or queries the distance decimal places.	5-29
:DISPlay:DISTance:LEFT	Sets or queries the minimum value of the display's horizontal scale.	5-29
:DISPlay:DIVide:DECibel	Sets or queries the magnification of the vertical axis.	5-30
:DISPlay:DIVide:DISTance	Sets or queries the magnification of the horizontal axis.	5-30
:DISPlay:INITialize:SCALe	Initializes the display scales.	5-30
:DISPlay:MARKer:INFormation	Sets or queries the marker information display.	5-30
FIPRobe Group		
:FIPRobe:PASSfail:CLADding:DEF1:ENABle	Sets or queries whether defects that are 2 µm or less in length are to be judged in the cladding area of the fiber inspection probe.	5-31
:FIPRobe:PASSfail:CLADding:DEF1:THReshold	Sets or queries the threshold for judging defects that are 2 µm or less in length in the cladding area of the fiber inspection probe.	5-31
:FIPRobe:PASSfail:CLADding:DEF2:ENABle	Sets or queries whether defects that are 2 µm to 5 µm in length are to be judged in the cladding area of the fiber inspection probe.	5-31
:FIPRobe:PASSfail:CLADding:DEF2:THReshold	Sets or queries the threshold for judging defects that are 2 µm to 5 µm or less in length in the cladding area of the fiber inspection probe.	5-31
:FIPRobe:PASSfail:CLADding:DEF3:ENABle	Sets or queries whether defects that are longer than 5 µm in length are to be judged in the cladding area of the fiber inspection probe.	5-31
:FIPRobe:PASSfail:CLADding:DEF3:THReshold	Sets or queries the threshold for judging defects that are longer than 5 µm in length in the cladding area of the fiber inspection probe.	5-31
:FIPRobe:PASSfail:CLADding:SCR1:ENABle	Sets or queries whether scratches that are 3 µm or less in length are to be judged in the cladding area of the fiber inspection probe.	5-32
:FIPRobe:PASSfail:CLADding:SCR1:THReshold	Sets or queries the threshold for judging scratches that are 3 µm or less in length in the cladding area of the fiber inspection probe.	5-32
:FIPRobe:PASSfail:CLADding:SCR2:ENABle	Sets or queries whether scratches that are longer than 3 µm in length are to be judged in the cladding area of the fiber inspection probe.	5-32
:FIPRobe:PASSfail:CLADding:SCR2:THReshold	Sets or queries the threshold for judging scratches that are longer than 3 µm in length in the cladding area of the fiber inspection probe.	5-32

Command	Function	Page
:FIPRobe:PASSfail:CONTact:DEF:ENABle	Sets or queries whether defects that are longer than 10 µm in length are to be judged in the contact area of the fiber inspection probe.	5-32
:FIPRobe:PASSfail:CONTact:DEF:THReshold	Sets or queries the threshold for judging defects that are longer than 10 µm in length in the contact area of the fiber inspection probe.	5-32
:FIPRobe:PASSfail:CONTact:SCR:ENABle	Sets or queries whether scratches are to be judged in the contact area of the fiber inspection probe.	5-33
:FIPRobe:PASSfail:CONTact:SCR:THReshold	Sets or queries the threshold for judging scratches in the contact area of the fiber inspection probe.	5-33
:FIPRobe:PASSfail:CORE:DEF1:ENABle	Sets or queries whether defects that are 3 µm or less in length are to be judged in the core area of the fiber inspection probe.	5-33
:FIPRobe:PASSfail:CORE:DEF1:THReshold	Sets or queries the threshold for judging defects that are 3 µm or less in length in the core area of the fiber inspection probe.	5-33
:FIPRobe:PASSfail:CORE:DEF2:ENABle	Sets or queries whether defects that are longer than 3 µm in length are to be judged in the core area of the fiber inspection probe.	5-33
:FIPRobe:PASSfail:CORE:DEF2:THReshold	Sets or queries the threshold for judging defects that are longer than 3 µm in length in the core area of the fiber inspection probe.	5-33
:FIPRobe:PASSfail:CORE:DEFAny:ENABle	Sets or queries whether defects are to be judged in the core area of the fiber inspection probe.	5-33
:FIPRobe:PASSfail:CORE:DEFAny:THReshold	Sets or queries the threshold for judging defects in the core area of the fiber inspection probe.	5-34
:FIPRobe:PASSfail:CORE:SCR1:ENABle	Sets or queries whether scratches that are 3 µm or less in length are to be judged in the core area of the fiber inspection probe.	5-34
:FIPRobe:PASSfail:CORE:SCR1:THReshold	Sets or queries the threshold for judging scratches that are 3 µm or less in length in the core area of the fiber inspection probe.	5-34
:FIPRobe:PASSfail:CORE:SCR2:ENABle	Sets or queries whether scratches that are longer than 3 µm in length are to be judged in the core area of the fiber inspection probe.	5-34
:FIPRobe:PASSfail:CORE:SCR2:THReshold	Sets or queries the threshold for judging scratches that are longer than 3 µm in length in the core area of the fiber inspection probe.	5-34
:FIPRobe:PASSfail:CORE:SCRAny:ENABle	Sets or queries whether scratches are to be judged in the core area of the fiber inspection probe.	5-34
:FIPRobe:PASSfail:CORE:SCRAny:THReshold	Sets or queries the threshold for judging scratches in the core area of the fiber inspection probe.	5-34
:FIPRobe:PASSfail:EXECute	Executes pass/fail judgment of the fiber inspection probe.	5-34
:FIPRobe:PASSfail:FIBertype	Sets or queries the fiber type of the fiber inspection probe.	5-35
:FIPRobe:PASSfail:STANdard	Sets or queries the standard that is used for fiber inspection probe pass/fail judgment.	5-35
:FIPRobe:STATe	Queries the image type of the fiber inspection probe.	5-35

Command	Function	Page
FSYStem Group		
:FSYStem:DATE?	Gets the date of a file.	5-36
:FSYStem:EXISt?	Queries whether the specified folder or file exists.	5-36
:FSYStem:FILE:COPY	Copies a file.	5-36
:FSYStem:FILE:DELete	Deletes a file.	5-36
:FSYStem:FILE:GET?	Gets a file from the OTDR. (file reception)	5-37
:FSYStem:FILE:MOVE	Moves a file.	5-37
:FSYStem:FILE:RENAme	Renames a file.	5-37
:FSYStem:FILE:SEND	Sends a file to the OTDR.	5-37
:FSYStem:FOLDer:COPY	Copies a folder.	5-38
:FSYStem:FOLDer:DELete	Deletes a folder.	5-38
:FSYStem:FOLDer:MAKE	Makes a folder.	5-38
:FSYStem:FOLDer:MOVE	Moves a folder.	5-38
:FSYStem:FOLDer:RENAme	Renames a folder.	5-39
:FSYStem:FREE?	Gets the free space on the drive.	5-39
:FSYStem:LIST?	Gets a list of the folders and files in the specified path.	5-39
:FSYStem:LOAD	Loads a file stored in the OTDR into the OTDR.	5-39
:FSYStem:PARAm:DESTination:PATH	Specifies the destination folder for file operations with an absolute path.	5-39
:FSYStem:PARAm:NAME	Specify the name of the target folder or file for file operations.	5-40
:FSYStem:PARAm:PATH	Specifies the folder path containing the target folder or file for file operations with an absolute path.	5-40
:FSYStem:SAVE	Saves the current waveform data (SOR, CSV, BMP, PDF, etc.) to the OTDR.	5-40
:FSYStem:SIZE?	Gets the size of a file.	5-40
:FSYStem:WAVE:GET?	Gets the current waveform data (SOR, CSV, BMP, PDF, etc.). (data reception)	5-41
:FSYStem:WAVE:TYPE	Specifies the file type for acquiring the current OTDR waveform with a communication command.	5-41
:FSYStem:WAVE:VALid?	Checks whether the measurement waveform data exists.	5-41
LABel Group		
:LABel:CABLe:CODE	Sets or queries the cable code.	5-42
:LABel:CABLe:ID	Sets or queries the cable ID.	5-42
:LABel:COMPany	Sets or queries the company name.	5-42
:LABel:DFLag:CURRent	Sets or queries the current data flag.	5-42
LABel:DIRection	Sets or queries the measurement direction.	5-42
:LABel:FIBer:ID	Sets or queries the fiber ID.	5-42
:LABel:FIBer:TYPE	Sets or queries the fiber type.	5-42
:LABel:ID:USEFile	Sets or queries whether file IDs are used as fiber IDs or whether fiber IDs are used separately from file IDs.	5-42
:LABel:LABel	Sets or queries the label.	5-43
:LABel:LOCation:ORIGinating	Sets or queries the start position label.	5-43
:LABel:LOCation:TERMinating	Sets or queries the end position label.	5-43
:LABel:OPERator	Sets or queries the label's operator name.	5-43

Command	Function	Page
LIGHtsource Group		
:LIGHtsource:ABORt	Turns the light source setting off. Stops the emission of the measurement light.	5-44
:LIGHtsource:EXECute	Turns the light source setting on. Starts the emission of the measurement light.	5-44
:LIGHtsource:MODulation	Sets or queries the internal modulation frequency of the light source.	5-44
:LIGHtsource:STATe	Turns the light source setting on and off. Stops/starts the emission of the measurement light.	5-44
:LIGHtsource:WAVelength	Sets or queries the wavelength of the light source.	5-44
MENU Group		
:MENU:ERRor:CLEar	Clears the error dialog box.	5-45
:MENU:FUNCtion	Sets or queries the function displayed in the data display screen.	5-45
:MENU:MARKer	Sets or queries the marker mode.	5-45
MISC Group		
:MISC:ALARmsound	Sets or queries the alarm sound.	5-46
:MISC:BACKlightoff	Sets or queries the backlight auto-off function for battery operation.	5-46
:MISC:BRIGhtness:AC	Sets or queries the LCD brightness for USB-AC adapter operation.	5-46
:MISC:BRIGhtness:BATTery	Sets or queries the LCD brightness for battery operation.	5-46
:MISC:DATE:DAY	Sets or queries the date.	5-46
:MISC:DATE:GET?	Retrieves the date and time information.	5-46
:MISC:DATE:HOUR	Sets or queries the hour.	5-46
:MISC:DATE:MINute	Sets or queries the minute.	5-46
:MISC:DATE:MODE	Sets or queries the year/month/day display type.	5-46
:MISC:DATE:MONTh	Sets or queries the month.	5-47
:MISC:DATE:SECond	Sets or queries the second.	5-47
:MISC:DATE:SET	Executes the changing of the date and time.	5-47
:MISC:DATE:YEAR	Sets or queries the year.	5-47
:MISC:LANGuage	Sets or queries the language.	5-47
:MISC:POWersave:BATTery	Sets or queries the power save function for battery operation.	5-48
:MISC:RLOSsmode	Sets or queries the return loss display.	5-48
NETWork Group		
:NETWork:CONTrol:PASSword	Sets or queries the network password.	5-49
:NETWork:CONTrol:TIMeout	Sets or queries the network timeout value.	5-49
:NETWork:CONTrol:USERname	Sets or queries the network user name.	5-49
:NETWork:DHCP	Sets or queries the network DHCP.	5-49
:NETWork:GATeway	Sets or queries the network gateway.	5-49
:NETWork:IPADdress	Sets or queries the network IP address.	5-49
:NETWork:NETMask	Sets or queries the network subnet mask.	5-50

Command	Function	Page
:NETWork:STATe	Sets or queries the network usage.	5-50
:NETWork:WLAN:ACCesspoint:SSID	Sets or queries the ID name (SSID) of the network WLAN application.	5-50
:NETWork:WLAN:ACCesspoint:ENCrypt	Sets or queries the encryption mode of the network WLAN application.	5-50
:NETWork:WLAN:ACCesspoint:PASSword	Sets or queries the password of the network WLAN application.	5-50
:NETWork:WLAN:ACCesspoint:CHannel	Sets or queries the channel of the network WLAN application.	5-50
:NETWork:WLAN:ACCesspoint:IPADdress	Sets or queries the IP address of the network WLAN application.	5-50
:NETWork:WLAN:ACCesspoint:NETMask	Sets or queries the subnet mask of the network WLAN application.	5-50
:NETWork:WLAN:ACCesspoint:DHCPStartaddr	Sets or queries the start address of the IP addresses that are assigned by the DHCP server of the network WLAN application.	5-51
:NETWork:WLAN:ACCesspoint:STATe	Sets or queries the network WLAN application usage.	5-51
:NETWork:WLAN:COMMon:FWVersion?	Queries the WLAN module's firmware version.	5-51
:NETWork:WLAN:COMMon:HWVersion?	Queries the WLAN module's hardware version.	5-51
:NETWork:WLAN:COMMon:MACaddress?	Queries the WLAN module's MAC address.	5-51
:NETWork:WLAN:STATion:AUToconnect	Sets or queries the auto network connection on/off state.	5-51
:NETWork:WLAN:STATion:CONNect	Starts the network WLAN station mode.	5-51
:NETWork:WLAN:STATion:DHCP	Sets or queries the WLAN station mode's DHCP on/off state.	5-51
:NETWork:WLAN:STATion:DISConnect	Stops the WLAN station mode.	5-51
:NETWork:WLAN:STATion:GATeway	Sets or queries the WLAN station mode's gateway.	5-51
:NETWork:WLAN:STATion:IPADdress	Sets or queries the WLAN station mode's IP address.	5-51
:NETWork:WLAN:STATion:NETMask	Sets or queries the WLAN station mode's netmask.	5-52
:NETWork:WLAN:STATion:PASSword	Sets or queries the WLAN station mode's password.	5-52
:NETWork:WLAN:STATion:SSID	Sets or queries the WLAN station mode's SSID.	5-52
:NETWork:WLAN:STATion:STATe?	Sets or queries the WLAN station mode's state.	5-52
:NETWork:WLAN:STATion:USEFixedssid	Sets or queries the WLAN station mode's fixed SSID on/off state.	5-52

OSWitch Group

:OSWitch:ENABle?	Queries whether an optical switch is connected to this instrument.	5-53
:OSWitch:LINK:ENABle	Sets or queries the optical switch linking.	5-53
:OSWitch:LINK:NUMBer<x>	Sets or queries the optical switch link channel.	5-53
:OSWitch:CHANnel:NUMBer?	Queries the number of channels (input and output ports) available on the optical switch connected to this instrument.	5-53
:OSWitch:CHANnel:SELect	Sets or queries the selected channel number of the optical switch connected to this instrument.	5-53

PMETer Group

:PMETer:AVERage:TIMes	Sets or queries the power meter average count.	5-54
:PMETer:DREF	Executes Dref on the power meter.	5-54
:PMETer:LINK:STATe	Sets or queries the light source power meter setting interlock.	5-54
:PMETer:MAXMin?	Queries all settings related to the maximum and minimum values of the power meter.	5-54

Command	Function	Page
:PMETer:MAXMin:MAX?	Sets or queries the maximum value.	5-54
:PMETer:MAXMin:MIN?	Sets or queries the minimum value.	5-54
:PMETer:MAXMin:STATe	Sets or queries the MAX and MIN display on/off state.	5-54
:PMETer:MEASurement:DATA?	Queries the power meter's measured results.	5-54
:PMETer:MODulation	Sets or queries the power meter modulation.	5-54
:PMETer:OFFSet	Sets or queries the power meter offset.	5-54
:PMETer:REFerence	Sets or queries the power meter reference value.	5-54
:PMETer:THReshold:LOWer	Sets or queries the power meter lower threshold value.	5-54
:PMETer:THReshold:UPPer	Sets or queries the power meter upper threshold value.	5-55
:PMETer:UNIT	Sets or queries the power meter display unit.	5-55
:PMETer:WAVelength:DETail	Sets or queries the wavelength when the wavelength mode is set to Detail.	5-55
:PMETer:ZERoset	Executes zero set on the power meter.	5-55

PMONitor Group

:PMONitor:DREF	Executes Dref (display reference) on the power monitor.	5-56
:PMONitor:MAX?	Queries the maximum scale value of the power monitor.	5-56
:PMONitor:MEASurement:DATA?	Queries the measured results of the power monitor.	5-56
:PMONitor:MIN?	Queries the minimum scale value of the power monitor.	5-56
:PMONitor:OFFSet	Sets or queries the power monitor offset.	5-56
:PMONitor:REFerence	Sets or queries the power monitor reference.	5-56
:PMONitor:THReshold:LOWer	Sets or queries the power monitor lower threshold limit.	5-56
:PMONitor:THReshold:Upper	Sets or queries the power monitor upper threshold limit.	5-56
:PMONitor:UNIT	Sets or queries the power unit display unit.	5-56
:PMONitor:WAVelength	Sets or queries the power monitor wavelength.	5-56
:PMONitor:ZERoset	Executes zero-set on the power monitor.	5-56

REPOrt Group

:REPOrt:EXECute	Creates a report.	5-57
:REPOrt:SET:EVENt:CONDition	Sets or queries whether event search conditions will be included in report files.	5-57
:REPOrt:SET:EVENt:LIST	Sets or queries the fiber end face image in report settings.	5-57
:REPOrt:SET:FIBEr	Sets or queries the mode for including fiber surface image files in report files.	5-57
:REPOrt:SET:JUDGe	Sets or queries whether pass/fail judgment conditions will be included in report files.	5-57
:REPOrt:SET:MAP	Sets or queries the MAP display in report settings.	5-57
:REPOrt:SET:MEASure:CONDition	Sets or queries whether measurement conditions will be included in report files.	5-57
:REPOrt:SET:MEASure:INFO	Sets or queries whether measurement information will be included in report files.	5-57
:REPOrt:SET:MEASure:RESUlt	Sets or queries whether measurement results will be included in report files.	5-57
:REPOrt:SET:OVERview	Sets or queries whether overviews will be included in report files.	5-57
:REPOrt:SET:WAVE	Sets or queries whether waveforms will be included in report files.	5-58

Command	Function	Page
SETup Group		
:SETup:ALLinit	Initializes all settings to their factory default values and formats the internal memory.	5-59
:SETup:INITialize	Initializes all settings to their factory default values.	5-59
STATus Group		
:STATus?	Queries all the settings for the communication status feature.	5-60
:STATus:CONDition?	Queries the contents of the condition register.	5-60
:STATus:ERRor?	Queries the error code and message information (top of the error queue).	5-60
:STATus:QENable	Sets or queries whether or not messages other than errors will be stored to the error queue (on/off).	5-60
:STATus:QMESsage	Sets or queries whether or not message information will be attached to the response to the STATus:ERRor? query (on/off).	5-60
SYSTem Group		
:SYSTem:REBoot	Restarts the system.	5-61
:SYSTem:SAVE	Saves system settings.	5-61
:SYSTem:SHUTdown	Shuts down the instrument.	5-61
VLS Group		
:VLS:ABORt	Stops the emission of the visible light source.	5-62
:VLS:EXECute	Starts the emission of the visible light source.	5-62
:VLS:MODulation	Sets or queries the modulation of the visible light source.	5-62
:VLS:STATe	Sets or queries the emission of the visible light source.	5-62
WAVedata Group		
:WAVedata:DISPlay:SEND:ASCii?	Queries the data of the displayed waveform in ASCII format.	5-63
:WAVedata:DISPlay:SEND:BINary?	Queries the displayed waveform data in binary format (unit array).	5-63
:WAVedata:DISPlay:SEND:BINUshort?	Queries the displayed waveform data in binary format (ushort array)	5-63
:WAVedata:LENGth?	Queries the number of waveform data points.	5-63
:WAVedata:OLDType:DISPlay:SEND?	Queries the data of the displayed waveform in Dot4 format.	5-63
:WAVedata:OLDType:SEND?	Queries the waveform data in Dot4 format.	5-63
:WAVedata:SEND:ASCii?	Queries the waveform data in ASCII format.	5-63
:WAVedata:SEND:BINary?	Queries all the waveform data in binary format (unit array).	5-63
:WAVedata:SEND:BINUshort?	Queries the waveform data in binary format (ushort array).	5-64
:WAVedata:SEND:SIZE	Sets or queries the number of acquired data points.	5-64
:WAVedata:SEND:STARt	Sets or queries the starting distance of the waveform data.	5-64

Command	Function	Page
Common Commands		
*CLS (Clear Status)	Clears all event status registers that are displayed in the status byte register summary.	5-65
*ESE (Standard Event Status Enable)	Sets or queries the standard event enable register.	5-65
*ESR? (Standard Event Status Register)	Queries and clears the standard event status register.	5-65
*IDN? (Identification)	Queries the instrument type and firmware version.	5-65
*OPT? (Option)	Queries the option information.	5-65
*RST (Reset)	Returns all the settings except the communication settings to their factory default values.	5-65
*SRE (Service Request Enable)	Sets or queries the service request enable register.	5-66
*STB? (Read Status Byte)	Queries the current status byte register value.	5-66
*TST? (Self Test)	Executes a self-test.	5-66

5.2 ACQuire Group

The commands in this group deal with waveform acquisition. You can make the same settings and queries that you can by using the front panel.

- However, you need to stop measurement before you change measurement conditions remotely.
- You cannot change the measurement conditions during real-time measurement or during averaged measurement. During measurement, only the :REALtime:STOP or AVERage:STOP command is valid.

:ACQuire:ADSave

Function Sets or queries auto data saving to files after executing averaged measurement.

Syntax
```
:ACQuire:ADSave <Boolean>
:ACQuire:ADSave?
```

Example
```
:ACQUIRE:ADSAVE ON
:ACQUIRE:ADSAVE? -> :ACQUIRE:ADSAVE 1
```

Description If automatic saving is set to ON, the auto save mode is set to user defined.

:ACQuire:AESearch

Function Sets or queries auto event detection after executing averaged measurement.

Syntax
```
:ACQuire:AESearch <Boolean>
:ACQuire:AESearch?
```

Example
```
:ACQUIRE:AESEARCH OFF
:ACQUIRE:AESEARCH?
-> :ACQUIRE:AESEARCH 0
```

:ACQuire:ATTenuation

Function Sets or queries the attenuation value.

Syntax
```
:ACQuire:ATTenuation {<NRf>|AUTO}
:ACQuire:ATTenuation?
```
<NRf> = 0.00 to 35.00 (in 2.5 dB steps)

Example
```
:ACQUIRE:ATTENUATION 2.5
:ACQUIRE:ATTENUATION AUTO
:ACQUIRE:ATTENUATION?
-> :ACQUIRE:ATTENUATION 25.00
```

Description
- The range that can be specified varies depending on the wavelength, distance range, and pulse width.
- You cannot set the attenuation when the average method is Hi-Reflection.
- You cannot set the attenuation when the distance range is set to auto.

:ACQuire:AUTO:ATTenuation?

Function Queries the attenuation value for auto mode.

Syntax
```
:ACQuire:AUTo:ATTenuation?
```

Example
```
:ACQUIRE:AUTO:ATTENUATION?
-> :ACQ:AUTO:ATTENUATION 0.00
```

:ACQuire:AUTO:DRANge?

Function Queries the distance range value for auto mode.

Syntax
```
:ACQuire:AUTo:DRANge?
```

Example
```
:ACQUIRE:AUTO:DRANGE?
-> :ACQ:AUTO:DRANGE 20000
```

Description The unit is m (meters).

:ACQuire:AUTO:PWIDth?

Function Queries the pulse width for auto mode.

Syntax
```
:ACQuire:AUTo:PWIDth?
```

Example
```
:ACQUIRE:AUTO:PWIDTH?
-> :ACQ:AUTO:PWIDTH 100E-09
```

Description The unit is s (seconds).

:ACQuire:AVERage:COUNt

Function Queries the current average count.

Syntax
```
:ACQuire:AVERage:COUNt?
```

Example
```
:ACQUIRE:AVERAGE:COUNT?
-> ACQUIRE:AVERAGE:COUNT 0
```

:ACQuire:AVERage:INDex

Function Sets or queries the average times.

Syntax
```
:ACQuire:AVERage:INDex {AUTO|E2 _ 10|
E2 _ 11|……|E2 _ 20}
:ACQuire:AVERage:INDex?
```
AUTO: Auto
E2_10: 2E+10
E2_11: 2E+11
: :
E2_20: 2E+20

Example
```
:ACQUIRE:AVERAGE:INDEX AUTO
:ACQUIRE:AVERAGE:INDEX E2 _ 12
:ACQUIRE:AVERAGE:INDEX?
-> :ACQ:AVER:IND E2 _ 15
```

:ACQuire:AVERage:MODE

Function Sets or queries the average method.

Syntax
```
:ACQuire:AVERage:MODE {HIREFLECTION|
HISPEED}
:ACQuire:AVERage:MODE?
```
HIREFLECTION: High reflection
HISPEED: High speed

Example
```
:ACQUIRE:AVERAGE:MODE HIREFLECTION
:ACQUIRE:AVERAGE:MODE?
-> :ACQUIRE:AVERAGE:MODE HISPEED
```

:ACQuire:AVERage:STARt

Function Executes averaged measurement.
Syntax :ACQuire:AVERage:STARt
Example :ACQUIRE:AVERAGE:START
Description This command can be used only when measurement is stopped.

:ACQuire:AVERage:STOP

Function Stops averaged measurement.
Syntax :ACQuire:AVERage:STOP
Example :ACQUIRE:AVERAGE:STOP
Description This command can be used only when averaged measurement is in progress.

:ACQuire:AVERage:TIME

Function Sets or queries the averaged time.
Syntax :ACQuire:AVERage:TIME {<NRf>|AUTO}
:ACQuire:AVERage:TIME?
<NRf> = 5, 10, 20, 30, 60, 180, 300, 600, 1200, 1800 (s)
AUTO: Auto
5: 5 s
10: 10 s
20: 20 s
30: 30 s
60: 1 min
180: 3 min
300: 5 min
600: 10 min
1200: 20 min
1800: 30 min
Example :ACQUIRE:AVERAGE:TIME AUTO
:ACQUIRE:AVERAGE:TIME 1200
:ACQUIRE:AVERAGE:TIME?
-> :ACQUIRE:AVERAVE:TIME 60

:ACQuire:AVERage:TYPE

Function Sets or queries the average unit.
Syntax :ACQuire:AVERage:TYPE {TIMES|DURATION}
ACQuire:AVERage:TYPE?
TIMES: Number of times
DURATION: Time
Example :ACQUIRE:AVERAGE:TYPE TIMES
:ACQUIRE:AVERAGE:TYPE DURATION
:ACQUIRE:AVERAGE:TYPE?
-> :ACQUIRE:AVERAGE:TYPE DURATION

:ACQuire:DRANge

Function Sets or queries the distance range.
Syntax :ACQuire:DRANge {<NRf>|AUTO}
:ACQuire:DRANge?
<NRf> = 100 m to 512000 m (0.1 km to 512 km)
AUTO: Auto
100: 100 m
200: 200 m
500: 500 m
1000: 1 km
2000: 2 km
5000: 5 km
10000: 10 km
20000: 20 km
30000: 30 km
40000: 40 km
50000: 50 km
100000: 100 km
200000: 200 km
256000: 256 km
300000: 300 km
400000: 400 km
512000: 512 km
Example :ACQUIRE:DRANGE AUTO
:ACQUIRE:DRANGE 500
:ACQUIRE:DRANGE 500M
:ACQUIRE:DRANGE 5000
:ACQUIRE:DRANGE 100KM
:ACQUIRE:DRANGE?
-> :ACQUIRE:DRANGE 1000
Description The values that you can specify vary depending on the wavelength setting. You can set the distance unit to m (meters) or km (kilometers). If the unit is omitted, m (meters) will be set.

:ACQuire:OFFSet

Function Sets or queries the measurement start position on the horizontal axis.
Syntax :ACQuire:OFFSet <NRf>
:ACQuire:OFFSet?
Example :ACQUIRE:OFFSET 1000
:ACQUIRE:OFFSET?
-> :ACQUIRE:OFFSET 999.99265E+00
Description The unit is m (meters).

:ACQuire:PLUGcheck

Function Sets or queries the optical plug connection check.
Syntax :ACQuire:PLUGcheck <Boolean>
:ACQuire:PLUGcheck?
Example :ACQUIRE:PLUGCHECK ON
:ACQUIRE:PLUGCHECK?
-> :ACQUIRE:PLUGCHECK 1

:ACQuire:PWIDth

Function Sets or queries the pulse width.

Syntax
```
:ACQuire:PWIDth {<NRf>|AUTO}
:ACQuire:PWIDth?
```
<NRf> = 3 ns to 20 µs (3E-9 to 20E-6)

AUTO:	Auto
3E-09:	3 ns
5E-09:	5 ns
10E-09:	10 ns
20E-09:	20 ns
30E-09:	30 ns
50E-09:	50 ns
100E-09:	100 ns
200E-09:	200 ns
300E-09:	300 ns
500E-09:	500 ns
1E-06:	1 µs
2E-06:	2 µs
5E-06:	5 µs
10E-06:	10 µs
20E-06:	20 µs

Example
```
:ACQUIRE:PWIDTH AUTO
:ACQUIRE:PWIDTH 3E-9
:ACQUIRE:PWIDTH 3NS
:ACQUIRE:PWIDTH 10US
:ACQUIRE:PWIDTH 10E-6
:ACQUIRE:PWIDTH?
-> :ACQUIRE:PWIDTH 10.0E-06
```

Description
- The value that can be specified varies depending on the wavelength and distance range.
- You cannot set the attenuation when the distance range is set to auto.

:ACQuire:REALtime:STARt

Function Executes real-time measurement.

Syntax `:ACQuire:REALtime:STARt`

Example `:ACQUIRE:REALTIME:START`

Description You can only use this command when measurement is stopped.

:ACQuire:REALtime:STOP

Function Stops real-time measurement.

Syntax `:ACQuire:REALtime:STOP`

Example `:ACQUIRE:REALTIME:STOP`

Description This command can be used only when real-time measurement is in progress.

:ACQuire:REALtime:MODE

Function Sets or queries the average mode for real-time measurement.

Syntax
```
:ACQuire:REALtime:MODE {HIREFLECTION|
HISPEED}
:ACQuire:REALtime:MODE?
```
HIREFLECTION: High reflection
HISPEED: High speed

Example
```
:ACQUIRE:REALTIME:MODE HIREFLECTION
:ACQUIRE:REALTIME:MODE?
-> :ACQuire:REALtime:
MODE HIREFLECTION
```

:ACQuire:RESult:INDex?

Function Queries the actual averaging count after the measurement.

Syntax `:ACQuire:RESult:INDex?`

Example
```
:ACQUIRE:RESULT:INDEX?
-> :ACQUIRE:RESULT:INDEX 13312
```

Description The unit is count.
- If the average count/duration is set to Auto, the average count at the point when the measurement automatically stopped is returned.
- Otherwise, the average count normally matches the specified value. It may not match, for example, if the measurement is interrupted.

:ACQuire:RESult:TIME?

Function Queries the actual average duration after the measurement.

Syntax `:ACQuire:RESult:TIME?`

Example
```
:ACQUIRE:RESULT:TIME?
-> :ACQUIRE:RESULT:TIME 30
```

Description The unit is seconds.
- If the average count/duration is set to Auto, the average duration at the point when the measurement automatically stopped is returned.
- Otherwise, the average count normally matches the specified value. It may not match, for example, if the measurement is interrupted.

:ACQuire:SELection:ATTenuation?

Function Queries the selectable attenuation values.

Syntax `:ACQuire:SELection:ATTenuation?`

Example
```
:ACQUIRE:SELECTION:ATTENUATION?
-> :ACQUIRE:SELECTION:
ATTENUATION 0.00,2.50,5.00,…,27.50
```

Description The unit is dB.

:ACQuire:SELection:DRANge?

Function Queries the selectable distance range values.
Syntax :ACQuire:SELection:DRANge?
Example :ACQUIRE:SELECTION:DRANGE?
-> :ACQUIRE:SELECTION:
DRANGE 500,1000,…,4000000
Description The unit is m (meters).

:ACQuire:SELection:PWIDth?

Function Queries a list of selectable pulse widths.
Syntax :ACQuire:SELection:PWIDth?
Example :ACQUIRE:SELECTION:PWIDTH?
-> :ACQUIRE:SELECTION:
PWIDTH 3E-9.10E-9,…,20E-6
Description The unit is s (seconds).

:ACQuire:SELection:SMPinterval?

Function Queries the selectable sampling intervals.
Syntax :ACQuire:SELection:SMPinterval?
Example :ACQUIRE:SELECTION:SMPINTERVAL?
-> :ACQUIRE:SELECTION:
SMPINTERVAL 0.05,0.10,0.20,…,4.00

:ACQuire:SELection:WAVelength?

Function Queries the selectable measurement wavelengths.
Syntax :ACQuire:SELection:WAVelength?
Example :ACQUIRE:SELECTION:WAVELENGTH?
-> :ACQUIRE:SELECTION:
WAVELENGTH 1.310E-6,1.550E-6
Description The wavelengths that you can select vary depending on the model.

:ACQuire:SMPinterval:DATA

Function Sets or queries the sampling interval.
Syntax :ACQuire:SMPinterval:DATA {<NRf>|NORMAL|HI}
:ACQuire:SMPinterval:DATA?
<NRf> = 0.02 m to 32 m
NORMAL: Normal
HI: High resolution
0.02: 2 cm
0.05: 5 cm
0.10: 10 cm
0.20: 20 cm
0.50: 50 cm
1.00: 1 m
2.00: 2 m
4.00: 4 m
8.00: 8 m
16.00: 16 m
32.00: 32 m
Example :ACQUIRE:SMPINTERVAL:DATA 2.0
:ACQUIRE:SMPINTERVAL:DATA NORMAL
:ACQUIRE:SMPINTERVAL:DATA?
-> :ACQ:SMP:DATA HI
Description You can only set the attenuation to NORMAL or HI when the distance range is set to auto.

:ACQuire:SMPinterval:VALue?

Function Queries the sampling interval.
Syntax :ACQuire:SMPinterval:VALue?
Example :ACQUIRE:SMPINTERVAL:VALUE?
-> :ACQUIRE:SMPINTERVAL:VALUE 8.0

:ACQuire:WAVelength

Function Sets or queries the measurement wavelength.
Syntax :ACQuire:WAVelength <NRf>
:ACQuire:WAVelength?
<NRf> = 0.850 µm to 1.650 µm
(850E-9 to 1650E-9)
1310E-09: 1310 nm
1550E-09: 1550 nm
Example :ACQUIRE:WAVELENGTH 0.85UM
:ACQUIRE:WAVELENGTH 1650E-9
:ACQUIRE:WAVELENGTH 1.650E-6
:ACQUIRE:WAVELENGTH?
-> :ACQUIRE:WAVELENGTH 1550E-09
Description The wavelengths that you can specify vary depending on the model.

5.3 ANALysis Group

The commands in this group deal with waveform analysis.
You can make the same settings and queries that you can by using the front panel.

:ANALysis:ASEarch:EXECute

Function Executes auto event detection.
Syntax :ANALysis:ASEarch:EXECute
Example :ANALYSIS:ASEARCH:EXECUTE
Description If you execute auto detection, check that the AS flag value is zero with a :STATus:CONDition? query, and then query the event results.

:ANALysis:ASEarch:NUMBer?

Function Queries the number of automatically detected events.
Syntax :ANALysis:ASEarch:NUMber?
Example :ANALYSIS:ASEARCH:NUMBER?
-> :ANALYSIS:ASEARCH:NUMBER 2

:ANALysis:BCOefficient

Function Sets or queries a backscattering light level of the current wavelength.
Syntax :ANALysis:BCOefficient <NRf>
:ANALysis:BCOefficient?
<NRf> = -64.99 to -10.00 (in 0.01 steps)
Example :ANALYSIS:BCOEFFICIENT -25.00
:ANALYSIS:BCOEFFICIENT?
-> :ANALYSIS:BCOEFFICIENT -50.00

:ANALysis:CUMLoss

Function Sets or queries the cumulative loss type.
Syntax :ANALysis:CUMLoss {TYPE1|TYPE2|TYPE3}
:ANALysis:CUMLoss?
TYPE1: Type 1
TYPE2: Type 2
TYPE3: Type 3
Example :ANALYSIS:CUMLOSS TYPE2
:ANALYSIS:CUMLOSS?
-> :ANALYSIS:CUMLOSS TYPE2

:ANALysis:CURSor:DECibel?

Function Queries the cursor dB (decibels).
Syntax :ANALysis:CURSor:DECibel?
Example :ANALYSIS:CURSOR:DECIBEL?
-> :ANALYSIS:CURSOR:DECIBEL 32.878

:ANALysis:CURSor:DELete

Function Deletes the cursor.
Syntax :ANALysis:CURSor:DELete
Example :ANALYSIS:CURSOR:DELETE

:ANALysis:CURSor:DISTance

Function Sets or queries the cursor position.
Syntax :ANALysis:CURSor:DISTance <NRf>
:ANALysis:CURSor:DISTance?
<NRf> = 0 to distance range (m)
Example :ANALYSIS:CURSOR:DISTANCE 10.00
:ANALYSIS:CURSOR:DISTANCE?
-> :ANALYSIS:CURS:
DISTANCE 11.529900E+00
Description This depends on the distance range, distance reference, and index of refraction.

:ANALysis:CURSor:INDex

Function Sets or queries the cursor position using a data number.
Syntax :ANALysis:CURSor:INDex <NRf>
:ANALysis:CURSor:INDex?
<NRf> = 0 to maximum number of data points
Example :ANALYSIS:CURSOR:INDEX 1000
:ANALYSIS:CURSOR:INDEX?
-> :ANALYSIS:CURSOR:INDEX 1000
Description Depends on the number of measurement data points.

:ANALysis:CURSor:LINK

Function Sets or queries the cursor link.
Syntax :ANALysis:CURSor:LINK <Boolean>
:ANALysis:CURSor:LINK?
Example :ANALYSIS:CURSOR:LINK ON
:ANALYSIS:CURSOR:LINK?
-> :ANALYSIS:CURSOR:LINK 1

:ANALysis:DUMMy:DISTance:STARt

Function Sets or queries the launch fiber start position.
Syntax :ANALysis:DUMMy:DISTance:STARt <NRf>
:ANALysis:DUMMy:DISTance:STARt?
<NRf> = 0 to 999999 (cm)
Example :ANALYSIS:DUMMY:DISTANCE:START 100000
:ANALYSIS:DUMMY:DISTANCE:START?
-> :ANALYSIS:DUMMY:DISTANCE:
START 100000

:ANALysis:DUMMy:DISTance:END

Function Sets or queries the launch fiber end position.
Syntax :ANALysis:DUMMy:DISTance:END <NRf>
:ANALysis:DUMMy:DISTance:END?
<NRf> = 0 to 999999 (cm)
Example :ANALYSIS:DUMMY:DISTANCE:END 100000
:ANALYSIS:DUMMY:DISTANCE:END?
-> :ANALYSIS:DUMMY:DISTANCE:END 10000

:ANALysis:DUMMy:END

Function Sets the end event of the Launch fiber event number.

Syntax
```
:ANALysis:DUMMy:END <NRf>
:ANALysis:DUMMy:END?
```
<NRf> = 0 to 2

Example
```
:ANALYSIS:DUMMY:END 0
:ANALYSIS:DUMMY:END?
-> :ANALYSIS:DUMMY:END 1
```

:ANALysis:DUMMy:MODE

Function Sets or queries the launch fiber setup mode.

Syntax
```
:ANALysis:DUMMy:MODE {NUMB|DIST}
ANALysis:DUMMy:MODE?
```
NUNB: Event number
DIST: Position

Example
```
:ANALYSIS:DUMMY:MODE DIST
:ANALYSIS:DUMMY:MODE?
-> :ANALYSIS:DUMMY:MODE DIST
```

:ANALysis:DUMMy:STARt

Function Sets the start event of the Launch fiber event number.

Syntax
```
:ANALysis:DUMMy:STARt <NRf>
:ANALysis:DUMMy:STARt?
```
<NRf> = 0 to 2

Example
```
:ANALYSIS:DUMMY:START 0
:ANALYSIS:DUMMY:START?
-> :ANALYSIS:DUMMY:START 1
```

:ANALysis:DUNit

Function Sets or queries the distance unit.

Syntax
```
:ANALysis:DUNit {KM|MILE|KF}
:ANALysis:DUNit?
```
KM: km
MILE: mile
KF: kf

Example
```
:ANALYSIS:DUNIT KM
:ANALYSIS:DUNIT?
-> :ANALYSIS:DUNIT KM
```

Description The unit cannot be set to mile, kf in Japanese mode.

:ANALysis:EMARker:LMTechnique

Function Sets or queries the approximation method (event).

Syntax
```
:ANALysis:EMARker:LMTechnique {LSA|
TPA}
:ANALysis:EMARker:LMTechnique?
```
LSA: Least squares method
TPA: 2-point method

Example
```
:ANALYSIS:EMARKER:LMTECHNIQUE LSA
:ANALYSIS:EMARKER:LMTECHNIQUE?
-> :ANALYSIS:EMARKER:LMTECHINIQUE TPA
```

:ANALysis:EMARker:SET:M1

Function Sets or queries marker M1 of the current event.

Syntax
```
:ANALysis:EMARker:SET:M1
:ANALysis:EMARker:SET:M1?
```

Example
```
:ANALYSIS:EMARKER:SET:M1
:ANALYSIS:EMARKER:SET:M1?
-> :ANALYSIS:EMARKER:SET:
M1 5.0471900E+03
```

Description The marker is set to the cursor position.

:ANALysis:EMARker:SET:M2

Function Sets or queries marker M2 of the current event.

Syntax
```
:ANALysis:EMARker:SET:M2
:ANALysis:EMARker:SET:M2?
```

Example
```
:ANALYSIS:EMARKER:SET:M2
:ANALYSIS:EMARKER:SET:M2?
-> :ANALYSIS:EMARKER:SET:
M2 7.2463500E+03
```

Description The marker is set to the cursor position.

:ANALysis:EMARker:SET:M3

Function Sets or queries marker M3 of the current event.

Syntax
```
:ANALysis:EMARker:SET:M3
:ANALysis:EMARker:SET:M3?
```

Example
```
:ANALYSIS:EMARKER:SET:M3
:ANALYSIS:EMARKER:SET:M3?
-> :ANALYSIS:EMARKER:SET:
M3 9.0471600E+03
```

Description The marker is set to the cursor position.

:ANALysis:EMARker:SET:Y2

Function Sets or queries marker Y2 of the current event.

Syntax
```
:ANALysis:EMARker:SET:Y2
:ANALysis:EMARker:SET:Y2?
```

Example
```
:ANALYSIS:EMARKER:SET:Y2
:ANALYSIS:EMARKER:SET:Y2?
-> :ANALYSIS:EMARKER:SET:
Y2 7.3777700E+03
```

Description The marker is set to the cursor position.

:ANALysis:ENABle:FEB2sploss

Function Sets or queries the pass/fail judgment on/off state of the two-split splitter loss threshold.

Syntax
```
:ANALysis:ENABle:FEB2sploss <Boolean>
:ANALysis:ENABle:FEB2sploss?
```

Example
```
:ANALYSIS:ENABLE:FEB2SPLOSS ON
:ANALYSIS:ENABLE:FEB2SPLOSS?
-> :ANALYSIS:ENABLE:FEB2SPLOSS 1
```

Description After setting the value, wait for the AS flag in the value obtained with the :STATus:CONDition? command to become zero, and then obtain the event results.

:ANALysis:ENABle:FEB4sploss

Function Sets or queries the pass/fail judgment on/off state of the four-split splitter loss threshold.

Syntax
```
:ANALysis:ENABle:FEB4SPloss <Boolean>
:ANALysis:ENABle:FEB4SPloss?
```

Example
```
:ANALYSIS:ENABLE:FEB4SPLOSS ON
:ANALYSIS:ENABLE:FEB4SPLOSS?
-> :ANALYSIS:ENABLE:FEB4SPLOSS 1
```

Description After setting the value, wait for the AS flag in the value obtained with the :STATus:CONDition? command to become zero, and then obtain the event results.

:ANALysis:ENABle:FEB8sploss

Function Sets or queries the pass/fail judgment on/off state of the eight-split splitter loss threshold.

Syntax
```
:ANALysis:ENABle:FEB8sploss <Boolean>
:ANALysis:ENABle:FEB8sploss?
```

Example
```
:ANALYSIS:ENABLE:FEB8SPLOSS ON
:ANALYSIS:ENABLE:FEB8SPLOSS?
-> :ANALYSIS:ENABLE:FEB8SPLOSS 1
```

Description After setting the value, wait for the AS flag in the value obtained with the :STATus:CONDition? command to become zero, and then obtain the event results.

:ANALysis:ENABle:FEB16sploss

Function Sets or queries the pass/fail judgment on/off state of the 16-split splitter loss threshold.

Syntax
```
:ANALysis:ENABle:
FEB16sploss <Boolean>
:ANALysis:ENABle:FEB16sploss?
```

Example
```
:ANALYSIS:ENABLE:FEB16SPLOSS ON
:ANALYSIS:ENABLE:FEB16SPLOSS?
-> :ANALYSIS:ENABLE:FEB16SPLOSS 1
```

Description After setting the value, wait for the AS flag in the value obtained with the :STATus:CONDition? command to become zero, and then obtain the event results.

:ANALysis:ENABle:FEB32sploss

Function Sets or queries the pass/fail judgment on/off state of the 32-split splitter loss threshold.

Syntax
```
:ANALysis:ENABle:
FEB32sploss <Boolean>
:ANALysis:ENABle:FEB32sploss?
```

Example
```
:ANALYSIS:ENABLE:FEB32SPLOSS ON
:ANALYSIS:ENABLE:FEB32SPLOSS?
-> :ANALYSIS:ENABLE:FEB32SPLOSS 1
```

Description After setting the value, wait for the AS flag in the value obtained with the :STATus:CONDition? command to become zero, and then obtain the event results.

:ANALysis:ENABle:FEB64sploss

Function Sets or queries the pass/fail judgment on/off state of the 64-split splitter loss threshold.

Syntax
```
:ANALysis:ENABle:
FEB64SPloss <Boolean>
:ANALysis:ENABle:FEB64SPloss?
```

Example
```
:ANALYSIS:ENABLE:FEB64SPLOSS ON
:ANALYSIS:ENABLE:FEB64SPLOSS?
-> :ANALYSIS:ENABLE:FEB64SPLOSS 1
```

Description After setting the value, wait for the AS flag in the value obtained with the :STATus:CONDition? command to become zero, and then obtain the event results.

:ANALysis:ENABle:FECLoss

Function Sets or queries the pass/fail judgment on/off state of the connector loss threshold.

Syntax
```
:ANALysis:ENABle:FECLoss <Boolean>
:ANALysis:ENABle:FECLoss?
```

Example
```
:ANALYSIS:ENABLE:FECLOSS ON
:ANALYSIS:ENABLE:FECLOSS?
-> :ANALYSIS:ENABLE:FECLOSS 1
```

Description After setting the value, wait for the AS flag in the value obtained with the :STATus:CONDition? command to become zero, and then obtain the event results.

:ANALysis:ENABle:FERLoss

Function Sets or queries the pass/fail judgment on/off state of the return loss threshold.

Syntax
```
:ANALysis:ENABle:FERLoss <Boolean>
:ANALysis:ENABle:FERLoss?
```

Example
```
:ANALYSIS:ENABLE:FERLOSS ON
:ANALYSIS:ENABLE:FERLOSS?
-> :ANALYSIS:ENABLE:FERLOSS 1
```

Description After setting the value, wait for the AS flag in the value obtained with the :STATus:CONDition? command to become zero, and then obtain the event results.

:ANALysis:ENABle:FESLoss

Function Sets or queries the pass/fail judgment on/off state of the splice loss threshold.

Syntax
```
:ANALysis:ENABle:FESLoss <Boolean>
:ANALysis:ENABle:FESLoss?
```

Example
```
:ANALYSIS:ENABLE:FESLOSS ON
:ANALYSIS:ENABLE:FESLOSS?
-> :ANALYSIS:ENABLE:FESLOSS 1
```

Description After setting the value, wait for the AS flag in the value obtained with the :STATus:CONDition? command to become zero, and then obtain the event results.

:ANALysis:ENABle:FETLoss

Function Sets or queries the pass/fail judgment on/off state of the total loss threshold.

Syntax
```
:ANALysis:ENABle:FETLoss <Boolean>
:ANALysis:ENABle:FETLoss?
```

Example
```
:ANALYSIS:ENABLE:FETLOSS ON
:ANALYSIS:ENABLE:FETLOSS?
 -> :ANALYSIS:ENABLE:FETLOSS 1
```

Description After setting the value, wait for the AS flag in the value obtained with the :STATus:CONDition? command to become zero, and then obtain the event results.

:ANALysis:ENABle:FEULoss

Function Sets or queries the pass/fail judgment on/off state of the dB/km threshold.

Syntax
```
:ANALysis:ENABle:FEULoss <Boolean>
:ANALysis:ENABle:FEULoss?
```

Example
```
:ANALYSIS:ENABLE:FEULOSS ON
:ANALYSIS:ENABLE:FEULOSS?
-> :ANALYSIS:ENABLE:FEULOSS 1
```

Description After setting the value, wait for the AS flag in the value obtained with the :STATus:CONDition? command to become zero, and then obtain the event results.

:ANALysis:ENDLoss

Function Sets or queries the loss indication of the END point.

Syntax
```
:ANALysis:ENDLoss <Boolean>
:ANALysis:ENDLoss?
```
0: The loss at the END point is not displayed.
1: The loss at the END point is displayed.

Example
```
:ANALYSIS:ENDLOSS 1
:ANALYSIS:ENDLOSS?
-> :ANALYSIS:ENDLOSS 1
```

:ANALysis:EVENt:CURRent:BRANch?

Function Gets the number of splitter splits of the current event.

Syntax
```
:ANALysis:EVENt:CURRent:BRANch?
```

Example
```
:ANALYSIS:EVENT:CURRENT:BRANCH?
-> :ANALYSIS:EVENT:CURRENT:BRANCH 2
```

escription Use the :ANALysis:EVENt:CURRent:INDex command to specify the current event first, and then use this command.

:ANALysis:EVENt:CURRent:CUMLoss?

Function Queries the cumulate loss of the current event.

Syntax
```
:ANALysis:EVENt:CURRent:CUMLoss?
```

Example
```
:ANALYSIS:EVENT:CURRENT:CUMLOSS?
-> :ANALYSIS:EVENT:CURRENT:
  CUMLOSS 1.810
```

Description Use the :ANALysis:EVENt:CURRent:INDex command to specify the current event first, and then use this command.

:ANALysis:EVENt:CURRent:DISTance?

Function Queries the distance of the current event.

Syntax
```
:ANALysis:EVENt:CURRent:DISTance?
```

Example
```
:ANALYSIS:EVENT:CURRENT:DISTANCE?
-> :ANALYSIS:EVENT:CURRENT:
  DISTANCE 987.000
```

Description Use the :ANALysis:EVENt:CURRent:INDex command to specify the current event first, and then use this command.

:ANALysis:EVENt:CURRent:INDex

Function Specifies the current event.

Syntax
```
:ANALysis:EVENt:CURRent:INDex <NRf>
```
<NRf> = event number (-1 to 100)
1 to 100: Event number
-1: Point S, point R
0: Point E
-2: Point O
-3: Point T

Example
```
:ANALYSIS:EVENT:CURRENT:INDEX 5
```

Description The event of the specified number becomes the current event.

:ANALysis:EVENt:CURRent:IOR?

Function Queries the section group index of the current event.

Syntax
```
:ANALysis:EVENt:CURRent:IOR?
```

Example
```
:ANALYSIS:EVENT:CURRENT:IOR?
-> :ANALYSIS:EVENT:CURRENT:IOR 1.46000
```

Description Use the :ANALysis:EVENt:CURRent:INDex command to specify the current event first, and then use this command.

:ANALysis:EVENt:CURRent:LOSS?

Function Queries the splice loss of the current event.

Syntax
```
:ANALysis:EVENt:CURRent:LOSS?
```

Example
```
:ANALYSIS:EVENT:CURRENT:LOSS?
-> :ANALYSIS:EVENT:CURRENT:LOSS 2.000
```

Description Use the :ANALysis:EVENt:CURRent:INDex command to specify the current event first, and then use this command.

:ANALysis:EVENt:CURRent:RETurnloss?

Function Queries the return loss of the current event.

Syntax
```
:ANALysis:EVENt:CURRent:RETurnloss?
```

Example
```
:ANALYSIS:EVENT:CURRENT:RETURNLOSS?
-> :ANALYSIS:EVENT:CURRENT:
RETURNLOSS 2.000
```

Description Use the :ANALysis:EVENt:CURRent:INDex command to specify the current event first, and then use this command.

:ANALysis:EVENt:CURRent:TYPE?

Function Queries the event type of the current event.
Syntax :ANALysis:EVENt:CURRent:TYPE?

REFLectance:	Reflection
SPLus:	Positive loss
SMINus:	Negative loss
SPLITter:	Splitter
BENDing:	Macro bending
DREFLectance:	Consecutive reflection
DSPLITter:	Consecutive splitter

Example
```
:ANALYSIS:EVENT:CURRENT:TYPE?
-> :ANALYSIS:EVENT:CURRENT:
  TYPE REFLECTANCE
```
Description Use the :ANALysis:EVENt:CURRent:INDex command to specify the current event first, and then use this command.

:ANALysis:EVENt:CURRent:UNITloss?

Function Queries the dB/km of the current event.
Syntax :ANALysis:EVENt:CURRent:UNITloss?
Example
```
:ANALYSIS:EVENT:CURRENT:UNITLOSS?
-> :ANALYSIS:EVENT:CURRENT:
  UNITLOSS 2.000
```
Description Use the :ANALysis:EVENt:CURRent:INDex command to specify the current event first, and then use this command.

:ANALysis:EVENt:DELete

Function Deletes the current event.
Syntax :ANALysis:EVENt:DELete
Example
```
:ANALYSIS:EVENT:DELETE
```

:ANALysis:EVENt:FIX:MODE

Function Sets or queries the event fix mode.
Syntax :ANALysis:EVENt:FIX:MODE {MODE1|MODE2}
:ANALysis:EVENt:FIX:MODE?
MODE1: Mode 1 (add mode)
MODE2: Mode 2 (master mode)
Example
```
:ANALYSIS:EVENT:FIX:MODE MODE1
:ANALYSIS:EVENT:FIX:MODE?
-> :ANALYSIS:EVENT:FIX:MODE MODE1
```

:ANALysis:EVENt:FIX:STATe

Function Sets or queries the event fix.
Syntax :ANALysis:EVENt:FIX:STATe <Boolean>
:ANALysis:EVENt:FIX:STATe?
0: Event fix is disabled.
1: Event fix is enabled.
Example
```
:ANALYSIS:EVENT:FIX:STATE 1
:ANALYSIS:EVENT:FIX:STATE?
-> :ANALYSIS:EVENT:FIX:STATE 1
```

:ANALysis:EVENt:INSert

Function Inserts an event at the cursor position.
Syntax :ANALysis:EVENt:INSert
Example
```
:ANALYSIS:EVENT:INSERT
```

:ANALysis:EVENt:IOR

Function Sets or queries the section group index of the current event.
Syntax :ANALysis:EVENt:IOR <NRf>
:ANALysis:EVENt:IOR?
<NRf> = 1.30000 to 1.79999 (in 0.00001 steps)
Example
```
:ANALYSIS:EVENT:IOR 1.47
:ANALYSIS:EVENT:IOR?
-> :ANALYSIS:EVENT:IOR 1.46000
```

:ANALysis:FEDetection

Function Sets or queries the fault event display.
Syntax :ANALysis:FEDetection <Boolean>
:ANALysis:FEDetection?
Example
```
:ANALYSIS:FEDETECTION ON
:ANALYSIS:FEDETECTION?
-> :ANALYSIS:FEDETECTION 1
```

:ANALysis:FESPlitter:BRANch<x>

Function Sets or queries the fault event's number of splitter splits.
Syntax :ANALysis:FESPlitter:BRANch {UNKNOWN|2|4|8|16|32|64}
:ANALysis:FESPlitter:BRANch?
<NRf> = 1, 2
Example
```
:ANALYSIS:FESPLITTER:BRANCH 32
:ANALYSIS:FESPLITTER:BRANCH?
-> :ANALYSIS:FESPLITTER:BRANCH 32
```

:ANALysis:FESPlitter:DETection

Function Sets or queries the fault event's network configuration on/off state.
Syntax :ANALysis:FESPlitter:DETection <Boolean>
:ANALysis:FESPlitter:DETection?
Example
```
:ANALYSIS:FESPLITTER:DETECTION ON
:ANALYSIS:FESPLITTER:DETECTION?
-> :ANALYSIS:FESPLITTER:DETECTION 1
```

:ANALysis:FESPlitter:STAGe

Function Sets or queries the fault event's number of splitter stages.
Syntax :ANALysis:FESPlitter:STAGe {0|1|2}
:ANALysis:FESPlitter:STAGe?
Example
```
:ANALYSIS:FESPLITTER:STAGE 1
:ANALYSIS:FESPLITTER:STAGE?
-> :ANALYSIS:FESPLITTER:STAGE 1
```

:ANALysis:FMARker:DELete

Function Deletes markers.
Syntax :ANALysis:FMARker:DELete
Example :ANALYSIS:FMARKER:DELETE

:ANALysis:FMARker:LEFT:DISTance?

Function Queries the distance between markers 1 and 2.
Syntax :ANALysis:FMARker:LEFT:DISTance?
Example :ANALYSIS:FMARKER:LEFT:DISTANCE?
-> :ANALYSIS:FMARKER:LEFT:
DISTANCE 490.20000E+00

:ANALysis:FMARker:LEFT:LOSS?

Function Queries the loss between markers 1 and 2.
Syntax :ANALysis:FMARker:LEFT:LOSS?
Example :ANALYSIS:FMARKER:LEFT:LOSS?
-> :ANALYSIS:FMARKER:LEFT:
LOSS 137.00000E-03

:ANALysis:FMARker:LEFT:UNITloss?

Function Queries the slope between markers 1 and 2.
Syntax :ANALysis:FMARker:LEFT:UNITloss?
Example :ANALYSIS:FMARKER:LEFT:UNITLOSS?
-> :ANALYSIS:FMARKER:LEFT:
UNITLOSS 279.00000E-03

:ANALysis:FMARker:LMTechnique

Function Sets or queries the marker approximation method.
Syntax :ANALysis:FMARker:LMTechnique {LSA|TPA}
:ANALysis:FMARker:LMTechnique?
LSA: Least squares method
TPA: 2-point method
Example :ANALYSIS:FMARKER:LMTECHNIQUE LSA
:ANALYSIS:FMARKER:LMTECHNIQUE?
-> :ANALYSIS:FMARKER:LMTECHNIQUE TPA

:ANALysis:FMARker:LOSS?

Function Queries the splice loss.
Syntax :ANALysis:FMARker:LOSS?
Example :ANALYSIS:FMARKER:LOSS?
-> :ANALYSIS:FMARKER:
LOSS 162.00000E-03

:ANALysis:FMARker:REFLection:SATurated?

Function Queries the saturated state of the reflection level.
Syntax :ANALysis:FMARker:REFLection:SATurated?
0: Not saturated
1: Saturated
Example :ANALYSIS:FMARKER:REFLECTION:SATURATED?
-> :ANALYSIS:FMARKER:REFLECTION:
SATURATED 0

:ANALysis:FMARker:REFLection:VALue?

Function Queries the reflection level.
Syntax :ANALysis:FMARker:REFLection:VALue?
Example :ANALYSIS:FMARKER:REFLECTION:VALUE?
-> :ANALYSIS:FMARKER:REFLECTION:
VALUE 1.0640000E+00

:ANALysis:FMARker:RETurnloss:SATurated?

Function Queries the saturated state of the return loss.
Syntax :ANALysis:FMARker:RETurnloss:SATurated?
0: Not saturated
1: Saturated
Example :ANALYSIS:FMARKER:RETURNLOSS:SATURATED?
-> :ANALYSIS:FMARKER:RETURN:
SATURATED 1

:ANALysis:FMARker:RETurnloss:VALue?

Function Queries the return loss.
Syntax :ANALysis:FMARker:RETurnloss:VALue?
Example :ANALYSIS:FMARKER:RETURNLOSS:VALUE?
-> :ANALYSIS:FMARKER:RETURNLOSS:
VALUE 47.003000E+00

:ANALysis:FMARker:RIGHt:DISTance?

Function Queries the distance between markers 2 and 3.
Syntax :ANALysis:FMARker:RIGHt:DISTance?
Example :ANALYSIS:FMARKER:RIGHT:DISTANCE?
-> :ANALYSIS:FMARKER:RIGHT:
DISTANCE 232.95000E+00

:ANALysis:FMARker:RIGHt:LOSS?

Function Queries the loss between markers 2 and 3.
Syntax :ANALysis:FMARker:RIGHt:LOSS?
Example :ANALYSIS:FMARKER:RIGHT:LOSS?
-> :ANALYSIS:FMARKER:RIGHT:
LOSS -159.00000E-03

:ANALysis:FMARker:RIGHt:UNITloss?

Function Queries the slope between markers 2 and 3.
Syntax :ANALysis:FMARker:RIGHt:UNITloss?
Example :ANALYSIS:FMARKER:RIGHT:UNITLOSS?
-> :ANALYSIS:FMARKER:RIGHT:
UNITLOSS -683.00000E-03

:ANALysis:FMARker:SET:M<x>

Function Sets or queries markers.
Syntax :ANALysis:FMARker:SET:M<x>
:ANALysis:FMARker:SET:M<x>?
<x> = 1, 2, 3
Example :ANALYSIS:FMARKER:SET:M1
:ANALYSIS:FMARKER:SET:M1?
-> :ANALYSIS:FMARKER:SET:
M1 7.2648300E+03

:ANALysis:FMARker:SET:Y<x>

Function Sets or queries auxiliary markers.
Syntax :ANALysis:FMARker:SET:Y<x>
:ANALysis:FMARker:SET:Y<x>?
<x> = 1, 2, 3
Example :ANALYSIS:FMARKER:SET:Y2
:ANALYSIS:FMARKER:SET:Y2?
-> ANALYSIS:FMARKER:SET:
Y2 6.1786000E+03

:ANALysis:IOR

Function Sets or queries the index of refraction of the current wavelength.
Syntax :ANALysis:IOR <NRf>
:ANALysis:IOR?
<NRf> = 1.30000 to 1.79999 (in 0.00001 steps)
Example :ANALYSIS:IOR 1.48000
:ANALYSIS:IOR? -> :ANALYSIS:IOR 1.48000

:ANALysis:MACRobending:DISPlay

Function Sets or queries the macro bending display on/off state.
Syntax :ANALysis:MACRobending:
DISPlay <Boolean>
Example :ANALYSIS:MACROBENDING:DISPLAY ON
:ANALYSIS:MACROBENDING:DISPLAY?
-> :ANALYSIS:MACROBENDING:DISPLAY 1

:ANALysis:MACRobending:THReshold

Function Sets or queries macro bending threshold value.
Syntax :ANALysis:MACRobending:
THReshold <NRf>
<NRf> = 0.001 to 99.999 (in 0.001 steps)
Example :ANALYSIS:MACROBENDING:THRESHOLD 1
:ANALYSIS:MACROBENDING:THRESHOLD?
-> :ANALYSIS:MACROBENDING:
THRESHOLD 100.00000E-03

:ANALysis:REFerence:DELete

Function Delete the distance reference.
Syntax :ANALysis:REFerence:DELete
Example :ANALYSIS:REFERENCE:DELETE

:ANALysis:REFerence:DISTance

Function Sets the distance reference or queries its current position.
Syntax :ANALysis:REFerence:DISTance
:ANALysis:REFerence:DISTance?
Example :ANALYSIS:REFERENCE:DISTANCE
:ANALYSIS:REFERENCE:DISTANCE?
-> :ANAL:REF:DIST 39.999710E+03
Description When querying, the value is returned in meters in reference to the OTDR port.

:ANALysis:SECTion:BASelevel?

Function Queries the dB value of the interval data reference point.
Syntax :ANALysis:SECTion:BASelevel?
Example :ANALYSIS:SECTION:BASELEVEL?
-> :ANALYSIS:SECTION:
BASELEVEL 34.268000E+00

:ANALysis:SECTion:DELete

Function Deletes the interval analysis data.
Syntax :ANALysis:SECTion:DELete
Example :ANALYSIS:SECTION:DELETE

:ANALysis:SECTion:DISTance?

Function Queries the distance of the interval data.
Syntax :ANALysis:SECTion:DISTance?
Example :ANALYSIS:SECTION:DISTANCE?
-> :ANALYSIS:SECTION:
DISTANCE 80.262240E+03

:ANALysis:SECTion:END

Function Sets or queries the end position of the interval data.
Syntax :ANALysis:SECTion:END <NRf>
<NRf> = End position
:ANALysis:SECTion:END?
Example :ANALYSIS:SECTION:END
:ANALYSIS:SECTION:END?
-> :ANALYSYS:SECTION:END 119.99912E+03

:ANALysis:SECTion:LMTechnique

Function Sets or queries the interval analysis approximation method.
Syntax :ANALysis:SECTion:
LMTechnique {LSA|TPA}
:ANALysis:SECTion:LMTechnique?
LSA: Least squares method
TPA: 2-point method
Example :ANALYSIS:SECTION:LMTECHNIQUE
:ANALYSIS:SECTION:LMTECHNIQUE?
-> :ANALYSIS:SECTION:LMTECHNIQUE LSA

:ANALysis:SECTion:LOSS?

Function Queries the loss of the interval data.
Syntax :ANALysis:SECTion:LOSS?
Example :ANALYSIS:SECTION:LOSS?
-> :ANALYSIS:SECTION:
LOSS 7.9300000E+00

:ANALysis:SECTion:REFerence

Function Sets the interval data reference point.
Syntax :ANALysis:SECTion:REFerence
Example :ANALYSIS:SECTION:REFERENCE

:ANALysis:SECTion:RETurnloss:SATurated?

Function Queries the optical return loss saturation of the interval data.
Syntax :ANALysis:SECTion:RETurnloss:SATurated?
Example :ANALYSIS:SECTION:RETURNLOSS:SATURATED?
-> :ANALYSIS:SECTION:RETURN:
SATURATED 1

:ANALysis:SECTion:RETurnloss:VALue?

Function Queries the optical return loss of the interval data.
Syntax :ANALysis:SECTion:RETurnloss:VALue?
Example :ANALYSIS:SECTION:RETURNLOSS:VALUE?
-> :ANALYSIS:SECTION:RETURNLOSS:
VALUE 48.250000E+00

:ANALysis:SECTion:STARt

Function Sets or queries the start position of the interval data.
Syntax :ANALysis:SECTion:STARt <NRf>
<NRf> = Start position
:ANALysis:SECTion:STARt?
Example :ANALYSIS:SECTION:START
:ANALYSIS:SECTION:START?
-> :ANALYSIS:SECTION:
START 39.736870E+03

:ANALysis:SNAPshot:FIX

Function Sets or queries the snapshot.
Syntax :ANALysis:SNAPshot:FIX <Boolean>
:ANALysis:SNAPshot:FIX?
0: Snapshot invalid
1: Snapshot valid
Example :ANALYSIS:SNAPSHOT:FIX 1
:ANALYSIS:SNAPSHOT:FIX?
-> :ANALYSIS:SNAPSHOT:FIX 1
Description
- If 1 is specified, a snapshot of the current waveform is taken.
- If 0 is specified, the snapshot is cleared.

:ANALysis:THReshold:B2SPloss

Function Sets or queries the two-split splitter loss threshold.
Syntax :ANALysis:THReshold:B2SPloss <NRf>
:ANALysis:THReshold:B2SPloss?
<NRf> = 1 to (four-split threshold)
Example :ANALYSIS:THRESHOLD:B2SPLOSS 1.0
:ANALYSIS:THRESHOLD:B2SPLOSS?
-> :ANALYSIS:THRESHOLD:B2SPLOSS 3.0
Description After setting the value, wait for the AS flag in the value obtained with the :STATus:CONDition? command to become zero, and then obtain the event results.

:ANALysis:THReshold:B4SPloss

Function Sets or queries the four-split splitter loss threshold.
Syntax :ANALysis:THReshold:B4SPloss <NRf>
:ANALysis:THReshold:B4SPloss?
<NRf> = (two-split threshold) to (eight-split threshold)
Example :ANALYSIS:THRESHOLD:B4SPLOSS 1.0
:ANALYSIS:THRESHOLD:B4SPLOSS?
-> :ANALYSIS:THRESHOLD:B4SPLOSS 6.0
Description After setting the value, wait for the AS flag in the value obtained with the :STATus:CONDition? command to become zero, and then obtain the event results.

:ANALysis:THReshold:B8SPloss

Function Sets or queries the eight-split splitter loss threshold.
Syntax :ANALysis:THReshold:B8SPloss <NRf>
:ANALysis:THReshold:B8SPloss?
<NRf> = (four-split threshold) to (16-split threshold)
Example :ANALYSIS:THRESHOLD:B8SPLOSS 1.0
:ANALYSIS:THRESHOLD:B8SPLOSS?
-> :ANALYSIS:THRESHOLD:B8SPLOSS 9.0
Description After setting the value, wait for the AS flag in the value obtained with the :STATus:CONDition? command to become zero, and then obtain the event results.

:ANALysis:THReshold:B16Sploss

Function Sets or queries the 16-split splitter loss threshold.
Syntax :ANALysis:THReshold:B16Sploss <NRf>
:ANALysis:THReshold:B16Sploss?
<NRf> = (eight-split threshold) to (32-split threshold)
Example :ANALYSIS:THRESHOLD:B16SPLOSS 1.0
:ANALYSIS:THRESHOLD:B16SPLOSS?
-> :ANALYSIS:THRESHOLD:B16SPLOSS 12.0
Description After setting the value, wait for the AS flag in the value obtained with the :STATus:CONDition? command to become zero, and then obtain the event results.

:ANALysis:THReshold:B32Sploss

Function Sets or queries the 32-split splitter loss threshold.

Syntax
```
:ANALysis:THReshold:B32Sploss <NRf>
:ANALysis:THReshold:B32Sploss?
```
<NRf> = (16-split threshold) to (64-split threshold)

Example
```
:ANALYSIS:THRESHOLD:B32SPLOSS 1.0
:ANALYSIS:THRESHOLD:B32SPLOSS?
-> :ANALYSIS:THRESHOLD:B32SPLOSS 15.0
```

Description After setting the value, wait for the AS flag in the value obtained with the :STATus:CONDition? command to become zero, and then obtain the event results.

:ANALysis:THReshold:B64Sploss

Function Sets or queries the 64-split splitter loss threshold.

Syntax
```
:ANALysis:THReshold:B64Sploss <NRf>
:ANALysis:THReshold:B64Sploss?
```
<NRf> = (64-split thresholds) to 20

Example
```
:ANALYSIS:THRESHOLD:B64SPLOSS 1.0
:ANALYSIS:THRESHOLD:B64SPLOSS?
-> :ANALYSIS:THRESHOLD:B64SPLOSS 18.0
```

Description After setting the value, wait for the AS flag in the value obtained with the :STATus:CONDition? command to become zero, and then obtain the event results.

:ANALysis:THReshold:EOFiber

Function Sets or queries the end-of-fiber threshold.

Syntax
```
:ANALysis:THReshold:EOFiber <NRf>
:ANALysis:THReshold:EOFiber?
```
<NRf> = 3 to 65

Example
```
:ANALYSIS:THRESHOLD:EOFIBER 5
:ANALYSIS:THRESHOLD:EOFIBER?
-> :ANALYSIS:THRESHOLD:EOFIBER 10
```

Description After setting the value, wait for the AS flag in the value obtained with the :STATus:CONDition? command to become zero, and then obtain the event results.

:ANALysis:THReshold:FEB2sploss

Function Sets or queries the fault event's 2-split splitter loss threshold.

Syntax
```
:ANALysis:THReshold:FEB2sploss <NRf>
:ANALysis:THReshold:FEB2sploss?
```
<NRf> = 1.0 to 30.0

Example
```
:ANALYSIS:THRESHOLD:FEB2SPLOSS 1.0
:ANALYSIS:THRESHOLD:FEB2SPLOSS?
-> :ANALYSIS:THRESHOLD:
  FEB2SPLOSS 20.0
```

Description After setting the value, wait for the AS flag in the value obtained with the :STATus:CONDition? command to become zero, and then obtain the event results.

:ANALysis:THReshold:FEB4sploss

Function Sets or queries the fault event's 4-split splitter loss threshold.

Syntax
```
:ANALysis:THReshold:FEB4sploss <NRf>
:ANALysis:THReshold:FEB4sploss?
```
<NRf> = 1.0 to 30.0

Example
```
:ANALYSIS:THRESHOLD:FEB4SPLOSS 1.0
:ANALYSIS:THRESHOLD:FEB4SPLOSS?
-> :ANALYSIS:THRESHOLD:
  FEB4SPLOSS 20.0
```

Description After setting the value, wait for the AS flag in the value obtained with the :STATus:CONDition? command to become zero, and then obtain the event results.

:ANALysis:THReshold:FEB8sploss

Function Sets or queries the fault event's 8-split splitter loss threshold.

Syntax
```
:ANALysis:THReshold:FEB8sploss <NRf>
:ANALysis:THReshold:FEB8sploss?
```
<NRf> = 1.0 to 30.0

Example
```
:ANALYSIS:THRESHOLD:FEB8SPLOSS 1.0
:ANALYSIS:THRESHOLD:FEB8SPLOSS?
-> :ANALYSIS:THRESHOLD:
  FEB8SPLOSS 20.0
```

Description After setting the value, wait for the AS flag in the value obtained with the :STATus:CONDition? command to become zero, and then obtain the event results.

:ANALysis:THReshold:FEB16sploss

Function Sets or queries the fault event's 16-split splitter loss threshold.

Syntax
```
:ANALysis:THReshold:FEB16sploss <NRf>
:ANALysis:THReshold:FEB16sploss?
```
<NRf> = 1.0 to 30.0

Example
```
:ANALYSIS:THRESHOLD:FEB16SPLOSS 1.0
:ANALYSIS:THRESHOLD:FEB16SPLOSS?
-> :ANALYSIS:THRESHOLD:
  FEB16SPLOSS 20.0
```

Description After setting the value, wait for the AS flag in the value obtained with the :STATus:CONDition? command to become zero, and then obtain the event results.

:ANALysis:THReshold:FEB32sploss

Function Sets or queries the fault event's 32-split splitter loss threshold.

Syntax
```
:ANALysis:THReshold:FEB32SPloss <NRf>
:ANALysis:THReshold:FEB32SPloss?
```
<NRf> = 1.0 to 30.0

Example
```
:ANALYSIS:THRESHOLD:FEB32SPLOSS 1.0
:ANALYSIS:THRESHOLD:FEB32SPLOSS?
-> :ANALYSIS:THRESHOLD:
  FEB32SPLOSS 20.0
```

Description After setting the value, wait for the AS flag in the value obtained with the :STATus:CONDition? command to become zero, and then obtain the event results.

:ANALysis:THReshold:FEB64sploss

Function Sets or queries the fault event's 64-split splitter loss threshold.

Syntax
```
:ANALysis:THReshold:FEB64sploss <NRf>
:ANALysis:THReshold:FEB64sploss?
```
<NRf> = 1.0 to 30.0

Example
```
:ANALYSIS:THRESHOLD:FEB64SPLOSS 1.0
:ANALYSIS:THRESHOLD:FEB64SPLOSS?
-> :ANALYSIS:THRESHOLD:
  FEB64SPLOSS 20.0
```

Description After setting the value, wait for the AS flag in the value obtained with the :STATus:CONDition? command to become zero, and then obtain the event results.

:ANALysis:THReshold:FECLoss

Function Sets or queries the connector loss threshold of the fault event.

Syntax
```
:ANALysis:THReshold:FECLoss <NRf>
:ANALysis:THReshold:FECLoss?
```
<NRf> = 0.01 to 9.99

Example
```
:ANALYSIS:THRESHOLD:FECLOSS 0.01
:ANALYSIS:THRESHOLD:FECLOSS?
-> :ANALYSIS:THRESHOLD:FECLOSS 1.23
```

Description After setting the value, wait for the AS flag in the value obtained with the :STATus:CONDition? command to become zero, and then obtain the event results.

:ANALysis:THReshold:FERLoss

Function Sets or queries the return loss threshold of the fault event.

Syntax
```
:ANALysis:THReshold:FERLoss <NRf>
:ANALysis:THReshold:FERLoss?
```
<NRf> = 20 to 70

Example
```
:ANALYSIS:THRESHOLD:FERLOSS 50
:ANALYSIS:THRESHOLD:FERLOSS?
-> :ANALYSIS:THRESHOLD:FERLOSS 30
```

Description After setting the value, wait for the AS flag in the value obtained with the :STATus:CONDition? command to become zero, and then obtain the event results.

:ANALysis:THReshold:FESLoss

Function Sets or queries the splice loss threshold of the fault event.

Syntax
```
:ANALysis:THReshold:FESLoss <NRf>
:ANALysis:THReshold:FESLoss?
```
<NRf> = 0.01 to 9.99

Example
```
:ANALYSIS:THRESHOLD:FESLOSS 0.01
:ANALYSIS:THRESHOLD:FESLOSS?
-> :ANALYSIS:THRESHOLD:FESLOSS 1.23
```

Description After setting the value, wait for the AS flag in the value obtained with the :STATus:CONDition? command to become zero, and then obtain the event results.

:ANALysis:THReshold:FESPloss

Function Sets or queries the splitter loss threshold of the fault event.

Syntax
```
:ANALysis:THReshold:FESPloss <NRf>
:ANALysis:THReshold:FESPloss?
```
<NRf> = 1.0 to 30.0

Example
```
:ANALYSIS:THRESHOLD:FESPLOSS 1.0
:ANALYSIS:THRESHOLD:FESPLOSS?
-> :ANALYSIS:THRESHOLD:FESPLOSS 20.0
```

Description After setting the value, wait for the AS flag in the value obtained with the :STATus:CONDition? command to become zero, and then obtain the event results.

:ANALysis:THReshold:FETLoss

Function Sets or queries the total loss threshold of the fault event.

Syntax
```
:ANALysis:THReshold:FETLoss <NRf>
:ANALysis:THReshold:FETLoss?
```
<NRf> = 1 to 65

Example
```
:ANALYSIS:THRESHOLD:FETLOSS 1
:ANALYSIS:THRESHOLD:FETLOSS?
-> :ANALYSIS:THRESHOLD:FETLOSS 65
```

Description After setting the value, wait for the AS flag in the value obtained with the :STATus:CONDition? command to become zero, and then obtain the event results.

:ANALysis:THReshold:FEULoss

Function Sets or queries the dB/km threshold of the fault event.

Syntax
```
:ANALysis:THReshold:FEULoss <NRf>
:ANALysis:THReshold:FEULoss?
```
<NRf> = 0.01 to 9.99

Example
```
:ANALYSIS:THRESHOLD:FEULOSS 0.01
:ANALYSIS:THRESHOLD:FEULOSS?
-> :ANALYSIS:THRESHOLD:FEULOSS 1.23
```

Description After setting the value, wait for the AS flag in the value obtained with the :STATus:CONDition? command to become zero, and then obtain the event results.

:ANALysis:THReshold:RLOSs

Function Sets or queries the return loss threshold.
Syntax `:ANALysis:THReshold:RLOSs <NRf>`
`:ANALysis:THReshold:RLOSs?`
<NRf> = 20 to 70
Example `:ANALYSIS:THRESHOLD:RLOSS 50`
`:ANALYSIS:THRESHOLD:RLOSS?`
`-> :ANALYSIS:THRESHOLD:RLOSS 30`
Description After setting the value, wait for the AS flag in the value obtained with the :STATus:CONDition? command to become zero, and then obtain the event results.

:ANALysis:THReshold:SLOSs

Function Sets or queries the splice loss threshold.
Syntax `:ANALysis:THReshold:SLOSs <NRf>`
`:ANALysis:THReshold:SLOSs?`
<NRf> = 0.01 to 9.99
Example `:ANALYSIS:THRESHOLD:SLOSS 0.01`
`:ANALYSIS:THRESHOLD:SLOSS?`
`-> :ANALYSIS:THRESHOLD:SLOSS 1.23`
Description After setting the value, wait for the AS flag in the value obtained with the :STATus:CONDition? command to become zero, and then obtain the event results.

:ANALysis:THReshold:SPLOss

Function Sets or queries the splitter loss threshold.
Syntax `:ANALysis:THReshold:SPLOSs <NRf>`
`:ANALysis:THReshold:SPLOSs?`
<NRf> = 1.0 to 20.0
Example `:ANALYSIS:THRESHOLD:SPLOSS 1.0`
`:ANALYSIS:THRESHOLD:SPLOSS?`
`-> :ANALYSIS:THRESHOLD:SPLOSS 20.0`
Description After setting the value, wait for the AS flag in the value obtained with the :STATus:CONDition? command to become zero, and then obtain the event results.

:ANALysis:TOTal:DISTance?

Function Queries the total distance of the measurement result.
Syntax `:ANALysis:TOTal:DISTance?`
Example `:ANALYSIS:TOTAL:DISTANCE?`
`-> :ANALYSIS:TOTAL:`
`DISTANCE 49.490000E+03`
Description Use this command after executing an auto event detection using :ANALysis:ASEarch:EXECute.

:ANALysis:TOTal:LOSS?

Function Queries the total loss of the measurement result.
Syntax `:ANALysis:TOTal:LOSS?`
Example `:ANALYSIS:TOTAL:LOSS?`
`-> :ANALYSIS:TOTAL:LOSS 1.271000E+01`
Description Use this command after executing an auto event detection using :ANALysis:ASEarch:EXECute.

:ANALysis:TOTal:RETurnloss?

Function Queries the total return loss of the measurement result.
Syntax `:ANALysis:TOTal:RETurnloss?`
Example `:ANALYSIS:TOTAL:RETURNLOSS?`
`-> :ANALYSIS:TOTAL:`
`RETURNLOSS 63.22200E+00`
Description Use this command after executing an auto event detection using :ANALysis:ASEarch:EXECute.

:ANALysis:TOTal:UNITloss?

Function Queries the dB/km value for the total distance of the measured result.
Syntax `:ANALysis:TOTal:UNITloss?`
Example `:ANALYSIS:TOTAL:UNITLOSS?`
`-> :ANALYSIS:TOTAL:`
`UNITLOSS 17.417000E+00`
Description Use this command after executing an auto event detection using :ANALysis:ASEarch:EXECute.

:ANALysis:TOTLoss

Function Sets or queries the total loss mode.
Syntax `:ANALysis:TOTLoss {ACCUMULATE|TPA}`
`:ANALysis:TOTLoss?`
ACCUMULATE: Cumulate loss
TPA: Loss between two S/E points
Example `:ANALYSIS:TOTLOSS TPA`
`:ANALYSIS:TOTLOSS?`
`-> :ANALYSIS:TOTLOSS TPA`

:ANALysis:TOTRloss

Function Sets or queries the total return loss mode.
Syntax `:ANALysis:TOTRloss {INCLUDE_END|`
`EXCLUDE_END}`
`:ANALysis:TOTRloss?`
INCLUDE_END: END point included
EXCLUDE_END: END point not included
Example `:ANALYSIS:TOTRLOSS EXCLUDE_END`
`:ANALYSIS:TOTRLOSS?`
`->:ANALYSIS:TOTRLOSS EXCLUDE_END`

5.4 COMMunicate Group

The commands in this group deal with communications. There are no front panel keys that correspond to the commands in this group.

:COMMunicate?

Function Queries all communication settings.

Syntax
```
:COMMunicate?
```

Example
```
:COMMUNICATE?
-> :COMMUNICATE:HEADER 1;VERBOSE 0
```

:COMMunicate:HEADer

Function Sets or queries whether a header is added to the response to a query (example with header: "ACQuire:DRANge 5000"; example without header: "5000").

Syntax
```
:COMMunicate:HEADer <Boolean>
:COMMunicate:HEADer?
```

Example
```
:COMMUNICATE:HEADER ON
:COMMUNICATE:HEADER?
-> :COMMUNICATE:HEADER 1
```

:COMMunicate:VERBose

Function Sets or queries whether the response to a query is returned fully spelled out (e.g., ACQuire:DRANge 5000) or in its abbreviated form (e.g., ACQ:DRAN 5000).

Syntax
```
:COMMunicate:VERBose <Boolean>
:COMMunicate:VERBose?
```

Example
```
:COMMUNICATE:VERBOSE ON
:COMMUNICATE:VERBOSE?
-> :COMMUNICATE:VERBOSE 1
```

5.5 DISPlay Group

The commands in this group deal with the screen display.
You can make the same settings and queries that you can by using the front panel.

:DISPlay:ALINe

Function Sets or queries the approximated line display.
Syntax `:DISPlay:ALINe <Boolean>`
`:DISPlay:ALINe?`
Example `:DISPLAY:ALINE ON`
`:DISPLAY:ALINE? -> :DISPLAY:ALINE 1`

:DISPlay:COLor

Function Sets or queries the screen color.
Syntax `:DISPlay:COLor {COLOR1|COLOR2}`
`:DISPlay:COLor?`
COLOR1: Color 1 (Dark theme)
COLOR2: Color 2 (Light theme)
Example `:DISPLAY:COLOR COLOR1`
`:DISPLAY:COLOR?`
`-> :DISPLAY:COLOR COLOR1`

:DISPlay:CURSor:DBValue

Function Sets or queries the cursor dB value.
Syntax `:DISPlay:CURSor:DBValue <Boolean>`
`:DISPlay:CURSor:DBValue?`
Example `:DISPLAY:CURSOR:DBVALUE ON`
`:DISPLAY:CURSOR:DBVALUE?`
`-> :DISPLAY:CURSOR:DBVALUE 1`

:DISPlay:CURSor:SECond

Function Sets or queries the ghost cursor display.
Syntax `:DISPlay:CURSor:SECond <Boolean>`
`:DISPlay:CURSor:SECond?`
Example `:DISPLAY:CURSOR:SECOND ON`
`:DISPLAY:CURSOR:SECOND?`
`-> :DISPLAY:CURSOR:SECOND 1`

:DISPlay:CURSor:TYPE

Function Sets or queries the cursor display format.
Syntax `:DISPlay:CURSor:TYPE {CROSS|LINE}`
`:DISPlay:CURSor:TYPE?`
CROSS: CROSS (+)
LINE: LINE (¦)
Example `:DISPLAY:CURSOR:TYPE LINE`
`:DISPLAY:CURSOR:TYPE?`
`-> :DISPLAY:CURSOR:TYPE CROSS`

:DISPlay:DECibel:UPPer

Function Sets or queries the maximum value of the display's vertical scale.
Syntax `:DISPlay:DECibel:UPPer <NRf>`
`:DISPlay:DECibel:UPPer?`
<NRf> = 1.6 to 70 (dB)
Example `:DISPLAY:DECIBEL:UPPER 60`
`:DISPLAY:DECIBEL:UPPER?`
`-> :DISPLAY:DECIBEL:UPPER 50.0`
Description The range that can be specified varies depending on magnification of the vertical axis.

:DISPlay:DIGit:DECibel

Function Sets or queries the dB decimal places.
Syntax `:DISPlay:DIGit:DECibel {DIGIT1|DIGIT2|DIGIT3}`
`:DISPlay:DIGit:DECibel?`
DIGIT1: **.*
DIGIT2: **.**
DIGIT3: **.***
Example `:DISPLAY:DIGIT:DECIBEL DIGIT3`
`:DISPLAY:DIGIT:DECIBEL?`
`-> :DISPLAY:DIGIT:DECIBEL DIGIT3`

:DISPlay:DIGit:DISTance

Function Sets or queries the distance decimal places.
Syntax `:DISPlay:DIGit:DISTance {DIGIT3|DIGIT4|DIGIT5}`
`:DISPlay:DIGit:DISTance?`
DIGIT3: **.***
DIGIT4: **.****
DIGIT5: **.*****
Example `:DISPLAY:DIGIT:DISTANCE DIGIT5`
`:DISPLAY:DIGIT:DISTANCE?`
`-> :DISPLAY:DIGIT:DISTANCE DIGIT5`

:DISPlay:DISTance:LEFT

Function Sets or queries the minimum value of the display's horizontal scale.
Syntax `:DISPlay:DISTance:LEFT <NRf>`
`:DISPlay:DISTance:LEFT?`
Example `:DISPLAY:DISTANCE:LEFT 1000`
`:DISPLAY:DISTANCE:LEFT?`
`-> :DISPLAY:DISTANCE:LEFT 498.97000E+00`
Description The range that can be specified varies depending on magnification of the horizontal axis.

:DISPlay:DIVide:DECibel

Function Sets or queries the magnification of the vertical axis.

Syntax
```
:DISPlay:DIVide:DECibel <NRf>
:DISPlay:DIVide:DECibel?
```
<NRf> = 0.2 to 7.5 (0.2, 0.5, 1.0, 2.0, 5.0, 7.5) (dB)

Example
```
:DISPLAY:DIVIDE:DECIBEL 1.0
:DISPLAY:DIVIDE:DECIBEL?
-> :DISPLAY:DIVIDED:DECIBEL 1.0
```

Description The unit is dB (decibels).

:DISPlay:DIVide:DISTance

Function Sets or queries the magnification of the horizontal axis.

Syntax
```
:DISPlay:DIVide:DISTance <NRf>
:DISPlay:DIVide:DISTance?
```
<NRf> = 1 m to 40 km

Example
```
:DISPLAY:DIVIDE:DISTANCE 2000
:DISPLAY:DIVIDE:DISTANCE?
-> :DISPLAY:DIVIDE:DISTNCE 10.0E+03
```

Description The range that can be specified varies depending on the distance range.

100 k range: 50 m to 10 km
50 k range: 20 m to 5 km
20 k range: 10 m to 2 km
10 k range: 5 m to 1 km
5 k range: 5 m to 500 m
2 k range: 2 m to 200 m
1 k range: 1 m to 100 m
500 m range: 1 m to 50 m

:DISPlay:INITialize:SCALe

Function Initializes the display scales.

Syntax
```
:DISPlay:INITialize:SCale
```

Example
```
:DISPLAY:INITIALIZE:SCALE
```

:DISPlay:MARKer:INFormation

Function Sets or queries the marker information display.

Syntax
```
:DISPlay:MARKer:INFomation <Boolean>
:DISPlay:MARKer:INFomation?
```

Example
```
:DISPLAY:MARKER:INFOMATION ON
:DISPLAY:MARKER:INFOMATION?
-> :DISPLAY:MARKER:INFOMATION 1
```

5.6 FIPRobe Group

The commands in this group deal with the pass/fail judgment of fiber end faces. You can perform the same operations and make the same settings and queries that you can from the front panel. This function is valid when the /FST option is installed.

:FIPRobe:PASSfail:CLADding:DEF1:ENABle

Function Sets or queries whether defects that are 2 µm or less in length are to be judged in the cladding area of the fiber inspection probe.

Syntax
```
:FIPRobe:PASSfail:CLADding:DEF1:ENABle <Boolean>
:FIPRobe:PASSfail:CLADding:DEF1:ENABle?
```

Example
```
:FIPROBE:PASSFAIL:CLADDING:DEF1:ENABLE ON
:FIPROBE:PASSFAIL:CLADDING:DEF1:ENABLE?
-> :FIPROBE:PASSFAIL:CLADDING:DEF1:ENABLE 1
```

:FIPRobe:PASSfail:CLADding:DEF1:THReshold

Function Sets or queries the threshold for judging defects that are 2 µm or less in length in the cladding area of the fiber inspection probe.

Syntax
```
:FIPRobe:PASSfail:CLADding:DEF1:THReshold <NRf>
:FIPRobe:PASSfail:CLADding:DEF1:THReshold?
```
<NRf> = 0 to 100

Example
```
:FIPROBE:PASSFAIL:CLADDING:DEF1:THRESHOLD 0
:FIPROBE:PASSFAIL:CLADDING:DEF1:THRESHOLD?
-> :FIPROBE:PASSFAIL:CLADDING:DEF1:THRESHOLD 0
```

:FIPRobe:PASSfail:CLADding:DEF2:ENABle

Function Sets or queries whether defects that are 2 µm to 5 µm in length are to be judged in the cladding area of the fiber inspection probe.

Syntax
```
:FIPRobe:PASSfail:CLADding:DEF2:ENABle <Boolean>
:FIPRobe:PASSfail:CLADding:DEF2:ENABle?
```

Example
```
:FIPROBE:PASSFAIL:CLADDING:DEF2:ENABLE ON
:FIPROBE:PASSFAIL:CLADDING:DEF2:ENABLE?
-> :FIPROBE:PASSFAIL:CLADDING:DEF2:ENABLE 1
```

:FIPRobe:PASSfail:CLADding:DEF2:THReshold

Function Sets or queries the threshold for judging defects that are 2 µm to 5 µm or less in length in the cladding area of the fiber inspection probe.

Syntax
```
:FIPRobe:PASSfail:CLADding:DEF2:THReshold <NRf>
:FIPRobe:PASSfail:CLADding:DEF2:THReshold?
```
<NRf> = 0 to 100

Example
```
:FIPROBE:PASSFAIL:CLADDING:DEF2:THRESHOLD 0
:FIPROBE:PASSFAIL:CLADDING:DEF2:THRESHOLD?
-> :FIPROBE:PASSFAIL:CLADDING:DEF2:THRESHOLD 0
```

:FIPRobe:PASSfail:CLADding:DEF3:ENABle

Function Sets or queries whether defects that are longer than 5 µm in length are to be judged in the cladding area of the fiber inspection probe.

Syntax
```
:FIPRobe:PASSfail:CLADding:DEF3:ENABle <Boolean>
:FIPRobe:PASSfail:CLADding:DEF3:ENABle?
```

Example
```
:FIPROBE:PASSFAIL:CLADDING:DEF3:ENABLE ON
:FIPROBE:PASSFAIL:CLADDING:DEF3:ENABLE?
-> :FIPROBE:PASSFAIL:CLADDING:DEF3:ENABLE 1
```

:FIPRobe:PASSfail:CLADding:DEF3:THReshold

Function Sets or queries the threshold for judging defects that are longer than 5 µm in length in the cladding area of the fiber inspection probe.

Syntax
```
:FIPRobe:PASSfail:CLADding:DEF3:THReshold <NRf>
:FIPRobe:PASSfail:CLADding:DEF3:THReshold?
```
<NRf> = 0 to 100

Example
```
:FIPROBE:PASSFAIL:CLADDING:DEF3:THRESHOLD 0
:FIPROBE:PASSFAIL:CLADDING:DEF3:THRESHOLD?
-> :FIPROBE:PASSFAIL:CLADDING:DEF3:THRESHOLD 0
```

:FIPRobe:PASSfail:CLADding:SCR1:ENABle

Function Sets or queries whether scratches that are 3 µm or less in length are to be judged in the cladding area of the fiber inspection probe.

Syntax
```
:FIPRobe:PASSfail:CLADding:SCR1:
ENABle <Boolean>
:FIPRobe:PASSfail:CLADding:SCR1:
ENABle?
```

Example
```
:FIPROBE:PASSFAIL:CLADDING:SCR1:
ENABLE ON
:FIPROBE:PASSFAIL:CLADDING:SCR1:
ENABLE?
-> :FIPROBE:PASSFAIL:CLADDING:SCR1:
  ENABLE 1
```

:FIPRobe:PASSfail:CLADding:SCR1:THReshold

Function Sets or queries the threshold for judging scratches that are 3 µm or less in length in the cladding area of the fiber inspection probe.

Syntax
```
:FIPRobe:PASSfail:CLADding:SCR1:
THReshold <NRf>
:FIPRobe:PASSfail:CLADding:SCR1:
THReshold?
```
<NRf> = 0 to 100

Example
```
:FIPROBE:PASSFAIL:CLADDING:SCR1:
THRESHOLD 0
:FIPROBE:PASSFAIL:CLADDING:SCR1:
THRESHOLD?
-> :FIPROBE:PASSFAIL:CLADDING:SCR1:
  THRESHOLD 0
```

:FIPRobe:PASSfail:CLADding:SCR2:ENABle

Function Sets or queries whether scratches that are longer than 3 µm in length are to be judged in the cladding area of the fiber inspection probe.

Syntax
```
:FIPRobe:PASSfail:CLADding:SCR2:
ENABle <Boolean>
:FIPRobe:PASSfail:CLADding:SCR2:ENAB
le?
```

Example
```
:FIPROBE:PASSFAIL:CLADDING:SCR2:
ENABLE ON
:FIPROBE:PASSFAIL:CLADDING:SCR2:
ENABLE?
-> :FIPROBE:PASSFAIL:CLADDING:SCR2:
  ENABLE 1
```

:FIPRobe:PASSfail:CLADding:SCR2:THReshold

Function Sets or queries the threshold for judging scratches that are longer than 3 µm in length in the cladding area of the fiber inspection probe.

Syntax
```
:FIPRobe:PASSfail:CLADding:SCR2:
THReshold <NRf>
:FIPRobe:PASSfail:CLADding:SCR2:
THReshold?
```
<NRf> = 0 to 100

Example
```
:FIPROBE:PASSFAIL:CLADDING:SCR2:
THRESHOLD 0
:FIPROBE:PASSFAIL:CLADDING:SCR2:
THRESHOLD?
-> :FIPROBE:PASSFAIL:CLADDING:SCR2:
  THRESHOLD 0
```

:FIPRobe:PASSfail:CONTact:DEF:ENABle

Function Sets or queries whether defects that are longer than 10 µm in length are to be judged in the contact area of the fiber inspection probe.

Syntax
```
:FIPRobe:PASSfail:CONTact:DEF:
ENABle <Boolean>
:FIPRobe:PASSfail:CONTact:DEF:
ENABle?
```

Example
```
:FIPROBE:PASSFAIL:CONTACT:DEF:
ENABLE ON
:FIPROBE:PASSFAIL:CONTACT:DEF:ENABLE?
-> :FIPROBE:PASSFAIL:CONTACT:DEF:
  ENABLE 1
```

:FIPRobe:PASSfail:CONTact:DEF:THReshold

Function Sets or queries the threshold for judging defects that are longer than 10 µm in length in the contact area of the fiber inspection probe.

Syntax
```
:FIPRobe:PASSfail:CONTact:DEF:
THReshold <NRf>
:FIPRobe:PASSfail:CONTact:DEF:
THReshold?
```
<NRf> = 0 to 100

Example
```
:FIPROBE:PASSFAIL:CONTACT:DEF:
THRESHOLD 0
:FIPROBE:PASSFAIL:CONTACT:DEF:
THRESHOLD?
-> :FIPROBE:PASSFAIL:CONTACT:DEF:
  THRESHOLD 0
```

:FIPRobe:PASSfail:CONTact:SCR:ENABle

Function Sets or queries whether scratches are to be judged in the contact area of the fiber inspection probe.

Syntax
```
:FIPRobe:PASSfail:CONTact:SCR:
ENABle <Boolean>
:FIPRobe:PASSfail:CONTact:SCR:
ENABle?
```

Example
```
:FIPROBE:PASSFAIL:CONTACT:SCR:
ENABLE ON
:FIPROBE:PASSFAIL:CONTACT:SCR:ENABLE?
-> :FIPROBE:PASSFAIL:CONTACT:SCR:
  ENABLE 1
```

:FIPRobe:PASSfail:CONTact:SCR:THReshold

Function Sets or queries the threshold for judging scratches in the contact area of the fiber inspection probe.

Syntax
```
:FIPRobe:PASSfail:CONTact:SCR:
THReshold <NRf>
:FIPRobe:PASSfail:CONTact:SCR:
THReshold?
```
<NRf> = 0 to 100

Example
```
:FIPROBE:PASSFAIL:CONTACT:SCR:
THRESHOLD 0
:FIPROBE:PASSFAIL:CONTACT:SCR:
THRESHOLD?
-> :FIPROBE:PASSFAIL:CONTACT:SCR:
  THRESHOLD 0
```

:FIPRobe:PASSfail:CORE:DEF1:ENABle

Function Sets or queries whether defects that are 3 µm or less in length are to be judged in the core area of the fiber inspection probe.

Syntax
```
:FIPRobe:PASSfail:CORE:DEF1:
ENABle <Boolean>
:FIPRobe:PASSfail:CORE:DEF1:ENABle?
```

Example
```
:FIPROBE:PASSFAIL:CORE:DEF1:ENABLE ON
:FIPROBE:PASSFAIL:CORE:DEF1:ENABLE?
-> :FIPROBE:PASSFAIL:CORE:DEF1:
  ENABLE 1
```

:FIPRobe:PASSfail:CORE:DEF1:THReshold

Function Sets or queries the threshold for judging defects that are 3 µm or less in length in the core area of the fiber inspection probe.

Syntax
```
:FIPRobe:PASSfail:CORE:DEF1:
THReshold <NRf>
:FIPRobe:PASSfail:CORE:DEF1:
THReshold?
```
<NRf> = 0 to 100

Example
```
:FIPROBE:PASSFAIL:CORE:DEF1:
THRESHOLD 0
:FIPROBE:PASSFAIL:CORE:DEF1:THRESHOLD?
 -> :FIPROBE:PASSFAIL:CORE:DEF1:
  THRESHOLD 0
```

:FIPRobe:PASSfail:CORE:DEF2:ENABle

Function Sets or queries whether defects that are longer than 3 µm in length are to be judged in the core area of the fiber inspection probe.

Syntax
```
:FIPRobe:PASSfail:CORE:DEF2:
ENABle <Boolean>
:FIPRobe:PASSfail:CORE:DEF2:ENABle?
```

Example
```
:FIPROBE:PASSFAIL:CORE:DEF2:ENABLE ON
:FIPROBE:PASSFAIL:CORE:DEF2:ENABLE?
-> :FIPROBE:PASSFAIL:CORE:DEF2:
  ENABLE 1
```

:FIPRobe:PASSfail:CORE:DEF2:THReshold

Function Sets or queries the threshold for judging defects that are longer than 3 µm in length in the core area of the fiber inspection probe.

Syntax
```
:FIPRobe:PASSfail:CORE:DEF2:
THReshold <NRf>
:FIPRobe:PASSfail:CORE:DEF2:
THReshold?
```
<NRf> = 0 to 100

Example
```
:FIPROBE:PASSFAIL:CORE:DEF2:
THRESHOLD 0
:FIPROBE:PASSFAIL:CORE:DEF2:THRESHOLD?
 -> :FIPROBE:PASSFAIL:CORE:DEF2:
  THRESHOLD 0
```

:FIPRobe:PASSfail:CORE:DEFAny:ENABle

Function Sets or queries whether defects are to be judged in the core area of the fiber inspection probe.

Syntax
```
:FIPRobe:PASSfail:CORE:DEFAny:
ENABle <Boolean>
:FIPRobe:PASSfail:CORE:DEFAny:ENABle?
```

Example
```
:FIPROBE:PASSFAIL:CORE:DEFANY:
ENABLE ON
:FIPROBE:PASSFAIL:CORE:DEFANY:ENABLE?
-> :FIPROBE:PASSFAIL:CORE:DEFANY:
  ENABLE 1
```

:FIPRobe:PASSfail:CORE:DEFAny:THReshold

Function Sets or queries the threshold for judging defects in the core area of the fiber inspection probe.

Syntax
```
:FIPRobe:PASSfail:CORE:DEFAny:
THReshold <NRf>
:FIPRobe:PASSfail:CORE:DEFAny:
THReshold?
```
<NRf> = 0 to 100

Example
```
:FIPROBE:PASSFAIL:CORE:DEFANY:
THRESHOLD 0
:FIPROBE:PASSFAIL:CORE:DEFANY:
THRESHOLD?
-> :FIPROBE:PASSFAIL:CORE:DEFANY:
  THRESHOLD 0
```

:FIPRobe:PASSfail:CORE:SCR1:ENABle

Function Sets or queries whether scratches that are 3 µm or less in length are to be judged in the core area of the fiber inspection probe.

Syntax
```
:FIPRobe:PASSfail:CORE:SCR1:
ENABle <Boolean>
:FIPRobe:PASSfail:CORE:SCR1:ENABle?
```

Example
```
:FIPROBE:PASSFAIL:CORE:SCR1:ENABLE ON
:FIPROBE:PASSFAIL:CORE:SCR1:ENABLE?
-> :FIPROBE:PASSFAIL:CORE:SCR1:
  ENABLE  1
```

:FIPRobe:PASSfail:CORE:SCR1:THReshold

Function Sets or queries the threshold for judging scratches that are 3 µm or less in length in the core area of the fiber inspection probe.

Syntax
```
:FIPRobe:PASSfail:CORE:SCR1:
THReshold <NRf>
:FIPRobe:PASSfail:CORE:SCR1:
THReshold?
```
<NRf> = 0 to 100

Example
```
:FIPROBE:PASSFAIL:CORE:SCR1:
THRESHOLD 0
:FIPROBE:PASSFAIL:CORE:SCR1:
THRESHOLD?
-> :FIPROBE:PASSFAIL:CORE:SCR1:
  THRESHOLD 0
```

:FIPRobe:PASSfail:CORE:SCR2:ENABle

Function Sets or queries whether scratches that are longer than 3 µm in length are to be judged in the core area of the fiber inspection probe.

Syntax
```
:FIPRobe:PASSfail:CORE:SCR2:
ENABle <Boolean>
:FIPRobe:PASSfail:CORE:SCR2:ENABle?
```

Example
```
:FIPROBE:PASSFAIL:CORE:SCR2:ENABLE ON
:FIPROBE:PASSFAIL:CORE:SCR2:ENABLE?
-> :FIPROBE:PASSFAIL:CORE:SCR2:
  ENABLE 1
```

:FIPRobe:PASSfail:CORE:SCR2:THReshold

Function Sets or queries the threshold for judging scratches that are longer than 3 µm in length in the core area of the fiber inspection probe.

Syntax
```
:FIPRobe:PASSfail:CORE:SCR2:
THReshold <NRf>
:FIPRobe:PASSfail:CORE:SCR2:
THReshold?
```
<NRf> = 0 to 100

Example
```
:FIPROBE:PASSFAIL:CORE:SCR2:
THRESHOLD 0
:FIPROBE:PASSFAIL:CORE:SCR2:
THRESHOLD?
-> :FIPROBE:PASSFAIL:CORE:SCR2:
  THRESHOLD 0
```

:FIPRobe:PASSfail:CORE:SCRAny:ENABle

Function Sets or queries whether scratches are to be judged in the core area of the fiber inspection probe.

Syntax
```
:FIPRobe:PASSfail:CORE:SCRAny:
ENABle <Boolean>
:FIPRobe:PASSfail:CORE:SCRAny?
```

Example
```
:FIPROBE:PASSFAIL:CORE:SCRANY:
ENABLE ON
:FIPROBE:PASSFAIL:CORE:SCRANY:ENABLE?
-> :FIPROBE:PASSFAIL:CORE:SCRANY:
  ENABLE 1
```

:FIPRobe:PASSfail:CORE:SCRAny:THReshold

Function Sets or queries the threshold for judging scratches in the core area of the fiber inspection probe.

Syntax
```
:FIPRobe:PASSfail:CORE:SCRAny:
THReshold <NRf>
:FIPRobe:PASSfail:CORE:SCRAny:
THReshold?
```
<NRf> = 0 to 100

Example
```
:FIPROBE:PASSFAIL:CORE:SCRANY:
THRESHOLD 0
:FIPROBE:PASSFAIL:CORE:SCRANY:
THRESHOLD?
-> :FIPROBE:PASSFAIL:CORE:SCRANY:
  THRESHOLD 0
```

:FIPRobe:PASSfail:EXECute

Function Executes pass/fail judgment of the fiber inspection probe.

Syntax
```
:FIPRobe:PASSfail:EXECute
```

Example
```
:FIPROBE:PASSFAIL:EXECUTE
```

:FIPRobe:PASSfail:FIBertype

Function Sets or queries the fiber type of the fiber inspection probe.

Syntax
```
:FIPRobe:PASSfail:FIBertype {SM|MM}
:FIPRobe:PASSfail:FIBertype?
```
SM: Single-mode fiber
MM: Multi-mode fiber

Example
```
:FIPROBE:PASSFAIL:FIBERTYPE SM
:FIPROBE:PASSFAIL:FIBERTYPE?
-> :FIPROBE:PASSFAIL:FIBERTYPE SM
```

:FIPRobe:PASSfail:STANdard

Function Sets or queries the standard that is used for fiber inspection probe pass/fail judgment.

Syntax
```
:FIPRobe:PASSfail:STANdard {SPC|UPC|
APC|USER}
```
SPC: Spherical surface polishing
UPC: Spherical surface polishing
APC: Angled spherical surface polishing
USER: Manual execution of judgment and threshold assignment

Example
```
:FIPROBE:PASSFAIL:STANDARD SPC
:FIPROBE:PASSFAIL:STANDARD?
-> :FIPROBE:PASSFAIL:STANDARD SPC
```

:FIPRobe:STATe

Function Queries the image type of the fiber inspection probe.

Syntax
```
:FIPRobe:STATe {VIDEO|STILL}
```
VIDEO : Video
STILL : Still picture

Example
```
:FIPROBE:STATE VIDEO
:FIPROBE:STATE?
-> :FIPROBE:STATE VIDEO
```

5.7 FSYStem Group

The commands in this group deal with file saving and operations.
The commands provide for file operations with consistent absolute paths throughout the system. They provide flexible file access by absolute paths.

:FSYStem:DATE?

Function Gets the date of a file.

Syntax `:FSYStem:DATE?`

Example
1. `:FSYSTEM:PARAM:PATH "/INTERNAL/AAA/BBB"`
2. `:FSYSTEM:PARAM:NAME "xxx.SOR"`
3. `:FSYSTEM:DATE?`
   `-> :FSYSTEM:DATE "2024/11/06 13:10"`

Result:
Gets the creation date of /INTERNAL/AAA/BBB/xxx.SOR.

Description Use this command after setting the path with the following command.
:FSYStem:PARAm:PATH
:FSYStem:PARAm:NAME

:FSYStem:EXISt?

Function Queries whether the specified folder or file exists.

Syntax `:FSYStem:EXISt?`
1: The specified folder or file exists.
0: The specified folder or file does not exist.

Example **Case 1: Specifying a folder**
1. `:FSYSTEM:PARAM:PATH "/INTERNAL/AAA/BBB"`
2. `:FSYSTEM:PARAM:NAME "CCC"`
3. `:FSYSTEM:EXIST?`
   `->:FSYSTEM:EXIST 1`

Result:
Returns 1 if the /INTERNAL/AAA/BBB/CCC folder exists, 0 otherwise.

Case 2: Specifying a file
1. `:FSYSTEM:PARAM:PATH "/INTERNAL/AAA/BBB"`
2. `:FSYSTEM:PARAM:NAME "xxx.SOR"`
3. `:FSYSTEM:EXIST? ->:FSYSTEM:EXIST 1`

Result:
Returns 1 if the /INTERNAL/AAA/BBB/xxx.SOR file exists, 0 otherwise.

Description Use this command after setting the path with the following command.
:FSYStem:PARAm:PATH
:FSYStem:PARAm:NAME

:FSYStem:FILE:COPY

Function Copies a file.

Syntax `:FSYStem:FOLDer:COPY`

Example
1. `:FSYSTEM:PARAM:PATH "/INTERNAL/AAA/BBB"`
2. `:FSYSTEM:PARAM:NAME "xxx.SOR"`
3. `:FSYSTEM:PARAM:DESTINATION:PATH "/INTERNAL/DDD/EEE"`
4. `:FSYSTEM:FILE:COPY`

Result:
/INTERNAL/AAA/BBB/xxx.SOR is copied to the /INTERNAL/DDD/EEE folder, and placed as /INTERNAL/DDD/EEE/xxx.SOR.

Description
- Use this command after setting the path with the following command.
  :FSYStem:PARAm:PATH
  :FSYStem:PARAm:NAME
  :FSYStem:PARAm:DESTination:PATH
- If a file with the same name exists in the copy destination, it is overwritten.

:FSYStem:FILE:DELete

Function Deletes a file.

Syntax `:FSYStem:FILE:DELete`

Example
1. `:FSYSTEM:PARAM:PATH "/INTERNAL/AAA/BBB"`
2. `:FSYSTEM:PARAM:NAME "xxx.SOR"`
3. `:FSYSTEM:FILE:DELETE`

Result:
xxx.SOR is deleted from /INTERNAL/AAA/BBB/.

Description Use this command after setting the path with the following command.
:FSYStem:PARAm:PATH
:FSYStem:PARAm:NAME

:FSYStem:FILE:GET?

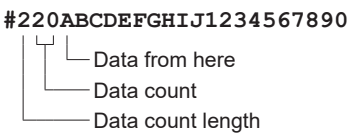
Function Gets a file from the OTDR. (file reception)

Syntax `:FSYStem:FILE:GET?`

Example
1. `:FSYSTEM:PARAM:PATH "/INTERNAL/AAA/BBB"`
2. `:FSYSTEM:PARAM:NAME "xxx.SOR"`
3. `:FSYSTEM:FILE:GET?`
   `-> #220ABCDEFGHIJ1234567890`

Result:
Gets /INTERNAL/AAA/BBB/xxx.SOR in block data format.

Description
- Use this command after setting the path with the following command.
  :FSYStem:PARAm:PATH
  :FSYStem:PARAm:NAME
- The specified file is sent in block data format.



:FSYStem:FILE:MOVE

Function Moves a file.

Syntax `:FSYStem:FILE:MOVE`

Example
1. `:FSYTEM:PARAM:PATH "/INTERNAL/AAA/BBB"`
2. `:FSYSTEM:PARAM:NAME "xxx.SOR"`
3. `:FSYSTEM:PARAM:DESTINATION:PATH "/INTERNAL/DDD/EEE"`
4. `:FSYSTEM:FILE:MOVE`

Result:
/INTERNAL/AAA/BBB/xxx.SOR is moved to the /INTERNAL/DDD/EEE folder, and placed as /INTERNAL/DDD/EEE/xxx.SOR.

Description
- Use this command after setting the path with the following command.
  :FSYStem:PARAm:PATH
  :FSYStem:PARAm:NAME
  :FSYStem:PARAm:DESTination:PATH
- If a file with the same name exists in the move destination, it is overwritten.

:FSYStem:FILE:RENAme

Function Renames a file.

Syntax `:FSYStem:FILE:RENAme <String>`
<String> = New file name

Example
1. `:FSYSTEM:PARAM:PATH "/INTERNAL/AAA/BBB"`
2. `:FSYSTEM:PARAM:NAME "xxx.SOR"`
3. `:FSYSTEM:FILE:RENAME "yyy.SOR"`

Result:
The name of xxx.SOR in /INTERNAL/AAA/BBB/xxx.SOR is changed to yyy.SOR and becomes /INTERNAL/AAA/BBB/yyy.SOR.

Description
- Use this command after setting the path with the following command.
  :FSYStem:PARAm:PATH
  :FSYStem:PARAm:NAME
- Enter the new file name in the parameter.
- If a file with the same name exits, renaming will fail.

:FSYStem:FILE:SEND

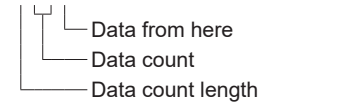
Function Sends a file to the OTDR.

Syntax `:FSYStem:FILE:SEND <Block data>`

Example
1. `:FSYSTEM:PARAM:PATH "/INTERNAL/AAA/BBB"`
2. `:FSYSTEM:PARAM:NAME "xxx.SOR"`
3. `:FSYSTEM:FILE:SEND #220ABCDEFGHIJ1234567890`

Result:
The /INTERNAL/AAA/BBB/xxx.SOR file is sent to the OTDR.

Description
- Use this command after setting the path with the following command.
  :FSYStem:PARAm:PATH
  :FSYStem:PARAm:NAME
- Enter the binary data of the file you want to send to the OTDR in block data format.



- If the folder specified by :FSYStem:PARAm:PATH does not exist, it is created automatically.

:FSYStem:FOLDer:COPY

Function Copies a folder.

Syntax :FSYStem:FOLDer:COPY

Example
1. :FSYSTEM:PARAM:PATH "/INTERNAL/AAA/BBB"
2. :FSYSTEM:PARAM:NAME "CCC"
3. :FSYTEM:PARAM:DESTination:PATH "/INTERNAL/DDD/EEE"
4. :FSYSTEM:FOLDER:COPY

Result:
/INTERNAL/AAA/BBB/CCC is copied to the /INTERNAL/DDD/EEE folder and placed as /INTERNAL/DDD/EEE/CCC.

Description
- Use this command after setting the path with the following command.
  :FSYStem:PARAm:PATH
  :FSYStem:PARAm:NAME
  :FSYStem:PARAm:DESTination:PATH
- All folders and files in the copy source folder are copied.
- If a file with the same name exists in the copy destination, it is overwritten.

:FSYStem:FOLDer:DELete

Function Deletes a folder.

Syntax :FSYStem:FOLDer:DELete

Example
1. :FSYSTEM:PARAM:PATH "/INTERNAL/AAA/BBB"
2. :FSYSTEM:PARAM:NAME "CCC"
3. :FSYSTEM:FOLDER:DELete

Result:
The /INTERNAL/AAA/BBB/CCC folder is deleted.

Description Use this command after setting the path with the following command.
:FSYStem:PARAm:PATH
:FSYStem:PARAm:NAME

:FSYStem:FOLDer:MAKE

Function Makes a folder.

Syntax :FSYStem:FOLDer:MAKE

Example
1. :FSYSTEM:PARAM:PATH "/INTERNAL/AAA/BBB"
2. FSYSTEM:PARAM:NAME "CCC"
3. :FSYSTEM:FOLDER:MAKE

Result:
The /INTERNAL/AAA/BBB/CCC folder is created. If the /INTERNAL/AAA and /INTERNAL/AAA/BBB folders do not exist, they are created automatically.

Description
- Use this command after setting the path with the following command.
  :FSYStem:PARAm:PATH
  :FSYStem:PARAm:NAME
- If the folder specified by :FSYStem:PARAm:PATH does not exist, it is created automatically.

:FSYStem:FOLDer:MOVE

Function Moves a folder.

Syntax :FSYStem:FOLDer:MOVE

Example
1. :FSYSTEM:PARAM:PATH "/INTERNAL/AAA/BBB"
2. :FSYSTEM:PARAM:NAME "CCC"
3. :FSYSTEM:PARAM:DESTination:PATH "/INTERNAL/DDD/EEE"
4. :FSYStem:FOLDer:MOVE

Result:
/INTERNAL/AAA/BBB/CCC is moved to the /INTERNAL/DDD/EEE folder, and placed as /INTERNAL/DDD/EEE/CCC.

Description
- Use this command after setting the path with the following command.
  :FSYStem:PARAm:PATH
  :FSYStem:PARAm:NAME
  :FSYStem:PARAm:DESTination:PATH
- All folders and files in the move source folder are moved.
- If a file with the same name exists in the move destination, it is overwritten.

:FSYStem:FOLDer:RENAme

Function Renames a folder.

Syntax `:FSYStem:FOLDer:RENAme <String>`
<String> = New folder name

Example
1. `:FSYSTEM:PARAM:PATH "/INTERNAL/AAA/BBB"`
2. `:FSYSTEM:PARAM:NAME "CCC"`
3. `:FSYSTEM:FOLDER:RENAME "FFF"`

Result:
The name of CCC in /INTERNAL/AAA/BBB/CCC is changed to FFF and becomes /INTERNAL/AAA/BBB/FFF.

Description
- Use this command after setting the path with the following command.
  :FSYStem:PARAm:PATH
  :FSYStem:PARAm:NAME
- Enter the new folder name in the parameter.
- If a folder with the same name exits, renaming will fail.

:FSYStem:FREE?

Function Gets the free space on the drive.

Syntax `:FSYStem:FREE?`

Example
1. `:FSYSTEM:PARAM:PATH "/INTERNAL/AAA/BBB/CCC"`
2. `:FSYSTEM:FREE?`
   `-> :FSYSTEM:FREE 190.709E+06`

Result:
Gets the free space on the /INTERNAL drive. (190.709MB)

Description
- Use this command after setting the path with the following command.
  :FSYStem:PARAm:PATH
- The free space in the drive belonging to the path specified by :FSYStem:PARAm:PATH is displayed.
  The unit is bytes.
  Example:
  `"/INTERNAL"` -> Size of /INTERNAL
  `"/INTERNAL/AAA/BBB/CCC"`
  -> Size of /INTERNAL
  `"/INTERNAL/AAA/BBB/xxx.SOR"`
  -> Size of /INTERNAL
  `"USB1/DDD/EEE"`
  -> Size of USB memory 1

:FSYStem:LIST?

Function Gets a list of the folders and files in the specified path.

Syntax `:FSYStem:LIST?`

Example
1. `:FSYSTEM:PARAM:PATH "/INTERNAL/AAA/BBB/CCC"`
2. `:FSYSTEM:LIST?`
   `-> :FSYS:LIST 5,EEE/,FFF/,aaa.SOR,bbb.SOR,ccc.BMP`

Result:
Gets a list of folders and files in /INTERNAL/AAA/BBB/CCC as a string.

Description
- Use this command after setting the path with the following command.
  :FSYStem:PARAm:PATH
- The information is returned as follows: number of items, file (or folder), . . . , file (or folder). Folders have a slash at the end.
- Returns -1 if a file is specified by :FSYStem:PARAm:PATH.

:FSYStem:LOAD

Function Loads a file stored in the OTDR into the OTDR.

Syntax `:FSYStem:LOAD`

Example
1. `:FSYSTEM:PARAM:PATH "/INTERNAL/AAA/BBB"`
2. `:FSYSTEM:PARAM:NAME "xxx.SOR"`
3. `:FSYSTEM:LOAD`

Result:
Replays /INTERNAL/AAA/BBB/xxx.SOR.

Description Use this command after setting the path with the following command.
:FSYStem:PARAm:PATH
:FSYStem:PARAm:NAME

:FSYStem:PARAm:DESTination:PATH

Function Specifies the destination folder for file operations with an absolute path.

Syntax
`:FSYStem:PARAm:DESTination:PATH <String>`
`:FSYStem:PARAm:DESTination:PATH?`

Example
`:FSYSTEM:PARAM:DESTINATION:PATH "/INTERNAL/DDD/EEE"`
`:FSYSTEM:PARAM:DESTINATION:PATH?`
`-> :FSYSTEM:PARAM:DESTINATION:PATH "/INTERNAL/DDD/EEE"`

Description
- Specify with an absolute path that includes the drive name.
- Paths must be delimited and prefixed with slashes.
- Identifier of each drive
  /INTERNAL: internal memory
  /SD: External microSD memory card
  /USB1: USB memory 1
  /USB2: USB memory 2

:FSYStem:PARAm:NAME

Function Specify the name of the target folder or file for file operations.
Include the extension for file names.

Syntax
```
:FSYStem:PARAm:NAME <String>
:FSYStem:PARAm:NAME?
```

Example **Case 1: Specifying a folder**
```
:FSYSTEM:PARAM:NAME "CCC"
:FSYSTEM:PARAM:NAME?
-> :FSYSTEM:PARAM:NAME "CCC"
```
Case 2: Specifying a file
```
:FSYSTEM:PARAM:NAME "xxx.SOR"
:FSYSTEM:PARAM:NAME?
-> :FSYSTEM:PARAM:NAME "xxx.SOR"
```

Description Specify a folder or file name.
Include the extension for file names.

:FSYStem:PARAm:PATH

Function Specifies the folder path containing the target folder or file for file operations with an absolute path.

Syntax
```
:FSYStem:PARAm:PATH <String>
:FSYStem:PARAm:PATH?
```

Example
```
:FSYSTEM:PARAM:PATH "/INTERNAL/AAA/
BBB"
:FSYSTEM:PARAM:PATH?
-> :FSYSTEM:PARAM:PATH "/INTERNAL/
AAA/BBB"
```

Description
- Specify with an absolute path that includes the drive name.
- Paths must be delimited and prefixed with slashes.
- Identifier of each drive

  /INTERNAL: internal memory
  /SD: External microSD memory card
  /USB1: USB memory 1
  /USB2: USB memory 2

:FSYStem:SAVE

Function Saves the current waveform data (SOR, CSV, BMP, PDF, etc.) to the OTDR.

Syntax
```
:FSYStem:SAVE
```

Example
1. `:FSYSTEM:PARAM:PATH "/INTERNAL/AAA/BBB/CCC"`
2. `:FSYSTEM:PARAM:NAME "1550 _ 0123.BMP"`
3. `:FSYSTEM:SAVE`

Result:
A BMP file is saved as 1550_0123.BMP in the /INTERNAL/AAA/BBB/CCC folder.

Description
- Use this command after setting the path with the following command.
  :FSYStem:PARAm:PATH
  :FSYStem:PARAm:NAME
- The save type is determined automatically from the extension of the specified file name.
  The following extensions are supported.
  .SOR: SOR format
  .CSV: CSV (waveform)
  .SET: Measurement setup file
  .CFG: Instrument setup file
  .BMP: Bitmap format
  .JPG: JPEG format
  .PDF: PDF (report) format
- If the folder specified by :FSYStem:PARAm:PATH does not exist, it is created automatically.
- To check for errors after saving, use :STATus:CONDition? to verify that file access is finished, and then use :STATus:ERRor?.

:FSYStem:SIZE?

Function Gets the size of a file.

Syntax
```
:FSYStem:SIZE?
```

Example
1. `:FSYSTEM:PARAM:PATH "/INTERNAL/AAA/BBB"`
2. `:FSYSTEM:PARAM:NAME "xxx.SOR"`
3. `:FSYSTEM:SIZE?`
   `-> :FSYSTEM:SIZE 230781`

Result:
Gets the file size of /INTERNAL/AAA/BBB/xxx.SOR.

Description
- Use this command after setting the path with the following command.
  :FSYStem:PARAm:PATH
  :FSYStem:PARAm:NAME
- The unit is bytes.

:FSYStem:WAVE:GET?

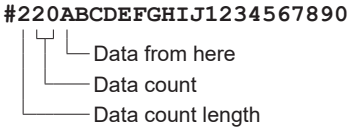
Function Gets the current waveform data (SOR, CSV, BMP, PDF, etc.). (data reception)

Syntax `:FSYStem:WAVE:GET?`

Example
1. `:FSYSTEM:WAVE:TYPE BMP`
2. `:FSYSTEM:WAVE:GET?`
   `-> #220ABCDEFGHIJ1234567890`

Result:
The BMP data of the OTDR's measured result screen is acquired as block data.

Description
- Use this command after setting the path with the following command.
  :FSYStem:WAVE:TYPE
- The current waveform data is sent in block data in the format specified by :FSYStem:WAVE:TYPE.



:FSYStem:WAVE:TYPE

Function Specifies the file type for acquiring the current OTDR waveform with a communication command.

Syntax
`:FSYStem:WAVE:TYPE {SOR|CSV_WAVE|CSV_EVENT|SET|CFG|BMP|JPG|PDF}`
`:FSYStem:WAVE:TYPE?`

SOR:	SOR format
CSV_WAVE:	CSV (waveform)
CSV_EVENT:	CSV (event)
SET:	Measurement setup file
CFG:	Instrument setup file
BMP:	Bitmap format
JPG:	JPEG format
PDF:	PDF format

Example
`:FSYSTEM:WAVE:TYPE SOR`
`:FSYSTEM:WAVE:TYPE?`
`-> :FSYSTEM:WAVE:TYPE SOR`

Description Use this command to specify the file type before using :FSYStem:WAVE:GET?.

:FSYStem:WAVE:VALid?

Function Checks whether the measurement waveform data exists.

Syntax `:FSYStem:WAVE:VALid?`

Example
`:FSYSTEM:WAVE:VALID?`
`-> :FSYSTEM:WAVE:VALID 1`

Description Measurement waveform data
0: Invalid (measurement waveform data does not exist)
1: Valid (measurement waveform data exists)

5.8 LABel Group

The commands in this group deal with the settings of the labels used in file identification. You can perform the same operations and make the same settings and queries that you can by using the front panel.

:LABel:CABLe:CODE

Function Sets or queries the cable code.
Syntax `:LABel:CABLe:CODE <String>`
`:LABel:CABLe:CODE?`
<String> = Up to 36 characters
Example `:LABEL:CABLE:CODE "123"`
`:LABEL:CABLE:CODE`
`->:LAB:CABLE:CODE "123"`

:LABel:CABLe:ID

Function Sets or queries the cable ID.
Syntax `:LABel:CABLe:ID <String>`
`:LABel:CABLe:ID?`
<String> = Up to 36 characters
Example `:LABEL:CABLE:ID "123"`
`:LABEL:CABLE:ID?`
`->:LABEL:CABLE:ID "123"`

:LABel:COMPany

Function Sets or queries the company name.
Syntax `:LABel:COMPany <String>`
`:LABel:COMPany?`
<String> = Up to 36 characters
Example `:LABEL:COMPANY "123"`
`:LABEL:COMPANY?`
`-> :LABEL:COMPANY "123"`

:LABel:DFLag:CURRent

Function Sets or queries the current data flag.
Syntax `:LABel:DFLag:CURRent {BC|RC|OT|CC}`
`:LABel:DFLag:CURRent?`
BC: as-Built Condition
RC: as-Repaired Condition
OT: Other
CC: Current state
Example `:LABEL:DFLAG:CURRENT BC`
`:LABEL:DFLAG:CURRENT?`
`-> :LABEL:DFLAG:CURRENT BC`

LABel:DIRection

Function Sets or queries the measurement direction.
Syntax `:LABel:DIRection {DIRECTION_AB|DIRECTION_BA}`
`:LABel:DIRection?`
DIRECTION_AB: A -> B
DIRECTION_BA: B -> A
Example `:LABEL:DIRECTION DIRECTION_AB`
`:LABEL:DIRECTION?`
`-> :LABEL:DIRECTION DIRECTION_AB`

:LABel:FIBer:ID

Function Sets or queries the fiber ID.
Syntax `:LABel:FIBer:ID <String>`
`:LABel:FIBer:ID?`
<String> = Up to 36 characters
Example `:LABEL:FIBER:ID "123"`
`:LABEL:FIBER:ID?`
`-> :LABEL:FIBER:ID "123"`

:LABel:FIBer:TYPE

Function Sets or queries the fiber type.
Syntax `:LABel:FIBer:TYPE {SMF|DSF|NZ_DSF|MMF}`
`:LABel:FIBer:TYPE?`
SMF: Single-mode optical fiber
DSF: Dispersion-shifted, single-mode fiber
NZ_DSF: Non-zero, dispersion-shifted, single-mode fiber
MMF: Multi-mode optical fiber
Example `:LABEL:FIBER:TYPE SMF`
`:LABEL:FIBER:TYPE?`
`-> :LABEL:FIBER:TYPE SMF`

:LABel:ID:USEFile

Function Sets or queries whether file IDs are used as fiber IDs or whether fiber IDs are used separately from file IDs.
Syntax `:LABel:ID:USEFile <Boolean>`
`:LABel:ID:USEFile?`
1: Use file IDs as fiber IDs
0: Use fiber IDs separately from file IDs
Example `:LABEL:ID:USEFILE 1`
`:LABEL:ID:USEFILE?`
`-> :LABEL:ID:USEFILE 1`

:LABel:LABel

Function Sets or queries the label.

Syntax
```
:LABel:LABel <String>
:LABel:LABel?
```
<String> = Up to 36 characters

Example
```
:LABEL:LABEL "123"
:LABEL:LABEL? -> :LABEL:LABEL "123"
```

:LABel:LOCation:ORIGinating

Function Sets or queries the start position label.

Syntax
```
:LABel:LOCation:ORIGinating <String>
:LABel:LOCation:ORIGinating?
```
<String> = Up to 36 characters

Example
```
:LABEL:LOCATION:ORIGINATING "123"
:LABEL:LOCATION:ORIGINATING?
-> :LABEL:LOC:ORIGINATING "123"
```

:LABel:LOCation:TERMinating

Function Sets or queries the end position label.

Syntax
```
:LABel:LOCation:TERMinating <String>
:LABel:LOCation:TERMinating?
```
<String> = Up to 36 characters

Example
```
:LABEL:LOCATION:TERMINATING "123"
:LABEL:LOCATION:TERMINATING?
-> :LABEL:LOCATION:TERMINATING "123"
```

:LABel:OPERator

Function Sets or queries the label's operator name.

Syntax
```
:LABel:OPERator <String>
:LABel:OPERator?
```
<String> = Up to 36 characters

Example
```
:LABEL:OPERATOR "123"
:LABEL:OPERATOR?
-> :LABEL:OPERATOR "123"
```

5.9 LIGHtsource Group

The commands in this group deal with the light source. You can perform the same operations and make the same settings and queries that you can by using the front panel.

:LIGHtsource:ABORt

Function Turns the light source setting off. Stops the emission of the measurement light.

Syntax
```
:LIGHtsource:ABORt
```

Example
```
:LIGHTSOURCE:ABORT
```

:LIGHtsource:EXECute

Function Turns the light source setting on. Starts the emission of the measurement light.

Syntax
```
:LIGHtsource:EXECute
```

Example
```
:LIGHTSOURCE:EXECUTE
```

:LIGHtsource:MODulation

Function Sets or queries the internal modulation frequency of the light source.

Syntax
```
:LIGHtsource:MODulation {MOD_CW|
MOD_270HZ|MOD_1KHZ|MODE_2KHZ}
:LIGHtsource:MODulation?
```

Example
```
:LIGHTSOURCE:MODULATION MOD_CW
:LIGHTSOURCE:MODULATION?
-> :LIGHTSOURCE:MODULATION MOD_CW
```

Description When :LIGHtsource:WAVelength is set to 0.850 um or 1.300 um, the modulation frequency can only be set to CW or 270 Hz.

:LIGHtsource:STATe

Function Turns the light source setting on and off. Stops/starts the emission of the measurement light.

Syntax
```
:LIGHtsource:STATe <Boolean>
```

Example
```
:LIGHT:SOURCE ON
:LIGHT:SOURCE? -> LIGHT:SOURCE ON
```

Description OFF: The same as :LIGHtsource:ABORt.
ON: The same as :LIGHtsource:EXECute.

:LIGHtsource:WAVelength

Function Sets or queries the wavelength of the light source.

Syntax
```
:LIGHtsource:WAVelength <NRf>
:LIGHtsource:WAVelength?
```
<NRf> = 0.850 um to 1.650 um
(850E-9 to 1650E-9)

Example
```
:LIGHTSOURCE:WAVELENGTH 0.85UM
:LIGHTSOURCE:WAVELENGTH 1650E-9
:LIGHTSOURCE:WAVELENGTH 1.650E-6
:LIGHTSOURCE:WAVELENGTH?
-> :LIGHTSOURCE:WAVELENGTH 1550E-09
```

Description The wavelengths that you can specify vary depending on the model.

5.10 MENU Group

The commands in this group are used to set the function or marker mode or query the settings.

:MENU:ERRor:CLEar

Function Clears the error dialog box.

Syntax `:MENU:ERRor:CLEar`

Example `:MENU:ERROR:CLEAR`

:MENU:FUNCtion

Function Sets or queries the function displayed in the data display screen.

Syntax `:MENU:FUNCtion {TOP|OTDR|LIGHt|OPC|PMETER|VLS|FIP}`

- TOP: MENU
- OTDR: OTDR
- LIGHt: Light source
- OPC: Power checker[1]
- PMETER: Power meter[2]
- VLS: Visible light source[3]
- FIP: Fiber inspection probe

Example
```
:MENU:FUNCTION TOP
:MENU:FUNCTION?
-> :MENU:FUNCTION TOP
```

Description
1. This is available when a power checker option (/PC) is installed.
2. This is available when a power meter (/SPM or /HPM option) is installed.
3. This is available when the visible light source (VLS option) is installed

:MENU:MARKer

Function Sets or queries the marker mode.

Syntax
```
:MENU:MARKer {MARKER|LINE}
:MENU:MARKer?
```

- MARKER: Marker
- LINE: Line

Example
```
:MENU:MARKER MARKER
:MENU:MARKER? -> :MENU:MARKER MARKER
```

5.11 MISC Group

The commands in this group deal with the date, language, and power management. You can make the same settings and queries that you can by using the front panel.

:MISC:ALARmsound

Function Sets or queries the alarm sound.
Syntax `:MISC:ALARmsound <Boolean>`
`:MISC:ALARmsound?`
Example `:MISC:ALARMSOUND OFF`
`:MISC:ALARMSOUND?`
`-> :MISC:ALARMSOUND 0`

:MISC:BACKlightoff

Function Sets or queries the backlight auto-off function for battery operation.
Syntax `:MISC:BACKlightoff <Boolean>`
`:MISC:BACKlightoff?`
Example `:MISC:BACKlightoff ON`
`:MISC:BACKlightoff?`
`-> :MISC:BACKLIGHTOFF 1`
Description Screen saver is not activated when an OTDR measurement is in progress.

:MISC:BRIGhtness:AC

Function Sets or queries the LCD brightness for USB-AC adapter operation.
Syntax `:MISC:BRIGhtness:AC {BRIGHT|NORMAL|DARK|OFF}`
`:MISC:BRIGhtness:AC?`
BRIGHT: Bright
NORMAL: Normal
DARK: Power save
OFF: None
Example `:MISC:BRIGHTNESS:AC NORMAL`
`:MISC:BRIGHTNESS:AC?`
`-> :MISC:BRIGHTNESS:AC NORMAL`

:MISC:BRIGhtness:BATTery

Function Sets or queries the LCD brightness for battery operation.
Syntax `:MISC:BRIGhtness:BATTery {BRIGHT|NORMAL|DARK|OFF}`
`:MISC:BRIGhtness:BATTery?`
BRIGHT: Bright
NORMAL: Normal
DARK: Power save
OFF: None
Example `:MISC:BRIGHTNESS:BATTERY NORMAL`
`:MISC:BRIGHTNESS:BATTERY?`
`-> :MISC:BRIGHTNESS:BATTERY NORMAL`
Description The parameters are the same as those for :MISC:BRIGhtness:AC.

:MISC:DATE:DAY

Function Sets or queries the date.
Syntax `:MISC:DATE:DAY <NRf>`
`:MISC:DATE:DAY?`
<NRf> = 1 to 31 (in 1 steps)
Example `:MISC:DATE:DAY 1`
`:MISC:DATE:DAY? -> :MISC:DATE:DAY 1`
Description When you execute the :MISC:DATE:SET command the date setting takes effect.

:MISC:DATE:GET?

Function Retrieves the date and time information.
Syntax `:MISC:DATE:GET?`
Example `:MISC:DATE:GET?`
`-> :MISC:DATE:GET 2025/01/31 23:59:59`

:MISC:DATE:HOUR

Function Sets or queries the hour.
Syntax `:MISC:DATE:HOUR <NRf>`
`:MISC:DATE:HOUR?`
<NRf> = 0 to 23 (in 1 steps)
Example `:MISC:DATE:HOUR 17`
`:MISC:DATE:HOUR?`
`-> :MISC:DATE:HOUR 12`
Description When you execute the :MISC:DATE:SET command the time setting takes effect.

:MISC:DATE:MINute

Function Sets or queries the minute.
Syntax `:MISC:DATE:MINute <NRf>`
`:MISC:DATE:MINute?`
<NRf> = 0 to 59 (in 1 steps)
Example `:MISC:DATE:MINUTE 5`
`:MISC:DATE:MINUTE?`
`-> :MISC:DATE:MINUTE 59`
Description When you execute the :MISC:DATE:SET command the time setting takes effect.

:MISC:DATE:MODE

Function Sets or queries the year/month/day display type.
Syntax `:MISC:DATE:MODE {TYPE1|TYPE2|TYPE3}`
`:MISC:DATE:MODE?`
TYPE1: 2024/08/29 12:16
TYPE2: 08/29/2024 12:16
TYPE3: 2024/AUG/29 12:16
Example `:MISC:DATE:MODE TYPE1`
`:MISC:DATE:MODE?`
`-> :MISC:DATE:MODE TYPE1`

:MISC:DATE:MONTh

Function Sets or queries the month.
Syntax `:MISC:DATE:MONTh <NRf>`
`:MISC:DATE:MONTh?`
<NRf> = 1 to 12
Example `:MISC:DATE:MONTH 1`
`:MISC:DATE:MONTH?`
`-> :MISC:DATE:MONTH 1`
Description When you execute the :MISC:DATE:SET command the month setting takes effect.

:MISC:DATE:SECond

Function Sets or queries the second.
Syntax `:MISC:DATE:SECond <NRf>`
`:MISC:DATE:SECond?`
<NRf> = 0 to 59 (in 1 steps)
Example `:MISC:DATE:SECOND 0`
`:MISC:DATE:SECOND?`
`-> :MISC:DATE:SECOND 0`
Description When you execute the :MISC:DATE:SET command the second setting takes effect.

:MISC:DATE:SET

Function Executes the changing of the date and time.
Syntax `:MISC:DATE:SET`
Example `:MISC:DATE:SET`

:MISC:DATE:YEAR

Function Sets or queries the year.
Syntax `:MISC:DATE:YEAR <NRf>`
`:MISC:DATE:YEAR?`
<NRf> = 2006 to 2035 (in 1 steps)
Example `:MISC:DATE:YEAR 2025`
`:MISC:DATE:YEAR?`
`-> :MISC:DATE:YEAR 2025`
Description When you execute the :MISC:DATE:SET command the year setting takes effect.

:MISC:LANGuage

Function Sets or queries the language.
Syntax `:MISC:LANGuage {JAPANESE|ENGLISH}`
`:MISC:LANGuage?`
Example `:MISC:LANGUAGE ENGLISH`
`:MISC:LANGUAGE?`
`-> :MISC:LANGUAGE ENGLISH`
Description The available language settings vary depending on the suffix code.

- **When the Suffix Code Is -HE**
  ENGLISH: English
  FRENCH: French
  GERMANY: German
  ITALY: Italian
  SPANISH: Spanish
  DUTCH: Dutch
  FINNISH: Finnish
  NORWEGIAN: Norwegian
  POLISH: Polish
  CZECH: Czech
  SWEDISH: Swedish
  TURKEY: Turkish
  PORTUGUESE: Portuguese
  TCHINESE: Traditional Chinese
  BRAZILIAN: Portuguese (Brazilian)
  ROMANIAN: Romanian
  INDONESIAN: Indonesian
  VIETNAMESE: Vietnamese
  HUNGARIAN: Hungarian
- **When the Suffix Code Is -HJ**
  JAPANESE: Japanese
  ENGLISH: English
- **When the Suffix Code Is -HC**
  CHINESE: Simplified Chinese
  ENGLISH: English
- **When the Suffix Code Is -HM**
  CHINESE: Simplified Chinese
- **When the Suffix Code Is -HK**
  KOREAN: Korean
  ENGLISH: English
- **When the Suffix Code Is -HR**
  RUSSIAN: Russian
  ENGLISH: English

:MISC:POWersave:BATTery

Function Sets or queries the power save function for battery operation.

Syntax
```
:MISC:POWersave:BATTery {OFF|A1MIN|
A5MIN|A10MIN|A30MIN}
:MISC:POWersave:BATTery?
```
OFF: None
A1MIN: Auto power off after 1 minute
A5MIN: Auto power off after 5 minutes
A10MIN: Auto power off after 10 minutes
A30MIN: Auto power off after 30 minutes

Example
```
:MISC:POWERSAVE:BATTERY S20MIN
:MISC:POWERSAVE:BATTERY?
-> :MISC:POWERSAVE:BATTERY OFF
```

Description Auto power off is not performed while an OTDR measurement is in progress.

:MISC:RLOSsmode

Function Sets or queries the return loss display.

Syntax
```
:MISC:RLOSsmode {NORMAL|NTT}
:MISC:MARKer?
```
NORMAL: Return loss
NTT: Reflection level

Example
```
:MISC:RLOSSMODE  NORMAL
:MISC:RLOSSMODE?
-> :MISC:RLOSSMODE NORMAL
```

5.12 NETWork Group

The commands in this group can only be used on models with the /LAN option.

:NETWork:CONTrol:PASSword

Function Sets or queries the network password.

Syntax
```
:NETWork:CONTrol:PASSword <String>
:NETWork:CONTrol:PASSword?
```

Example
```
:NETWORK:CONTROLl:PASSWORD "ABC"
:NETWORK:CONTROLl:PASSWORD?
-> :NETWORK:CONTROLl:PASSWORD "ABC"
```

Description
- This command is invalid when the network is turned off with the :NETWork:STATe command.
- This command will be valid from the next connection.
- You do not need to set the password when the user name is set to “anonymous.”

:NETWork:CONTrol:TIMeout

Function Sets or queries the network timeout value.

Syntax
```
:NETWork:CONTrol:TIMeout {<NRf>|
INFinite}
:NETWork:CONTrol:TIMeout?
```
<NRf> = 1 to 7200 (s)

Example
```
:NETWORK:CONTROL:TIMEOUT 30
:NETWORK:CONTROL:TIMEOUT?
-> :NETWORK:CONTROL:TIMEOUT 30
```

Description
- The unit is seconds
- This command is invalid when the network is turned off with the :NETWork:STATe command.
- This command will be valid from the next connection.
- Valid for models with the /LAN option.

:NETWork:CONTrol:USERname

Function Sets or queries the network user name.

Syntax
```
:NETWork:CONTrol:USERname <String>
:NETWork:CONTrol:USERname?
```

Example
```
:NETWORK:CONTROL:
USERNAME "anonymous"
:NETWORK:CONTROL:USERNAME?
-> :NETWORK:CONTROL:
USERNAME "anonyumous"
```

Description
- This command is invalid when the network is turned off with the :NETWork:STATe command.
- This command will be valid from the next

:NETWork:DHCP

Function Sets or queries the network DHCP.

Syntax
```
:NETWork:DHCP <Boolean>
:NETWork:DHCP?
```

Example
```
:NETWORK:DHCP ON
:NETWORK:DHCP? -> :NETWORK:DHCP 1
```

Description
- This command is invalid when the network is turned off with the :NETWork:STATe command.
- This command will be valid from the next connection.
- Valid for models with the /LAN option.

:NETWork:GATeway

Function Sets or queries the network gateway.

Syntax
```
:NETWork:GATeway <String>
:NETWork:GATeway?
```

Example
```
:NETWORK:GATEWAY "255.255.255.0"
:NETWORK:GATEWAY?
-> :NETWORK:GATEWAY "255.255.255.0"
```

Description
- This command is invalid when the network is turned off with the :NETWork:STATe command.
- This command is invalid when DHCP is turned off with the :NETWork:DHCP command.
- This command will be valid from the next connection.
- Valid for models with the /LAN option.

:NETWork:IPADdress

Function Sets or queries the network IP address.

Syntax
```
:NETWork:IPADdress <String>
:NETWork:IPADdress?
```

Example
```
:NETWORK:IPADDRESS "192.168.0.1"
:NETWORK:IPADDRESS?
-> :NETWORK:IPADDRESS "192.168.0.1"
```

Description
- This command is invalid when the network is turned off with the :NETWork:STATe command.
- This command is invalid when DHCP is turned off with the :NETWork:DHCP command.
- This command will be valid from the next connection.
- Valid for models with the /LAN option.

:NETWork:NETMask

Function Sets or queries the network subnet mask.

Syntax
```
:NETWork:NETMask <string>
:NETWork:NETMask?
```

Example
```
:NETWORK:NETMASK "255.255.255.0"
:NETWORK:NETMASK?
-> :NETWORK:NETMASK "255.255.255.0"
```

Description
- This command is invalid when the network is turned off with the :NETWork:STATe command.
- This command is invalid when DHCP is turned off with the :NETWork:DHCP command.
- This command will be valid from the next connection.
- Valid for models with the /LAN option.

:NETWork:STATe

Function Sets or queries the network usage.

Syntax
```
:NETWork:STATe <Boolean>
:NETWork:STATe?
```
OFF or 0: Invalid
ON or 1: Valid

Example
```
:NETWORK:STATE ON
:NETWORK:STATE? -> :NETWORK:STATE 1
```

Description
- This command will be valid from the next connection.
- Valid for models with the /LAN option.

:NETWork:WLAN:ACCesspoint:SSID

Function Sets or queries the ID name (SSID) of the network WLAN application.

Syntax
```
:NETWork:WLAN:ACCesspoint:
SSID <String>
:NETWork:WLAN:ACCesspoint:SSID?
```

Example
```
:NETWORK:WLAN:ACCESSPOINT:
SSID "12345678"
:NETWork:WLAN:ACCESSPOINT:SSID?
-> :NETWork:WLAN:ACCESSPOINT:
  SSID "12345678"
```

:NETWork:WLAN:ACCesspoint:ENCrypt

Function Sets or queries the encryption mode of the network WLAN application.

Syntax
```
:NETWork:WLAN:ACCesspoint:
ENCrypt {NONE|WPA2 _ PSK}
:NETWork:WLAN:ACCesspoint:ENCrypt?
```

Example
```
:NETWORK:WLAN:ACCESSPOINT:
ENCRYPT WPA2 _ PSK
:NETWORK:WLAN:ACCESSPOINT:ENCRYPT?
-> :NETWORK:WLAN:ACCESSPOINT:
  ENCRYPT WPA2 _ PSK
```

:NETWork:WLAN:ACCesspoint:PASSword

Function Sets or queries the password of the network WLAN application.

Syntax
```
:NETWork:WLAN:ACCesspoint:
PASSword <String>
:NETWork:WLAN:ACCesspoint:PASSword?
```

Example
```
:NETWORK:WLAN:ACCESSPOINT:
PASSWORD "12345678"
:NETWORK:WLAN:ACCESSPOINT:PASSWORD?
-> :NETWORK:WLAN:ACCESSPOINT:
  PASSWORD "12345678"
```

:NETWork:WLAN:ACCesspoint:CHannel

Function Sets or queries the channel of the network WLAN application.

Syntax
```
:NETWork:WLAN:ACCesspoint:
CHannel {AUTO|CH<x>}
:NETWork:WLAN:ACCesspoint:CHannel?
```
<x> = 1 to 11

Example
```
:NETWORK:WLAN:ACCESSPOINT:
CHANNEL CH1
:NETWORK:WLAN:ACCESSPOINT:CHANNEL?
-> :NETWORK:WLAN:ACCESSPOINT:
  CHANNEL CH1
```

:NETWork:WLAN:ACCesspoint:IPADdress

Function Sets or queries the IP address of the network WLAN application.

Syntax
```
:NETWork:WLAN:ACCesspoint:
IPADdress <String>
:NETWork:WLAN:ACCesspoint:IPADdress?
```

Example
```
:NETWORK:WLANAPMODE:
IPADDRESS "192.168.0.1"
:NETWORK:WLAN:ACCESSPOINT:IPADDRESS?
-> :NETWORK:WLAN:ACCESSPOINT:
  IPADDRESS "192.168.0.1"
```

:NETWork:WLAN:ACCesspoint:NETMask

Function Sets or queries the subnet mask of the network WLAN application.

Syntax
```
:NETWork:WLAN:ACCesspoint:
NETMask <String>
:NETWork:WLAN:ACCesspoint:NETMask?
```

Example
```
:NETWORK:WLAN:ACCESSPOINT:
NETMASK "255.255.255.0"
:NETWORK:WLAN:ACCESSPOINT:NETMASK?
-> :NETWORK:WLAN:ACCESSPOINT:
  NETMASK "255.255.255.0"
```

:NETWork:WLAN:ACCesspoint:DHCPStartaddr

Function Sets or queries the start address of the IP addresses that are assigned by the DHCP server of the network WLAN application.

Syntax
```
:NETWork:WLAN:ACCesspoint:
DHCPStartaddr <String>
:NETWork:WLAN:ACCesspoint:
DHCPStartaddr?
```

Example
```
:NETWORK:WLAN:ACCESSPOINT:
DHCPSTARTADDR "192.168.0.100"
:NETWORK:WLAN:ACCESSPOINT:
DHCPSTARTADDR?
-> :NETWORK:WLAN:ACCESSPOINT:
  DHCPSTARTADDR "192.168.0.100"
```

:NETWork:WLAN:ACCesspoint:STATe

Function Sets or queries the network WLAN application usage.

Syntax
```
:NETWork:WLAN:ACCesspoint:
STATe <Boolean>
:NETWork:WLAN:ACCesspoint:STATe?
```

Example
```
:NETWORK:WLAN:ACCESSPOINT:STATE ON
:NETWORK:WLAN:ACCESSPOINT:STATE?
-> :NETWORK:WLAN:ACCESSPOINT:STATE 1
```

Description The WLAN application becomes available when the WLAN application usage is set to ON.

:NETWork:WLAN:COMMon:FWVersion?

Function Queries the WLAN module's firmware version.

Syntax
```
:NETWork:WLAN:COMMon:FWVersion?
```

Example
```
:NETWork:WLAN:COMMon:FWVersion?
-> :NETWork:WLAN:COMMon:
  FWVersion 1695438116
```

:NETWork:WLAN:COMMon:HWVersion?

Function Queries the WLAN module's hardware version.

Syntax
```
:NETWork:WLAN:COMMon:HWVersion?
```

Example
```
:NETWork:WLAN:COMMon:HWVersion?
-> :NETWork:WLAN:COMMon:
  HWVersion 18448
```

:NETWork:WLAN:COMMon:MACaddress?

Function Queries the WLAN module's MAC address.

Syntax
```
:NETWork:WLAN:COMMon:MACaddress?
```

Example
```
:NETWork:WLAN:COMMon:MACADDRESS?
-> :NETWork:WLAN:COMMon:
  MACADDRESS "XXXXXXXXXXXX"
```

:NETWork:WLAN:STATion:AUToconnect

Function Sets or queries the auto network connection on/off state.

Syntax
```
:NETWork:WLAN:STATion:
AUToconnect <Boolean>
:NETWork:WLAN:STATion:AUToconnect?
```

Example
```
:NETWORK:WLAN:STATION:AUTOCONNECT ON
:NETWORK:WLAN:STATION:AUTOCONNECT?
-> :NETWORK:WLAN:STATION:
  AUTOCONNECT 1
```

:NETWork:WLAN:STATion:CONNect

Function Starts the network WLAN station mode.

Syntax
```
:NETWork:WLAN:STATion:CONNect
```

Example
```
:NETWORK:WLAN:STATION:CONNECT
```

:NETWork:WLAN:STATion:DHCP

Function Sets or queries the WLAN station mode's DHCP on/off state.

Syntax
```
:NETWork:WLAN:STATion:DHCP <Boolean>
:NETWork:WLAN:STATion:DHCP?
```

Example
```
:NETWORK:WLAN:STATION:DHCP ON
:NETWORK:WLAN:STATION:DHCP?
-> :NETWORK:WLAN:STATION:DHCP 1
```

:NETWork:WLAN:STATion:DISConnect

Function Stops the WLAN station mode.

Syntax
```
:NETWork:WLAN:STATion:DISConnect
```

Example
```
:NETWORK:WLAN:STATION:DISCONNECT
```

:NETWork:WLAN:STATion:GATeway

Function Sets or queries the WLAN station mode's gateway.

Syntax
```
:NETWork:WLAN:STATion:
GATeway <String>
:NETWork:WLAN:STATion:GATeway?
```

Example
```
:NETWORK:WLAN:STATION:
GATEWAY "255.255.255.0"
:NETWORK:WLAN:STATION:GATEWAY?
-> :NETWORK:WLAN:STATION:
  GATEWAY "255.255.255.0"
```

:NETWork:WLAN:STATion:IPADdress

Function Sets or queries the WLAN station mode's IP address.

Syntax
```
:NETWork:WLAN:STATion:
IPADdress <String>
:NETWork:WLAN:STATion:IPADdress?
```

Example
```
:NETWORK:WLAN:STATION:
IPADDRESS "192.168.0.1"
:NETWORK:WLAN:STATION:IPADDRESS?
-> :NETWORK:WLAN:STATION:
  IPADDRESS "192.168.0.1"
```

:NETWork:WLAN:STATion:NETMask

Function Sets or queries the WLAN station mode's netmask.

Syntax
```
:NETWork:WLAN:STATion:
NETMask <String>
:NETWork:WLAN:STATion:NETMask?
```

Example
```
:NETWORK:WLAN:STATION:
NETMASK "255.255.255.0"
:NETWORK:WLAN:STATION:NETMASK?
-> :NETWORK:WLAN:STATION:
  NETMASK "255.255.255.0"
```

:NETWork:WLAN:STATion:PASSword

Function Sets or queries the WLAN station mode's password.

Syntax
```
:NETWork:WLAN:STATion:
PASSword <String>
:NETWork:WLAN:STATion:PASSword?
```

Example
```
:NETWORK:WLAN:STATION:
PASSWORD "12345678"
:NETWORK:WLAN:STATION:PASSWORD?
-> :NETWORK:WLAN:STATION:
  PASSWORD "12345678"
```

:NETWork:WLAN:STATion:SSID

Function Sets or queries the WLAN station mode's SSID.

Syntax
```
:NETWork:WLAN:STATion:SSID <String>
:NETWork:WLAN:STATion:SSID?
```

Example
```
:NETWORK:WLAN:STATION:SSID "12345678"
:NETWORK:WLAN:STATION:SSID?
-> :NETWORK:WLAN:STATION:
  SSID "12345678"
```

:NETWork:WLAN:STATion:STATe?

Function Sets or queries the WLAN station mode's state.

Syntax
```
:NETWork:WLAN:STATion:STATe?
```

Example
```
:NETWORK:WLAN:STATION:STATE?
-> :NETWORK:WLAN:STATION:STATE 1
```

:NETWork:WLAN:STATion:USEFixedssid

Function Sets or queries the WLAN station mode's fixed SSID on/off state.

Syntax
```
:NETWork:WLAN:STATion:
USEFixedssid <Boolean>
:NETWork:WLAN:STATion:USEFixedssid?
```

Example
```
:NETWORK:WLAN:STATION:USEFIXEDSSID ON
:NETWORK:WLAN:STATION:USEFIXEDSSID?
-> :NETWORK:WLAN:STATION:
  USEFIXEDSSID 1
```

5.13 OSWitch Group

The commands in this group deal with optical switch control. You can make the same settings and queries that you can from the front panel. This feature is available when an AQ3550 optical switch box is connected.

:OSWitch:ENABle?

Function Queries whether an optical switch is connected to this instrument.

Syntax `:OSWitch:ENABle?`
0: Not connected
1: Connected

Example `:OSWITCH:ENABLE? -> :OSWITCH:ENABLE 1`

Description Outputs the connection status of the optical switch.

:OSWitch:LINK:ENABle

Function Sets or queries the optical switch linking.

Syntax `:OSWitch:LINK:ENABle <Boolean>`
`:OSWitch:LINK:ENABle?`

Example `:OSWITCH:LINK:ENABLE 1`
`:OSWITCH:LINK:ENABLE?`
`-> :OSWITCH:LINK:ENABLE 1`

Description Enables or disables the optical switch link function for the optical switch connected to the instrument.

:OSWitch:LINK:NUMBer<x>

Function Sets or queries the optical switch link channel.

Syntax `:OSWitch:LINK:NUMBer<x> <Boolean>`
`:OSWitch:LINK:NUMBer<x>?`
<x> = 1 to 12

Example `:OSWITCH:LINK:NUMBER12 1`
`:OSWITCH:LINK:NUMBER12?`
`-> :OSWITCH:LINK:NUMBER12 1`

Description Enables or disables channel number-specific settings of the optical switch link function for the optical switch connected to the instrument.

:OSWitch:CHANnel:NUMBer?

Function Queries the number of channels (input and output ports) available on the optical switch connected to this instrument.

Syntax `:OSWitch:CHANnel:NUMBer?`

Example `:OSWITCH:CHANNEL:NUMBER?`
`-> :OSWITCH:CHANNEL:NUMBER 12`

Description Outputs the number of channels available on the optical switch.

:OSWitch:CHANnel:SELect

Function Sets or queries the selected channel number of the optical switch connected to this instrument.

Syntax `:OSWitch:CHANnel:SELect <NRf>`
<NRf> = 1 to 12
`:OSWitch:CHANnel:SELect?`

Example `:OSWITCH:CHANNEL:SELECT 8`
`:OSWITCH:CHANNEL:SELECT?`
`-> :OSWITCH:CHANNEL:SELECT 8`

Description You can select from the channel numbers of the optical switch that the measurement target optical fiber cables are connected to.

5.14 PMETer Group

The commands in this group deal with the power meter. You can make the same settings and queries that you can by using the front panel. This function is valid on models with the /SPM or /HPM option.

:PMETer:AVERage:TIMes

Function Sets or queries the power meter average count.
Syntax `:PMETer:AVERage:TIMes {T1|T10|T50|T100}`
`:PMETer:AVERage:TIMes?`
T1: 1
T10: 10
T50: 50
T100: 100
Example `:PMETER:AVERAGE:TIMES T1`
`:PMETER:AVERAGE:TIMES?`
`-> :PMETER:AVERAGE:TIMES T1`

:PMETer:DREF

Function Executes Dref on the power meter.
Syntax `:PMETer:DREF`
Example `:PMETER:DREF`

:PMETer:LINK:STATe

Function Sets or queries the light source power meter setting interlock.
Syntax `:PMETer:LINK:STATe <Boolean>`
Example `:PMETER:LINK:STATE ON`
`:PMETER:LINK:STATE?`
`-> :PMETER:LINK:STATE 1`

:PMETer:MAXMin?

Function Queries all settings related to the maximum and minimum values of the power meter.
Syntax `:PMETer:MAXMin?`
Example `:PMETER:MAXMIN?`
`-> :PMETER:MAXMIN`

:PMETer:MAXMin:MAX?

Function Sets or queries the maximum value.
Syntax `:PMETer:MAXMin:MAX?`
Example `:PMETER:MAXMIN:MAX?`
`-> :PMETER:MAXMIN:MAX 5.00`
Description You can only make this query when :PMETer:MAXMin:STATe is set to ON.

:PMETer:MAXMin:MIN?

Function Sets or queries the minimum value.
Syntax `:PMETer:MAXMin:MIN?`
Example `:PMETER:MAXMIN:MIN?`
`-> :PMETER:MAXMIN:MIN -5.00`
Description You can only make this query when :PMETer:MAXMin:STATe is set to ON.

:PMETer:MAXMin:STATe

Function Sets or queries the MAX and MIN display on/off state.
Syntax `:PMETer:MAXMin:STATe <Boolean>`
Example `:PMETER:MAXMIN:STATE ON`
`:PMETER:MAXMIN:STATE?`
`-> :PMETER:MAXMIN:STATE 1`

:PMETer:MEASurement:DATA?

Function Queries the power meter's measured results.
Syntax `:PMETer:MEASurement:DATA?`
Example `:PMETER:MEASUREMENT:DATA?`
`-> :PMETER:MEASUREMENT:DATA -26.5`

:PMETer:MODulation

Function Sets or queries the power meter modulation.
Syntax `:PMETer:MODulation {MOD_270HZ|`
`MOD_CW|MOD_1KHZ|MOD_2KHZ}`
Example `:PMETER:MODULATION MOD_270HZ`
`:PMETER:MODULATION?`
`-> :PMETER:MODULATION MOD_270HZ`

:PMETer:OFFSet

Function Sets or queries the power meter offset.
Syntax `:PMETer:OFFSet <NRf>`
`:PMETer:OFFSet?`
<NRf> = -9.900 to 9.900 (in steps of 0.001)
Example `:PMETER:OFFSET -9.900`
`:PMETER:OFFSET?`
`-> :PMETER:OFFSET -9.900`

:PMETer:REFerence

Function Sets or queries the power meter reference value.
Syntax `:PMETer:REFerence <NRf>`
`:PMETer:REFerence?`
<NRf> = -80.00 to 40.00 (in steps of 0.01) [dBm]
Example `:PMETER:REFERENCE -80.00`
`:PMETER:REFERENCE?`
`-> :PMETER:REFERENCE -80.00`

:PMETer:THReshold:LOWer

Function Sets or queries the power meter lower threshold value.
Syntax `:PMETer:THReshold:LOWer <NRf>`
<NRf> = -80.00 to 40.00 (in steps of 0.01) [dBm]
Example `:PMETER:THRESHOLD:LOWER -80.00`
`:PMETER:THRESHOLD:LOWER?`
`-> :PMETER:THRESHOLD:LOWER -80.00`

:PMETer:THReshold:UPPer

Function Sets or queries the power meter upper threshold value.

Syntax `:PMETer:THReshold:UPPer <NRf>`
<NRf> = -80.00 to 40.00 (in steps of 0.01) [dBm]

Example
```
:PMETER:THRESHOLD:UPPER -80.00
:PMETER:THRESHOLD:UPPER?
-> :PMETER:THRESHOLD:UPPER -80.00
```

:PMETer:UNIT

Function Sets or queries the power meter display unit.

Syntax
```
:PMETer:UNIT {DB|DBM|W}
:PMETer:UNIT?
```

Example
```
:PMETER:UNIT DBM
:PMETER:UNIT? -> :PMETER:UNIT DB
```

:PMETer:WAVelength:DETail

Function Sets or queries the wavelength when the wavelength mode is set to Detail.

Syntax
```
:PMETer:WAVelength:DETail <NRf>
:PMETer:WAVelength:DETail?
```
<NRf> = 0.850um to 1.650um
(850E-09 to 1650E-09)

Example
```
:PMETer:WAVELENGTH:DETAIL 0.85UM
:PMETer:WAVELENGTH:DETAIL 1650E-09
:PMETer:WAVELENGTH:DETAIL 1.650E-06
:PMETer:WAVELENGTH:DETAIL?
-> :PMETER:WAVELENGTH:DETAIL 1550E-09
```

Description If the wavelength mode is set to Simple or CWDM, executing this command sets the mode to Detail.

:PMETer:ZERoset

Function Executes zero set on the power meter.

Syntax `:PMETer:ZERoset`

Example `:PMETER:ZEROSET`

5.15 PMONitor Group

The commands in this group deal with the power monitor. You can make the same settings and inquiries as when the front panel is used.

:PMONitor:DREF

Function Executes Dref (display reference) on the power monitor.
Syntax :PMONitor:DREF
Example :PMONITOR:DREF

:PMONitor:MAX?

Function Queries the maximum scale value of the power monitor.
Syntax :PMONitor:MAX?
Example :PMONITOR:MAX? -> :PMONITOR:MAX -5.0

:PMONitor:MEASurement:DATA?

Function Queries the measured results of the power monitor.
Syntax :PMONitor:MEASurement:DATA?
Example :PMONITOR:MEASUREMENT:DATA?
-> :PMONITOR:MEASUREMENT:DATA -26.5

:PMONitor:MIN?

Function Queries the minimum scale value of the power monitor.
Syntax :PMONitor:MIN?
Example :PMONITOR:MIN? -> :PMONITOR:MIN -50.0

:PMONitor:OFFSet

Function Sets or queries the power monitor offset.
Syntax :PMONitor:OFFSet <NRf>
:PMONitor:OFFSet?
<NRf> = -9.9 to 9.9
Example :PMONITOR:OFFSET -5.0
:PMONITOR:OFFSET?
-> :PMONITOR:OFFSET -3.0

:PMONitor:REFerence

Function Sets or queries the power monitor reference.
Syntax :PMONitor:REFerence <NRf>
:PMONitor:REFerence?
<NRf> = -50.0 to -5.0
Example :PMONITOR:REFERENCE -45.00
:PMONITOR:REFERENCE?
-> :PMONITOR:REFERENCE -44.1

:PMONitor:THReshold:LOWer

Function Sets or queries the power monitor lower threshold limit.
Syntax :PMONitor:THReshold:Lower <NRf>
:PMONitor:THReshold:Lower?
<NRf> = -50.0 to -5.0
Example :PMONITOR:THRESHOLD:LOWER -10
:PMONITOR:THRESHOLD:LOWER?
-> :PMONITOR:THRESHOLD:LOWER -10.0

:PMONitor:THReshold:Upper

Function Sets or queries the power monitor upper threshold limit.
Syntax :PMONitor:THReshold:Upper <NRf>
:PMONitor:THReshold:Upper?
<NRf> = -50.0 to -5.0
Example :PMONITOR:THRESHOLD:UPPER -10
:PMONITOR:THRESHOLD:UPPER?
-> :PMONITOR:THRESHOLD:UPPER -10.0

:PMONitor:UNIT

Function Sets or queries the power unit display unit.
Syntax :PMONitor:UNIT {DB|DBM|W}
:PMONitor:UNIT?
DB: dB Unit
DBM: dBm Unit
W: W Unit
Example :PMONITOR:UNIT DBM
:PMONITOR:UNIT ? -> :PMONITOR:UNIT DB

:PMONitor:WAVelength

Function Sets or queries the power monitor wavelength.
Syntax :PMONitor:WAVelength <NRf>
:PMONitor:WAVelength?
<NRf> = 0.850 um to 1.650 um
(850E-09 to 1650E-09)
Example :PMONITOR:WAVELENGTH 0.85UM
:PMONITOR:WAVELENGTH 1650E-09
:PMONITOR:WAVELENGTH 1.650E-06
:PMONITOR:WAVELENGTH?
-> PMONITOR:WAVELENGTH 1550E-09

:PMONitor:ZERoset

Function Executes zero-set on the power monitor.
Syntax :PMONitor:ZERoset
Example :PMONITOR:ZEROSET

5.16 REPOrt Group

The commands in this group deal with report. You can make the same settings and queries that you can by using the front panel.

:REPOrt:EXECute

Function Creates a report.
Syntax :REPOrt:EXECute
Example :REPORT:EXECUTE
Description The report file save destination is set to the path specified by :FSYStem:PARAm:PATH.

:REPOrt:SET:EVENt:CONDition

Function Sets or queries whether event search conditions will be included in report files.
Syntax :REPOrt:SET:EVENt:CONDition <Boolean>
:REPOrt:SET:EVENt:CONDition?
Example :REPORT:SET:EVENT:CONDITION ON
:REPORT:SET:EVENT:CONDITION?
-> :REPORT:SET:EVENT:CONDITION 1

:REPOrt:SET:EVENt:LIST

Function Sets or queries the fiber end face image in report settings.
Syntax :REPOrt:SET:EVENt:LIST <Boolean>
:REPOrt:SET:EVENt:LIST?
Example :REPORT:SET:EVENT:LIST ON
:REPORT:SET:EVENT:LIST?
-> :REPORT:SET:EVENT:LIST 1

:REPOrt:SET:FIBEr

Function Sets or queries the mode for including fiber surface image files in report files.
Syntax :REPOrt:SET:FIBEr {OFF|AUTO}
:REPOrt:SET:FIBEr?
Example :REPORT:SET:FIBER AUTO
:REPORT:SET:FIBER?
-> :REPORT:SET:FIBER AUTO

:REPOrt:SET:JUDGe

Function Sets or queries whether pass/fail judgment conditions will be included in report files.
Syntax :REPOrt:SET:JUDGe <Boolean>
:REPOrt:SET:JUDGe?
Example :REPORT:SET:JUDGE ON
:REPORT:SET:JUDGE?
-> :REPORT:SET:JUDGE 1

:REPOrt:SET:MAP

Function Sets or queries the MAP display in report settings.
Syntax :REPOrt:SET:MAP <Boolean>
:REPOrt:SET:MAP?
Example :REPORT:SET:MAP ON
:REPORT:SET:MAP?
-> :REPORT:SET:MAP 1

:REPOrt:SET:MEASure:CONDition

Function Sets or queries whether measurement conditions will be included in report files.
Syntax :REPOrt:SET:MEASure:
CONDition <Boolean>
:REPOrt:SET:MEASure:CONDition?
Example :REPORT:SET:MEASURE:CONDITION ON
:REPORT:SET:MEASURE:CONDITION?
-> :REPORT:SET:MEASURE:CONDITION 1

:REPOrt:SET:MEASure:INFO

Function Sets or queries whether measurement information will be included in report files.
Syntax :REPOrt:SET:MEASure:INFO <Boolean>
:REPOrt:SET:MEASure:INFO?
Example :REPORT:SET:MEASURE:INFO ON
:REPORT:SET:MEASURE:INOF?
-> :REPORT:SET:MEASURE:INFO 1

:REPOrt:SET:MEASure:RESUlt

Function Sets or queries whether measurement results will be included in report files.
Syntax :REPOrt:SET:MEASure:RESUlt <Boolean>
:REPOrt:SET:MEASure:RESUlt?
Example :REPORT:SET:MEASURE:RESULT ON
:REPORT:SET:MEASURE:RESULT?
-> :REPORT:SET:MEASURE:RESULT 1

:REPOrt:SET:OVERview

Function Sets or queries whether overviews will be included in report files.
Syntax :REPOrt:SET:OVERview <Boolean>
:REPOrt:SET:OVERview?
Example :REPORT:SET:OVERview ON
:REPORT:SET:OVERview?
-> :REPORT:SET:OVERview 1

:REPOrt:SET:WAVE

Function Sets or queries whether waveforms will be included in report files.

Syntax
```
:REPOrt:SET:WAVE <Boolean>
:REPOrt:SET:WAVE?
```

Example
```
:REPORT:SET:WAVE ON
:REPORT:SET:WAVE?
-> :REPORT:SET:WAVE 1
```

5.17 SETup Group

The commands in this group deal with the initialization of the settings. The AQ7290 settings can be returned to their factory defaults.

:SETup:ALLinit

Function Initializes all settings to their factory default values and formats the internal memory.
Syntax `:SETup:ALLInit`
Example `:SETup:ALLINIT`

:SETup:INITialize

Function Initializes all settings to their factory default values.
Syntax `:SETup:INITialize`
Example `:SETUP:INITIALIZE`

5.18 STATus Group

The commands in this group are used to make settings and queries related to the communication status feature. There are no front panel keys that correspond to the commands in this group.

:STATus?

Function Queries all the settings for the communication status feature.

Syntax `:STATus?`

Example
```
:STATUS?
-> :STATUS:QENABLE 1;QMESSAGE 1
```

:STATus:CONDition?

Function Queries the contents of the condition register.

Syntax `:STATus:CONDition?`

Example `:STATUS:CONDITION? -> 16`

Description For information about the condition register, see section 6.1, “Condition Register.”

:STATus:ERRor?

Function Queries the error code and message information (top of the error queue).

Syntax `:STATus:ERRor?`

Example
```
:STATUS:ERROR?
-> 113,"Undefined header"
```

:STATus:QENable

Function Sets or queries whether or not messages other than errors will be stored to the error queue (on/off).

Syntax
```
:STATus:QENable <Boolean>
:STATus:QENable?
```

Example
```
:STATUS:QENABLE ON
:STATUS:QENABLE? -> :STATUS:QENABLE 1
```

:STATus:QMESsage

Function Sets or queries whether or not message information will be attached to the response to the STATus:ERRor? query (on/off).

Syntax
```
:STATus:QMESsage <Boolean>
:STATus:QMESsage?
```

Example
```
:STATUS:QMESSAGE OFF
:STATUS:QMESSAGE?
-> :STATUS:QMESSAGE 0
```

5.19 SYSTem Group

The commands in this group deal with startup. There are no front panel keys that correspond to the commands in this group.

:SYSTem:REBoot

Function Restarts the system.
Syntax `:SYSTem:REBoot`
Example `:SYSTEM:REBOOT`
Description The communication will be disconnected, so execute disconnection (TmcFinish).

:SYSTem:SAVE

Function Saves system settings.
Syntax `:SYSTem:SAVE`
Example `:SYSTEM:SAVE`

:SYSTem:SHUTdown

Function Shuts down the instrument.
Syntax `:SYSTem:SHUTdown`
Example `:SYSTEM:SHUTDOWN`
Description The communication will be disconnected, so execute disconnection (TmcFinish).

5.20 VLS Group

The commands in this group deal with the visible light source. This function is valid on models with the /VLS option. You can perform the same operations and make the same settings and queries that you can by using the front panel.

:VLS:ABORt

Function Stops the emission of the visible light source.
Syntax :VLS:ABORt
Example :VLS:ABORT

:VLS:EXECute

Function Starts the emission of the visible light source.
Syntax :VLS:EXECute
Example :VLS:EXECUTE

:VLS:MODulation

Function Sets or queries the modulation of the visible light source.
Syntax :VLS:MODulation {CW|2Hz}
:VLS:MODulation?
Example :VLS:MODULATION CW
:VLS:MODULATION? -> :VLS:MODULATION CW

:VLS:STATe

Function Sets or queries the emission of the visible light source.
Syntax :VLS:STATe <Boolean>
:VLS:STATe?
Example :VLS:STATE ON
:VLS:STATE? -> :VLS:STATE 1
Description :VLS:STATE ON is equivalent to the :VLS:EXECUTE command.
:VLS:STATE OFF is equivalent to the :VLS:ABORT command.

5.21 WAVedata Group

The commands in this group deal with waveform data. There are no front panel keys that correspond to the commands in this group.

:WAVedata:DISPlay:SEND:ASCii?

Function Queries the data of the displayed waveform in ASCII format.
Syntax `:WAVedata:DISPlay:SEND:ASCii?`
Example `:WAVEDATA:DISPLAY:SEND:ASCII?`
`-> 0.000,1.234,...`
Description Queries the waveform data that is displayed on the screen in ASCII format.
Return value format: XXXXX, XXXXX, etc.
Return value example: 45000 is returned for a value of 45.000 dB.

:WAVedata:DISPlay:SEND:BINary?

Function Queries the displayed waveform data in binary format (unit array).
Syntax `:WAVedata:DISPlay:SEND:BINary?`
Example `:WAVEDATA:DISPLAY:SEND:BINARY?`
`-> #6123456ABCDEFGHJ1234567890...`
Description Queries the waveform data that is displayed on the screen in block data (in binary format).
A returned value is the product of multiplying 1000 by a decibel value that is specified to the thousandths place. One value consists of 4 bytes and starts with the most-significant byte (to the thousandths place). For example, a returned value of 1000 indicates a value of 1.000 dB.

:WAVedata:DISPlay:SEND:BINUshort?

Function Queries the displayed waveform data in binary format (ushort array)
Syntax `:WAVedata:DISPlay:SEND:BINUshort?`
Example `:WAVEDATA:DISPLAY:SEND:BINUSHORT?`
`-> #6123456ABCDEFGHJ1234567890`
Description Queries the waveform data that is displayed on the screen in block data (in binary format).
A returned value is the product of multiplying 1000 by a decibel value that is specified to the thousandths place. One value consists of 2 bytes and starts with the most-significant byte (to the thousandths place). For example, a returned value of 1000 indicates a value of 1.000 dB.

:WAVedata:LENGth?

Function Queries the number of waveform data points.
Syntax `:WAVedata:LENGth?`
Example `:WAVEDATA:LENGTH?`
`-> :WAVEDATA:LENGTH 25000`

:WAVedata:OLDType:DISPlay:SEND?

Function Queries the data of the displayed waveform in Dot4 format.
Syntax `:WAVedata:OLDType:DISPlay:SEND?`
Example `:WAVEDATA:OLDTYPE:DISPLAY:SEND?`
`-> #42000...`
Description Queries the waveform data that is displayed on the screen in block data (in binary format).
The returned value is an integer value that is the product of multiplying 250 by the data value.
The most-significant byte and least-significant byte are produced in alternation, starting with the most-significant byte (to the thousandths place).

:WAVedata:OLDType:SEND?

Function Queries the waveform data in Dot4 format.
Syntax `:WAVedata:OLDType:SEND?`
Example `:WAVEDATA:OLDTYPE:SEND?`
`-> #6123456ABCDEFGHJ1234567890...`
Description Queries all the waveform data in block data (in binary format).
The returned value is an integer value that is the product of multiplying 250 by the data value.
The most-significant byte and least-significant byte are produced in alternation, starting with the most-significant byte (to the thousandths place).

:WAVedata:SEND:ASCii?

Function Queries the waveform data in ASCII format.
Syntax `:WAVedata:SEND:ASCii?`
Example `:WAVEDATA:SEND:ASCII?`
`-> 0.000,1.234,...`
Description Queries all the waveform data in ASCII format.
Return value format: XX.XXX, XX.XXX, etc.

:WAVedata:SEND:BINary?

Function Queries all the waveform data in binary format (unit array).
Syntax `:WAVedata:SEND:BINary?`
Example `:WAVEDATA:SEND:BINARY?`
`-> #6123456ABCDEFGHJ1234567890...`
Description Queries all the waveform data in block data (in binary format).
A returned value is the product of multiplying 1000 by a decibel value that is specified to the thousandths place. One value consists of 4 bytes and starts with the most-significant byte (to the thousandths place). For example, a returned value of 1000 indicates a value of 1.000 dB.

:WAVedata:SEND:BINUshort?

Function Queries the waveform data in binary format (ushort array).

Syntax `:WAVedata:SEND:BINUshort?`

Example
```
:WAVedata:SEND:BINUshort?
-> #6123456ABCDEFGHJ1234567890
```

Description Queries all the waveform data in block data (in binary format).
A returned value is the product of multiplying 1000 by a decibel value that is specified to the thousandths place. One value consists of 2 bytes and starts with the most-significant byte (to the thousandths place). For example, a returned value of 1000 indicates a value of 1.000 dB.

:WAVedata:SEND:SIZE

Function Sets or queries the number of acquired data points.

Syntax
```
:WAVedata:SEND:SIZE <NRf>
:WAVedata:SEND:SIZE?
```
<NRf> = 1 to the number of measured points

Example
```
:WAVEDATA:SEND:SIZE 20000
:WAVEDATA:SEND:SIZE?
-> :WAVEDATA:SEND:SIZE 20000
```

Description Values that exceed the range are rounded.

:WAVedata:SEND:STARt

Function Sets or queries the starting distance of the waveform data.

Syntax
```
:WAVedata:SEND:STARt <NRf>
:WAVedata:SEND:STARt?
```
<NRf> = 0 to measured distance (m)

Example
```
:WAVEDATA:SEND:START 20000
:WAVEDATA:SEND:START?
-> :WAVEDATA:SEND:START 20000
```

Description Values that exceed the range are rounded.

5.22 Common Commands

*CLS (Clear Status)

Function Clears all event status registers that are displayed in the status byte register summary.

Syntax `*CLS`

Example `*CLS`

Description
- Clears all queues except the output queue and all event registers except the MAV summary message.
- After this command is executed, the AQ7290 is set to OCIS (Operation Complete Command Idle State) and OQIS (Operation Complete Query Idle State).

*ESE (Standard Event Status Enable)

Function Sets or queries the standard event enable register.

Syntax `*ESE<wsp><integer>`
`*ESE?`
<integer> = 0 to 255

Example `*ESE 251`
`*ESE? -> 251`

Description
- Each item whose bit is set is enabled.
- The ESE is set to its default value when:
  The power is turned on.
  Zero is set.
- The ESE does not change its value when the following commands are executed:
  *RST (Reset)
  *CLS (Clear Status)
  Device clear (DCL, SDC)
- The default value is zero.

*ESR? (Standard Event Status Register)

Function Queries and clears the standard event status register.

Syntax `*ESR?`

Example `*ESR? -> 251`

Description The returned value of this query is not affected by the *ESE (Event Status Enable Register).

*IDN? (Identification)

Function Queries the instrument type and firmware version.

Syntax `*IDN?`

Example `*IDN?`
`-> YOKOGAWA,AQ7290,SN123456789,F1.01`

Description Four data fields delimited by commas are returned.

Field 1:	
Manufacturer	"YOKOGAWA"
Field 2:	
Model	"AQ7290"
Field 3:	
Instrument serial number	"SN123456789"
Field 4:	
Firmware version	"F1.01"

*OPT? (Option)

Function Queries the option information.

Syntax `*OPT?`

Example `*OPT? -> SPM,LAN`
SPM: Standard optical power meter
HPM: High power optical power meter
PC: Power checker
(Port 2 of the AQ7293F is not supported.)
VLS: Visible light source
FST: Fiber surface test function
LAN: Ethernet
SB: Shoulder belt

Description The installed options are returned delimited by commas.

*RST (Reset)

Function Returns all the settings except the communication settings to their factory default values.

Syntax `*RST`

Example `*RST`

Description
- This command stops the operation in progress and resets the AQ7290 to its factory defaults.
- The following items do not change:
  Output queue
  *SRE (Service Request Enable)
  *ESE (Standard Event Status Enable)
  Calibration data that affects the AQ7290 specifications
- Communication settings are not reset.

*SRE (Service Request Enable)

Function Sets or queries the service request enable register.

Syntax
```
*SRE <wsp><integer>
*SRE?
```
<integer> = 0 to 255

Example
```
*SRE 250
*SRE? -> 250
```

Description
- Each item whose bit is set is enabled.
- The SRE is set to its default value when:
  The power is turned on.
  Zero is set.
- The SRE does not change its value when the following commands are executed:
  *RST (Reset)
  *CLS (Clear Status)
  Device clear (DCL, SDC)
- The default value is zero.

*STB? (Read Status Byte)

Function Queries the current status byte register value.

Syntax
```
*STB?
```

Example
```
*STB? -> 251
```

Description The STB is not cleared even if it is read.

*TST? (Self Test)

Function Executes a self-test.

Syntax
```
*TST?
```

Example
```
*TST? -> 0
```

Description The AQ7290 returns 0 if the self-test is successful and 1 if it is not.

6.1 Condition Register

The condition register indicates the internal condition of the instrument.

Condition register
`:STATus:CONDition?`

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	PRT	LS	PME	PMZ	PMM	0	AR	PC	FIA	AS	FILE	AVE	REAL

The meanings of each bit of the condition register are as follows:

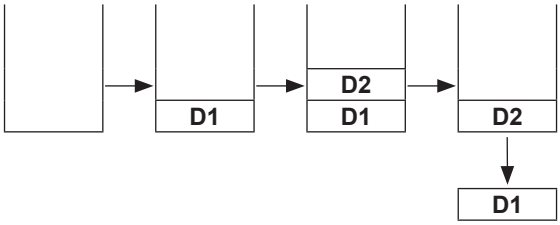
Bit 0	REAL	Set to 1 during real-time measurement.
Bit 1	AVE	Set to 1 during averaged measurement.
Bit 2	FILE	Set to 1 while a file is being accessed.
Bit 3	AS	Set to 1 during auto searching.
Bit 4	FIA	Set to 1 while the instrument checks whether the fiber is in use.
Bit 5	PC	Set to 1 during plug checking.
Bit 6	AR	Set to 1 during auto range measurement.
Bit 8	PMM	Set to 1 during power meter measurement.
Bit 9	PMZ	Set to 1 while the power meter is being reset to 0.
Bit 10	PME	Set to 1 while a measurement error is occurring in the power meter.
Bit 11	LS	Set to 1 while the light source is ON.
Bit 12	PRT	Set to 1 during printing.

6.2 Output and Error Queue

Output queue

The output queue stores query response messages. As shown below, data is stored in order and read from the oldest message first. The output queue is cleared when the instrument is turned off and then back on.

The output queue cannot be emptied using the `*CLS` command. To see whether the output queue is emptyor not, check bit 4 (MAV) of the status byte.



Error queue

When an error occurs, the error queue stores the error number and message. For example, if the instrument receives an incorrect program message from the controller, the error number (113) and the error message (“Undefined header”) are stored in the error queue when the instrument displays the error message.
You can use the `:STATus:ERRor?` query to read the contents of the error queue. Like the output queue, the messages in the error queue are read from the oldest one first.
If the error queue overflows, the last message is replaced with the following message: 350, “Queue overflow”
The error queue is cleared when:

- A *CLS command is received.
- The instrument is turned off and then back on.

You can determine whether or not the error queue is empty by checking bit 2 in the status byte (EAV).

Appendix 1 Error Messages

This section explains communication error messages.

- Messages can be displayed in English or in another language on the instrument. However, when they are read from a PC or other similar device, messages are displayed in English.
- If servicing is necessary to solve the problem indicated by a message, contact your nearest YOKOGAWA dealer.
- Only communication error messages are listed here. For other error messages, see the User's Manual IM AQ7290-01EN.

Error in Execution

Code	Message
50	*OPC/? exists in message.
102	Syntax error.
103	Invalid separator.
104	Data type error.
108	Parameter not allowed.
109	Missing parameter.
111	Header separator error.
112	Program mnemonic too long.
113	Undefined header.
114	Header suffix out of range.
120	Numeric data error.
123	Exponent too large.
124	Too many digits.
128	Numeric data not allowed.
131	Invalid suffix.
134	Suffix too long.
138	Suffix not allowed.
141	Invalid character data.
144	Character data too long.
148	Character data not allowed.
150	String data error.
151	Invalid string data.
158	String data not allowed.
161	Invalid block data.
168	Block data not allowed.
171	Invalid expression.
178	Expression data not allowed.
181	Invalid outside macro definition.
221	Setting conflict.
222	Data out of range.
223	Data invalid
224	Illegal parameter value.
241	Hardware missing.
260	Expression error.
270	Macro error.
272	Macro execution error.
273	Improper macro label.
275	Macro definition too long.
276	Macro recursion error.
277	Macro redefinition not allowed
278	Macro header not found.

Code	Message
350	Queue overflow.
410	Query INTERRUPTED.
420	Query UNTERMINATED.
430	Query DEADLOCKED.
440	Query UNTERMINATED after indefinite response.

Appendix 2 Command Compatibility of Older Models (AQ7280, AQ1210 series, etc.)

Some commands of older models such as the AQ7280 and AQ1210 series are not recommended for use with this instrument due to compatibility of functions. Use the following alternative commands for this instrument.

Deprecated commands (commands for older models)		Alternative commands (commands for this instrument)
• FILE group	→	FSYStem group
• NETWork:WLANApmode group	→	NETWork:WLAN:ACCesspoint group
• NETWork:WLANCommon group	→	NETWork:WLAN:COMMon group

Table of Deprecated and Alternative Commands

FILE group

The commands in this group save, load, and manipulate files by relative paths with respect to the current folder. On this instrument, use the FSYStem group commands.

Note

- The **FSYStem group** commands manipulate files with absolute paths that are consistent throughout the system.
- Commands in the FSYStem group allow operations similar to those in the FILE group of older models, but some operations and parameters are different. For details, see the commands in the FSYStem group in section 5.7.

Function	
Deprecated command **FILE group** (see page App-6)	→ **Alternative command** **FSYStem group** (see section 5.7)
Deletes files.	
`:FILE:DELete:EXECute`	`:FSYStem:FOLDer:DELete` `:FSYStem:FILE:DELete`
Queries the free space on the current drive.	
`:FILE:DRIVe:FREE?`	`:FSYStem:FREE?`
Sets or queries the current drive setting.	

`:FILE:DRIVe:SET`	Not required in FSYStem group (no concept of current)
	`:FSYStem:PARAm:PATH`
	`:FSYStem:PARAm:DESTination:PATH`

Gets the specified file.	
`:FILE:FILE:GET?`	`:FSYStem:FILE:GET?`
Checks whether the specified file exists.	
`:FILE:FILE:EXISt?`	`:FSYStem:EXISt?`
Specifies the name of the file that will be handled with file retrieve, file size retrieve, and file send commands.	
`:FILE:FILE:NAME`	`:FSYStem:PARAm:PATH` `:FSYStem:PARAm:DESTination:PATH`
Gets the size of the specified file.	
`:FILE:FILE:SIZE?`	`:FSYStem:SIZE?`
Gets a list of the current folder.	
`:FILE:FOLDer:LIST?`	`:FSYStem:LIST?`

Function	
Deprecated command **FILE group** (see page App-6)	**Alternative command** → **FSYStem group** (see section 5.7)
Makes a folder.	
`:FILE:FOLDer:MAKE`	`:FSYStem:FOLDer:MAKE`
Sets or queries the current folder name. (Moves one level at a time from the current position. ".." returns to one level higher).	

`:FILE:FOLDer:PATH`	Not required in FSYStem group (no concept of current)
	`:FSYStem:PARAm:PATH`
	`:FSYStem:PARAm:DESTination:PATH`

Replays the file.	
`:FILE:LOAD:EXECute`	`:FSYStem:LOAD`
Sets or queries the comment for saving files.	
`:FILE:SAVE:COMMent1` to `:FILE:SAVE:COMMent10`	Not required for `:FSYStem:SAVE` `:FSYStem:PARAm:NAME`
Saves a file.	
`:FILE:SAVE:EXECute`	`:FSYStem:SAVE`
Sets or queries the ID number for saving files.	
`:FILE:SAVE:ID`	Not required for `:FSYStem:SAVE` `:FSYStem:PARAm:NAME`
Sets or queries the item used in the file name for saving files.	
`:FILE:SAVE:ITEM<x>`	Not required for `:FSYStem:SAVE` `:FSYStem:PARAm:NAME`
Sets or queries the file save separator.	
`:FILE:SAVE:SEParator`	Not required for `:FSYStem:SAVE` `:FSYStem:PARAm:NAME`
Sets or queries the sub number for saving files.	
`:FILE:SAVE:SUB`	Not required for `:FSYStem:SAVE` `:FSYStem:PARAm:NAME`
Sets or queries the file name type for saving files.	
`:FILE:SAVE:TYPE`	Not required for `:FSYStem:SAVE` `:FSYStem:PARAm:NAME`
Queries whether the SOR file image is enabled.	
`:FILE:SOR:VALid?`	`:FSYStem:WAVE:VALid?`
Acquires an SOR file image.	
`:FILE:SOR:GET?`	`:FSYStem:WAVE:GET?`
Gets a list of the sub folders in the current folder.	
`:FILE:SUBFolder:LIST?`	`:FSYStem:LIST?`
Sets or queries the file type for saving files	
`:FILE:TYPE`	Not required for `:FSYStem:SAVE` It is determined automatically from the extension of `:FSYStem:PARAm:NAME.`

NETWork:WLANApmode group

The commands in this group are AQ1210 series commands related to wireless LAN access point mode settings.

On this instrument, use the NETWork:WLAN:ACCesspoint group commands.

Function	
Deprecated command	**Alternative command**
NETWork:WLANApmode group (see page App-12) →	**NETWork:WLAN:ACCesspoint group** (see page 5-50)
Sets or queries the WLAN AP Mode's SSID.	
`:NETWork:WLANApmode:SSID`	`:NETWork:WLAN:ACCesspoint:SSID`
Sets or queries the WLAN AP Mode's encryption type.	
`:NETWork:WLANApmode:ENCrypt`	`:NETWork:WLAN:ACCesspoint:ENCrypt`
Sets or queries the WLAN AP Mode's password.	
`:NETWork:WLANApmode:PASSword`	`:NETWork:WLAN:ACCesspoint:PASSword`
Sets the WLAN AP Mode's channel.	
`:NETWork:WLANApmode:CHannel`	`:NETWork:WLAN:ACCesspoint:CHannel`
Sets or queries the WLAN AP Mode's IP address.	
`:NETWork:WLANApmode:IPADdress`	`:NETWork:WLAN:ACCesspoint:IPADdress`
Sets or queries the WLAN AP Mode's netmask.	
`:NETWork:WLANApmode:NETMask`	`:NETWork:WLAN:ACCesspoint:NETMask`
Sets or queries the start address of the IP addresses that are assigned by the WLAN AP Mode's DHCP server.	
`:NETWork:WLANApmode:DHCPStartaddr`	`:NETWork:WLAN:ACCesspoint:DHCPStartaddr`
Turns the WLAN AP Mode on and off.	
`:NETWork:WLANApmode:STATe`	`:NETWork:WLAN:ACCesspoint:STATe`

NETWork:WLANCommon group

The commands in this group are AQ1210 series commands related to the WLAN module settings.

On this instrument, use the NETWork:WLAN:COMMon group commands.

Function	
Deprecated command	**Alternative command**
NETWork:WLANCommon group (see page App-13) →	**NETWork:WLAN:COMMon group** (see page 5-51)
Queries the WLAN module's FW version.	
`:NETWork:WLANCommon:FWVersion?`	`:NETWork:WLAN:COMMon:FWVersion?`
Queries the WLAN module's HW version.	
`:NETWork:WLANCommon:HWVersion?`	`:NETWork:WLAN:COMMon:HWVersion?`
Queries the WLAN module's MAC address.	
`:NETWork:WLANCommon:MACaddress?`	`:NETWork:WLAN:COMMon:MACaddress?`

Appendix 3 Deprecated Commands (commands for older models)

The following command groups are used on older models such as the AQ7280 and AQ1210.

- FILE Group
- NETWork:WLANApmode Group
- NETWork:WLANCommon Group

The use of these commands is not recommended for this instrument because they have been changed to other commands due to compatibility of functions.
For the alternative commands that can be used on this instrument, see the explanation of each command and “Table of Deprecated and Alternative Commands” in appendix 2.

FILE Group

FILE group commands are those that have been used in the AQ7280 and AQ1210 series. These commands deal with operations that store data and read that data, and performs file operations based on the current folder.
You can perform the same operations and make the same settings and queries that you can from the front panel.

Note

Commands in the FSYStem group of this instrument allow operations similar to those in the FILE group of older models, but some operations and parameters are different.
For details, see section 5.7, “FSYStem Group.”

:FILE:DELete:EXECute

Function Deletes a file or folder.
Syntax `:FILE:DELete:EXECute <String>`
<String> = File name or folder name
Example `:FILE:DELETE:EXECUTE "0001.SOR"`
Description You can also delete folders.

On this instrument, use the following commands.
`:FSYStem:FOLDer:DELete`
`:FSYStem:FILE:DELete`

:FILE:DRIVe:FREE?

Function Queries the free space on the current drive.
Syntax `:FILE:DRIVe:FREE?`
Example `:FILE:DRIVE:FREE? -> 1234567`
Description The value is the number of bytes.

On this instrument, use the following commands.
`:FSYStem:FREE?`

:FILE:DRIVe:SET

Function Sets or queries the current drive setting.
Syntax `:FILE:DRIVe:SET {INTERNAL|SD|USB1|USB2}`
`:FILE:DRIVe:SET?`

INTERNAL:	Internal memory
SD:	SD card (external memory)
USB1:	USB memory 1
USB2:	USB memory 1

Example `:FILE:DRIVE:SET USB1`
`:FILE:DRIVE:SET?`
`-> :FILE:DRIVE:SET INTERNAL`
Description The report output destination will also be the drive specified by this command.

On this instrument, use the following commands.
The **`FSYStem`** group commands do not require a current drive setting (there is no concept of current).
`:FSYStem:PARAm:PATH`
`:FSYStem:PARAm:DESTination:PATH`

Note

Commands in the FSYStem group of this instrument allow operations similar to those in the FILE group of older models, but some operations and parameters are different.
For details, see section 5.7, “FSYStem Group.”

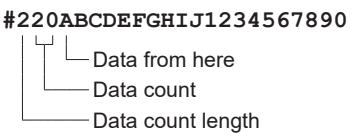
:FILE:FILE:EXISt?

Function Checks whether or not the specified file exists.
Syntax `:FILE:FILE:EXISt?`
Example `:FILE:FILE:EXIST?`
`-> :FILE:FILE:EXIST 1`
Description Returns 1 if the file exists and 0 if the file does not exist.

On this instrument, use the following commands.
`:FSYStem:EXISt?`

:FILE:FILE:GET?

Function Acquires the specified file.
Syntax `:FILE:FILE:GET?`
Example `:FILE:FILE:GET?`
`-> #220ABCDEFGHIJ1234567890`
Description A block data header is attached to the start of the loaded data.



On this instrument, use the following commands.
`:FSYStem:FILE:GET?`

:FILE:FILE:NAME

Function Specifies the file name used for file acquisition, file size acquisition, and file transfer.
Syntax `:FILE:FILE:NAME <String>`
`:FILE:FILE:NAME?`
Example `:FILE:FILE:NAME "0.SOR"`
`:FILE:FILE:NAME?`
`-> :FILE:FILE:NAME "0.sor"`
Description
- The following characters cannot be used in folder names:
  ", *, /, :, <, >, ?, \, |
- The following character strings cannot be used in folder names:
  “AUX”, “CON”, “PRN”, “NUL”, “CLOCK”, “LPT1”, “LPT2”, “LPT3”, “LPT4”, “LPT5”, “LPT6”, “LPT7”, “LPT8”, “LPT9”, “COM1”, “COM2”, “COM3”, “COM4”, “COM5”, “COM6”, “COM7”, “COM8”, “COM9”

On this instrument, use the following commands.
`:FSYStem:PARAm:PATH`
`:FSYStem:PARAm:DESTination:PATH`

:FILE:FILE:SIZE?

Function Acquires the size of the specified file.
Syntax `:FILE:FILE:SIZE?`
Example `:FILE:FILE:SIZE?`
`-> :FILE:FILE:SIZE 230781`
Description **On this instrument, use the following commands.**
`:FSYStem:SIZE?`

:FILE:FOLDer:LIST?

Function Acquires a list of the contents of the current folder
Syntax `:FILE:FOLDer:LIST?`
Example `:FILE:FOLDER:LIST?`
`-> :FILE:FOLDER:`
`LIST "3,ABC.SOR,DEF.SOR,MACRO/"`
Description The information is returned as follows: number of items, file (or folder), . . . , file (or folder).
A slash is attached to the end of folder names.

On this instrument, use the following commands.
`:FSYStem:LIST?`

:FILE:FOLDer:MAKE

Function You can make folders.
Syntax `:FILE:FOLDer:MAKE <String>`
<String> = Folder name
Example `:FILE:FOLDER:MAKE "Data"`
Description **On this instrument, use the following commands.**
`:FSYStem:FOLDer:MAKE`

Note

Commands in the FSYStem group of this instrument allow operations similar to those in the FILE group of older models, but some operations and parameters are different.
For details, see section 5.7, “FSYStem Group.”

:FILE:FOLDer:PATH

Function Sets or queries the current folder name.
Syntax `:FILE:FOLDer:PATH <String>`
`:FILE:FOLDer:PATH?`
<String> = Folder name
Example `:FILE:FOLDER:PATH "AQ7280"`
`:FILE:FOLDER:PATH?`
`-> :FILE:FOLD:PATH "AQ7280"`
Description
- Moves one level at a time relative to the current position (“..” returns to one level higher).
- The following characters cannot be used in folder names.
  ", *, /, :, <, >, ?, \, |
- The following character strings cannot be used in folder names.
  “AUX”, “CON”, “PRN”, “NUL”, “CLOCK”, “LPT1”, “LPT2”, “LPT3”, “LPT4”, “LPT5”, “LPT6”, “LPT7”, “LPT8”, “LPT9”, “COM1”, “COM2”, “COM3”, “COM4”, “COM5”, “COM6”, “COM7”, “COM8”, “COM9”
- Report files are saved to the folder path set with this command.

On this instrument, use the following commands.
The **FSYStem** group commands do not require a current folder name setting (there is no concept of current).
:FSYStem:PARAm:PATH
:FSYStem:PARAm:DESTination:PATH

:FILE:LOAD:EXECute

Function Loads the specified file into the instrument.
Syntax `:FILE:LOAD:EXECute <String>`
<String> = File name
Example `:FILE:LOAD:EXECUTE "0001.SOR"`
Description **On this instrument, use the following commands.**
:FSYStem:LOAD

:FILE:SAVE:COMMent1

Function Sets or queries the save comment 1.
Syntax `:FILE:SAVE:COMMent1 <String>`
`:FILE:SAVE:COMMent1?`
Example `:FILE:SAVE:COMMENT1 "AQ7280"`
`:FILE:SAVE:COMMENT1? -> "AQ7280"`
Description You can enter up to 30 characters.
On this instrument, use the following commands.
There is no need to set comments for
:FSYStem:SAVE.
:FSYStem:PARAm:NAME

:FILE:SAVE:COMMent2

Function Sets or queries the save comment 2.
Syntax `:FILE:SAVE:COMMent2 <String>`
`:FILE:SAVE:COMMent2?`
Example `:FILE:SAVE:COMMENT2 "AQ7280"`
`:FILE:SAVE:COMMENT2? -> "AQ7280"`
Description You can enter up to 30 characters.
On this instrument, use the following commands.
There is no need to set comments for
:FSYStem:SAVE.
:FSYStem:PARAm:NAME

:FILE:SAVE:COMMent3

Function Sets or queries the save comment 3.
Syntax `:FILE:SAVE:COMMent3 <String>`
`:FILE:SAVE:COMMent3?`
Example `:FILE:SAVE:COMMENT3 "AQ7280"`
`:FILE:SAVE:COMMENT3? -> "AQ7280"`
Description You can enter up to 30 characters.
On this instrument, use the following commands.
There is no need to set comments for
:FSYStem:SAVE.
:FSYStem:PARAm:NAME

:FILE:SAVE:COMMent4

Function Sets or queries the save comment 4.
Syntax `:FILE:SAVE:COMMent4 <String>`
`:FILE:SAVE:COMMent4?`
Example `:FILE:SAVE:COMMENT4 "AQ7280"`
`:FILE:SAVE:COMMENT4? -> "AQ7280"`
Description You can enter up to 30 characters.
On this instrument, use the following commands.
There is no need to set comments for
:FSYStem:SAVE.
:FSYStem:PARAm:NAME

:FILE:SAVE:COMMent5

Function Sets or queries the save comment 5.
Syntax `:FILE:SAVE:COMMent5 <String>`
`:FILE:SAVE:COMMent5?`
Example `:FILE:SAVE:COMMENT5 "AQ7280"`
`:FILE:SAVE:COMMENT5? -> "AQ7280"`
Description You can enter up to 30 characters.
On this instrument, use the following commands.
There is no need to set comments for
:FSYStem:SAVE.
:FSYStem:PARAm:NAME

Note

Commands in the FSYStem group of this instrument allow operations similar to those in the FILE group of older models, but some operations and parameters are different.
For details, see section 5.7, “FSYStem Group.”

:FILE:SAVE:COMMent6

Function Sets or queries the save comment 6.
Syntax `:FILE:SAVE:COMMent6 <String>`
`:FILE:SAVE:COMMent6?`
Example `:FILE:SAVE:COMMENT6 "AQ7280"`
`:FILE:SAVE:COMMENT6? -> "AQ7280"`
Description You can enter up to 30 characters.
On this instrument, use the following commands.
There is no need to set comments for
:FSYStem:SAVE.
:FSYStem:PARAm:NAME

:FILE:SAVE:COMMent7

Function Sets or queries the save comment 7.
Syntax `:FILE:SAVE:COMMent7 <String>`
`:FILE:SAVE:COMMent7?`
Example `:FILE:SAVE:COMMENT7 "AQ7280"`
`:FILE:SAVE:COMMENT7? -> "AQ7280"`
Description You can enter up to 30 characters.
On this instrument, use the following commands.
There is no need to set comments for
:FSYStem:SAVE.
:FSYStem:PARAm:NAME

:FILE:SAVE:COMMent8

Function Sets or queries the save comment 8.
Syntax `:FILE:SAVE:COMMent8 <String>`
`:FILE:SAVE:COMMent8?`
Example `:FILE:SAVE:COMMENT8 "AQ7280"`
`:FILE:SAVE:COMMENT8? -> "AQ7280"`
Description You can enter up to 30 characters.
On this instrument, use the following commands.
There is no need to set comments for
:FSYStem:SAVE.
:FSYStem:PARAm:NAME

:FILE:SAVE:COMMent9

Function Sets or queries the save comment 9.
Syntax `:FILE:SAVE:COMMent9 <String>`
`:FILE:SAVE:COMMent9?`
Example `:FILE:SAVE:COMMENT9 "AQ7280"`
`:FILE:SAVE:COMMENT9? -> "AQ7280"`
Description You can enter up to 30 characters.
On this instrument, use the following commands.
There is no need to set comments for
:FSYStem:SAVE.
:FSYStem:PARAm:NAME

:FILE:SAVE:COMMent10

Function Sets or queries the save comment 10.
Syntax `:FILE:SAVE:COMMent10 <String>`
`:FILE:SAVE:COMMent10?`
Example `:FILE:SAVE:COMMENT10 "AQ7280"`
`:FILE:SAVE:COMMENT10? -> "AQ7280"`
Description You can enter up to 30 characters.
On this instrument, use the following commands.
There is no need to set comments for
:FSYStem:SAVE.
:FSYStem:PARAm:NAME

:FILE:SAVE:EXECute

Function Saves the file.
Syntax `:FILE:SAVE:EXECute`
Example `:FILE:SAVE:EXECUTE`
Description To check for errors after saving, first make sure that files are not being accessed by executing the STATus:CONDition? command, and then execute the STATus:ERRor command.
(See section 6.1, “Status Register.”)
On this instrument, use the following commands.
:FSYStem:SAVE

:FILE:SAVE:ID

Function Sets or queries the ID number saved with a file.
Syntax `:FILE:SAVE:ID <NRf>`
`:FILE:SAVE:ID?`
<NRf> = 0 to 9999
Example `:FILE:SAVE:ID 100`
`:FILE:SAVE:ID? -> :FILE:SAVE:ID 100`
Description **On this instrument, use the following commands.**
There is no need to set ID numbers for
:FSYStem:SAVE.
:FSYStem:PARAm:NAME

Note

Commands in the FSYStem group of this instrument allow operations similar to those in the FILE group of older models, but some operations and parameters are different.
For details, see section 5.7, “FSYStem Group.”

:FILE:SAVE:ITEM<x>

Function Sets or queries the items used in the name when a file is saved.

Syntax
```
:FILE:SAVE:ITEM<x> {NONE|WAVELENGTH|
ID|COMMENT1|COMMENT2|COMMENT3|
COMMENT4|COMMENT5|COMMENT6|COMMENT7|
COMMENT8|COMMENT9|COMMENT10|COMPANY|
OPERATOR|CABLEID|FIBERID|FIBERTYPE|
CABLECODE|ORIGINATING|TERMINATING}
:FILE:SAVE:ITEM<x>?
```
<x> = 1 to 10

NONE:	None
WAVELENGTH:	Wavelength
ID:	ID number
COMMENT1 to 10:	Comment 1 to 10
COMPANY:	Company name
OPERATOR:	Operator name
CABLEID:	Cable ID
FIBERID:	Fiber ID
FIBERTYPE:	Fiber type
CABLECODE:	Cable code
ORIGINATING:	Start position
TERMINATING:	End position

Example
```
:FILE:SAVE:ITEM1 WAVELENGTH
:FILE:SAVE:ITEM1?
-> :FILE:SAVE:ITEM WAVELENGTH
```

Description <If <x>=1, NONE cannot be specified.

On this instrument, use the following commands.
There is no need to saved item settings for
:FSYStem:SAVE.
:FSYStem:PARAm:NAME

:FILE:SAVE:SEParator

Function Sets or queries the file name separator when a file is saved.

Syntax
```
:FILE:SAVE:SEParator {NONE|UNDERBAR|
TILDE|HAT}
:FILE:SAVE:SEParator?
```

NONE:	None
UNDERBAR:	Underscore “_”
TILDE:	Tilde “~”
HAT:	Hat “^”

Example
```
:FILE:SAVE:SEPARATOR UNDERBAR
:FILE:SAVE:SEPARATOR?
-> :FILE:SAVE:SEPARATOR UNDERBAR
```

Description **On this instrument, use the following commands.**
There is no need to set separators for
:FSYStem:SAVE.
:FSYStem:PARAm:NAME

:FILE:SAVE:SUB

Function Sets or queries the tape number type saved with a file.

Syntax
```
:FILE:SAVE:SUB {OFF|AB|AC|AD|AE|AF|
AG|AH}
:FILE:SAVE:SUB?
```

OFF:	OFF	AE:	a-e
AB:	a-b	AF:	a-f
AC:	a-c	AG:	a-g
AD:	a-d	AH:	a-h

Example
```
:FILE:SAVE:SUB AH
:FILE:SAVE:SUB? -> :FILE:SAVE:SUB OFF
```

Description **On this instrument, use the following commands.**
There is no need to set sub numbers for
:FSYStem:SAVE.
:FSYStem:PARAm:NAME

:FILE:SAVE:TYPE

Function Sets or queries the file name type when a file is saved.

Syntax
```
:FILE:SAVE:TYPE {NO|COMMENT|CMNO|
NOCM|CMWLNO|NOCMWL|WLCMNO}
:FILE:SAVE:TYPE?
```

NO:	Number
COMMENT:	Comment
CMNO:	Comment + number
NOCM:	Number + comment
CMWLNO:	Comment + wavelength + number
NOCMWL:	Number + comment + wavelength
WLCMNO:	Wavelength + comment + number

Example
```
:FILE:SAVE:TYPE COMMENT
:FILE:SAVE:TYPE?
-> :FILE:SAVE:TYPE CMWLNO
```

Description
- This command is a simplified version of :FILE:SAVE:ITEM<x>.
- The instrument returns “UNKNOWN” for items other than number (ID no.), comment, and wavelength.

On this instrument, use the following commands.
There is no need to set file name types for
:FSYStem:SAVE.
:FSYStem:PARAm:NAME

Note

Commands in the FSYStem group of this instrument allow operations similar to those in the FILE group of older models, but some operations and parameters are different.
For details, see section 5.7, “FSYStem Group.”

:FILE:SOR:GET?

Function Acquires an SOR file image.

Syntax `:FILE:SOR:GET?`

Example `:FILE:SOR:GET? -> #6123456ABCDEFGHJ`

Description
- You can perform acquisition when SOR measurement data is enabled. You can use :FILE:SOR:VALid? to query whether the SOR measurement data is enabled.
- The SOR file image is acquired as block data (in binary format).

On this instrument, use the following commands.
`:FSYStem:WAVE:GET?`

:FILE:SOR:VALid?

Function Queries whether the SOR file image is enabled.

Syntax `:FILE:SOR:VALid?`

Example
```
:FILE:SOR:VALID?
-> :FILE:SOR:VALID 1
```

Description Measured data
0: Disabled
1: Enabled

On this instrument, use the following commands.
`:FSYStem:WAVE:VALid?`

:FILE:SUBFolder:LIST?

Function Acquires a list of the subfolders in the current folder.

Syntax `:FILE:SUBFolder:LIST?`

Example
```
:FILE:SUBFOLDER:LIST?
-> :FILE:SUBFOLDER:
LIST "3,ABC/,DEF/,MACRO/"
```

Description Returns the number of results followed by the file and folder names. Folder names are followed by slashes.

On this instrument, use the following commands.
`:FSYStem:LIST?`

:FILE:TYPE

Function Sets or queries the data save format when a file is saved.

Syntax
```
:FILE:TYPE {SET|SOR|CSV_WAVE|
CSV_EVENT|BMP|PNG|JPG|CFG}
:FILE:TYPE?
```
SET: A setup file
SOR: A file that conforms to Telcordia SR-4731
CSV_WAVE: A CSV waveform file
CSV_EVENT: A CSV event file
BMP: A BMP file
PNG: A PNG file
JPG: A JPG file
CFG: System setup data file

Example
```
:FILE:TYPE SOR
:FILE:TYPE? -> :FILE:TYPE CSV_WAVE
```

Description SOR, CSV_WAVE, and CSV_ENENT are only valid when the function mode is set to OTDR.

On this instrument, use the following commands.
There is no need to set file types for **`:FSYStem:SAVE.`**
It is determined automatically from the extension of **`:FSYStem:PARAm:NAME.`**

NETWork:WLANApmode Group

The commands in this group are AQ1210 series commands related to wireless LAN access point mode settings.

On this instrument, these commands are planned to be discontinued in the future.

:NETWork:WLANApmode:SSID

Function Sets or queries the ID name (SSID) of the network WLAN application.

Syntax
```
:NETWork:WLANApmode:SSID <String>
:NETWork:WLANApmode:SSID?
```

Example
```
:NETWORK:WLANAPMODE:SSID "12345678"
:NETWork:WLANAPMODE:SSID?
-> :NETWork:WLANAPMODE:SSID "12345678"
```

Description **On this instrument, use the following commands.**
:NETWork:WLAN:ACCesspoint:SSID

:NETWork:WLANApmode:ENCrypt

Function Sets or queries the encryption mode of the network WLAN application.

Syntax
```
:NETWork:WLANApmode:ENCrypt {NONE|
WPA2_PSK}
:NETWork:WLANApmode:ENCrypt?
```

Example
```
:NETWORK:WLANAPMODE:
ENCRYPT WPA2_PSK
:NETWORK:WLANAPMODE:ENCRYPT?
-> :NETWORK:WLANAPMODE:
ENCRYPT WPA2_PSK
```

Description **On this instrument, use the following commands.**
:NETWork:WLAN:ACCesspoint:ENCrypt

:NETWork:WLANApmode:PASSword

Function Sets or queries the password of the network WLAN application.

Syntax
```
:NETWork:WLANApmode:PASSword <String>
:NETWork:WLANApmode:PASSword?
```

Example
```
:NETWORK:WLANAPMODE:
PASSWORD "12345678"
:NETWORK:WLANAPMODE:PASSWORD?
-> :NETWORK:WLANAPMODE:
PASSWORD "12345678"
```

Description On this instrument, use the following commands.
:NETWork:WLAN:ACCesspoint:PASSword

:NETWork:WLANApmode:CHannel

Function Sets or queries the channel of the network WLAN application.

Syntax
```
:NETWork:WLANApmode:CHannel {AUTO|
CH<x>}
:NETWork:WLANApmode:CHannel?
```
<x> = 1 to 11

Example
```
:NETWORK:WLANAPMODE:CHANNEL CH1
:NETWORK:WLANAPMODE:CHANNEL?
-> :NETWORK:WLANAPMODE:CHANNEL CH1
```

Description **On this instrument, use the following commands.**
:NETWork:WLAN:ACCesspoint:CHannel

:NETWork:WLANApmode:IPADdress

Function Sets or queries the IP address of the network WLAN application.

Syntax
```
:NETWork:WLANApmode:IPADdress <String>
:NETWork:WLANApmode:IPADdress?
```

Example
```
:NETWORK:WLANAPMODE:
IPADDRESS "192.168.0.1"
:NETWORK:WLANAPMODE:IPADDRESS?
-> :NETWORK:WLANAPMODE:
IPADDRESS "192.168.0.1"
```

Description **On this instrument, use the following commands.**
:NETWork:WLAN:ACCesspoint:IPADdress

:NETWork:WLANApmode:NETMask

Function Sets or queries the subnet mask of the network WLAN application.

Syntax
```
:NETWork:WLANApmode:NETMask <String>
:NETWork:WLANApmode:NETMask?
```

Example
```
:NETWORK:WLANAPMODE:
NETMASK "255.255.255.0"
:NETWORK:WLANAPMODE:NETMASK?
-> :NETWORK:WLANAPMODE:
NETMASK "255.255.255.0"
```

Description **On this instrument, use the following commands.**
:NETWork:WLAN:ACCesspoint:NETMask

:NETWork:WLANApmode:DHCPStartaddr

Function Sets or queries the start address of the IP addresses that are assigned by the DHCP server of the network WLAN application.

Syntax
```
:NETWork:WLANApmode:
DHCPStartaddr <String>
:NETWork:WLANApmode:DHCPStartaddr?
```

Example
```
:NETWORK:WLANAPMODE:
DHCPSTARTADDR "192.168.0.100"
:NETWORK:WLANAPMODE:DHCPSTARTADDR?
-> :NETWORK:WLANAPMODE:
DHCPSTARTADDR "192.168.0.100"
```

Description **On this instrument, use the following commands.**
:NETWork:WLAN:ACCesspoint:DHCPStartaddr

:NETWork:WLANApmode:STATe

Function Sets or queries the network WLAN application usage.

Syntax
```
:NETWork:WLANApmode:STATe <Boolean>
:NETWork:WLANApmode:STATe?
```

Example
```
:NETWORK:WLANAPMODE:STATE ON
:NETWORK:WLANAPMODE:STATE?
-> :NETWORK:WLANAPMODE:STATE 1
```

Description The WLAN application becomes available when the WLAN application usage is set to ON.
On this instrument, use the following commands.
:NETWork:WLAN:ACCesspoint:STATe

NETWork:WLANCommon Group

The commands in this group are AQ1210 series commands related to the WLAN module settings. On this instrument, these commands are planned to be discontinued in the future.

:NETWork:WLANCommon:FWVersion?

Function Queries the WLAN module's firmware version.

Syntax
```
:NETWork:WLANCommon:FWVersion?
```

Example
```
:NETWORK:WLANCOMMON:FWVERSION?
-> :NETWORK:WLANCOMMON:
FWVERSION 1695438116
```

Description **On this instrument, use the following commands.**
:NETWork:WLAN:COMMon:FWVersion?

:NETWork:WLANCommon:HWVersion?

Function Queries the WLAN module's hardware version.

Syntax
```
:NETWork:WLANCommon:HWVersion?
```

Example
```
:NETWORK:WLANCOMMON:HWVERSION?
-> :NETWORK:WLANCOMMON:
HWVERSION 18448
```

Description **On this instrument, use the following commands.**
:NETWork:WLAN:COMMon:HWVersion?

:NETWork:WLANCommon:MACaddress?

Function Queries the WLAN module's MAC address.

Syntax
```
:NETWork:WLANCommon:MACaddress?
```

Example
```
:NETWORK:WLANCOMMON:MACADDRESS?
-> :NETWORK:WLANCOMMON:
MACADDRESS "XXXXXXXXXXXX"
```

Description **On this instrument, use the following commands.**
:NETWork:WLAN:COMMon:MACaddress?

Index