

SCPI Programming Manual

S331P Site Master™

Ultraportable Cable and Antenna Analyzer Featuring Classic and Advanced Modes

150 kHz to 4 GHz (S331P-0704 Option)

150 kHz to 6 GHz (S331P-0706 Option)



NOTICE

Anritsu Company has prepared this manual for use by Anritsu Company personnel and customers as a guide for the proper installation, operation and maintenance of Anritsu Company equipment and computer programs. The drawings, specifications, and information contained herein are the property of Anritsu Company, and any unauthorized use or disclosure of these drawings, specifications, and information is prohibited; they shall not be reproduced, copied, or used in whole or in part as the basis for manufacture or sale of the equipment or software programs without the prior written consent of Anritsu Company.

UPDATES

Updates, if any, can be downloaded from the Anritsu Website at:

<http://www.anritsu.com>

For the latest service and sales contact information in your area, please visit:

<http://www.anritsu.com/contact-us>

Table of Contents

Chapter 1—General Information

1-1	Introduction	1-1
1-2	Remote Operation Setup and Interface	1-1
	Ethernet Interface Connection and Setup	1-2
1-3	Sending SCPI Commands	1-3
	Ethernet Connectivity	1-3

Chapter 2—Programming with SCPI

2-1	Introduction	2-1
2-2	Introduction to SCPI Programming	2-1
	SCPI Common Commands	2-2
	SCPI Required Commands	2-2
	SCPI Optional Commands	2-2
2-3	Subsystem Commands	2-3
	Command Names	2-3
	Hierarchical Command Structure	2-4
	Query Commands	2-5
	Data Parameters	2-6
	Unit Suffixes	2-6
2-4	Notational Conventions	2-7
2-5	Notational Examples	2-8
	Command Terminators	2-8
2-6	Formatting Conventions	2-9

Chapter 3—All Mode Commands

Chapter 4—Cable & Antenna Analyzer Mode Commands

Appendix A—List of Commands by Mode

Appendix B—List of Commands, Alphabetical

Chapter 1 — General Information

1-1 Introduction

This SCPI Programming Manual provides information for remote operation of the Site Master S331P Cable and Antenna Analyzer using SCPI commands sent from a user application on an external controller through an Ethernet connection or on the local S331P controller.

This Programming Manual includes the following:

- An overview of the Ethernet connection to the instrument.
- An overview of Standard Commands for Programmable Instruments (SCPI) command structure and conventions.
- The IEEE common commands that are supported by the instruments.
- A complete listing and description of all the SCPI commands that can be used to remotely control functions of the instrument. The commands are organized by measurement mode starting in [Chapter 3](#).

This manual is intended to be used in conjunction with the Site Master S331P User Guide. Refer to the instrument user guide for general information about the instrument, including equipment setup and operating instructions. This chapter provides a general description of remote programming setup and interface using Ethernet, and sending SCPI commands to the instrument.

1-2 Remote Operation Setup and Interface

Remote operation of the instrument is accomplished via an Ethernet interface to the S331P controlling tablet or PC. The Site Master S331P software fully supports the IEEE-802.3 standard. Make sure to install the S331P software onto your computer in order to communicate with S331P instrument using SCPI commands, since S331P doesn't support SCPI interface without the S331P PC software. Most Site Master functions can be controlled via an Ethernet connection to a PC that is connected directly to the S331P controller (with an Ethernet cross-over cable) or through a network. Alternatively, the user application can be installed on the S331P controller to interface directly with the S331P software.

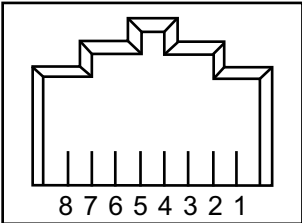
Ethernet networking uses a bus or star topology in which all of the interfacing devices are connected to a central cable called the bus, or are connected to a hub. Ethernet uses the CSMA/CD access method to handle simultaneous transmissions over the bus. CSMA/CD stands for Carrier Sense Multiple Access/Collision Detection. This standard enables network devices to detect simultaneous data channel usage (called a collision) and provides for a contention protocol. When a network device detects a collision, the CSMA/CD standard dictates that the data is retransmitted after waiting a random amount of time. If a second collision is detected, then the data are again retransmitted after waiting twice as long. This is known as exponential back off.

Ethernet Interface Connection and Setup

The TCP/IP setup requires the following:

- **IP Address:** Every computer/electronic device in a TCP/IP network requires an IP address. An IP address has four numbers (each between 0 and 255) separated by periods. For example: 128.111.122.42 is a valid IP address.
- **Subnet Mask:** The subnet mask distinguishes the portion of the IP address that is the network ID from the portion that is the station ID. The subnet mask 255.255.0.0, when applied to the IP address given above, would identify the network ID as 128.111 and the station ID as 122.42. All stations in the same local area network should have the same network ID, but different station IDs.
- **Default Gateway:** A TCP/IP network can have a gateway to communicate beyond the LAN that is identified by the network ID. A gateway is a computer or electronic device that is connected to two different networks and can move TCP/IP data from one network to the other. A single LAN that is not connected to other LANs requires a default gateway setting of 0.0.0.0. If you have a gateway, then the default gateway would be set to the appropriate value of your gateway.
- **Ethernet Address:** An Ethernet address (also known as a MAC address) is a unique 48-bit value that identifies a network interface card to the rest of the network. Every network card has a unique Ethernet address permanently stored into its memory.

Table 1-1. 8-pin Ethernet RJ45 Connector Pinout Diagram

			
Pin	Name	Description	Wire Color
1	TX+	Transmit data (> +3 volts)	White/Orange
2	TX–	Transmit data (< –3 volts)	Orange
3	RX+	Receive data (> +3 volts)	White/Green
4	—	Not used (common mode termination)	Blue
5	—	Not used (common mode termination)	White/Blue
6	RX–	Receive data (< –3 volts)	Green
7	—	Not used (common mode termination)	White/Brown
8	—	Not used (common mode termination)	Brown

Note

You may need to consult your network documentation or network administrator for assistance in configuring your network setup.

The S331P controller IP address is set automatically by using Dynamic Host Configuration Protocol (DHCP). DHCP is an Internet protocol that automates the process of setting IP addresses for devices that use TCP/IP, and is the most common method of configuring a device for network use. After the Ethernet cable is connected to the S331P controller, go to System, Status, Connectivity Info to view the IP address that the instrument has been assigned. If the user application runs on the S331P controller, the application should access the local host (127.0.0.1).

1-3 Sending SCPI Commands

SCPI commands can be sent to the instrument through any Virtual Instrument Software Architecture (VISA) controller. VISA is a commonly used API in the Test and Measurement industry for communicating with instruments from a PC. The physical connection between the PC and the instrument is USB or Ethernet. NI-VISA is the National Instruments implementation of the VISA I/O standard. Information and downloads are available at: <http://www.ni.com/visa/>

The following example describes the verification that a VISA controller can interact with the instrument. The images shown and the instructions for your instrument and software may differ from the examples.

Ethernet Connectivity

1. On the PC, run NI Measurement & Automation Explorer or VISA Interactive Control and create a new LAN Resource under Network Devices. Add the TCP/IP resource using a Manual Entry of Raw Socket, as shown in the example below.

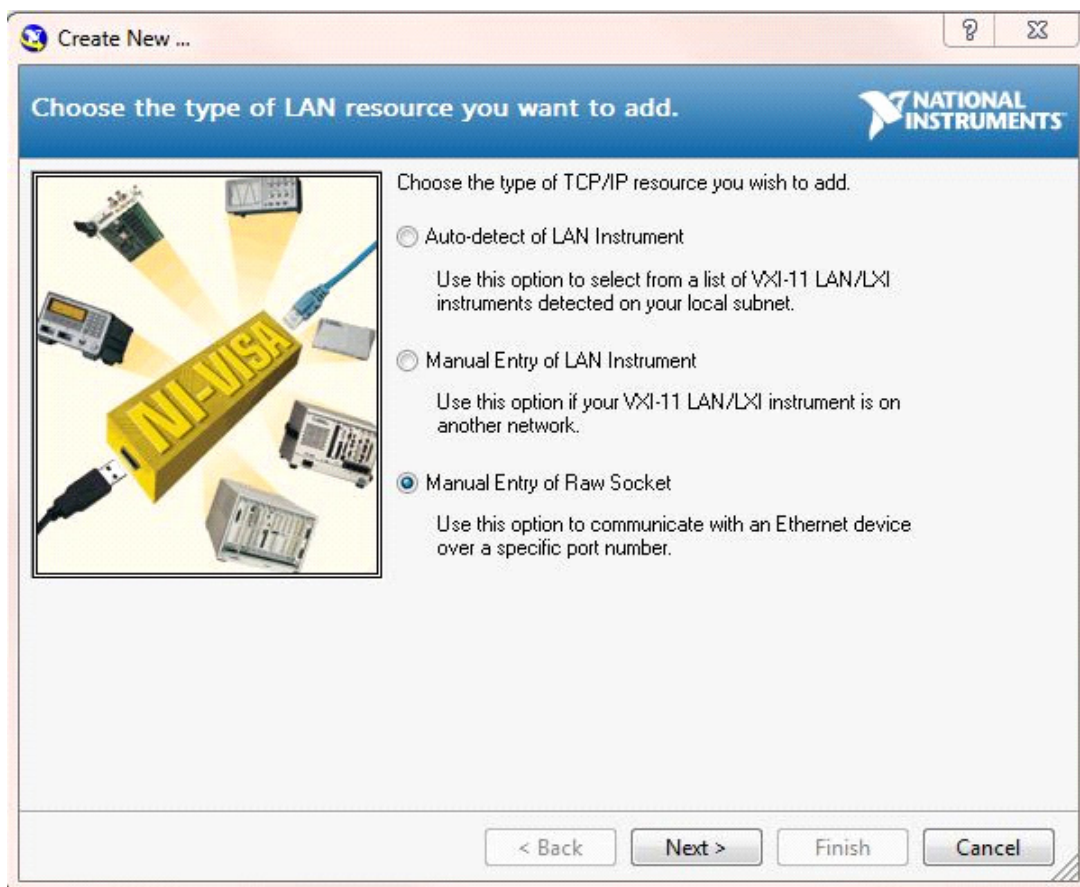


Figure 1-1. NI VISA Interactive Control LAN Resource Addition Using Raw Socket

- 2. Enter the IP address that the instrument has acquired (On the S331P, go to System, Status, Connectivity Info, then enter the IP address as shown in [Figure 1-2](#).
- 3. Enter port 9001 and validate the connection (note that the latest versions of Windows 10 and above may require port 59001). If the connection is not validated, enter the other port number and try validating the connection again.

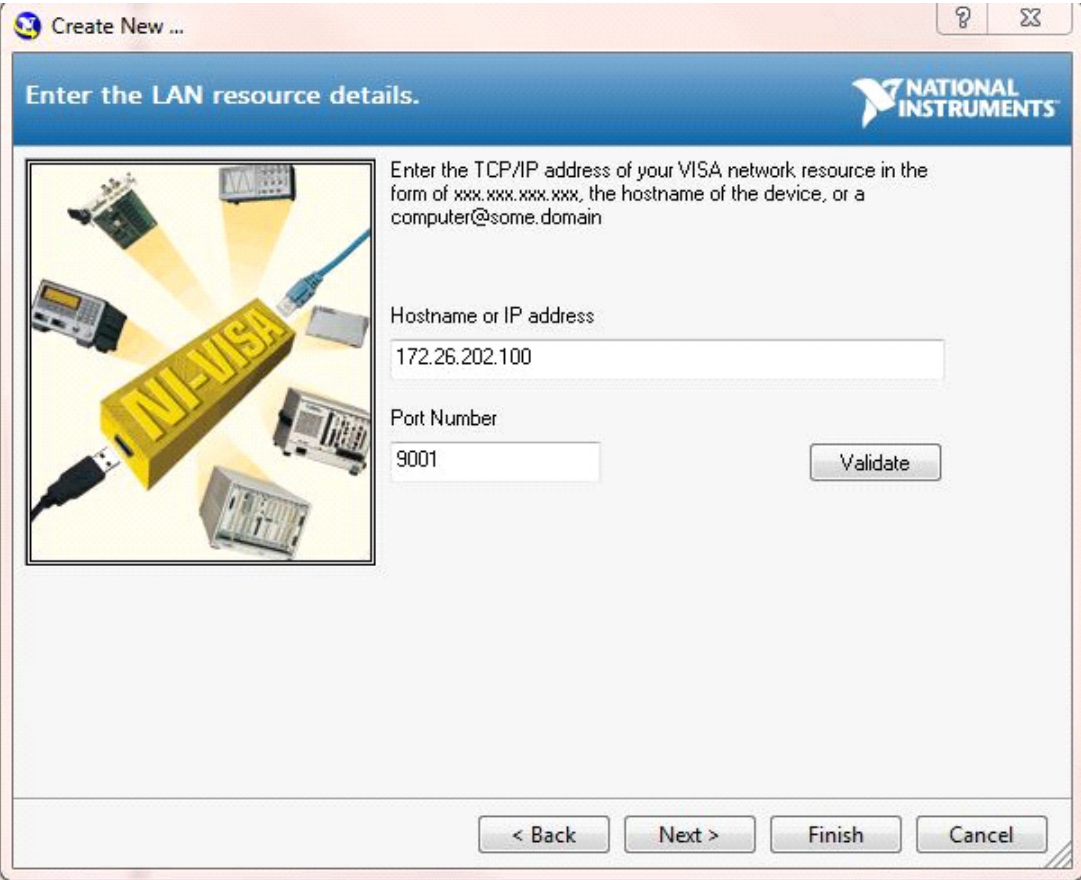


Figure 1-2. NI VISA Interactive Control LAN Resource Settings of IP Address and Port Number

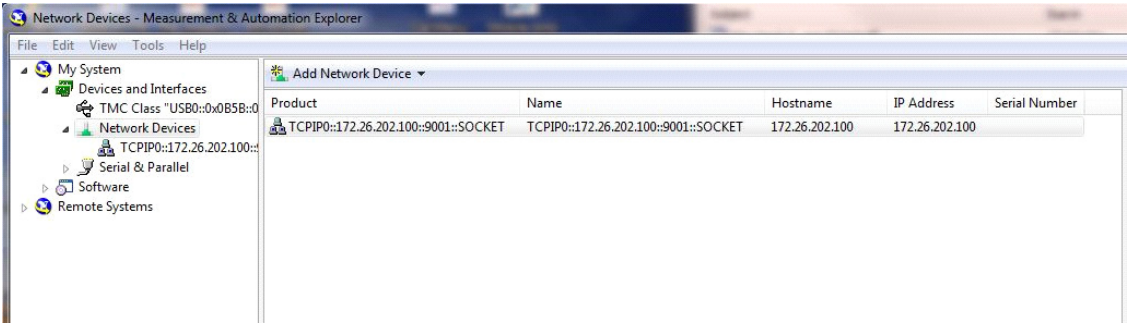


Figure 1-3. NI VISA Interactive Control LAN Resource Validated

4. Select the Configuration I/O settings tab and verify that the Termination Methods are set as shown in the example below.

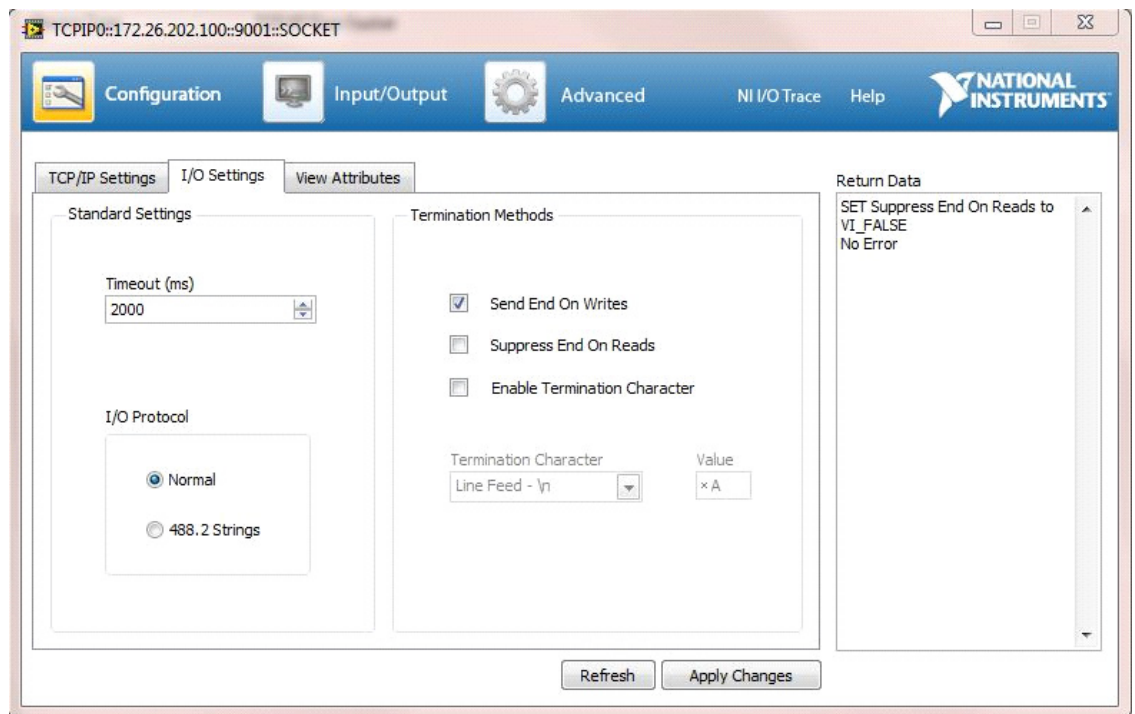


Figure 1-4. NI VISA Interactive Control LAN Resource I/O Termination Method Settings

5. Select the Input/Output Basic I/O tab and execute the default *IDN? Query. If the PC is connected to the instrument the command returns the following information from the Buffer: manufacturer name (“Anritsu”), model number/options, serial number, and firmware package number, as shown in the example below.

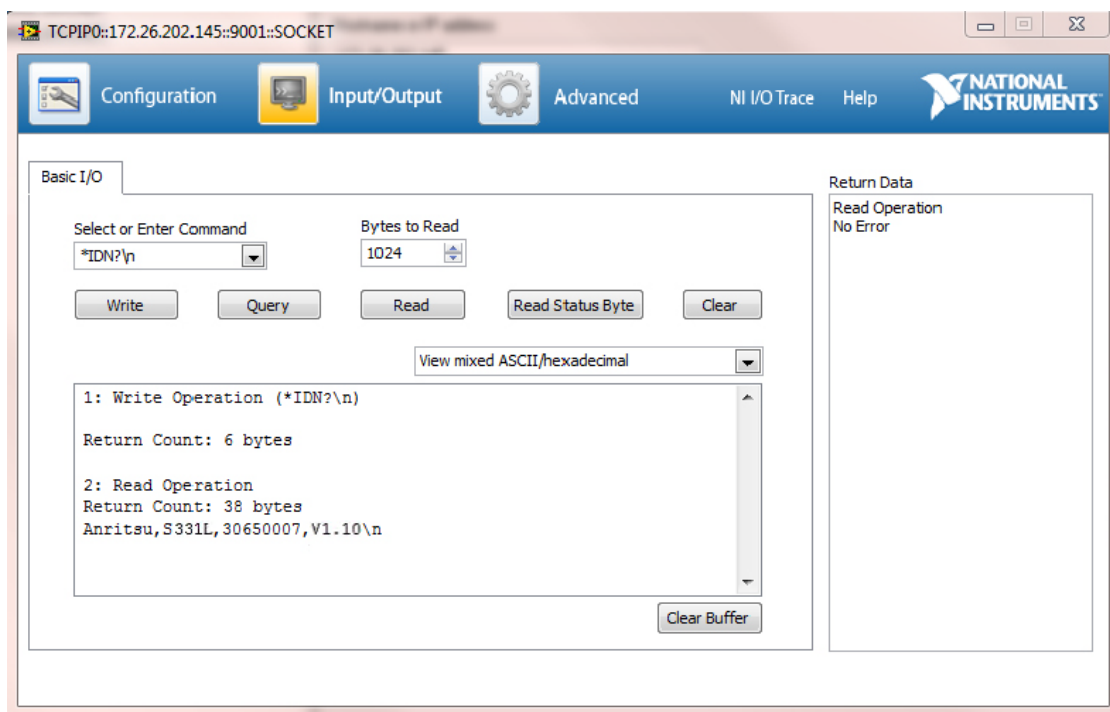


Figure 1-5. NI VISA Interactive Control USB Basic I/O Tab

Note

When sending SCPI commands over Ethernet, it is required to send a newline termination character at the end of each command. In the example above, a newline character (“\n” in this case, but could be different depending on your programming environment) was used to terminate the *IDN? command.

Note

When sending query commands over raw socket, the entire buffer must be read before the next query command is sent. Each query result is terminated by a newline to help identify the end of the query response. Query read operations could be broken into multiple reads, if necessary.

Note

When using raw socket connections, you must close a session before opening a new one or before switching to a new protocol (such as USB). If you try to open a new session or switch protocols without first closing the previously opened session, you may lose communications with the instrument and not be able to reconnect until you reboot the instrument.

Chapter 2 — Programming with SCPI

2-1 Introduction

This chapter provides an introduction to Standard Commands for Programming Instruments (SCPI) programming that includes descriptions of the command types, hierarchical command structure, command subsystems, data parameters, and notational conventions.

2-2 Introduction to SCPI Programming

Anritsu instruments can be operated with the use of SCPI commands. SCPI is intended to give the user a consistent environment for program development. It does so by defining controller messages, instrument responses, and message formats for all SCPI compatible instruments. SCPI commands are messages to the instrument to perform specific tasks. The command set includes:

- [“SCPI Common Commands” on page 2-2](#)
- [“SCPI Required Commands” on page 2-2](#)
- [“SCPI Optional Commands” on page 2-2](#)

Note

The Site Master follows the SCPI standard, but is not fully compliant with that standard. The main reason that the Site Master is not fully compliant is because it does not support all of the required SCPI commands, and because it uses some exceptions in the use of short form and long form command syntax.

SCPI Common Commands

Some common commands are defined in the IEEE-488.2 standard and must be implemented by all SCPI compatible instruments. These commands are identified by the asterisk (*) at the beginning of the command keyword. These commands are defined to control instrument status registers, status reporting, synchronization, and other common functions. For example, *IDN? is a common command supported by the Microwave Site Master.

SCPI Required Commands

The required SCPI commands supported by the instrument are listed in the [Table 2-1](#).

Table 2-1. SCPI Required Commands

:STATus
:SYSTem

SCPI Optional Commands

[Table 2-2](#) lists the optional SCPI commands that comprise the majority of the command set described in this document. These commands control most of the programmable functions of the instrument.

Table 2-2. SCPI Optional Commands

:ABORt	:FETCh	:MEASure	:TRACe
:CALCulate	:FORMat	:MMEMory	:TRIGger
:CALibration	:INITiate	:READ	:UNIT
:CONFigure	:INPut	:SENSe	: [SENSe]
:DISPlay	:INSTrument	:SOURce	

The SCPI optional commands are sorted by measurement modes and commands may be repeated in more than one mode.

- [Chapter 3, “All Mode Commands”](#)
- [Chapter 4, “Cable & Antenna Analyzer Mode Commands”](#)

2-3 Subsystem Commands

Subsystem commands control all instrument functions and some general purpose functions. All subsystem commands are identified by the colon used between keywords, as in `:INITiate:CONTinuous`.

The following information is provided for each subsystem command described in the following chapters.

- The command name, see “Command Names” on page 2-3.
- The path from the subsystem root command, see “Hierarchical Command Structure” on page 2-4.
- The query form of the command (if applicable), see “Query Commands” on page 2-5.
- A description of the purpose of the command.
- The data parameters used as arguments for the command, see “Data Parameters” on page 2-6. This may include the parameter type and the available parameter choices.

Command Names

Typical SCPI commands consist of one or more keywords, parameters, and punctuation. SCPI command keywords can be a mixture of upper and lower case characters. Except for common commands, each keyword has a long and a short form. In this manual, the long form is presented with the short form in upper case and the remainder in lower case. For example, the long form of the command keyword to control the instrument display is `:DISPlay`.

The short form keyword is usually the first four characters of the long form (example: `DISP` for `DISPlay`). The exception to this is when the long form is longer than four characters and the fourth character is a vowel. In such cases, the vowel is dropped and the short form becomes the first three characters of the long form. Example: the short form of the keyword `:POWer` is `:POW`.

Some command keywords may have a numeric suffix to differentiate between multiple instrument features such as multiple trace options. For example, `:CALCulate#:DATA? FDATA|SDATA|FMEM|SMEM` can result in two different commands, one for trace 1 `:CALC1:DATA? FDATA` and another for trace 2 `:CALC2:DATA? FMEM`.

Note	If a numeric suffix is not included in a command, the first option is implied. Curly brackets <code>{ }</code> designate optional keyword or command parameters. Square brackets <code>[]</code> designate optional command keywords. For example, the command <code>:TRACe[:DATA]? {1 2}</code> can be sent as <code>:TRACe?</code> or <code>:TRACe? 1</code> , or as <code>:TRAC?</code> or <code>:TRAC? 1</code> to obtain data from trace 1.
-------------	---

As with any programming language, the exact command keywords and command syntax must be used. The syntax of the individual commands is described in detail in the programming command chapters. Unrecognized versions of long form or short form commands, or improper syntax, will generate an error.

Long Format vs. Short Format

Each keyword has a long format and a short format. The start frequency can be specified by :SENSe:FREQuency:STARt or :SENS:FREQ:STAR. The capital letters in the command specification indicate the short form of the command. A mixture of the entire short form elements with entire long form elements of each command is acceptable. For example, :SENS:FREQuency:STAR is an acceptable form of the command. However, :SENS:FREQuen:STA is not an acceptable form of the command because :FREQuen is not the entire short or long form of the command element.

Hierarchical Command Structure

All SCPI commands, except the common commands, are organized in a hierarchical structure similar to the inverted tree file structure used in most computers. The SCPI standard refers to this structure as “the Command Tree.” The command keywords that correspond to the major instrument control functions are located at the top of the command tree. The root command keywords for the SCPI command set are shown in [Figure 2-1](#).

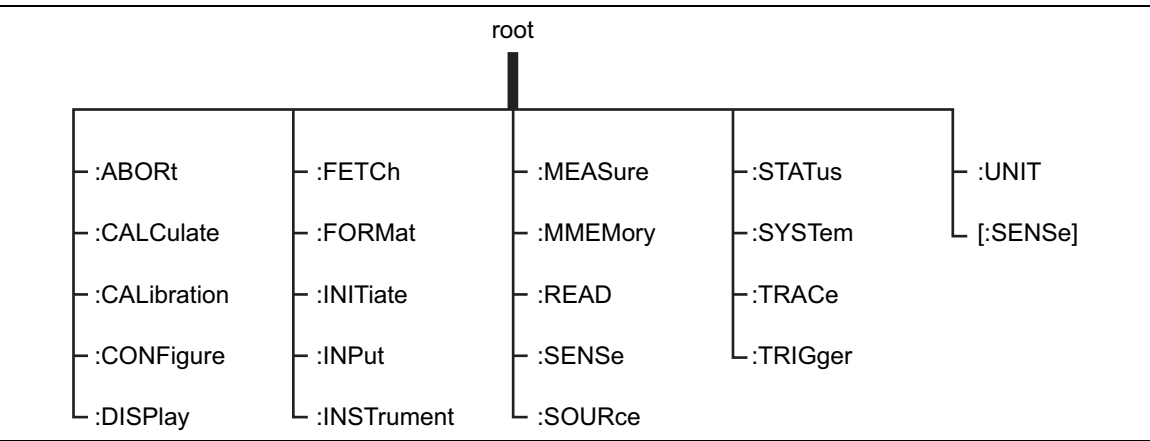


Figure 2-1. SCPI Command Tree

All instrument SCPI commands, except the :ABORt command, have one or more subcommands (keywords) associated with them to further define the instrument function to be controlled. The subcommand keywords may also have one or more associated subcommands (keywords). Each subcommand level adds another layer to the command tree. The command keyword and its associated subcommand keywords form a portion of the command tree called a command subsystem. The :CONFigure command subsystem is shown in [Figure 2-2](#).

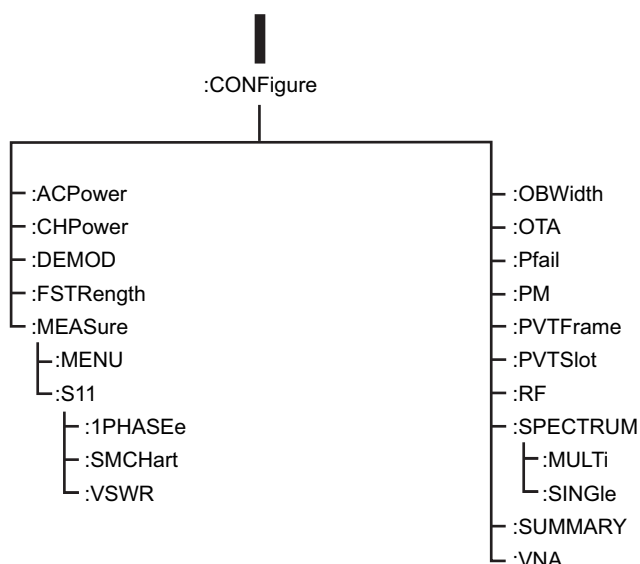


Figure 2-2. SCPI :CONFigure Subsystem

A colon (:) separates each subsystem. For example, the command :SENSe:FREQuency:START <freq> sets the start frequency. The start frequency is part of the :FREQuency subsystem which is part of the :SENSe subsystem. Stop frequency is also part of the :SENSe:FREQuency subsystem. It is specified by :SENSe:FREQuency:STOP.

Query Commands

All commands, unless specifically noted in the commands syntax descriptions, have a query form. As defined in IEEE-488.2, a query is a command with a question mark symbol appended (examples: *IDN? and :OPTions?). When a query form of a command is received, the current setting associated with the command is placed in the output buffer. Query commands always return the short form of the parameter unless otherwise specified. Boolean values are returned as 1 or 0, even when they can be set as on or off.

Data Parameters

Data parameters, referred to simply as “parameters,” are the quantitative values used as arguments for the command keywords. The parameter type associated with a particular SCPI command is determined by the type of information required to control the particular instrument function. For example, Boolean (ON | OFF) type parameters are used with commands that control switch functions.

Some command descriptions specify the type of data parameter to be used with each command. The most commonly used parameter types are numeric, extended numeric, discrete, and Boolean.

Numeric

Numeric parameters comprise integer numbers or any number in decimal or scientific notation, and may include polarity signs.

Discrete

Discrete parameters, such as INTernal and EXTernal, are used to control program settings to a predetermined finite value or condition.

Boolean

Boolean parameters represent binary conditions and may be expressed as ON, OFF or 1, 0. Boolean parameters are always returned by query commands as 1 or 0 in numeric value format.

Unit Suffixes

Unit suffixes are not required for data parameters, provided the values are scaled for the global default units. The instrument SCPI default units are: Hz (Hertz) for frequency related parameters s (seconds) for time related parameters, and m (meters) for distance related parameters.

2-4 Notational Conventions

The SCPI interface standardizes command syntax and style that simplifies the task of programming across a wide range of instrumentation. As with any programming language, the exact command keywords and command syntax must be used. Unrecognized commands or improper syntax will not function.

Table 2-3. Notational Conventions

:	A colon links command keywords together to form commands. The colon is not an actual part of the keyword, but is a signal to the SCPI interface parser. A colon must precede a root keyword immediately following a semicolon (see “Notational Examples” on page 2-8).
;	A semicolon separates commands if multiple commands are placed on a single program line.
[]	Square brackets enclose one or more optional keywords.
{ }	Braces enclose one or more keyword or command parameters that may be included zero or more times.
	A vertical bar indicates “or” and is used to separate alternative parameter options. Example: ON OFF is the same as ON or OFF.
< >	Angle brackets enclose parameter descriptions.
::=	Means “is defined as” For example: <a>::=<c> indicates that <c> can replace <a>.
<i>sp</i>	Space, referred to as <i>white space</i> , must be used to separate keywords from their associated data parameters. It must not be used between keywords or inside keywords.
xxx	Indicates a root command name
#	Indicates an integer value selection from a range of values

For further information about SCPI command syntax and style, refer to the Standard Commands for Programmable Instruments (SCPI) 1999.0 document.

2-5 Notational Examples

Table 2-4 provides examples of valid command syntax:

Table 2-4. Creating Valid Commands

Command Specification	Valid Forms
<code>[:SENSe] :FREQuency :STARt</code> <code><frequency> { Hz kHz MHz GHz }</code>	The following all produce the same result: <code>:SENSe:FREQuency:STARt 1 MHz</code> <code>:SENS:FREQ:STAR 1 MHz</code> <code>:sense:frequency:start 1000000</code> <code>:FREQ:STAR 1000 KHZ</code>
<code>:CALCulate:MARKer#:X</code> <code><value> { Hz kHz MHz GHz, m cm mm, ft }</code>	The first 2 commands set the location of marker 1. The third command sets the location of marker 2. <code>:CALC:MARK:X 1 GHz</code> <code>:CALC:MARK1:X 1 GHz</code> <code>:CALC:MARK2:X 2 GHz</code>
<code>:INITiate:CONTInuous OFF ON 0 1</code>	The following commands are identical: <code>:INITiate:CONTInuous OFF</code> <code>:init:cont 0</code>

Command statements read from left to right and from top to bottom. In the command statement above, the `:FREQuency` keyword immediately follows the `:SENSe` keyword with no separating space. A space (*sp*) is used between the command string and its argument.

Note that the first keyword in the command string does not require a leading colon; however, it is good practice to always use a leading colon for all keywords. Note also that the `:SENSe` keyword is optional. This is a SCPI convention for all voltage or signal source type instruments that allows shorter command statements to be used.

The following is an example of a multiple command statement that uses two separate commands in a single statement:

```
:FREQuency:STARt 10E6;:FREQuency:STOP 20E9
```

NoteA semicolon is used to join the commands and a leading colon used immediately after the semicolon to start the second command.

Command Terminators

The `<new line>` character (ASCII 10) in the last data byte of a command string is used as a command terminator. Use of a command terminator will reset the command path to the root of the tree.

2-6 Formatting Conventions

This manual uses the conventions listed below in describing SCPI commands. The abbreviations “Cmd” and “Param” are used to represent “Command” and “Parameter”.

Table 2-5. Formatting Conventions

<code>:COMMands:LOOK:LIKE:THIS</code>	Commands are formatted to differentiate them from their description.
<code>:COMMand:QUERies:LOOK:LIKE:THIS?</code>	The query form of the command is followed by a “?”
<code><identifier></code>	Identifiers are enclosed in “< >”. They indicate that some type of data must be provided.
<code> </code>	The “ ” indicates that a choice must be made.
<code>[optional input]</code>	Optional input is enclosed in “[]”. The “[]” are not part of the command.

Chapter 3 — All Mode Commands

The commands in this section apply to all instrument modes.

***IDN?**

Query Description: Provides information about the device. Return format:
Anritsu,<model-number>/<options>,<serial-number>,<package-version>

Example: To get device information:

*IDN?

***OPC?**

Query Description: Blocks the SCPI engine from receiving new commands until system completes all pending operations. Returns 1 for operation complete.

Example: To block until operation is complete:

*OPC?

:CONFigure:BASE:FACTorydefault

Cmd Description: This command sets system to factory default state.

Example: To set to factory default:

:CONF:BASE:FACT

:CONFigure:BASE:MASTerreset

Cmd Description: This command sets system to master reset state.

Example: To perform master reset:

:CONF:BASE:MAST

:FETCh:GPSData? {CURRENT|LAST}

Query Description: Query returns the current or the last GPS data. If no parameter is entered, it defaults to return current GPS data. Latitude and longitude units are in radians. Altitude units are in meters. Return format: No H/W Found|No Fix|Good Fix
(3D),<time-stamp>,<latitude>,<longitude>,<altitude>|Good Fix
(2D),<time-stamp>,<latitude>,<longitude>

Param Description: Optional Param: GPS data collection type, default is CURRENT.

Param Range: CURRENT|LAST

Example: Query the current GPS data:

:FETC:GPSD? CURRENT

:INSTrument:CATalog:FULL?

Query Description: Queries the available modes. 'CAA'2 for Advanced Cable Antenna Analyzer mode, 'Power Meter'6 for Power Meter, 'HI_PM'10 for High Accuracy Power Meter, 'cCAA'201 for Classic Cable Antenna Analyzer mode, 'VIP'202 for Visual Inspection Probe

Example: To query available application modes:

```
:INST:CAT:FULL?
```

:INSTrument:NSElect <application-mode-number>**:INSTrument:NSElect?**

Cmd Description: Command sets the required application mode.

Query Description: Query returns which mode is currently selected. 2 for Advanced Cable Antenna Analyzer mode, 6 for Power Meter, 10 for High Accuracy Power Meter, 201 for Classic Cable Antenna Analyzer mode, 202 for Visual Inspection Probe.

Param Type: Integer

Param Description: Application Mode

Param Range: 2|6|10|201|202

Example: To set current application mode to Hi Accuracy Power Meter:

```
:INST:NSEL 10
```

:INSTrument:SElect <"mode-identifier">**:INSTrument:SElect?**

Cmd Description: Command sets the required application mode.

Query Description: Query returns which mode is currently selected. "CAA" for Advanced Cable Antenna Analyzer mode, "cCAA" for Classic Cable Antenna Analyzer mode

Param Type: String

Param Description: Application mode identifier

Param Range: "CAA"|"cCAA"

Example: To set current application mode to Classic Cable Antenna Analyzer:

```
:INST:SEL "cCAA"
```

:PROGram:ETT:ABORTscript

Cmd Description: Command exits easy-test mode.

Example: To abort ETT mode:

```
:PROG:ETT:ABOR
```

:PROGram:ETT:LOADscript <"filename">

Cmd Description: Command begins easy-test mode with the given filename.

Param Description: Filename with .ett extension

Example: To enter ETT mode:

```
:PROG:ETT:LOAD "/internal/my-test.ett"
```

:PROGram:ETT:NEXTstep

Cmd Description: Command performs the next step while in easy-test mode.

Example: To proceed to the next step in ETT mode:

```
:PROG:ETT:NEXT
```

:PROGram:ETT:STATe?

Example: To get the ETT state:

```
:PROG:ETT:STAT?
```

:SYSTem:OPTions?

Query Description: Query returns option string delimited by "/".

Example: To query full option string:

```
:SYST:OPT?
```

:SYSTem:PRESet

Cmd Description: Presets the application to its default state. Use *OPC? to block till preset is complete.

Example: To perform preset:

```
:SYST:PRES
```

[:SENSe] :BASe:IPaddress?

Query Description: Query returns the IP address in a special format. Return value example: 2887436918 (to recover the actual IP address of 172.26.202.118, you need to convert the returned value to hex AC1ACA76 and then take the resulting 4 numbers (2 characters each, in hex) and convert them back to decimal.

Example: To query IP address:

```
:BAS:IP?
```

[:SENSe] :BASe:REMOte 0|1|2

Cmd Description: Command sets instrument into local lock-out for remote operation. Send 0 is to get out of remote mode and restore front-panel operation, 1 to set remote mode with synchronous communication, and 2 to set remote mode with asynchronous communication.

Param Type: Integer

Param Range: 1|0

Default Value: 0

Example: To set the instrument to remote synchronous mode and perform a local lock-out:

:BAS:REM 1

[:SENSe] :BASe:SCRNshot:SCHeMe INVerted|STANdard**[:SENSe] :BASe:SCRNshot:SCHeMe?**

Cmd Description: Command sets the screenshot setting scheme.

Query Description: Query returns the value that represents the current screenshot scheme.

Param Type: Character

Param Range: INVerted|STANdard

Default Value: INVerted

Example: To set the screen capture scheme to standard:

:BAS:SCRN:SCH STAN

Chapter 4 — Cable & Antenna Analyzer Mode Commands

The commands in this section apply to the Cable and Antenna Analysis mode.

:CALCulate#:DATA? FDATA|SDATA|FMEM|SMEM

Query Description: Query returns formatted data block of the specified trace.

Param Range: FDATA|SDATA|FMEM|SMEM

Example: To query trace 1 S-data:

:CALC1:DATA? SDATA

:CALCulate#:LIMit:LOWer:SEGMent:ADD <StartX><StopX><StartY><StopY>

Cmd Description: Command adds a segment to the lower limit line.

Param Type: Float

Default Units: X units: Hz for Frequency domain, m or ft for Distance domain; Y units: same as current active trace y-axis units

Example: To add a lower limit segment from 1MHz to 1GHz with value 20 dB for trace 1:

:CALC1:LIM:LOW:SEGM:ADD 1e6,1e9,20,20

:CALCulate#:LIMit:LOWer:SEGMent:DELeTe

Cmd Description: Command deletes the active lower segment.

Example: To delete the active lower segment for trace 1

:CALC1:LIM:LOW:SEGM:DEL

:CALCulate#:LIMit:LOWer:SEGMent:EDIT <StartX><StopX><StartY><StopY>

Cmd Description: Command edits the lower limit active segment. The active segment index must be updated when adding/deleting a segment.

Param Type: Float

Default Units: X units: Hz for Frequency domain, m or ft for Distance domain; Y units: same as current active trace y-axis units

Example: To edit the active segment and set it to 1MHz to 1GHz with value 20 dB for trace 1:

:CALC1:LIM:LOW:SEGM:EDIT 1e6,1e9,20,20

:CALCulate#:LIMit:LOWer:SEGMent:STARt:X <value>

:CALCulate#:LIMit:LOWer:SEGMent:STARt:X?

Cmd Description: Command sets the lower limit line X start value for current segment.

Query Description: Query returns the lower limit line X start value for active segment.

Param Type: Float

Default Units: Hz for Frequency domain, m or ft for Distance domain, s for Time domain

Example: To set limit X start value to 1 MHz for trace 1:

```
:CALC1:LIM:LOW:SEGM:STAR:X 1000000
```

:CALCulate#:LIMit:LOWer:SEGMent:STARt:Y <amplitude>

:CALCulate#:LIMit:LOWer:SEGMent:STARt:Y?

Cmd Description: Command sets the lower limit line Y start value for active segment.

Query Description: Query returns the lower limit line Y start value for active segment.

Param Type: Float

Default Units: Current active trace y-axis units

Example: To set limit start amplitude to 20 dB for trace 1:

```
:CALC1:LIM:LOW:SEGM:STAR:Y 20
```

:CALCulate#:LIMit:LOWer:SEGMent:STOP:X <value>

:CALCulate#:LIMit:LOWer:SEGMent:STOP:X?

Cmd Description: Command sets the lower limit line X stop value for active segment. Time units are available only if option 2 is installed.

Query Description: Query returns the upper limit line X stop value for active segment.

Param Type: Float

Default Units: Hz for Frequency domain, m or ft for Distance domain, s for Time domain

Example: To set limit X stop value to 1 MHz for trace 1:

```
:CALC1:LIM:LOW:SEGM:STOP:X 1000000
```

:CALCulate#:LIMit:LOWer:SEGMent:STOP:Y <amplitude>

:CALCulate#:LIMit:LOWer:SEGMent:STOP:Y?

Cmd Description: Command sets the lower limit line Y stop value for active segment.

Query Description: Query returns the lower limit line Y stop value for active segment.

Param Type: Float

Default Units: Current active trace y-axis units

Example: To set limit stop amplitude to 20 dB for trace 1:

```
:CALC1:LIM:LOW:SEGM:STOP:Y 20
```

:CALCulate#:LIMit:LOWer:SEGment:TOTal?

Query Description: Query returns the total segments in the lower limit for the given trace.

Example: To query how many lower limit segments are currently available for trace 1:

```
:CALC:LIM:LOW:SEGM:TOT?
```

:CALCulate#:LIMit:LOWer:Y <amplitude>{dB,s|ms|ns|ps}

:CALCulate#:LIMit:LOWer:Y?

Cmd Description: Command sets the lower limit line value. In group delay mode, the value is in units of time.

Query Description: Query returns the lower limit line value.

Param Range: x[,dB]

Default Units: Current active trace y-axis units

Example: To set limit amplitude to 20 dB for trace 1:

```
:CALC1:LIM:LOW:Y 20
```

:CALCulate#:LIMit:LOWer[:STATe] OFF|ON|0|1

:CALCulate#:LIMit:LOWer[:STATe]?

Cmd Description: Command sets the low limit line state for the specified trace.

Query Description: Query returns the lower limit line state for the specified trace.

Param Range: OFF|ON|0|1

Default Value: OFF|0

Example: To set limit on for trace 1:

```
:CALC1:LIM:LOW ON
```

:CALCulate#:LIMit:UPPer:SEGment:ADD

<StartX><StopX><StartY><StopY>

Cmd Description: Command adds a segment to the upper limit line.

Param Type: Float

Default Units: X units: Hz for Frequency domain, m or ft for Distance domain; Y units: same as current active trace y-axis units

Example: To add an upper limit segment from 1MHz to 1GHz with value 20 dB for trace 1:

```
:CALC1:LIM:UPP:SEGM:ADD 1e6,1e9,20,20
```

:CALCulate#:LIMit:UPPer:SEGment:DELeTe

Cmd Description: Command deletes the active upper segment.

Example: To delete the active upper segment for trace 1

```
:CALC1:LIM:UPP:SEGM:DEL
```

**:CALCulate#:LIMit:UPPer:SEGMent:EDIT
<StartX><StopX><StartY><StopY>**

Cmd Description: Command edits the upper limit active segment. The active segment index must be updated when adding/deleting a segment.

Param Type: Float

Default Units: X units: Hz for Frequency domain, m or ft for Distance domain; Y units: same as current active trace y-axis units

Example: To edit the active segment and set it to 1MHz to 1GHz with value 20 dB for trace 1:

```
:CALC1:LIM:UPP:SEGM:EDIT 1e6,1e9,20,20
```

**:CALCulate#:LIMit:UPPer:SEGMent:STARt:X <value>
:CALCulate#:LIMit:UPPer:SEGMent:STARt:X?**

Cmd Description: Command sets the upper limit line X start value for active segment. Time units are available only if option 2 is installed.

Query Description: Query returns the upper limit line X start value for active segment.

Param Type: Float

Default Units: Hz for Frequency domain, m or ft for Distance domain, s for Time domain

Example: To set limit X start value to 1 MHz for trace 1:

```
:CALC1:LIM:UPP:SEGM:STAR:X 1000000
```

**:CALCulate#:LIMit:UPPer:SEGMent:STARt:Y <amplitude>
:CALCulate#:LIMit:UPPer:SEGMent:STARt:Y?**

Cmd Description: Command sets the upper limit line Y start value for active segment.

Query Description: Query returns the upper limit line Y start value for active segment.

Param Type: Float

Default Units: Current active trace y-axis units

Example: To set limit start amplitude to 20 dB for trace 1:

```
:CALC1:LIM:UPP:SEGM:STAR:Y 20
```

:CALCulate#:LIMit:UPPer:SEGMent:STOP:X <value>

:CALCulate#:LIMit:UPPer:SEGMent:STOP:X?

Cmd Description: Command sets the upper limit line X stop value for active segment. Time units are available only if option 2 is installed.

Query Description: Query returns the upper limit line X stop value for active segment.

Param Type: Float

Default Units: Hz for Frequency domain, m or ft for Distance domain, s for Time domain

Example: To set limit X stop value to 1 MHz for trace 1:

:CALC1:LIM:UPP:SEGM:STOP:X 1000000

:CALCulate#:LIMit:UPPer:SEGMent:STOP:Y <amplitude>

:CALCulate#:LIMit:UPPer:SEGMent:STOP:Y?

Cmd Description: Command sets the upper limit line Y stop value for active segment.

Query Description: Query returns the upper limit line Y stop value for active segment.

Param Type: Float

Default Units: Current active trace y-axis units

Example: To set limit stop amplitude to 20 dB for trace 1:

:CALC1:LIM:UPP:SEGM:STOP:Y 20

:CALCulate#:LIMit:UPPer:SEGMent:TOTal?

Query Description: Query returns the total segments in the upper limit for the given trace.

Example: To query how many upper limit segments are currently available for trace 1:

:CALC:LIM:UPP:SEGM:TOT?

:CALCulate#:LIMit:UPPer:Y <amplitude>{dB,s|ms|ns|ps}

:CALCulate#:LIMit:UPPer:Y?

Cmd Description: Command sets the upper limit line value. In group delay mode, the value is in units of time.

Query Description: Query returns the upper limit line value.

Param Type: Float

Default Units: Current active trace y-axis units

Example: To set limit amplitude to 20 dB for trace 1:

:CALC1:LIM:UPP:Y 20

:CALCulate#:LIMit:UPPer[:STATe] OFF|ON|0|1

:CALCulate#:LIMit:UPPer[:STATe]?

Cmd Description: Command sets the upper limit line state for the specified trace.

Query Description: Query returns the upper limit line state for the specified trace.

Param Range: OFF|ON|0|1

Default Value: OFF|0

Example: To set limit on for trace 1:

:CALC1:LIM:UPP ON

:CALCulate:LIMit:ALARm OFF|ON|0|1

:CALCulate:LIMit:ALARm?

Cmd Description: Command sets the limit alarm state. The device beeps if a trace point crosses the limit line.

Query Description: Query returns the limit alarm state.

Param Range: OFF|ON|0|1

Default Value: OFF|0

Example: To set limit alarm on:

:CALC:LIM:ALAR ON

:CALCulate:LIMit:LOWer:SEGment:ACTive <Index>

Cmd Description: Command sets the given lower segment index active.

Param Type: int

Example: To set lower segment 1 active:

:CALC:LIM:LOW:SEGM:ACT 1

:CALCulate:LIMit:MESSage OFF|ON|0|1

:CALCulate:LIMit:MESSage?

Cmd Description: Command sets the limit message state. Displays a Pass/Fail message to indicate whether the trace crosses the limit line.

Query Description: Query returns the limit message state.

Param Range: OFF|ON|0|1

Default Value: OFF|0

Example: To set limit message on:

:CALC:LIM:MESS ON

:CALCulate:LIMit:MESSage:RESult?

Query Description: Query returns 0 if limit line says Fail. Otherwise, it returns 1 if limit line says Pass.

Example: To get limit line Pass/Fail result:

:CALC:LIM:MESS:RES?

:CALCulate:LIMit:MODE

Cmd Description: Command changes the limit line to either Line or Segmented mode.

Example: To change the limit line mode

:CALC:LIM:MOD SEGM

:CALCulate:LIMit:PRESet

Cmd Description: Command presets the limit line to its default state.

Example: To perform limit line preset:

:CALC:LIM:PRES

:CALCulate:LIMit:TYPE 0|1

:CALCulate:LIMit:TYPE?

Cmd Description: Command sets the active limit line type. 0 for upper limit and 1 for lower limit.

Query Description: Query returns the active limit line type.

Param Range: 0|1

Default Value: 0

Example: To set the active limit to lower limit:

:CALC:LIM:TYPE 1

:CALCulate:LIMit:UPPer:SEGment:ACTive <Index>

Cmd Description: Command sets the given upper segment index active.

Param Type: int

Example: To set upper segment 1 active:

:CALC:LIM:UPP:SEGM:ACT 1

:CALCulate:LIMit:VALue <amplitude>{dB}

:CALCulate:LIMit:VALue?

Cmd Description: Command sets the limit line value.

Query Description: Query returns the limit line value.

Param Type: Float

Default Units: Current active trace y-axis units

Example: To set limit line to 20 dB:

:CALC:LIM:VAL 20

:CALCulate:LIMit[:STATe] OFF|ON|0|1

:CALCulate:LIMit[:STATe]?

Cmd Description: Command sets the limit line state.

Query Description: Query returns the limit line state.

Param Range: OFF|ON|0|1

Default Value: OFF|0

Example: To set limit on:

:CALC:LIM ON

:CALCulate:MARKer#:PEAK

Cmd Description: This command sets specified marker to the trace peak value.

Example: To set marker 5 to the peak of the trace:

:CALC:MARK5:PEAK

:CALCulate:MARKer#:PEAK:BOUNded

Cmd Description: This command sets specified marker to the trace peak value between two markers. Markers 5 and 7 will search for the peak between markers 1 and 2. Markers 6 and 8 will search for the peak between markers 3 and 4.

Example: To set marker 5 to the peak between markers 1 and 2:

:CALC:MARK5:PEAK:BOUN

:CALCulate:MARKer#:TRACking
PEAK|VALLeY|BNDPeak|BNDValley|OFF
:CALCulate:MARKer#:TRACking?

Cmd Description: Command sets the tracking mode of the specified marker.

Query Description: Query returns the tracking mode of the specified marker. Query returns PEAK for peak tracking, or returns VALL for valley tracking, or returns BNDP for bounded peak tracking, or returns BNDV for bounded valley tracking.

Param Range: PEAK|VALLeY|BNDPeak|BNDValley|OFF

Default Value: OFF

Example: To set marker 2 to peak tracking:

:CALC:MARK2:TRAC PEAK

:CALCulate:MARKer#:TYPE REFERENCE|DELTA
:CALCulate:MARKer#:TYPE?

Cmd Description: Command sets the type of the specified marker.

Query Description: Query returns the type of the specified marker. Query returns REF if the specified marker is a reference marker, or returns DELT if the specified marker is a delta marker.

Param Range: REFERENCE|DELTA

Default Value: REF

Example: To set marker 2 type to delta:

:CALC:MARK2:TYPE DELT

:CALCulate:MARKer#:VALLeY

Cmd Description: This command sets specified marker to the trace valley value.

Example: To set marker 5 to the valley of the trace:

:CALC:MARK5:VALL

:CALCulate:MARKer#:VALLeY:BOUNded

Cmd Description: This command sets specified marker to the trace valley value between two markers. Markers 5 and 7 will search for the valley between markers 1 and 2. Markers 6 and 8 will search for the valley between markers 3 and 4.

Example: To set marker 5 to the valley between markers 1 and 2:

:CALC:MARK5:VALL:BOUN

:CALCulate:MARKer#:X

<value>{Hz|kHz|MHz|GHz,m|cm|mm,ft,ps|ns|us|ms|s}

:CALCulate:MARKer#:X?

Cmd Description: Command sets the position for the specified marker in the current sweep domain. For delta marker, it sets the relative position to the reference marker. Verify that the appropriate unit is used. Time units are available only if option 2 is installed.

Query Description: Query returns the marker position for the specified marker. For a delta marker, query returns the position relative to the reference marker.

Param Type: Float

Param Description: Frequency, distance, or time

Default Value: Model-dependent

Default Units: Hz

Example: To set marker 2 frequency to 1 GHz:

:CALC:MARK2:X 1 GHZ

:CALCulate:MARKer#:Y?

Query Description: Query returns the amplitude value for the specified marker.

Example: To query marker 3 amplitude value:

:CALC:MARK3:Y?

:CALCulate:MARKer#[:STATe] OFF|ON|0|1**:CALCulate:MARKer#[:STATe]?**

Cmd Description: Command sets the state of the specified marker.

Query Description: Query returns the state of the specified marker. Query returns 1 if the specified marker is on, or returns 0 if the specified marker is off.

Param Range: OFF|ON|0|1

Default Value: OFF|0

Example: To set marker 1 state ON:

:CALC:MARK1 ON

:CALCulate:MARKer:ACTive?

Query Description: Query returns active marker number. Returns NONE if no markers are active.

Example: To get active marker

:CALC:MARK:ACT?

:CALCulate:MARKer:DISPlay MKRTable|MKRonly|AOFF

:CALCulate:MARKer:DISPlay?

Cmd Description: Command sets the marker display mode.

Query Description: Query returns the marker display mode. Query returns MKRT if the markers and marker table are displayed, returns MKR if only the markers are displayed, or returns AOFF if markers and marker table are not displayed.

Param Range: MKRTable|MKRonly|AOFF

Default Value: MKRTable

Example: To set marker display to marker only:

:CALC:MARK:DISP MKR

:CALCulate:MARKer:PRESet

Cmd Description: Presets markers.

Example: To preset markers:

:CALC:MARK:PRES

:CALCulate:MATH:FUNCTION NONE|SUBTraction|ADDition|AVERage

:CALCulate:MATH:FUNCTION?

Cmd Description: Command sets the trace math function. NONE - no trace math applied; SUBT - memory trace subtracted from active trace; ADD - memory trace added to active trace; AVER - active and memory traces added and divided by 2.

Query Description: Query returns the trace math function.

Param Range: NONE|SUBTraction|ADDition|AVERage

Default Value: NONE

Example: To set trace math function to addition:

:CALC:MATH:FUNC ADD

:CALCulate:MATH:MEMorize

Cmd Description: Copies active trace into the memory buffer.

Example: To copy active trace into the memory buffer:

:CALC:MATH:MEM

:CALCulate:TRANSform:CLAVerage?

Query Description: Query returns cable loss average when the measurement mode is 'Cable Loss'.

Default Units: dB

Example: To query cable loss average:

:CALC:TRAN:CLAV?

**:CALCulate:TRANSform:DISTance:CABLoss <cable-loss>{
dB/m,dB/ft}**

:CALCulate:TRANSform:DISTance:CABLoss?

Cmd Description: Command sets the cable loss. Verify that the appropriate unit is used.

Query Description: Query returns the cable loss based on the current units.

Param Type: Float

Default Value: 0

Default Units: dB/m

Example: To set cable loss to 2:

:CALC:TRAN:DIST:CABL 2

:CALCulate:TRANSform:DISTance:DMAX?

Query Description: Query returns max distance in the current units.

Example: To query max distance:

:CALC:TRAN:DIST:DMAX?

**:CALCulate:TRANSform:DISTance:PVELocity
<propagation-velocity>**

:CALCulate:TRANSform:DISTance:PVELocity?

Cmd Description: Command sets the propagation velocity.

Query Description: Query returns the propagation velocity.

Param Type: Float

Default Value: 1

Example: To set propagation velocity to 0.05:

:CALC:TRAN:DIST:PVEL 0.05

:CALCulate:TRANSform:DISTance:RESolution?

Query Description: Query returns distance resolution in the current units for reflection.

Example: To query distance resolution:

:CALC:TRAN:DIST:RESolution?

:CALCulate:TRANSform:DISTance:STARt <distance>{ m|cm|mm,ft}
:CALCulate:TRANSform:DISTance:STARt?

Cmd Description: Command sets the start distance. Verify the appropriate unit is used.

Query Description: Query returns the start distance in the current units.

Param Type: Float

Param Description: Distance

Default Units: m(meter)

Example: To set start distance to 1 m:

:CALC:TRAN:DIST:STAR 1

:CALCulate:TRANSform:DISTance:STOP <distance>{ m|cm|mm,ft}
:CALCulate:TRANSform:DISTance:STOP?

Cmd Description: Command sets the stop distance. Verify the appropriate unit is used.

Query Description: Query returns the stop distance in the current units.

Param Type: Float

Param Description: Distance

Default Units: m(meter)

Example: To set stop distance to 2 m:

:CALC:TRAN:DIST:STOP 2 M

:CALCulate:TRANSform:DISTance:UNIT METers|FEET
:CALCulate:TRANSform:DISTance:UNIT?

Cmd Description: Command sets the distance unit.

Query Description: Query returns the current distance units.

Param Range: METers|FEET

Default Value: METers

Example: To set distance unit to feet:

:CALC:TRAN:DIST:UNIT FEET

:CALCulate:TRANSform:DISTance:WINDow
RECTangular | MSLobe | NSLobe | LSLobe
:CALCulate:TRANSform:DISTance:WINDow?

Cmd Description: Command sets the window type for time domain.

Query Description: Query returns the window type for time domain.

Param Type: Character

Param Range: RECTangular | MSLobe | NSLobe | LSLobe

Default Value: NSLobe

Example: To set time domain window type to rectangular:

:CALC:TRAN:DIST:WIND RECT

:CAPTure:STARt

Cmd Description: Command starts the data capture for continuous sweeps. It does not work in single sweep and hold mode.

Example: To start data capture

:CAPT:STAR

:CAPTure:STOP

Cmd Description: Command stops the data capture for continuous sweeps. It does not work in single sweep and hold mode.

Example: To stop capture:

:CAPT:STOP

:CAPTure:TRACe#:DATA? FDATA | SDATA | FMEM | SMEM

Cmd Description: Command is used to return the data block of the specified trace at the end of a sweep, while the instrument is in continuous sweep mode. This command must follow the :CAPTure:STARt command. A :CAPTure:STOP command is used to end the capture session. If the system is in single sweep and hold mode, then this command is ignored.

Example: To trigger a sweep:

:CAPT:TRAC1:DATA? FDATA

:CONFigure:MEASure:DISPlay SINGLE|DUAL

:CONFigure:MEASure:DISPlay?

Cmd Description: Command sets the display format to single or dual traces. Only supported in Advanced Cable & Antenna Analyzer mode.

Query Description: Query returns the display format. Only supported in Advanced Cable & Antenna Analyzer mode.

Param Range: SINGLE|DUAL

Default Value: SINGLE

Example: To set display format to dual traces:

```
:CONF:MEAS:DISP DUAL
```

:CONFigure:MEASure:MODE

RLFreq|RLDTf|SWRFreq|SWRDtf|CLFreq|TRES|SMITH|PHASE

:CONFigure:MEASure:MODE?

Cmd Description: Command sets the active trace measurement mode.

Query Description: Query returns the current measurement mode.

Param Description: RLFreq: Return Loss (Frequency domain sweep); RLDTf: Return Loss(DTF); SWRFreq: SWR(Frequency domain sweep); SWRDtf: SWR(DTF); CLFreq: Cable Loss - One Port (Frequency domain sweep); TRES: Transmission - External Sensor; SMITH: Smith Chart; PHASE: One-Port Phase.

Example: To set measurement mode to Cable Loss:

```
:CONF:MEAS:MOD CLFR
```

:DISPlay:UPDate ON|OFF|1|0

:DISPlay:UPDate?

Cmd Description: Command sets the display update on/off.

Query Description: Query returns display update status. Query returns 1 if display update is on. Otherwise, it returns 0.

Param Range: ON|OFF|1|0

Default Value: OFF|0

Example: To turn display update off:

```
:DISP:UPD OFF
```

:DISPlay:WINDow:TRACe:STATe TRACe|MEMory|BOTH
:DISPlay:WINDow:TRACe:STATe?

Cmd Description: Command sets the trace display type for the active trace. TRAC - only active trace is shown; MEM - only memory trace is shown; BOTH - both active and memory traces are shown. Use :CALCulate:MATH:MEMorize before changing the display type.

Query Description: Query returns the trace display type.

Param Range: TRACe|MEMory|BOTH

Default Value: TRACe

Example: To show both active and memory trace on the display:

:DISP:WIND:TRAC:STAT BOTH

:DISPlay:WINDow:TRACe:Y[:SCALE]:AUToscale

Cmd Description: This command sets the top and bottom values such that the trace occupies a significant portion of the grid.

Example: To perform auto-scale:

:DISP:WIND:TRAC:Y:AUT

:DISPlay:WINDow:TRACe:Y[:SCALE]:BOTTom <amplitude>{dB}
:DISPlay:WINDow:TRACe:Y[:SCALE]:BOTTom?

Cmd Description: Command sets the display bottom value.

Query Description: Query returns the display bottom value.

Param Type: Float

Default Value: Return Loss = 60dB; DTF Return Loss = 60dB; Cable Loss = 30dB; VSWR = 1; DTF VSWR = 1; Transmission = -90dB; Phase=-225

Default Units: dB

Example: To set trace bottom value in Return Loss to 40 dB:

:DISP:WIND:TRAC:Y:BOTT 40

:DISPlay:WINDow:TRACe:Y[:SCALE]:FULLscale

Cmd Description: Sets the top and bottom to their maximum values.

Example: To perform full-scale:

:DISP:WIND:TRAC:Y:FULL

:DISPlay:WINDow:TRACe:Y[:SCALe]:TOP <amplitude>{dB}
:DISPlay:WINDow:TRACe:Y[:SCALe]:TOP?

Cmd Description: Command sets the display top value.

Query Description: Query returns the display top value.

Param Type: Float

Default Value: Return Loss = 0dB; DTF Return Loss = 0dB; Cable Loss = 0dB; VSWR = 3; DTF VSWR = 3; Transmission = 10dB; Phase = 225

Default Units: dB

Example: To set trace top value in Return Loss to 10 dB:

:DISP:WIND:TRAC:Y:TOP 10

:FORMat[:READings][:DATA] ASCii|INTEger,32|REAL,32
:FORMat[:READings][:DATA]?

Cmd Description: Command sets the data format type. ASCii and REAL returns data in current unit. INTEger returns data with mdBm unit. This command applies to :TRACe[:DATA]? and :CALCulate:DATA?.

Query Description: Query returns the data format type as ASC or INT, 32 or REAL, 32.

Param Range: ASCii|INTEger,32|REAL,32

Example: To set data format type to integer:

:FORM INT,32

:FORMat[:READings][:DATA]:MULTiplier 1|1M
:FORMat[:READings][:DATA]:MULTiplier?

Cmd Description: Command sets the data format multiplier factor.

Query Description: Query returns the data format multiplier factor. If '1M' is passed as argument, then query results for :TRACe[:DATA]? and :CALCulate:DATA? will specify readings multiplied by 10⁶. If '1' is passed, then query results for :TRACe[:DATA]? and :CALCulate:DATA? will specify read.

Param Range: 1|1M

Default Value: 1M

Example: To set data format multiplier to 1:

:FORM:MULT 1

:INITiate#:DATA? FDATA|SDATA|FMEM|SMEM

Cmd Description: Command triggers a sweep and returns the data block of the specified trace at the end of sweep. If the system is in continuous mode, then this command is ignored.

Example: To trigger a sweep and wait for Formatted data to be returned:

:INIT1:DATA? FDATA

:INITiate:CONTinuous OFF|ON|0|1

:INITiate:CONTinuous?

Cmd Description: Command sets the sweep type between single sweep and continuous sweep.

Query Description: Query returns the sweep type, single or continuous. Query returns 1 for continuous sweep, and returns 0 for single sweep.

Param Range: OFF|ON|0|1

Default Value: ON|1

Example: To set the sweep type to single sweep:

:INIT:CONT OFF

:INITiate:HOLD OFF|ON|0|1

:INITiate:HOLD?

Cmd Description: Command sets the sweep state between run and hold. This command is ignored in external trigger mode.

Query Description: Query returns the sweep state, run or hold. Query returns 1 for hold mode, and returns 0 for run mode.

Param Range: OFF|ON|0|1

Default Value: OFF|0

Example: To set the sweep state to hold:

:INIT:HOLD ON

:INITiate[:IMMediate]

Cmd Description: Command triggers a single sweep if in single sweep mode. If the system is in continuous mode, then this command is ignored.

Example: To trigger a sweep:

:INIT

:STATus:OPERation?

Query Description: Queries sweep status. This is used in combination with :INITiate[:IMMediate]. Sweep complete bit is the bit 8. Query returns 256 if sweep has completed. Otherwise, it returns 0.

Example: To query sweep status:

:STAT:OPER?

:TRACe[:DATA]? {1|2}

Query Description: Query returns raw trace data in real and imaginary format.

Param Type: Integer

Param Description: Optional Param: Trace number, default is 1.

Param Range: 1|2

Example: To get trace 1 data:

:TRAC?

[:SENSe]:APPLication:TST:RESult?

Query Description: Query returns self-test details. The self-test command (See [:SENSe]:APPLication:TST? [NORMal]) must be sent before getting the self-test details.

Example: To retrieve application self-test result:

:APPL:TST:RES?

[:SENSe]:APPLication:TST? {NORMal}

Query Description: Query returns 0 if self-test succeeds with no error. Otherwise, it returns 255.

Param Description: Optional Param: Self-test type, default is NORMal.

Param Range: NORMal

Example: To run an application self-test:

:APPL:TST?

[:SENSe]:CALibration:STATe?

Query Description: Query returns the state of calibrated measurements depending on the calibration collection type. 0 means no calibration was done. For calibration types RFP1 & RRP1, return value is 1. For calibration types TRFP & TRES, return value is 4. For calibration types 2PFP & 2PES, return value is 5.

Example: To query state after calibration is completed:

:CAL:STAT?

[:SENSe]:CORRection:COLLect:ABORt

Cmd Description: Aborts all steps of the RF calibration.

Example: To abort RF calibration:

:CORR:COLL:ABOR

[:SENSe] :CORRection:COLLect:INFO:STATus?

Query Description: Query returns 0 if calibration is not in progress, 1 if calibration is in progress, 2 if calibration was aborted, 3 if cal-coefficients are being calculated, or 4 if calibration has completed.

Example: To query calibration progress status:

:CORR:COLL:INFO:STAT?

[:SENSe] :CORRection:COLLect:INITialize

Cmd Description: Command to initialize an RF calibration with the current calibration type.

Example: To initialize RF calibration:

:CORR:COLL:INIT

[:SENSe] :CORRection:COLLect:LOAD

Cmd Description: Perform 'Load' step of calibration.

Example: To measure load:

:CORR:COLL:LOAD

[:SENSe] :CORRection:COLLect:OPEN

Cmd Description: Perform 'Open' step of calibration.

Example: To measure open:

:CORR:COLL:OPEN

[:SENSe] :CORRection:COLLect:SAVe

Cmd Description: Complete an RF Cal and perform coefficient calculation.

Example: To complete a RF calibration and perform coefficient calculation:

:CORR:COLL:SAV

[:SENSe] :CORRection:COLLect:SHORt

Cmd Description: Perform 'Short' step of calibration.

Example: To measure short:

:CORR:COLL:SHOR

[:SENSe] :CORRection:COLLect:STATus?

{INITialize | OPEN | SHORT | LOAD | SAVE | ALL | THRU | ZERO}

Query Description: Query with step parameter returns 1 if a specified cal step is completed. Otherwise, it returns 0. Query without any parameter returns 0 if no cal started, 1 if cal already started, 2 if cal has been aborted, 3 if cal is calculating, or 4 if cal has been completed.

Param Range: INITialize | OPEN | SHORT | LOAD | SAVE | ALL | THRU | ZERO

Example: To query calibration step 'Open' status:

:CORR:COLL:STAT? OPEN

[:SENSe] :CORRection:COLLect:THRU

Cmd Description: Perform 'Thru' step of calibration.

Example: To measure Thru:

:CORR:COLL:THRU

[:SENSe] :CORRection:COLLect:TYPE RFP1 | TRES

[:SENSe] :CORRection:COLLect:TYPE?

Cmd Description: Command sets calibration collection type.

Query Description: Query returns calibration collection type.

Param Range: RFP1 | TRES

Default Value: RFP1

Example: To set collection type to RFP1:

:CORR:COLL:TYPE RFP1

[:SENSe] :CORRection:COLLect:ZERO

Cmd Description: Perform 'Zero Sensor' step of calibration.

Example: To zero sensor:

:CORR:COLL:ZERO

[:SENSe] :CORRection:TYPE STANdard | FLEX

[:SENSe] :CORRection:TYPE?

Cmd Description: Command sets correction type.

Query Description: Query returns correction type. Query returns STAN | FLEX.

Param Range: STANdard | FLEX

Default Value: STANdard

Example: To set correction type to flex:

:CORR:TYPE FLEX

[:SENSe] :CORRection[:STATe] ON|OFF|1|0

[:SENSe] :CORRection[:STATe] ?

Cmd Description: Command sets correction status.

Query Description: Query returns correction status. Query returns 1 if RF cal is on. Otherwise, it returns 0.

Param Range: ON|OFF|1|0

Default Value: OFF|0

Example: To turn calibration correction on:

:CORR:STAT ON

[:SENSe] :FREQuency:CABLe < cable-list-index >

[:SENSe] :FREQuency:CABLe ?

Cmd Description: Command selects a cable with the index number from the sorted cable list.

Query Description: Query returns the index number of the selected cable based on the sorted cable list.

Param Type: Integer

Param Description: Index number from cable list

Default Value: 0

Example: To set the cable index to 10:

:FREQ:CABL 10

[:SENSe] :FREQuency:CABLe:CATalog:FULL ?

Query Description: Query returns the entire cable list that is stored in memory.

Example: To get the cable list:

:FREQ:CABL:CAT:FULL?

[:SENSe] :FREQuency:CABLe:NAME <"cable-name">

[:SENSe] :FREQuency:CABLe:NAME ?

Cmd Description: Command sets the cable name.

Query Description: Query returns the cable name.

Param Type: String

Param Description: Cable name

Default Value: NONE

Example: To set the cable to "AVA5-50 7/8":

:FREQ:CABL:NAM "AVA5-50 7/8"

[:SENSe] :FREQuency:STARt <frequency>{ Hz | kHz | MHz | GHz }
[:SENSe] :FREQuency:STARt?

Cmd Description: Command sets the start frequency.

Query Description: Query returns the start frequency in Hertz.

Param Type: Float

Param Description: Start frequency

Default Value: Model-dependent

Default Units: Hz

Example: To set the start frequency to 2 GHz:

:FREQ:STAR 2 GHz

[:SENSe] :FREQuency:STOP <frequency>{ Hz | kHz | MHz | GHz }
[:SENSe] :FREQuency:STOP?

Cmd Description: Command sets the stop frequency.

Query Description: Query returns the stop frequency in Hertz.

Param Type: Float

Param Description: Stop frequency

Default Value: Model-dependent

Default Units: Hz

Example: To set the stop frequency to 2 GHz:

:FREQ:STOP 2 GHz

[:SENSe] :SWEep:RESolution 130 | 259 | 517 | 1033 | 2065
[:SENSe] :SWEep:RESolution?

Cmd Description: Command sets the data points.

Query Description: Query returns the data points as a number.

Param Range: 130 | 259 | 517 | 1033 | 2065

Default Value: 259

Example: To set data points to 1033:

:SWE:RES 1033

[:SENSe] :SWEep:RFIMmunity HIGH|LOW|1|0

[:SENSe] :SWEep:RFIMmunity?

Cmd Description: Command sets the RF immunity state.

Query Description: Query returns the RF immunity state, 1 for low immunity and 0 for high immunity.

Param Range: HIGH|LOW|1|0

Default Value: LOW|1

Example: To set RF immunity to low:

:SWE:RFIM LOW

[:SENSe] :SWEep:TYPE CONTInuous|SINGLe|EXTErnal

Cmd Description: Command sets the sweep trigger type to be Internal (Continuous or Single) or External.

Param Range: CONTInuous|SINGLe|EXTErnal

Default Value: CONTInuous

Example: To set the sweep trigger type to internal single sweep:

:SWE:TYPE SING

[:SENSe] :TRACe:SElect 1|2

[:SENSe] :TRACe:SElect?

Cmd Description: Command sets the active trace.

Query Description: Query returns the active trace.

Param Range: 1|2

Default Value: 1

Example: To set active trace to trace 2:

:TRAC:SEL 2

Appendix A — List of Commands by Mode

Chapter 1—General Information

Chapter 2—Programming with SCPI

Chapter 3—All Mode Commands

*IDN?	3-1
*OPC?	3-1
:CONFigure:BASE:FACTorydefault	3-1
:CONFigure:BASE:MASTerreset	3-1
:FETCh:GPSData? {CURRent LAST}	3-1
:INSTrument:CATalog:FULL?	3-2
:INSTrument:NSElect <application-mode-number>	3-2
:INSTrument:NSElect?	3-2
:INSTrument:SElect <"mode-identifier">	3-2
:INSTrument:SElect?	3-2
:PROGram:ETT:ABORTscript	3-2
:PROGram:ETT:LOADscript <"filename">	3-3
:PROGram:ETT:NEXTstep	3-3
:PROGram:ETT:STATe?	3-3
:SYSTem:OPTions?	3-3
:SYSTem:PRESet	3-3
[[:SENSE]:]BASE:IPaddress?	3-3
[[:SENSE]:]BASE:REMOte 0 1 2	3-4
[[:SENSE]:]BASE:SCRNshot:SCHeMe INVerted STANdard	3-4
[[:SENSE]:]BASE:SCRNshot:SCHeMe?	3-4

Chapter 4—Cable & Antenna Analyzer Mode Commands

:CALCulate#:DATA? FDATa SDATa FMEM SMEM	4-1
:CALCulate#:LIMit:LOWer:SEGMENT:ADD <StartX><StopX><StartY><StopY>	4-1
:CALCulate#:LIMit:LOWer:SEGMENT:DELeTe	4-1
:CALCulate#:LIMit:LOWer:SEGMENT:EDIT <StartX><StopX><StartY><StopY>	4-1
:CALCulate#:LIMit:LOWer:SEGMENT:STARt:X <value>	4-2
:CALCulate#:LIMit:LOWer:SEGMENT:STARt:X?	4-2
:CALCulate#:LIMit:LOWer:SEGMENT:STARt:Y <amplitude>	4-2
:CALCulate#:LIMit:LOWer:SEGMENT:STARt:Y?	4-2
:CALCulate#:LIMit:LOWer:SEGMENT:STOP:X <value>	4-2
:CALCulate#:LIMit:LOWer:SEGMENT:STOP:X?	4-2
:CALCulate#:LIMit:LOWer:SEGMENT:STOP:Y <amplitude>	4-2
:CALCulate#:LIMit:LOWer:SEGMENT:STOP:Y?	4-2
:CALCulate#:LIMit:LOWer:SEGMENT:TOTAl?	4-3
:CALCulate#:LIMit:LOWer:Y <amplitude> {dB,s ms ns ps}	4-3
:CALCulate#:LIMit:LOWer:Y?	4-3
:CALCulate#:LIMit:LOWer[:STATe] OFF ON 0 1	4-3

:CALCulate#:LIMit:LOWer[:STATe]?	4-3
:CALCulate#:LIMit:UPPer:SEGment:ADD <StartX><StopX><StartY><StopY>	4-3
:CALCulate#:LIMit:UPPer:SEGment:DELeTe	4-3
:CALCulate#:LIMit:UPPer:SEGment:EDIT <StartX><StopX><StartY><StopY>	4-4
:CALCulate#:LIMit:UPPer:SEGment:START:X <value>	4-4
:CALCulate#:LIMit:UPPer:SEGment:START:X?	4-4
:CALCulate#:LIMit:UPPer:SEGment:START:Y <amplitude>	4-4
:CALCulate#:LIMit:UPPer:SEGment:START:Y?	4-4
:CALCulate#:LIMit:UPPer:SEGment:STOP:X <value>	4-5
:CALCulate#:LIMit:UPPer:SEGment:STOP:X?	4-5
:CALCulate#:LIMit:UPPer:SEGment:STOP:Y <amplitude>	4-5
:CALCulate#:LIMit:UPPer:SEGment:STOP:Y?	4-5
:CALCulate#:LIMit:UPPer:SEGment:TOTal?	4-5
:CALCulate#:LIMit:UPPer:Y <amplitude> {dB,s ms ns ps}	4-5
:CALCulate#:LIMit:UPPer:Y?	4-5
:CALCulate#:LIMit:UPPer[:STATe] OFF ON 0 1	4-6
:CALCulate#:LIMit:UPPer[:STATe]?	4-6
:CALCulate:LIMit:ALARm OFF ON 0 1	4-6
:CALCulate:LIMit:ALARm?	4-6
:CALCulate:LIMit:LOWer:SEGment:ACTive <Index>	4-6
:CALCulate:LIMit:MESSAge OFF ON 0 1	4-6
:CALCulate:LIMit:MESSAge?	4-6
:CALCulate:LIMit:MESSAge:RESult?	4-7
:CALCulate:LIMit:MODE	4-7
:CALCulate:LIMit:PRESet	4-7
:CALCulate:LIMit:TYPE 0 1	4-7
:CALCulate:LIMit:TYPE?	4-7
:CALCulate:LIMit:UPPer:SEGment:ACTive <Index>	4-7
:CALCulate:LIMit:VALue <amplitude> {dB}	4-8
:CALCulate:LIMit:VALue?	4-8
:CALCulate:LIMit[:STATe] OFF ON 0 1	4-8
:CALCulate:LIMit[:STATe]?	4-8
:CALCulate:MARKer#:PEAK	4-8
:CALCulate:MARKer#:PEAK:BOUNded	4-8
:CALCulate:MARKer#:TRACking PEAK VALley BNDPeak BNDValley OFF	4-9
:CALCulate:MARKer#:TRACking?	4-9
:CALCulate:MARKer#:TYPE REFeRence DELTA	4-9
:CALCulate:MARKer#:TYPE?	4-9
:CALCulate:MARKer#:VALley	4-9
:CALCulate:MARKer#:VALley:BOUNded	4-9
:CALCulate:MARKer#:X <value> {Hz kHz MHz GHz,m cm mm,ft,ps ns us ms s}	4-10
:CALCulate:MARKer#:X?	4-10
:CALCulate:MARKer#:Y?	4-10
:CALCulate:MARKer#[:STATe] OFF ON 0 1	4-10
:CALCulate:MARKer#[:STATe]?	4-10
:CALCulate:MARKer:ACTive?	4-10
:CALCulate:MARKer:DISPlay MKRTable MKRonly AOFF	4-11
:CALCulate:MARKer:DISPlay?	4-11
:CALCulate:MARKer:PRESet	4-11
:CALCulate:MATH:FUNCTION NONE SUBTraction ADDition AVERage	4-11

:CALCulate:MATH:FUNCTION?	4-11
:CALCulate:MATH:MEMorize	4-11
:CALCulate:TRANSform:CLAVerage?	4-11
:CALCulate:TRANSform:DISTance:CABLoss <cable-loss> { dB/m,dB/ft}	4-12
:CALCulate:TRANSform:DISTance:CABLoss?	4-12
:CALCulate:TRANSform:DISTance:DMAX?	4-12
:CALCulate:TRANSform:DISTance:PVELocity <propagation-velocity>	4-12
:CALCulate:TRANSform:DISTance:PVELocity?	4-12
:CALCulate:TRANSform:DISTance:RESolution?	4-12
:CALCulate:TRANSform:DISTance:START <distance> { m cm mm,ft}	4-13
:CALCulate:TRANSform:DISTance:START?	4-13
:CALCulate:TRANSform:DISTance:STOP <distance> { m cm mm,ft}	4-13
:CALCulate:TRANSform:DISTance:STOP?	4-13
:CALCulate:TRANSform:DISTance:UNIT METers FEET	4-13
:CALCulate:TRANSform:DISTance:UNIT?	4-13
:CALCulate:TRANSform:DISTance:WINDow RECTangular MSLobe NSLobe LSLobe	4-14
:CALCulate:TRANSform:DISTance:WINDow?	4-14
:CAPTure:START	4-14
:CAPTure:STOP	4-14
:CAPTure:TRACe#:DATA? FDATA SDATA FMEM SMEM	4-14
:CONFigure:MEASure:DISPlay SINGLE DUAL	4-15
:CONFigure:MEASure:DISPlay?	4-15
:CONFigure:MEASure:MODE	
RLFreq RLDTf SWRFreq SWRDtf CLFreq TRES SMITH PHASe	4-15
:CONFigure:MEASure:MODE?	4-15
:DISPlay:UPDate ON OFF 1 0	4-15
:DISPlay:UPDate?	4-15
:DISPlay:WINDow:TRACe:STATe TRACe MEMory BOTH	4-16
:DISPlay:WINDow:TRACe:STATe?	4-16
:DISPlay:WINDow:TRACe:Y[:SCALE]:AUToscale	4-16
:DISPlay:WINDow:TRACe:Y[:SCALE]:BOTTom <amplitude> {dB}	4-16
:DISPlay:WINDow:TRACe:Y[:SCALE]:BOTTom?	4-16
:DISPlay:WINDow:TRACe:Y[:SCALE]:FULLscale	4-16
:DISPlay:WINDow:TRACe:Y[:SCALE]:TOP <amplitude> {dB}	4-17
:DISPlay:WINDow:TRACe:Y[:SCALE]:TOP?	4-17
:FORMat[:READings][:DATA] ASCii INTEger,32 REAL,32	4-17
:FORMat[:READings][:DATA]?	4-17
:FORMat[:READings][:DATA]:MULTiplier 1 1M	4-17
:FORMat[:READings][:DATA]:MULTiplier?	4-17
:INITiate#:DATA? FDATA SDATA FMEM SMEM	4-17
:INITiate:CONTInuous OFF ON 0 1	4-18
:INITiate:CONTInuous?	4-18
:INITiate:HOLD OFF ON 0 1	4-18
:INITiate:HOLD?	4-18
:INITiate[:IMMediate]	4-18
:STATus:OPERation?	4-18
:TRACe[:DATA]? {1 2}	4-19
[[:SENSe]:APPLication:TST:RESult?	4-19
[[:SENSe]:APPLication:TST? {NORMal}	4-19

[SENSe]:CALibration:STATe?	4-19
[SENSe]:CORRection:COLlect:ABORt	4-19
[SENSe]:CORRection:COLlect:INFO:STATus?	4-20
[SENSe]:CORRection:COLlect:INITialize	4-20
[SENSe]:CORRection:COLlect:LOAD	4-20
[SENSe]:CORRection:COLlect:OPEN	4-20
[SENSe]:CORRection:COLlect:SAVe	4-20
[SENSe]:CORRection:COLlect:SHORt	4-20
[SENSe]:CORRection:COLlect:STATus?	
{INITialize OPEN SHORt LOAD SAVe ALL THRU ZERO}	4-21
[SENSe]:CORRection:COLlect:THRU	4-21
[SENSe]:CORRection:COLlect:TYPE RFP1 TRES	4-21
[SENSe]:CORRection:COLlect:TYPE?	4-21
[SENSe]:CORRection:COLlect:ZERO	4-21
[SENSe]:CORRection:TYPE STANDARD FLEX	4-21
[SENSe]:CORRection:TYPE?	4-21
[SENSe]:CORRection[:STATe] ON OFF 1 0	4-22
[SENSe]:CORRection[:STATe]?	4-22
[SENSe]:FREQuency:CABLe <cable-list-index>	4-22
[SENSe]:FREQuency:CABLe?	4-22
[SENSe]:FREQuency:CABLe:CATalog:FULL?	4-22
[SENSe]:FREQuency:CABLe:NAME <"cable-name">	4-22
[SENSe]:FREQuency:CABLe:NAME?	4-22
[SENSe]:FREQuency:STARt <frequency> { Hz kHz MHz GHz}	4-23
[SENSe]:FREQuency:STARt?	4-23
[SENSe]:FREQuency:STOP <frequency> { Hz kHz MHz GHz}	4-23
[SENSe]:FREQuency:STOP?	4-23
[SENSe]:SWEep:RESolution 130 259 517 1033 2065	4-23
[SENSe]:SWEep:RESolution?	4-23
[SENSe]:SWEep:RFIMmunity HIGH LOW 1 0	4-24
[SENSe]:SWEep:RFIMmunity?	4-24
[SENSe]:SWEep:TYPE CONTinuous SINGle EXTeRnal	4-24
[SENSe]:TRACe:SELeCt 1 2	4-24
[SENSe]:TRACe:SELeCt?	4-24

Appendix B — List of Commands, Alphabetical

All SCPI Commands in Alphabetic List

:CALCulate:LIMit:ALARm OFF ON 0 1	4-6
:CALCulate:LIMit:ALARm?	4-6
:CALCulate:LIMit:LOWer:SEGment:ACTive <Index>	4-6
:CALCulate:LIMit:MESSAge OFF ON 0 1	4-6
:CALCulate:LIMit:MESSAge:RESult?	4-7
:CALCulate:LIMit:MESSAge?	4-6
:CALCulate:LIMit:MODE	4-7
:CALCulate:LIMit:PRESet	4-7
:CALCulate:LIMit:TYPE 0 1	4-7
:CALCulate:LIMit:TYPE?	4-7
:CALCulate:LIMit:UPPer:SEGment:ACTive <Index>	4-7
:CALCulate:LIMit:VALue <amplitude> {dB}	4-8
:CALCulate:LIMit:VALue?	4-8
:CALCulate:LIMit[:STATe] OFF ON 0 1	4-8
:CALCulate:LIMit[:STATe]?	4-8
:CALCulate:MARKer:ACTive?	4-10
:CALCulate:MARKer:DISPlay MKRTable MKRonly AOFF	4-11
:CALCulate:MARKer:DISPlay?	4-11
:CALCulate:MARKer:PRESet	4-11
:CALCulate:MARKer#:PEAK	4-8
:CALCulate:MARKer#:PEAK:BOUNded	4-8
:CALCulate:MARKer#:TRACking PEAK VALley BNDPeak BNDValley OFF	4-9
:CALCulate:MARKer#:TRACking?	4-9
:CALCulate:MARKer#:TYPE REFerence DELTA	4-9
:CALCulate:MARKer#:TYPE?	4-9
:CALCulate:MARKer#:VALley	4-9
:CALCulate:MARKer#:VALley:BOUNded	4-9
:CALCulate:MARKer#:X <value> {Hz kHz MHz GHz,m cm mm,ft,ps ns us ms s}	4-10
:CALCulate:MARKer#:X?	4-10
:CALCulate:MARKer#:Y?	4-10
:CALCulate:MARKer#[:STATe] OFF ON 0 1	4-10
:CALCulate:MARKer#[:STATe]?	4-10
:CALCulate:MATH:FUNCTion NONE SUBTraction ADDition AVERage	4-11
:CALCulate:MATH:FUNCTion?	4-11
:CALCulate:MATH:MEMorize	4-11
:CALCulate:TRANSform:CLAVerage?	4-11
:CALCulate:TRANSform:DISTance:CABLoss <cable-loss> { dB/m,dB/ft}	4-12
:CALCulate:TRANSform:DISTance:CABLoss?	4-12
:CALCulate:TRANSform:DISTance:DMAX?	4-12
:CALCulate:TRANSform:DISTance:PVELocity <propagation-velocity>	4-12
:CALCulate:TRANSform:DISTance:PVELocity?	4-12
:CALCulate:TRANSform:DISTance:RESolution?	4-12
:CALCulate:TRANSform:DISTance:STARt <distance> { m cm mm,ft}	4-13

:CALCulate:TRANSform:DISTance:START?	4-13
:CALCulate:TRANSform:DISTance:STOP <distance>{ m cm mm,ft}	4-13
:CALCulate:TRANSform:DISTance:STOP?	4-13
:CALCulate:TRANSform:DISTance:UNIT METers FEET	4-13
:CALCulate:TRANSform:DISTance:UNIT?	4-13
:CALCulate:TRANSform:DISTance:WINDow RECTangular MSLobe NSLobe LSLobe	4-14
:CALCulate:TRANSform:DISTance:WINDow?	4-14
:CALCulate#:DATA? FDATA SDATA FMEM SMEM	4-1
:CALCulate#:LIMit:LOWER:SEGMENT:ADD <StartX><StopX><StartY><StopY>	4-1
:CALCulate#:LIMit:LOWER:SEGMENT:DELeTe	4-1
:CALCulate#:LIMit:LOWER:SEGMENT:EDIT <StartX><StopX><StartY><StopY>	4-1
:CALCulate#:LIMit:LOWER:SEGMENT:START:X <value>	4-2
:CALCulate#:LIMit:LOWER:SEGMENT:START:X?	4-2
:CALCulate#:LIMit:LOWER:SEGMENT:START:Y <amplitude>	4-2
:CALCulate#:LIMit:LOWER:SEGMENT:START:Y?	4-2
:CALCulate#:LIMit:LOWER:SEGMENT:STOP:X <value>	4-2
:CALCulate#:LIMit:LOWER:SEGMENT:STOP:X?	4-2
:CALCulate#:LIMit:LOWER:SEGMENT:STOP:Y <amplitude>	4-2
:CALCulate#:LIMit:LOWER:SEGMENT:STOP:Y?	4-2
:CALCulate#:LIMit:LOWER:SEGMENT:TOTal?	4-3
:CALCulate#:LIMit:LOWER:Y <amplitude> {dB,s ms ns ps}	4-3
:CALCulate#:LIMit:LOWER:Y?	4-3
:CALCulate#:LIMit:LOWER[:STATe] OFF ON 0 1	4-3
:CALCulate#:LIMit:LOWER[:STATe]?	4-3
:CALCulate#:LIMit:UPPer:SEGMENT:ADD <StartX><StopX><StartY><StopY>	4-3
:CALCulate#:LIMit:UPPer:SEGMENT:DELeTe	4-3
:CALCulate#:LIMit:UPPer:SEGMENT:EDIT <StartX><StopX><StartY><StopY>	4-4
:CALCulate#:LIMit:UPPer:SEGMENT:START:X <value>	4-4
:CALCulate#:LIMit:UPPer:SEGMENT:START:X?	4-4
:CALCulate#:LIMit:UPPer:SEGMENT:START:Y <amplitude>	4-4
:CALCulate#:LIMit:UPPer:SEGMENT:START:Y?	4-4
:CALCulate#:LIMit:UPPer:SEGMENT:STOP:X <value>	4-5
:CALCulate#:LIMit:UPPer:SEGMENT:STOP:X?	4-5
:CALCulate#:LIMit:UPPer:SEGMENT:STOP:Y <amplitude>	4-5
:CALCulate#:LIMit:UPPer:SEGMENT:STOP:Y?	4-5
:CALCulate#:LIMit:UPPer:SEGMENT:TOTal?	4-5
:CALCulate#:LIMit:UPPer:Y <amplitude> {dB,s ms ns ps}	4-5
:CALCulate#:LIMit:UPPer:Y?	4-5
:CALCulate#:LIMit:UPPer[:STATe] OFF ON 0 1	4-6
:CALCulate#:LIMit:UPPer[:STATe]?	4-6
:CAPTure:START	4-14
:CAPTure:STOP	4-14
:CAPTure:TRACe#:DATA? FDATA SDATA FMEM SMEM	4-14
:CONFigure:BASE:FACTorydefault	3-1
:CONFigure:BASE:MASTerreset	3-1
:CONFigure:MEASure:DISPlay SINGLE DUAL	4-15
:CONFigure:MEASure:DISPlay?	4-15
:CONFigure:MEASure:MODE	
RLFReq RLDTf SWRFreq SWRDtf CLFReq TRES SMITH PHASe	4-15

:CONFigure:MEASure:MODE?	4-15
:DISPlay:UPDate ON OFF 1 0.	4-15
:DISPlay:UPDate?	4-15
:DISPlay:WINDow:TRACe:STATe TRACe MEMory BOTH.	4-16
:DISPlay:WINDow:TRACe:STATe?	4-16
:DISPlay:WINDow:TRACe:Y[:SCALe]:AUToscale	4-16
:DISPlay:WINDow:TRACe:Y[:SCALe]:BOTTom <amplitude> {dB}	4-16
:DISPlay:WINDow:TRACe:Y[:SCALe]:BOTTom?	4-16
:DISPlay:WINDow:TRACe:Y[:SCALe]:FULLscale	4-16
:DISPlay:WINDow:TRACe:Y[:SCALe]:TOP <amplitude> {dB}	4-17
:DISPlay:WINDow:TRACe:Y[:SCALe]:TOP?	4-17
:FETCh:GPSTData? {CURRENT LAST}	3-1
:FORMat[:READings][:DATA] ASCii INTEger,32 REAL,32	4-17
:FORMat[:READings][:DATA]:MULTiplier 1 1M	4-17
:FORMat[:READings][:DATA]:MULTiplier?	4-17
:FORMat[:READings][:DATA]?	4-17
:INITiate:CONTInuous OFF ON 0 1	4-18
:INITiate:CONTInuous?	4-18
:INITiate:HOLD OFF ON 0 1	4-18
:INITiate:HOLD?	4-18
:INITiate[:IMMediate]	4-18
:INITiate#:DATA? FDATA SDATA FMEM SMEM	4-17
:INSTrument:CATalog:FULL?	3-2
:INSTrument:NSElect <application-mode-number>	3-2
:INSTrument:NSElect?	3-2
:INSTrument:SElect <"mode-identifier">	3-2
:INSTrument:SElect?	3-2
:PROGram:ETT:ABORTscript.	3-2
:PROGram:ETT:LOADscript <"filename">	3-3
:PROGram:ETT:NEXTstep.	3-3
:PROGram:ETT:STATe?	3-3
:STATus:OPERation?	4-18
:SYSTem:OPTions?	3-3
:SYSTem:PRESet	3-3
:TRACe[:DATA]? {1 2}	4-19
[[:SENSe]:APPLication:TST:RESult?	4-19
[[:SENSe]:APPLication:TST? {NORMal}	4-19
[[:SENSe]:BASE:IPaddress?	3-3
[[:SENSe]:BASE:REMote 0 1 2	3-4
[[:SENSe]:BASE:SCRNshot:SCHeM InVerted STANdard	3-4
[[:SENSe]:BASE:SCRNshot:SCHeM?	3-4
[[:SENSe]:CALibration:STATe?	4-19
[[:SENSe]:CORRection:COLlect:ABORT.	4-19
[[:SENSe]:CORRection:COLlect:INFO:STATus?	4-20
[[:SENSe]:CORRection:COLlect:INITialize	4-20
[[:SENSe]:CORRection:COLlect:LOAD	4-20
[[:SENSe]:CORRection:COLlect:OPEN.	4-20
[[:SENSe]:CORRection:COLlect:SAVe	4-20
[[:SENSe]:CORRection:COLlect:SHORT	4-20
[[:SENSe]:CORRection:COLlect:STATus?	

{INITialize OPEN SHORT LOAD SAVE ALL THRU ZERO}	4-21
[:SENSe]:CORRection:COLLect:THRU	4-21
[:SENSe]:CORRection:COLLect:TYPE RFP1 TRES	4-21
[:SENSe]:CORRection:COLLect:TYPE?	4-21
[:SENSe]:CORRection:COLLect:ZERO	4-21
[:SENSe]:CORRection:TYPE STANdard FLEX	4-21
[:SENSe]:CORRection:TYPE?	4-21
[:SENSe]:CORRection[:STATe] ON OFF 1 0	4-22
[:SENSe]:CORRection[:STATe]?	4-22
[:SENSe]:FREQuency:CABLe <cable-list-index>	4-22
[:SENSe]:FREQuency:CABLe:CATalog:FULL?	4-22
[:SENSe]:FREQuency:CABLe:NAME <"cable-name">	4-22
[:SENSe]:FREQuency:CABLe:NAME?	4-22
[:SENSe]:FREQuency:CABLe?	4-22
[:SENSe]:FREQuency:STARt <frequency> { Hz kHz MHz GHz}	4-23
[:SENSe]:FREQuency:STARt?	4-23
[:SENSe]:FREQuency:STOP <frequency> { Hz kHz MHz GHz}	4-23
[:SENSe]:FREQuency:STOP?	4-23
[:SENSe]:SWEep:RESolution 130 259 517 1033 2065	4-23
[:SENSe]:SWEep:RESolution?	4-23
[:SENSe]:SWEep:RFIMmunity HIGH LOW 1 0	4-24
[:SENSe]:SWEep:RFIMmunity?	4-24
[:SENSe]:SWEep:TYPE CONTinuous SINGle EXtErnal	4-24
[:SENSe]:TRACe:SElect 1 2	4-24
[:SENSe]:TRACe:SElect?	4-24
*IDN?	3-1
*OPC?	3-1



10580-00474



A



Anritsu utilizes recycled paper and environmentally conscious inks and toner.

Anritsu Company
490 Jarvis Drive
Morgan Hill, CA 95037-2809
USA
<http://www.anritsu.com>