Keysight E4981A 120 Hz/1 kHz/1 MHz Capacitance Meter

Programming
Manual

KEYSIGHT
TECHNOLOGIES

Notices

Manual Part Number

E4981-90001

Edition

Edition 3, October 8, 2019

Printed in Malaysia

Published by:

Keysight Technologies International
Japan G.K,
1-3-3 Higashikawasaki-cho
Chuo-ku
Kobe-shi, Hyogo, Japan

Warranty

THE MATERIAL CONTAINED IN THIS DOCUMENT IS PROVIDED "AS IS," AND IS SUBJECT TO BEING CHANGED, WITHOUT NOTICE, IN FUTURE EDITIONS. FURTHER, TO THE MAXIMUM EXTENT PERMITTED BY APPLICABLE LAW, KEYSIGHT DISCLAIMS ALL WARRANTIES, EITHER EXPRESS OR IMPLIED WITH REGARD TO THIS MANUAL AND ANY INFORMATION CONTAINED HEREIN, INCLUDING BUT NOT LIMITED TO THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. KEYSIGHT SHALL NOT BE LIABLE FOR ERRORS OR FOR INCIDENTAL OR CONSEQUENTIAL DAMAGES IN CONNECTION WITH THE FURNISHING, USE, OR PERFORMANCE OF THIS DOCUMENT OR ANY INFORMATION CONTAINED HEREIN. SHOULD KEYSIGHT AND THE USER HAVE A SEPARATE WRITTEN AGREEMENT WITH WARRANTY TERMS COVERING THE MATERIAL IN THIS DOCUMENT THAT CONFLICT WITH THESE TERMS, THE WARRANTY TERMS IN THE SEPARATE AGREEMENT WILL CONTROL.

Technology Licenses

The hardware and/or software described in this document are furnished under a license and may be used or copied only in accordance with the terms of such license.

Declaration of Conformity

Declarations of Conformity for this product and for other Keysight products may be downloaded from the Web. Go to http://www.keysight.com/go/conformity. You can then search by product number to find the latest Declaration of Conformity.

U.S. Government Rights

The Software is "commercial computer software," as defined by Federal Acquisition Regulation ("FAR") 2.101. Pursuant to FAR 12.212 and 27.405-3 and Department of Defense FAR Supplement ("DFARS") 227.7202, the U.S. government acquires commercial computer software under the same terms by which the software is customarily provided to the public. Accordingly, Keysight provides the Software to U.S. government customers under its standard commercial license, which is embodied in its End User License Agreement (EULA), a copy of which can be found at http://www.keysight.com/find/sweula. The license set forth in the EULA represents the exclusive authority by which the U.S. government may use, modify, distribute, or disclose the Software. The EULA and the license set forth therein, does not require or permit, among other things, that Keysight: (1) Furnish technical information related to commercial computer software or commercial computer software documentation that is not customarily provided to the public; or (2) Relinquish to, or otherwise provide, the government rights in excess of these rights customarily provided to the public to use, modify, reproduce, release, perform, display, or disclose commercial computer software or commercial computer software documentation. No additional government requirements beyond those set forth in the EULA shall apply, except to the extent that those terms, rights, or licenses are explicitly required from all providers of commercial computer software pursuant to the FAR and the DFARS and are set forth specifically in writing elsewhere in the EULA. Keysight shall be under no obligation to update, revise or otherwise modify the Software. With respect to any technical data as defined by FAR 2.101, pursuant to FAR 12.211 and 27.404.2 and DFARS 227.7102, the U.S. government acquires no greater than Limited Rights as defined in FAR 27.401 or DFAR 227.7103-5 (c), as applicable in any technical data.

Safety Notices

CAUTION

A **CAUTION** notice denotes a hazard. It calls attention to an operating procedure, practice, or the like that, if not correctly performed or adhered to, could result in damage to the product or loss of important data. Do not proceed beyond a CAUTION notice until the indicated conditions are fully understood and met.

WARNING

A **WARNING** notice denotes a hazard. It calls attention to an operating procedure, practice, or the like that, if not correctly performed or adhered to, could result in personal injury or death. Do not proceed beyond a WARNING notice until the indicated conditions are fully understood and met.

Table of Contents

A.Manual Changes

B.Information for Replacing 4268A, 4288A with E4981A

C.Status Reporting System

D.Operations when Overload, No Contact, or Low C is Detected

E.Error Messages

1 Making Effective Use of This Manual

This chapter provides an overview of this manual as well as useful information to help you navigate through the manual. It also briefly describes how to use this manual, focusing on how you can look up particular commands.

Contents of This Manual

This manual is a programming guide for the Keysight E4981A 120 Hz/1 kHz/1 MHz capacitance meter. It provides detail explanations of how to remotely control the E4981A from a PC. The chapter-by-chapter contents of this manual are as follows:

Chapter 1 “Making Effective Use of This Manual”

This chapter provides an overview of this manual as well as useful information to help you navigate through the manual. It also briefly describes how to use this manual, focusing on how you can look up particular commands.

Chapter 2 “Overview of Remote Control”

This chapter provides an overview of the remote control system and the SCPI commands.

Chapter 3 “Setting Up Measurement Conditions and Display”

This chapter describes how to set up the measurement conditions and display. It also describes how to save/recall the instrument settings including the measurement conditions.

Chapter 4 “Preparation for Accurate Measurement (Executing Correction)”

This chapter describes how to execute the correction function.

Chapter 5 “Starting (Triggering) Measurement and Waiting for Completion of Measurement”

This chapter describes how to trigger the instrument to start measurement and how to detect completion of measurement.

Chapter 6 “Reading Out Measured Result”

This chapter describes how to read out the measured result and the measurement signal level monitor result.

Chapter 7 “Sorting Based on Measured Result (Comparator Function)”

This chapter describes how to use the comparator function to perform sorting based on the measured result.

Chapter 8 “Avoiding Mistakes Related to Work and Daily Checks”

This chapter describes how to avoid simple mistakes related to work, how to detect the occurrence of an error, and how to execute the self-test.

Chapter 9 “Measurement Applications (Sample Programs)”

This chapter contains sample programs in Excel VBA format for both basic measurement and measurement using a system integrated with the handler/scanner interface.

Chapter 10 “Command Reference”

This chapter provides the SCPI command reference for the Keysight E4981A. Each command is fully described and ordered alphabetically based on its abbreviated name format. Use the index to look up a SCPI command by its full syntax. To find a command according to its function, refer to the “SCPI Command Table” on page 307.

Appendix A “Manual Changes”

This appendix contains the information required to adapt this manual to earlier versions or configurations of the Keysight E4981A than that indicated by the current printing date of this manual. The information in this manual applies directly to the E4981A model that has the serial number prefix listed on the title page of this manual.

Appendix B “Information for Replacing 4268A, 4288A with E4981A”

This appendix describes information that is applicable when replacing the Keysight 4268A, 4288A with the Keysight E4981A. See the 4268A, 4288A Operation Manual for detailed information on the 4268A, 4288A. See the E4981A Operation Manual and other chapters of this manual (Programming Manual) for more detailed information on the E4981A.

Appendix C “Status Reporting System”

This appendix describes the status reporting system of the Keysight E4981A.

Appendix D “Operations when Overload, No Contact, or Low C is Detected”

This appendix describes display output, GPIB/LAN/USB output, and handler interface output when an Overload, No contact, or Low C is detected.

Appendix E “Error Messages”

The Keysight E4981A provides error messages to indicate its operating status. This appendix describes the error messages of the E4981A in order of error number. To search for error messages alphabetically, refer to **the Operation Manual**.

How To Use This Manual

Chapters 3 to 8 provide task-based descriptions of SCPI commands that are useful for programming and explain how you can use them. These chapters contain explanations and sample program listings that you can use to develop your custom programs. For more information on individual commands, see Chapter 10, “Command Reference.”

Looking up SCPI commands

Chapter 10, “Command Reference,” contains a complete reference of SCPI commands. You can look up a particular SCPI command in any of the following ways:

Lookup by Abbreviated Command Name

The command reference is organized alphabetically according to the abbreviated name used as the title for each command’s description.

Lookup by Full Command Name

You can use the index at the end of the manual to find full command names along with the page numbers where they appear.

Lookup by Command Function

Table 10-1 on page 307 provides a complete list of commands by function and indicates the page numbers where the commands appear in the command reference.

Lookup by Front panel key

Table 10-2 on page 318 provides a complete list of commands that correspond to the front panel key tree and indicates the page numbers where the commands appear in the command reference.

NOTE

Some SCPI commands supported by the E4981A have optional syntax elements. In the command reference conventions, these elements are enclosed between square brackets ([ ]) or printed in lowercase letters. See “Syntax” on page 155 for more information.

Using sample programs

This manual comes with source codes of sample programs.

Looking up a sample program

To look up the description of a sample program, see the listings under “Sample program” in the index.

2 Overview of Remote Control

This chapter provides an overview of the remote control system and the SCPI commands.

Types of Remote Control System

Depending on the system controller and the interface, you can configure three types of remote control system as shown in the table below.

System controller	Interface	Overview
External Controller (external computer such as PC)	GPIB	System to control the E4981A and other devices connected via GPIB from the external controller. For more information, refer to "GPIB Remote Control System" on page 19.
	LAN	System to control the E4981A and other devices connected via LAN from the external controller. For more information, refer to "LAN Remote Control System" on page 21.
	USB	System to control the E4981A and other devices connected via USB from the external controller. For more information, refer to "USB Remote Control System" on page 33.

NOTE

You must install Keysight I/O Libraries Suite in the external controller in advance.

Use Keysight I/O Libraries Suite 15.0 or higher.

For further information on I/O Libraries Suite, see the Keysight I/O Libraries Suite manual.

Keysight I/O Libraries Suite may not be available for certain external controllers or OS versions. For further details, refer to the Help guidance for Keysight I/O Libraries Suite.

GPIB Remote Control System

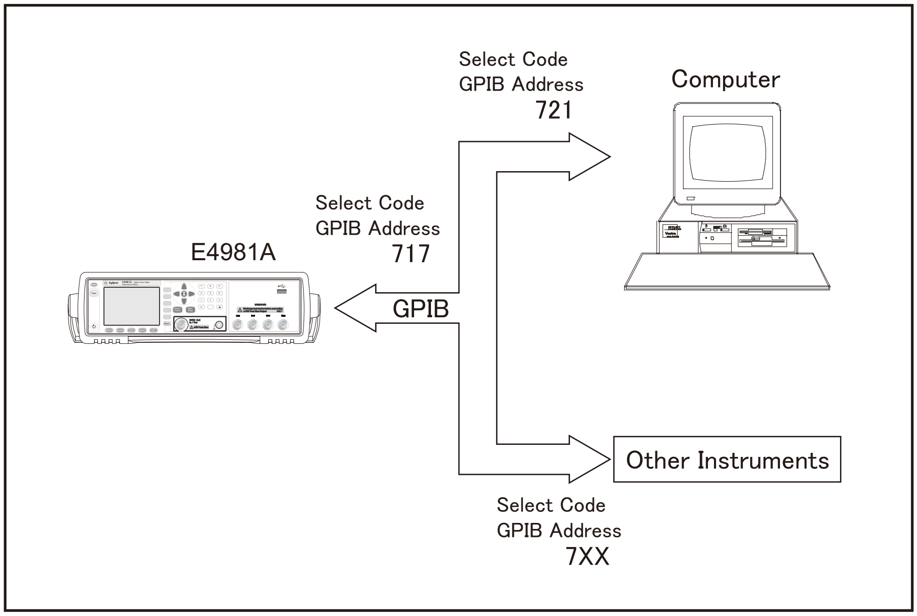
What is GPIB?

GPIB (General Purpose Interface Bus) is an interface standard for connecting computers and peripherals that supports the following international standards: IEEE 488.1, IEC-625, IEEE 488.2, and JIS-C1901. The GPIB interface allows you to control the Keysight E4981A from an external computer. The computer sends commands and instructions to the E4981A and receives data sent from the E4981A via the GPIB.

System configuration

Use GPIB cables to make connections between the E4981A, the external controller (computer), and peripherals. Figure 2-1 shows an overview of the GPIB remote control system's configuration.

Figure 2-1 Configuration of the GPIB remote control system



e4981aue0017

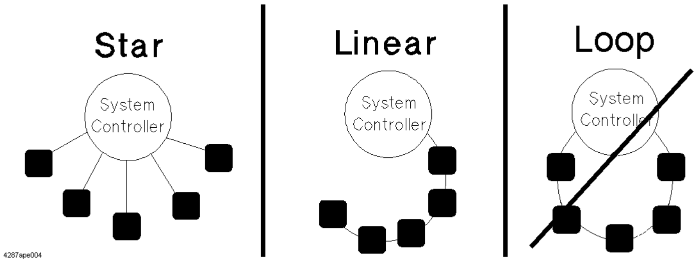
Required equipment

1. E4981A
2. External controller (computer)

   Use a personal computer or workstation equipped with the GPIB interface. You need to install software to control this instrument via the GPIB into the external controller (for example, HTBasic or Keysight VEE).
3. Other devices (other instruments and/or peripherals that serve your purpose)
4. GPIB cables for connecting the E4981A, the external controller, and other devices

Scale of the system you can construct

- You can connect up to 15 devices in a single GPIB system.
- The length of cables to make connections between devices must be 4 m or less. The total length of connecting cables in a single GPIB system must be 2 m × the number of connected devices (including the controller) or less. You cannot construct a system in which the total cable length exceeds 20 m.
- The number of connectors connected to an individual device must be four or less. If you connect five or more connectors, excessive force will be applied to the connector part, which may result in failure.
- You can choose the device connection topology from star, linear, and combined. Loop connection is not supported.



Device selector

The device selector is a unique value assigned to each device that is used by the controller to select the control target (to send/receive messages) among devices connected on the GPIB remote control system.

The device selector consists of a select code (usually, 7) and a GPIB address. For example, when the select code is 7 and the GPIB address is 17, the device selector is 717. The select code must be individually set for each system. The GPIB address must be set to a unique value for each device, and is used to identify devices on the same system. In the description and sample programs in this manual, it is assumed that the device selector is set to 717.

Setting the GPIB address of the E4981A

[System] - **SYSTEM CONFIG** - **GPIB ADDR**

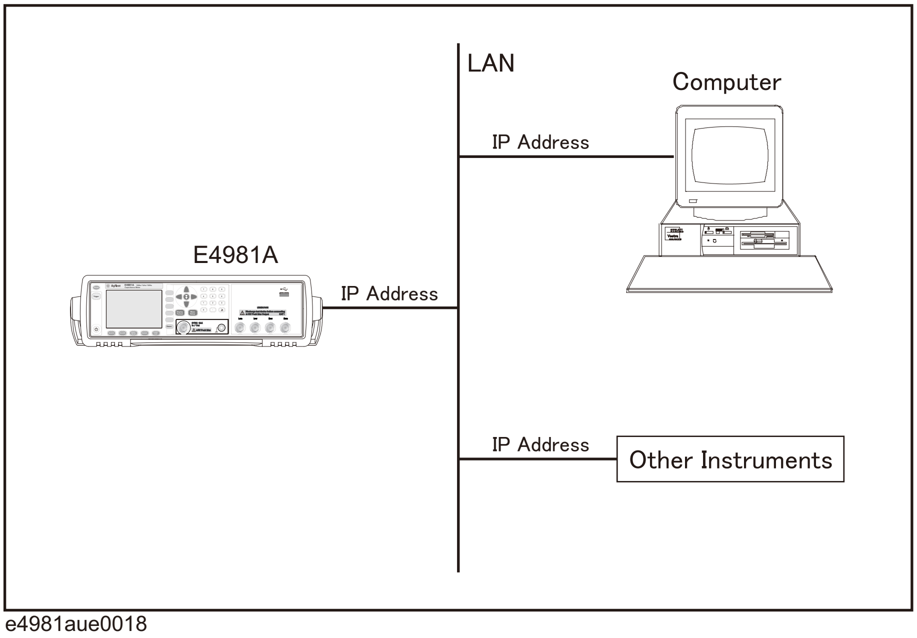
LAN Remote Control System

The LAN (Local Area Network) remote control system provides two methods: controlling the E4981A using the SICL-LAN server and controlling the E4981A using the telnet server.

System configuration

Use a LAN cable to connect the E4981A to the external controller (computer). Figure 2-2 shows an overview of the LAN remote control system's configuration.

Figure 2-2 Configuration of the LAN remote control system



Required equipment

1. E4981A
2. External controller (personal computer or workstation that can be connected to a LAN and with Keysight I/O Libraries Suite installed)
3. Other devices (other instruments and/or peripherals that serve your purpose)
4. LAN cable for connecting the E4981A with the external controller

Preparing the E4981A

Before controlling the E4981A via a LAN, you need to configure the network function. For detailed information on the procedure, refer to *Operation Manual*.

Control over SICL-LAN server

In a control system using the SICL-LAN server, communication between the external controller (client) and the E4981A (server) is performed using the SICL-LAN protocol. Communication is performed using SICL (Standard Instrument Control Library). You can control the E4981A by programming using SICL or VISA in C language in a UNIX environment, or Visual C++, Visual Basic, or VEE in a Windows environment.

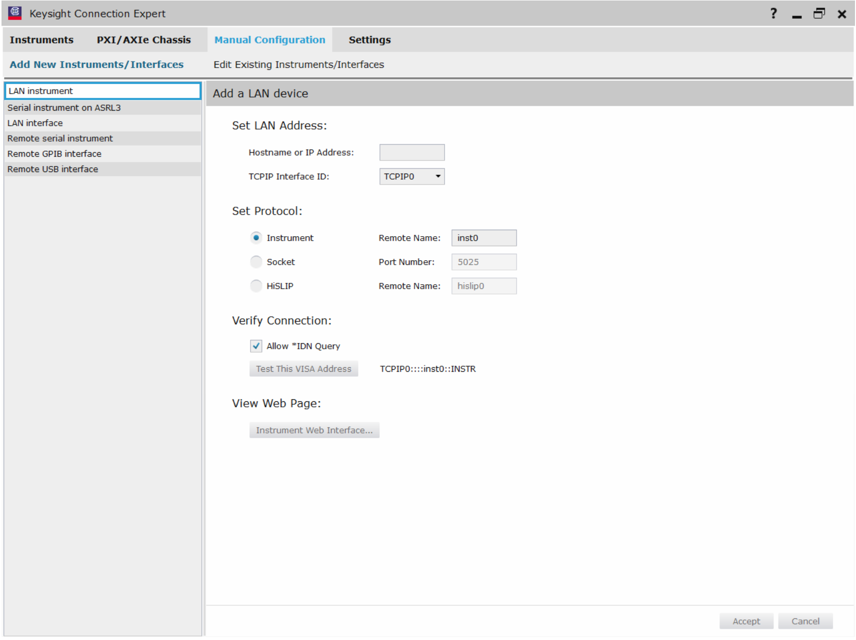
NOTE **In E4981A, SICL-LAN (VXI-11) value cannot be set to OFF.**

Preparing the external controller

To establish communication with the E4981A using the TCP/IP protocol, you need to set the I/O interface of the external controller in advance. This section shows the setting procedure when using the external controller in the Windows environment.

Step 1. From your PC’s Start menu, click Program - Keysight I/O Libraries Suite - Keysight Connection Expert to open the Keysight Connection Expert setting screen.

Step 2. In the Keysight Connection Expert setting screen, select **Manual Configuration** and at the **Add New Instruments/Interfaces** tab, select **LAN instrument.**



Add a LAN device

Set LAN Address:

Hostname or IP Address:

TCPIP Interface ID: TCPIP0

Set Protocol:

● Instrument Remote Name: inst0

○ Socket Port Number: 5025

○ HiSLIP Remote Name: hislip0

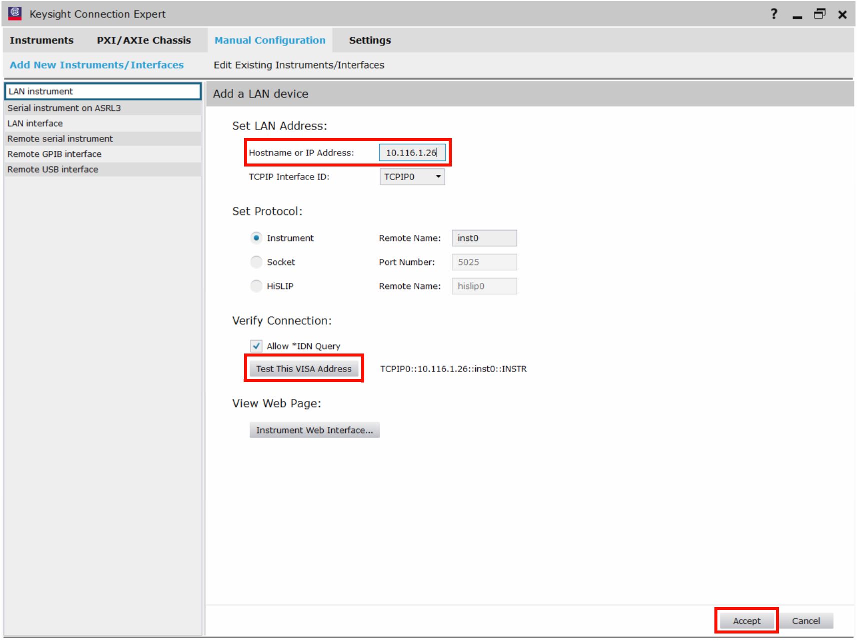
Verify Connection:

☑ Allow *IDN Query

TCPIP0::::inst0::INSTR

View Web Page:

Step 3. Under **Add a LAN device**, enter the **Hostname or IP Address** and click **Test This VISA Address**. Then, click **Accept**.



Add a LAN device

Set LAN Address:

Hostname or IP Address: 10.116.1.26

TCPIP Interface ID: TCPIP0

Set Protocol:

● Instrument — Remote Name: inst0

○ Socket — Port Number: 5025

○ HiSLIP — Remote Name: hislip0

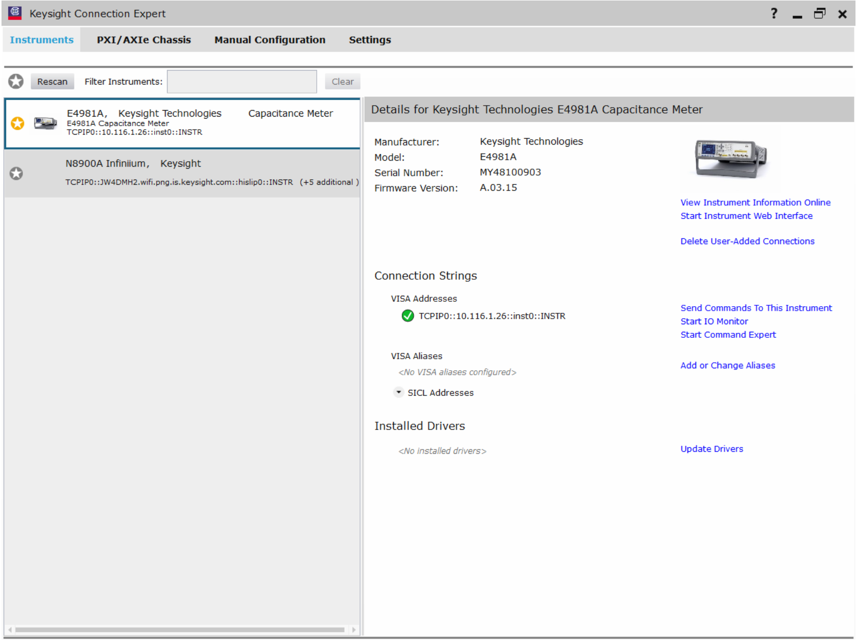
Verify Connection:

☑ Allow *IDN Query

TCPIP0::10.116.1.26::inst0::INSTR

View Web Page:

Step 4. In the Keysight Connection Expert's Instruments screen, check that the E4981A has been added.



E4981A, Keysight Technologies — Capacitance Meter
E4981A Capacitance Meter
TCPIP0::10.116.1.26::inst0::INSTR

N8900A Infiniium, Keysight
TCPIP0::JW4DMH2.wifi.png.is.keysight.com::hislip0::INSTR (+5 additional)

Details for Keysight Technologies E4981A Capacitance Meter

Manufacturer: Keysight Technologies
Model: E4981A
Serial Number: MY48100903
Firmware Version: A.03.15

Connection Strings

VISA Addresses

✓ TCPIP0::10.116.1.26::inst0::INSTR

VISA Aliases

<No VISA aliases configured>

SICL Addresses

Installed Drivers

<No installed drivers>

Control using C or Visual Basic

You can control the E4981A by programming using SICL/VISA in C language in a UNIX environment, or Visual C++ or Visual Basic in a Windows environment.

For more information on the control method, see a sample program using the VBA macro of Microsoft Excel described in “Section 9, Sample Program.”

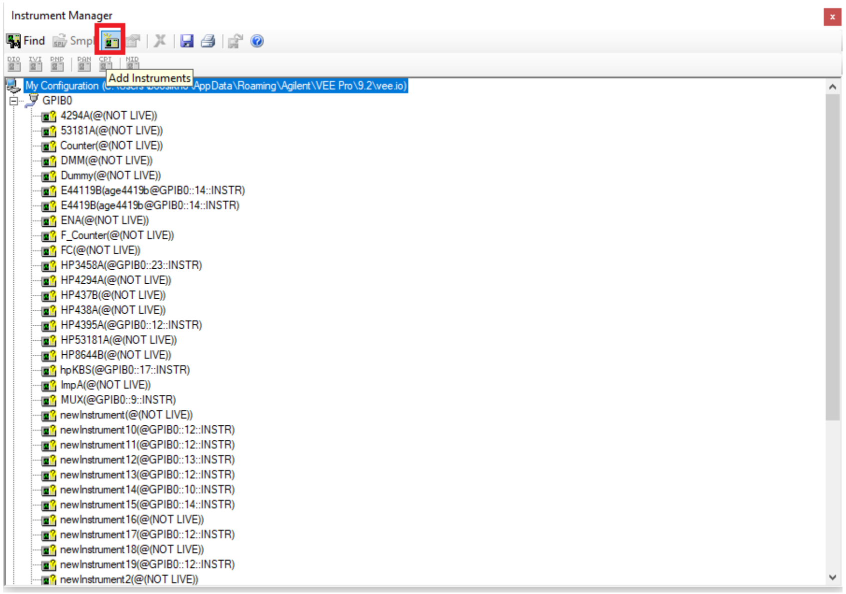
Control using Keysight VEE

Keysight VEE allows you to control the E4981A via the direct I/O interface. The following example shows how to control the E4981A whose IP address is set to 192.168.1.101.

NOTE When using Keysight VEE for PCs, use Keysight VEE Pro 6 for Windows or higher.

Step 1. On the Keysight VEE’s **I/O** menu, click **Instrument Manager....**

Step 2. In the Instrument Manager setting screen, click the **Add Instruments** icon.

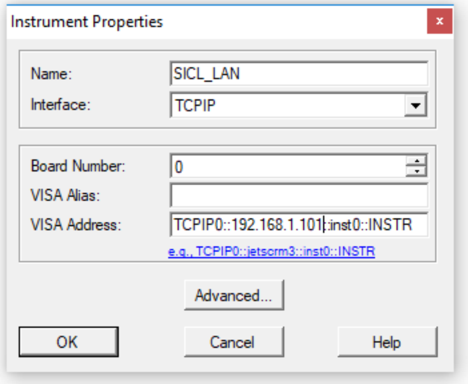


```
My Configuration (                 AppData\Roaming\Agilent\VEE Pro\9.2\vee.io)
  GPIB0
    4294A(@(NOT LIVE))
    53181A(@(NOT LIVE))
    Counter(@(NOT LIVE))
    DMM(@(NOT LIVE))
    Dummy(@(NOT LIVE))
    E44119B(age4419b@GPIB0::14::INSTR)
    E4419B(age4419b@GPIB0::14::INSTR)
    ENA(@(NOT LIVE))
    F_Counter(@(NOT LIVE))
    FC(@(NOT LIVE))
    HP3458A(@GPIB0::23::INSTR)
    HP4294A(@(NOT LIVE))
    HP437B(@(NOT LIVE))
    HP438A(@(NOT LIVE))
    HP4395A(@GPIB0::12::INSTR)
    HP53181A(@(NOT LIVE))
    HP8644B(@(NOT LIVE))
    hpKBS(@GPIB0::17::INSTR)
    ImpA(@(NOT LIVE))
    MUX(@GPIB0::9::INSTR)
    newInstrument(@(NOT LIVE))
    newInstrument10(@GPIB0::12::INSTR)
    newInstrument11(@GPIB0::12::INSTR)
    newInstrument12(@GPIB0::13::INSTR)
    newInstrument13(@GPIB0::12::INSTR)
    newInstrument14(@GPIB0::10::INSTR)
    newInstrument15(@GPIB0::14::INSTR)
    newInstrument16(@(NOT LIVE))
    newInstrument17(@GPIB0::12::INSTR)
    newInstrument18(@(NOT LIVE))
    newInstrument19(@GPIB0::12::INSTR)
    newInstrument2(@(NOT LIVE))
```

Step 3. The *Add Interface/Device* window appears. Select *Interface Type* as **TCPIP** and press **OK**. The *Instrument Properties* window will then appear.

Make the settings as follows:

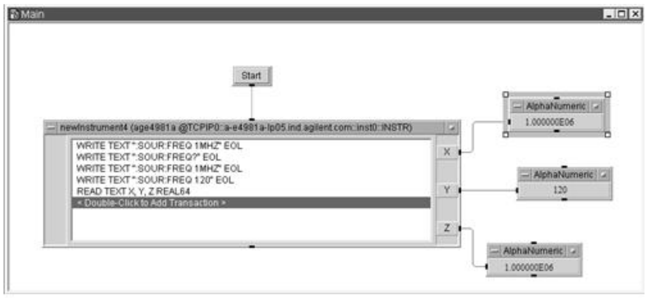
Name: **SICL_LAN** (you can specify any name), Interface: **TCPIP**,
Board Number: **0**, and VISA Address: **TCPIP::192.168.1.101::inst0::INSTR**.



Name: SICL_LAN
Interface: TCPIP
Board Number: 0
VISA Alias:
VISA Address: TCPIP0::192.168.1.101::inst0::INSTR

Figure 2-3 shows an example of control using the direct I/O interface that has been set in the above procedure.

Figure 2-3 Example of control using Keysight VEE


Main
Start
newInstrument4 (age4981a @TCPIP0::a-e4981a-lp05.ind.agilent.com::inst0::INSTR)
X
Y
Z
AlphaNumeric
1.000000E06
AlphaNumeric
120
AlphaNumeric
1.000000E06

```
WRITE TEXT ":SOUR:FREQ 1MHZ" EOL
WRITE TEXT ":SOUR:FREQ?" EOL
WRITE TEXT ":SOUR:FREQ 1MHZ" EOL
WRITE TEXT ":SOUR:FREQ 120" EOL
READ TEXT X, Y, Z REAL64
< Double-Click to Add Transaction >
```

e4981a0009

Control over telnet server

In a control system that operates over the telnet server, communications are performed through connection between the sockets provided by the processes of the external controller and the E4981A to establish a network path between them.

A socket is an endpoint for network connection; port 5024 and port 5025 are provided for the sockets for the E4981A. Port 5024 is provided for conversational control using telnet (user interface program for the TELNET protocol) and port 5025 for control from a program.

Preparing the external controller

As in the case of the SICL-LAN server, you need to set the I/O interface of the external controller in advance to establish communication to the E4981A using the TCP/IP protocol. For the procedure, refer to “Preparing the external controller” on page 23 of the “Control over SICL-LAN server” section.

Conversational control using telnet (using port 5024)

You can use telnet to perform conversational control by sending SCPI commands to the E4981A on a message-by-message basis. For telnet, the socket of port 5024 is used for communications.

NOTE

For port 5024, service requests are asynchronous. Also, use Ctrl+C to clear the device.

In this example, to show you the control procedure using telnet, in a Windows environment you control the E4981A from the external controller.

Step 1. Open the MS-DOS command prompt screen.

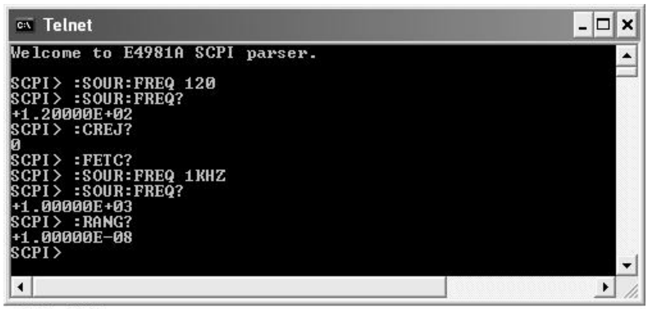
Step 2. At the MS-DOS prompt, type “telnet <IP Address>” and press Enter.

Step 3. The telnet screen opens.

Step 4. Type a command and press Enter; it is sent to the E4981A and executed. If you enter a command that queries some data, the query response is displayed below the line in which you entered the command.

Figure 2-4 shows the screen displaying the measured value with :FETC? after using the :SOUR:FREQ command to set measurement frequency to 120Hz\1kHz, :CREJ? to check the low C reject status. The setting check is made with the query after each setting.

Figure 2-4 Example of control using telnet



```
Welcome to E4981A SCPI parser.

SCPI> :SOUR:FREQ 120
SCPI> :SOUR:FREQ?
+1.20000E+02
SCPI> :CREJ?
0
SCPI> :FETC?
SCPI> :SOUR:FREQ 1KHZ
SCPI> :SOUR:FREQ?
+1.00000E+03
SCPI> :RANG?
+1.00000E-08
SCPI>
```

e4981a0001

Step 5. Press] while holding down Ctrl in the telnet screen to break the connection to the E4981A. The telnet prompt appears. Type “quit” at the telnet prompt and press Enter to finish using telnet.

NOTE In E4981A, Telnet cannot be set to OFF.

Control from a program (using port 5025)

When controlling the E4981A from a program on the external controller, use the socket of port 5025 for connection. Use `:SYSTem:COMMunicate:LAN[:SELF]:CONTrol?` on page 276 to obtain the connection port number.

NOTE In E4981A, Socket cannot be set to OFF.

Control using C or Visual Basic

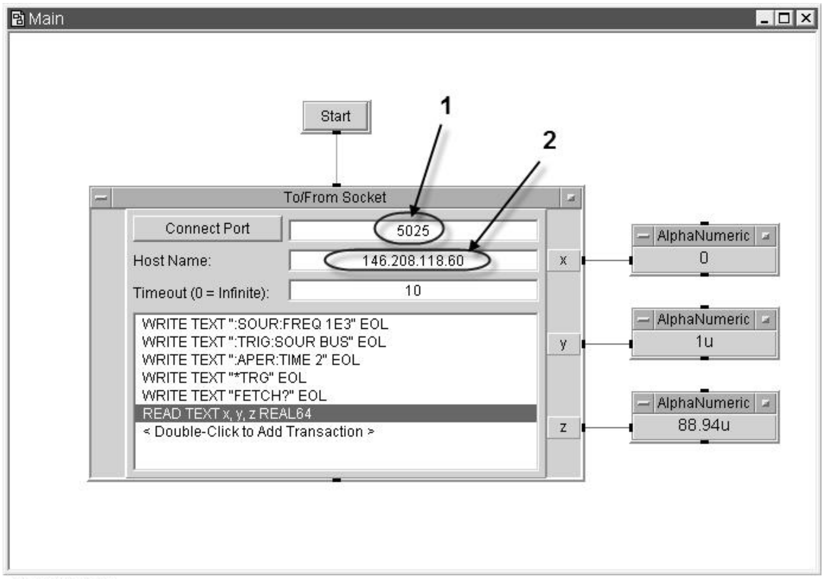
You can control the E4981A by socket programming using C language in a UNIX environment, or Visual C++ or Visual Basic in a Windows environment.

For socket programming, the library for network connection on the TCP/IP protocol is required. For a UNIX environment, BSD (Berkeley Software Distribution) Sockets API is available; for a Windows environment, WinSock (WinSock1.1 and WinSock2.0), created by porting BSD Sockets to Windows and expanding it, is available.

Control using Keysight VEE

Keysight VEE allows you to control the E4981A through the connection to the socket of port 5025 using the To/From Socket. Figure 2-5 shows an example (when the IP address of the E4981A is 192.168.1.101). Enter “5025” in the field to specify the port for connection (1 in Figure 2-5) and enter the IP address of the E4981A in the field to specify the host name (2 in Figure 2-5).

Figure 2-5 Example of control using Keysight VEE



Connect Port: 5025
Host Name: 146.208.118.60
Timeout (0 = Infinite): 10

```
WRITE TEXT ":SOUR:FREQ 1E3" EOL
WRITE TEXT ":TRIG:SOUR BUS" EOL
WRITE TEXT ":APER:TIME 2" EOL
WRITE TEXT "*TRG" EOL
WRITE TEXT "FETCH?" EOL
READ TEXT x, y, z REAL64
< Double-Click to Add Transaction >
```

x: 0
y: 1u
z: 88.94u

e4980auj1042

Control via Web server

For control over a Web server, communications are performed between the external controller and the E4981A through a LAN, regarding the E4981A as a Web server. You can control the E4981A and send SCPI commands from the external controller by displaying the E4981A’s front panel in the external controller with Internet Explorer (IE6.0 SP2 or later).

In addition, you can import the screen and recall the measurement data.

Preparing external controller

As in the case of the SICL-LAN server, you need to set the I/O interface of the external controller in advance to establish communication to the E4981A using the TCP/IP protocol. For the procedure, refer to“Preparing the external controller” on page 23 of “Control over SICL-LAN server” section.

Control using web server

The following steps show how to control the E4981A using Internet Explorer.

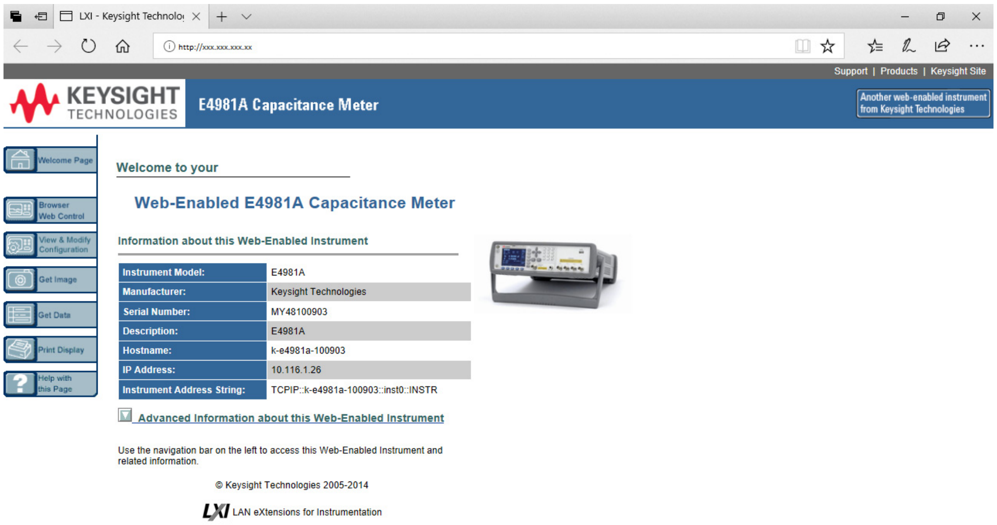
Step 1. Start Internet Explorer.

Step 2. Type <IP Address> in the address field and press the enter key.

Enter CURRENT IP ADDR on the E4981A’s SYSTEM CONFIG page for an IP address.

Step 3. The Web server’s start screen appears.

Figure 2-6 Web server start screen



Welcome to your

Web-Enabled E4981A Capacitance Meter

Information about this Web-Enabled Instrument

Instrument Model:	E4981A
Manufacturer:	Keysight Technologies
Serial Number:	MY48100903
Description:	E4981A
Hostname:	k-e4981a-100903
IP Address:	10.116.1.26
Instrument Address String:	TCPIP::k-e4981a-100903::inst0::INSTR

Advanced Information about this Web-Enabled Instrument

Use the navigation bar on the left to access this Web-Enabled Instrument and related information.

© Keysight Technologies 2005-2014

LXI LAN eXtensions for Instrumentation

Web Server function

The Web Server function consists of the following pages.

Page	Description
Welcome Page	Displays various setting information
Browser Web Control	Simulated front panel (on Web) and the function to input/output SCPI command and requires password.
View & Modify Configuration	Displays and modifies various setting remote connection information
Get Image	Displays current screen with gif format
Get Data	Requires password and displays the content of memory buffer/Buffer3
Print Display	Calls the printing function of the browser
Help with this Page	Displays the Help file

NOTE

To use the Web Server, refer to “Help with this Page.”

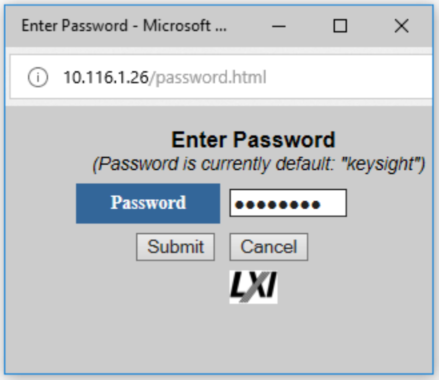
Password of Web Server function

For the Web Server function, a password must be entered when the following operation is performed.

The default password is “keysight”.

- When pressing the Modify Configuration button on the View & Modify Configuration page.
- When moving to the Browser Web Control page from a certain page.
- When moving to the Get Data page from a certain page.

Figure 2-7 Password entry screen


Enter Password - Microsoft ...

10.116.1.26/password.html

Enter Password

(Password is currently default: "keysight")

Password ••••••••

Changing the password of the Web Server function

You can change the password of the Web Server function.

NOTE

The password must contain four to eight alphanumeric characters.

Step 1. Click View & Modify Configuration button.

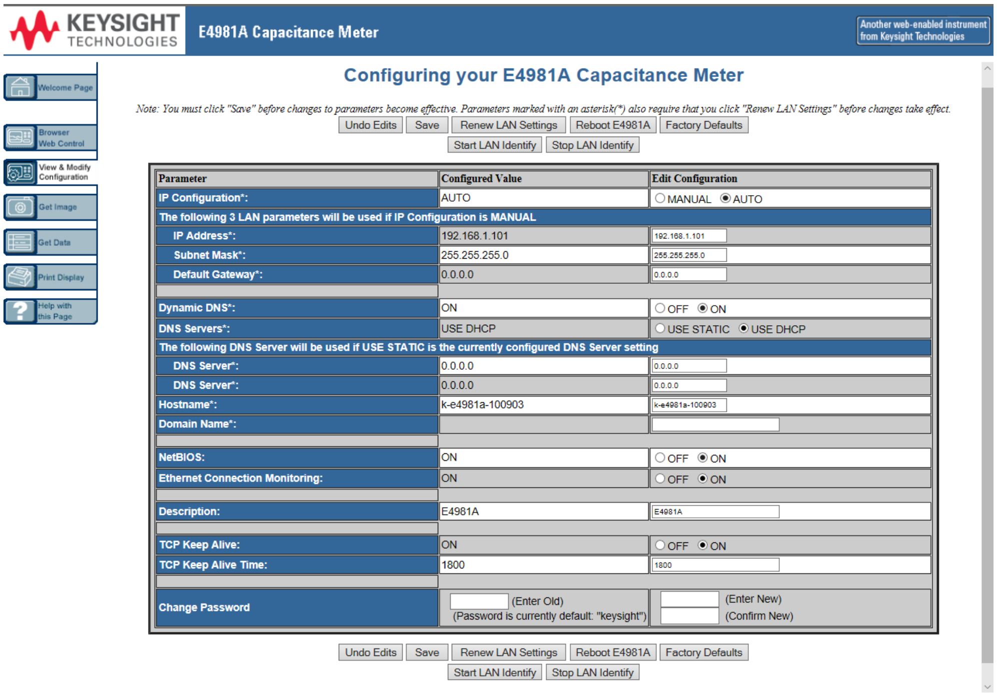
Step 2. Click Modify Configuration button.

Step 3. Enter the current password.

Step 4. Enter the current password and new password (twice) in the “Change Password” line.

Step 5. Press the Save button.

Figure 2-8 Web Server function



Configuring your E4981A Capacitance Meter

Note: You must click "Save" before changes to parameters become effective. Parameters marked with an asterisk(*) also require that you click "Renew LAN Settings" before changes take effect.

Parameter	Configured Value	Edit Configuration
IP Configuration*:	AUTO	○ MANUAL ● AUTO
The following 3 LAN parameters will be used if IP Configuration is MANUAL		
IP Address*:	192.168.1.101	192.168.1.101
Subnet Mask*:	255.255.255.0	255.255.255.0
Default Gateway*:	0.0.0.0	0.0.0.0
Dynamic DNS*:	ON	○ OFF ● ON
DNS Servers*:	USE DHCP	○ USE STATIC ● USE DHCP
The following DNS Server will be used if USE STATIC is the currently configured DNS Server setting		
DNS Server*:	0.0.0.0	0.0.0.0
DNS Server*:	0.0.0.0	0.0.0.0
Hostname*:	k-e4981a-100903	k-e4981a-100903
Domain Name*:		
NetBIOS:	ON	○ OFF ● ON
Ethernet Connection Monitoring:	ON	○ OFF ● ON
Description:	E4981A	E4981A
TCP Keep Alive:	ON	○ OFF ● ON
TCP Keep Alive Time:	1800	1800
Change Password	(Enter Old) (Password is currently default: "keysight")	(Enter New) (Confirm New)

NOTE

You don’t have to press either “Renew LAN Settings” or “Reboot E4981A.”

USB Remote Control System

The USB (Universal Serial Bus) remote control system provides device control via USB, which is equivalent to control via GPIB. Connection is made through an interface in compliance with USBTMC-USB488 and USB 2.0.

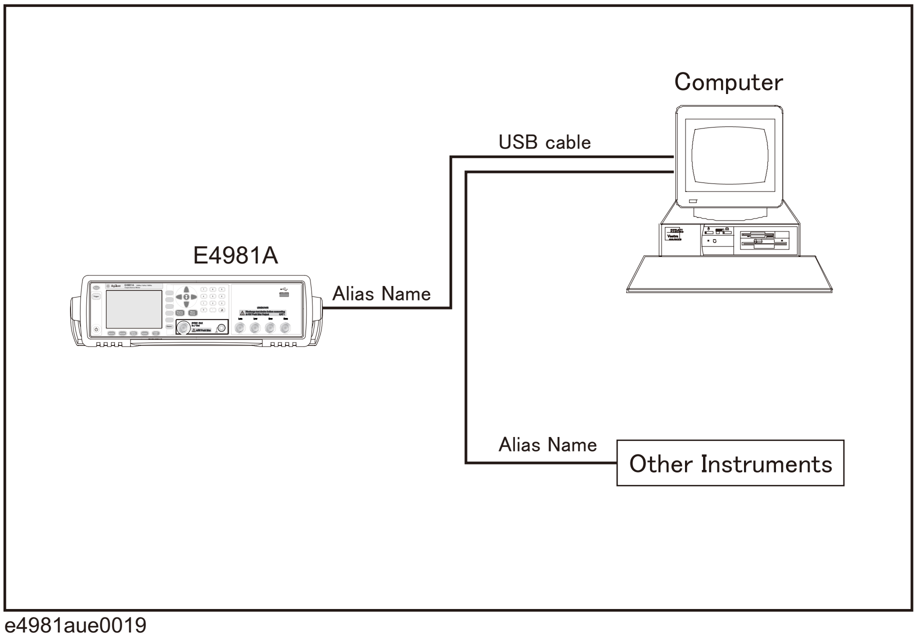
USBTMC (USB Test & Measurement Class) is a protocol whose design is based on USB for communication with a USB device, such as that via GPIB.

System configuration

The USB remote control system controls instruments that use the name “alias.” There is no such address for GPIB connections.

Use a USB cable to connect the E4981A to an external controller (personal computer). Figure 2-9 shows an overview of the system configuration for the USB remote control system

Figure 2-9 Configuration of the USB remote control system

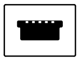


Required equipment

1. E4981A (models with USB interface port (type mini-B)).
2. External controller (personal computer with Keysight I/O Libraries Suite and USB host port (type A)) installed.
3. Other USB-compatible devices (instruments and/or peripherals for specific purposes).
4. USB cable for connecting the E4981A to the external controller (with type A/4-prong male or type mini-B/5-prong male connectors, depending on device used).

USB port types

There are two standard types of USB ports. The external controller (PC) must be connected via the USB host port (type A), while the E4981A and other USB-compatible devices must be connected via the USB interface port (type mini-B).

	Type A: USB host port
	Type mini-B: USB interface port

Preparing E4981A

You do not have to configure any softkey or command for the E4981A in order to control the E4981A from an external controller. Simply connect a USB cable to the USB interface port.

Preparing external controller

You must set up the I/O interface of the external controller in advance to establish communication with the E4981A via a USB. The USB can identify devices automatically, so once you connect a USB cable to a target device, a dialog box will appear for USB device registration.

NOTE

The E4981A is identified as a new device if its serial number is different.

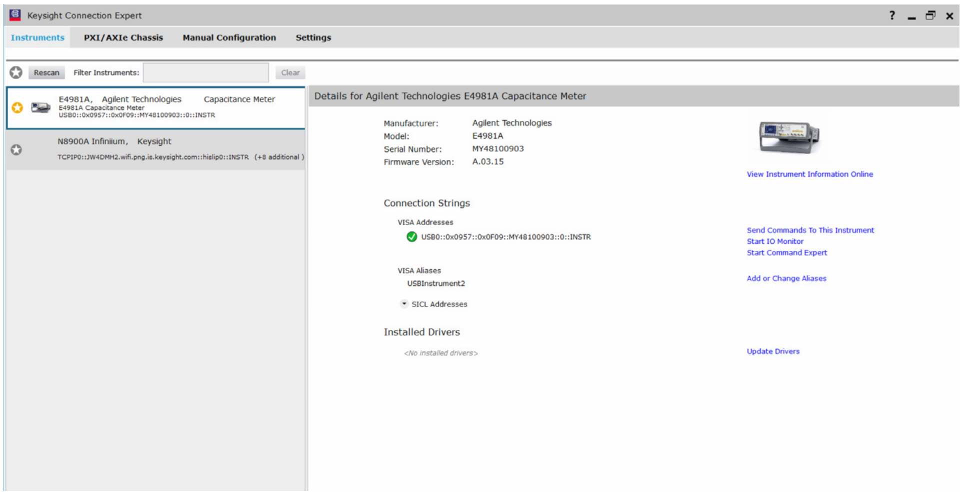
Changing alias on setting screen

The following are steps using Keysight I/O Libraries Suite 17.

Step 1. **From your PC’s Start menu, click Program - Keysight IO Libraries Suite -Keysight Connection Expert to open the setting screen.**

Step 2. In the Instruments screen the connected USB device is auto detected.

Figure 2-10 USB device is auto detected



E4981A, Agilent Technologies Capacitance Meter
E4981A Capacitance Meter
USB0::0x0957::0x0F09::MY48100903::0::INSTR

N8900A Infiniium, Keysight
TCPIP0::JW4DMH2.wifi.png.is.keysight.com::hislip0::INSTR (+8 additional)

Details for Agilent Technologies E4981A Capacitance Meter

Manufacturer:	Agilent Technologies
Model:	E4981A
Serial Number:	MY48100903
Firmware Version:	A.03.15

Connection Strings

VISA Addresses
✓ USB0::0x0957::0x0F09::MY48100903::0::INSTR

VISA Aliases
USBInstrument2

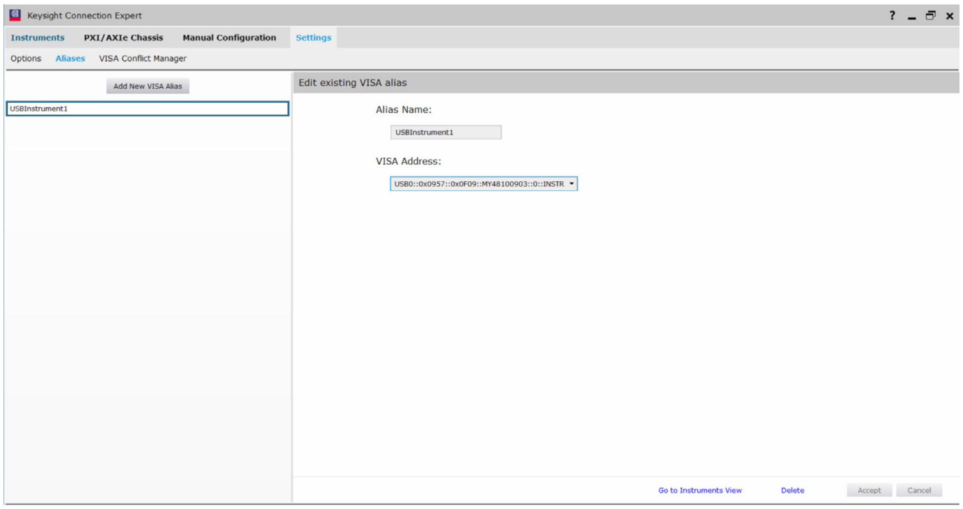
▾ SICL Addresses

Installed Drivers

<No installed drivers>

Step 3. Click **Add or Change Aliases** to change the VISA alias.

Figure 2-11 Edit existing VISA alias



USBInstrument1

Edit existing VISA alias

Alias Name:
USBInstrument1

VISA Address:
USB0::0x0957::0x0F09::MY48100903::0::INSTR

Control using C or Visual Basic

You can control the E4981A by programming using SICL/VISA with Visual C++ or Visual Basic in a Windows environment. For further information on controlling the E4981A, see the manual for SICL or VISA. For Keysight I/O Libraries Suite, use Keysight I/O Libraries Suite 15.

You may employ aliases in programming using SICL/VISA.

The following example shows an OPEN command to control the E4981A, for which the alias is E4981A_USBIF.

SICL	id = iopen ("E4981A_USBIF")
VISA	viOpen (...,"E4981A_USBIF",...)

NOTE

For further details on the programming using SICL/VISA, see the SICL Users Guide or the VISA Users Guide.

Control using Keysight VEE

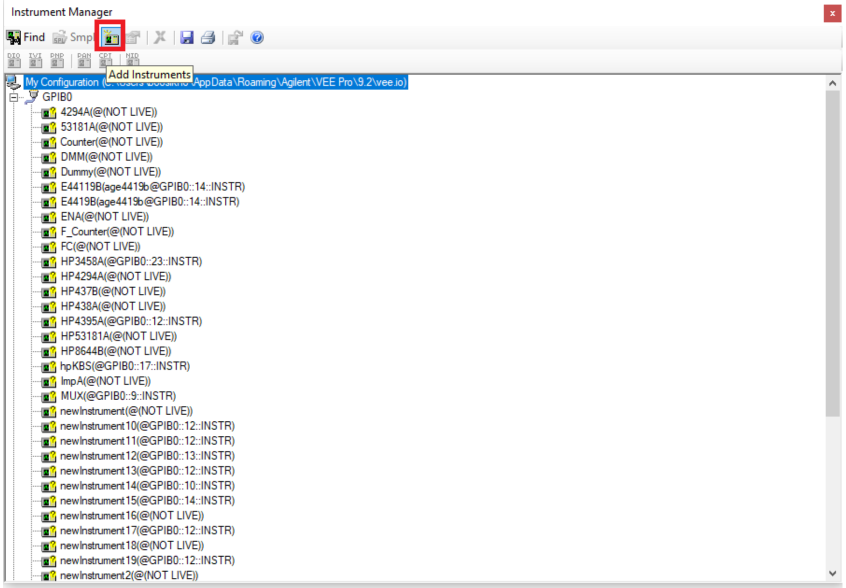
Keysight VEE allows you to control the E4981A via the direct I/O interface. The following example shows how to control the E4981A, for which alias is given as E4981A_USBIF.

NOTE

When using Keysight VEE for PCs, use Keysight VEE Pro 7 for Windows or a later version.

Step 1. On the Keysight VEE's **I/O** menu, click **Instrument Manager...**

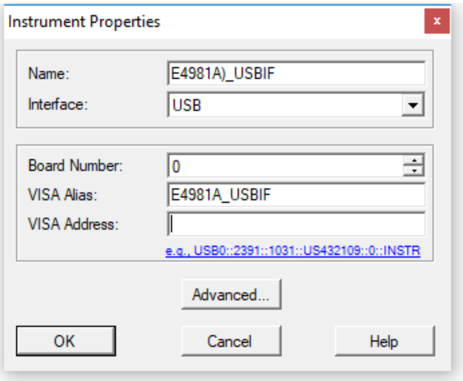
Step 2. In the Instrument Manager setting screen, click the **Add Instruments** icon and in the Add Interface/Device screen select **USB**.



My Configuration (C:\Users\[illegible]\AppData\Roaming\Agilent\VEE Pro\9.2\vee.io)
GPIB0
4294A(@(NOT LIVE))
53181A(@(NOT LIVE))
Counter(@(NOT LIVE))
DMM(@(NOT LIVE))
Dummy(@(NOT LIVE))
E44119B(age4419b@GPIB0::14::INSTR)
E4419B(age4419b@GPIB0::14::INSTR)
ENA(@(NOT LIVE))
F_Counter(@(NOT LIVE))
FC(@(NOT LIVE))
HP3458A(@GPIB0::23::INSTR)
HP4294A(@(NOT LIVE))
HP437B(@(NOT LIVE))
HP438A(@(NOT LIVE))
HP4395A(@GPIB0::12::INSTR)
HP53181A(@(NOT LIVE))
HP8644B(@(NOT LIVE))
hpKBS(@GPIB0::17::INSTR)
ImpA(@(NOT LIVE))
MUX(@GPIB0::9::INSTR)
newInstrument(@(NOT LIVE))
newInstrument10(@GPIB0::12::INSTR)
newInstrument11(@GPIB0::12::INSTR)
newInstrument12(@GPIB0::13::INSTR)
newInstrument13(@GPIB0::12::INSTR)
newInstrument14(@GPIB0::10::INSTR)
newInstrument15(@GPIB0::14::INSTR)
newInstrument16(@(NOT LIVE))
newInstrument17(@GPIB0::12::INSTR)
newInstrument18(@(NOT LIVE))
newInstrument19(@GPIB0::12::INSTR)
newInstrument2(@(NOT LIVE))

Step 3. The Instrument Properties dialog box appears.

Specify Name: **E4981A_USBIF** (any other name acceptable), Interface: **USB**, Board Number: **0** (USB port number), and Alias: **E4981A_USBIF** (which is registered in the IO Config setting screen), then click **OK**.

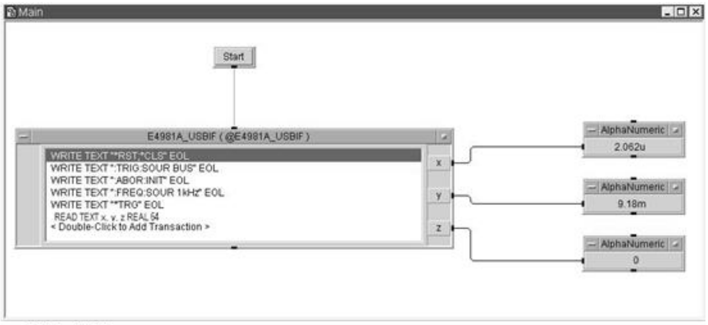


Instrument Properties

Name: E4981A)_USBIF
Interface: USB
Board Number: 0
VISA Alias: E4981A_USBIF
VISA Address:
e.g., USB0::2391::1031::US432109::0::INSTR

Figure 2-12 shows an example of control using the direct I/O interface that was set in the above procedures.

Figure 2-12 Sample control using Keysight VEE (USB)



Main

E4981A_USBIF (@E4981A_USBIF)

```
WRITE TEXT "*RST;*CLS" EOL
WRITE TEXT ":TRIG:SOUR BUS" EOL
WRITE TEXT ":ABOR:INIT" EOL
WRITE TEXT ":FREQ:SOUR 1kHz" EOL
WRITE TEXT "*TRG" EOL
 READ TEXT x, y, z REAL 64
< Double-Click to Add Transaction >
```

AlphaNumeric: 2.062u
AlphaNumeric: 9.18m
AlphaNumeric: 0

e4981a0011

Sending SCPI Command Messages

Types and structure of commands

The SCPI commands available for the E4981A are classified into two groups as follows.

E4981A commands

Commands specific to the E4981A. They cover all measurement functions that the E4981A has and some general-purpose functions. The commands in this group are arranged in a hierarchical structure called the command tree (see Table 10-3, “E4981A SCPI command tree,” on page 335). Each command consists of character strings (mnemonics) indicating each hierarchical level and colon (:) separators between hierarchical levels.

IEEE common commands

Commands to cover general-purpose functions defined in IEEE488.2 that are commonly available to instruments that support this standard. The commands in this group have an asterisk (*) at the beginning. There is no hierarchical structure for the commands in this group.

Concepts of the command tree

The commands at the top of the command tree are called “root commands” or simply “roots.” To access lower-level commands in the tree, you need to specify a specific path like a directory path in the DOS file system. After turning the power on or resetting your PC, the current path is set to the root. Special characters in messages change the path setting as described below.

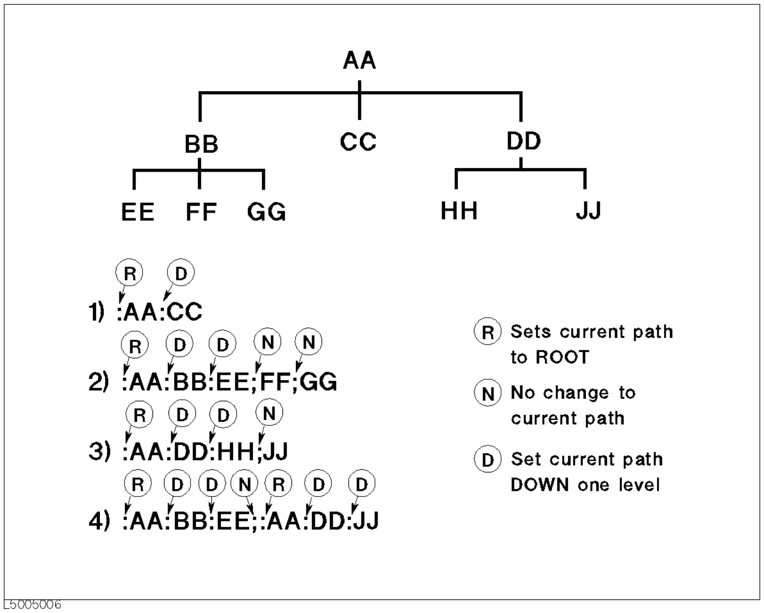
Message terminator A message terminator such as the <new line> character sets the current path to the root.

Colon (:) A colon between two command mnemonics lowers the level of the current path in the command tree. A colon used as the first character of a command specifies the command mnemonic that follows as the root-level command.

Semicolon (;) A semicolon does not change the current path but separates two commands in the same message.

Figure 2-13 shows an example of how to use colons and semicolons to efficiently access commands in the command tree.

Figure 2-13 Using colons and semicolons



Grammar of messages

This section describes the grammar to send program messages via GPIB. Program messages are messages that the user sends to the instrument from the external controller to control the instrument. A program message consists of one or more commands and their necessary parameters.

Upper/lower case sensitivity

Upper/lower case insensitive.

Program message terminator

A program message must be terminated with one of the three program message terminators: <new line>, <^END>, or <new line><^END>. <^END> indicates that EOI on the GPIB interface becomes active at the instant when the immediately previous data byte is sent. For example, the OUTPUT command of HTBasic automatically sends the message terminator after the last data byte.

Parameters

A space (ASCII code: 32) is required between a command and its first parameter. When sending several parameters in a single command, separate each parameter with a comma (,).

Message including several commands

When sending two or more commands in a single message, separate each command with a semicolon (;). The following example shows how to send the *CLS command and the :STAT:PRES command in a single message using HTBasic.

```
OUTPUT 717;"*CLS;:STAT:PRES"
```

Remote mode

The E4981A enters remote mode when controlled with commands from the controller and **RMT** is displayed in the status display area in the lower right of the screen.

Press **[Local/Lock]** to cancel the remote mode.

LXI

E4981A is complaint with LXI standard: version1.2 Class C.

About LXI

LXI (LAN eXtension for Instrumentation) is the LAN-based successor to GPIB and combines the advantages of Ethernet with the simplicity and familiarity of GPIB. The key features of LXI are as follows:

- The speed, simplicity, worldwide reach, low cost, ongoing enhancement and backward compatibility of LAN.
- Quick, easy configuration through the intuitive web interface built into compliant instruments.
- Simplified programming and greater software reuse through IVI drivers.
- The ability to create hybrid systems that include LXI, GPIB, VXI, PXI, CANbus, etc.
- Enhanced system performance and event handling via hardware- and LAN-based triggering modes.

NOTE

For more information on LXI, please visit www.lxistandard.org.

3 Setting Up Measurement Conditions and Display

This chapter describes how to set up the measurement conditions and display. It also describes how to save/recall the instrument settings including the measurement conditions.

Setting Up Measurement Conditions

Selecting measurement parameters

To select the measurement parameters, use the following commands.

- `:CALCulate1:FORMat` on page 188
- `:CALCulate2:FORMat` on page 191

You can select one of the following measurement parameter combinations shown in Table 3-1. If you select one parameter using the SCPI command and the resulting parameter combination is not among those in Table 3-1, the other parameter is automatically changed to a proper parameter. For example, when the primary parameter is Cp and you select Rs as the secondary parameter, the primary parameter is automatically changed to Cs.

Table 3-1 **Measurement parameter combinations**

Primary parameter	Secondary parameter
Cp	D, Q, G, Rp
Cs	D, Q, Rs

Each parameter is described below.

Cp: Capacitance value measured using the parallel equivalent circuit model

Cs: Capacitance value measured using the series equivalent circuit model

D: Dissipation factor

Q: Quality factor (inverse of D)

G:	Equivalent parallel conductance measured using the parallel equivalent circuit model
Rp:	Equivalent parallel resistance measured using the parallel equivalent circuit model
Rs:	Equivalent series resistance measured using the series equivalent circuit model

Setting up measurement signal (frequency and level)

Setting up frequency

To select the measurement signal frequency (120 Hz/1 kHz/1 MHz), use the following command.

- `:SOURce:FREQuency[:CW]` on page 267

Setting up level

To set the measurement signal level, use the following command.

- `:SOURce:VOLTage[:LEVel][:IMMediate][:AMPLitude]` on page 268

Selecting measurement range

Selecting measurement range mode

To select the measurement range mode (auto range/hold range), use the following command.

- `[:SENSe][:FIMPedance]:RANGe:AUTO` on page 265

Selecting measurement range

To select the measurement range, use the following command.

- `[:SENSe][:FIMPedance]:RANGe[:UPPer]` on page 266

The selectable measurement ranges differ depending on the measurement signal frequency. Therefore, if a newly selected measurement frequency conflicts with the current measurement range setting, the setting automatically changes to cover the allowable range.

NOTE **Setting up the measurement range automatically sets up the measurement range mode to the hold range mode.**

Selecting measurement time

To select the measurement time (1,2,4,6,8), use the following command.

- `[:SENSe][:FIMPedance]:APERture:TIME` on page 253

Selecting cable length

To set the cable length (0 m/1 m/2 m), use the following command.

- `:CALibration:CABLe[:LENGth]` on page 171

Setting up averaging function

Turning ON/OFF averaging function

To turn ON/OFF the averaging function, use the following command.

- `[:SENSe]:AVERage[:STATe]` on page 233

Setting up averaging count

To set the averaging count, use the following command.

- `[:SENSe]:AVERage:COUNt` on page 232

Setting up trigger delay time

To set the trigger delay time, use the following command.

- `:TRIGger:SEQ2:DELay` on page 306

Setting up source delay time

To set the source delay time, use the following command.

- `:TRIGger[:SEQ1]:DELay` on page 303

Setting trigger sync source

To set the trigger sync source, use the following command.

- `:SOURce:VOLTage:MODE` on page 269

Setting Signal Level Compensation (SLC)

To set the signal level compensation (SLC), use the following command.

- `:SOURce:VOLTage:ALC[:STATe]` on page 267

Sample program

See Example 9-1 to view an example of setting up measurement conditions and LCD display.

Setting Up Display

Turning ON/OFF display

To turn ON/OFF display of the measurement parameter obtained, the measurement signal level monitored, the handler output (comparator sorting result), the multi-correction settings, and the correction data, use the following command.

- `:DISPlay[:WINDow][:STATe]` on page 215

Setting up measurement result display

You can set up the following items related to the measurement result display.

- Fixed point display
  - ON/OFF
  - Value of highest digit

The following table shows the commands used to set up the above items.

Setup item		Command
Fixed point display	ON/OFF	`:DISPlay[:WINDow]:TEXT1[:DATA]:FMSD[:STATe]` on page 217
	Value of highest digit	`:DISPlay[:WINDow]:TEXT1[:DATA]:FMSD:DATA` on page 216

Selecting items displayed in instrument setup display area

To select a display page for the part that displays the instrument setup at the right of the display (instrument setup display area), use the following command.

- `:DISPlay:PAGE` on page 213

Displaying measurement result as deviation from reference value (deviation measurement mode)

You can use the deviation measurement mode to display a relative measurement result as deviation from the reference value instead of displaying its absolute value. To turn ON/OFF the deviation measurement mode, use the following commands.

- `:CALCulate1:MATH:STATe` on page 190
- `:CALCulate2:MATH:STATe` on page 193

You can select from two modes in the deviation measurement mode: displaying the deviation as it is or displaying the deviation as a percentage relative to the reference value. To set the deviation measurement mode, use the following commands.

- `:CALCulate1:MATH:EXPRession:NAME` on page 189
- `:CALCulate2:MATH:EXPRession:NAME` on page 192

To set the reference value in the deviation measurement mode, use the following command.

- `:DATA[:DATA]` on page 208

The table below shows the relationship between the setups using the above commands and the values displayed as the measurement result.

Setup of `:CALCulate1:MATH:STATe` or `:CALCulate2:MATH:STATe`	Setup of `:CALCulate1:MATH:EXPRession:NAME` or `:CALCulate2:MATH:EXPRession:NAME`	Value displayed as the measurement result
OFF	– – –	$Meas$
ON	DEV	$Meas - Ref$
	PCNT	$\frac{Meas - Ref}{Ref} \times 100$

Where, $Meas$ and Ref are:

$Meas$: Measured value

Ref: Reference value (set using the `:DATA[:DATA]` command)

NOTE

The measured value read out by the SCPI command is a calculation result based on the above setup. For judgment in the comparator function, the measurement result is used as is regardless of the setup. (Refer to the data processing flow in the *Operation Manual*.)

Setting Up Beep

Setting up conditions to make a beep sound

To turn ON/OFF the beep sound, use one of the following commands. You can use either of these two commands since both functions are identical.

- `:CALCulate1:COMParator:BEEPer[:STATe]` on page 174
- `:SYSTem:BEEPer:STATe` on page 273

To generate the beep sound regardless of ON/OFF setting, use `:SYSTem:BEEPer[:IMMediate]` on page 273. To change the tone of beep sound from five different types of available tones, use `:SYSTem:BEEPer:TONE` on page 274.

To set the conditions to make a beep sound according to the comparator sorting result, use the following command.

- `:CALCulate1:COMParator:BEEPer:CONDition` on page 173

The table below shows the relationship between the settings made by the above commands and the condition to make a beep.

Setting of :CALCulate1:COMParator:BEEPer[:STATe] or :SYSTem:BEEPer:STATe	Setting of :CALCulate1:COMParator:BEEPer:CONDition	Condition to make a beep	
OFF	–––	Never make a beep sound	
ON	FAIL	·When wrong key operation is performed. ·When an error, alarm, or other message is output.	·When the sorting judgment result of the comparator is OUT_OF_BIN, OVLD, LOWC_OR_NC or AUX_BIN.
	PASS		·When the sorting judgment result of the comparator is within BIN1 to BIN9.

Making a beep sound

To make a beep sound, use the following command:

- `:SYSTem:BEEPer[:IMMediate]` on page 273

Saving/Recalling Instrument Setup State (Save/Recall Function)

You can save/recall up to 20 instrument settings into/from the built-in nonvolatile memory (Flash memory, 0 through 9) and external USB memory (10 through 19).

NOTE

The ninth instrument setting corresponds to Auto Recall.

Auto recall is not executed when power is ON, by pressing the **[Preset]** key.

For information on the instrument setups you can save/recall, refer to *Operation Manual, Appendix C "Initial Setting"*.

Saving instrument setup state

To save the instrument setup state, use the following command.

- `*SAV` on page 161
- `:MMEMory:STORe:STATe[:REGister]` on page 229

Recalling saved instrument setup state

To recall the saved instrument setup state, use the following command.

- `*RCL` on page 160
- `:MMEMory:LOAD:STATe[:REGister]` on page 229

4 Preparation for Accurate Measurement (Executing Correction)

This chapter describes how to execute the correction function.

Executing OPEN/SHORT/LOAD Correction

Turning ON/OFF correction function

To turn ON/OFF each type of correction, use the corresponding command below.

Type of correction	Command
OPEN correction	`[:SENSe]:CORRection:OPEN[:STATe]` on page 248
SHORT correction	`[:SENSe]:CORRection:SHORt[:STATe]` on page 249
LOAD correction	`[:SENSe]:CORRection:LOAD[:STATe]` on page 242

Executing the measurement of correction data with the `[:SENSe]:CORRection:COLLect[:ACQuire]` command on page 237 automatically measures data and turns ON the correction.

The measured value and the standard value is available to display or input using the following parameters:

Type of correction	Parameter
OPEN correction	G-B, Cp-G
SHORT correction	R-X, Ls-Rs
LOAD correction	Cp-D, Cp-Q, Cp-G, Cp-Rp, Cs-D, Cs-Q, Cs-Rs

Measuring correction data

Measuring correction data

To measure each type of correction data, use the following command:

- `[:SENSe]:CORRection:COLLect[:ACQuire]` on page 237

For Open and SHORT correction, 120 Hz, 1 kHz and 1 MHz are measured and each data frequency is saved. Therefore, even if the measurement frequency is changed there is no need to reset the correction. The result is stored as the data for normal operation when the multi-correction function is OFF and as the data for multi-correction when it is ON (for the channel that has been selected at execution).

For LOAD correction, the correction measurement is performed only for the frequency specified at execution time.

Fixing the range at LOAD correction

When you want to fix the range at LOAD correction measurement, use the following command:

- `[:SENSe]:CORRection:COLLect:LOAD:RANGe:AUTO` on page 238

The load correction is performed at the range specified by `[:SENSe][:FIMPedance]:RANGe[:UPPer]` command on page 266, when the function is set to OFF.

Defining LOAD correction standard

Before measuring LOAD correction data, you must define the LOAD correction standard. To define the LOAD correction standard, use the following commands:

- `[:SENSe]:CORRection:CKIT:STAN3[:DATA]` on page 235
- `[:SENSe]:CORRection:CKIT:STAN3:FORMat` on page 236

Reading/writing correction data (saving/recalling correction conditions)

After saving the correction conditions to a file, you can recall these conditions on the E4981A at any time from the file.

The correction data is set up in the parameter-oriented format shown in the table below:

Type of correction	Parameter format
OPEN correction	G-B or Cp-G (select with the `[:SENSe]:CORRection:CKIT:STAN1:FORMat` command)
SHORT correction	R-X or Ls-Rs (select with the `[:SENSe]:CORRection:CKIT:STAN2:FORMat` command)
LOAD correction	Parameter format to define the LOAD correction standard (select with the `[:SENSe]:CORRection:CKIT:STAN3:FORMat` command)

To read/write the correction data, use the following command.

- `[:SENSe]:CORRection:DATA` on page 240

The correction data written using the above command is handled as follows:

- In case of STAN3: Saved as LOAD correction data for the measurement frequency specified when the command is executed.
- In case of STAN1/STAN2: Saved as OPEN/SHORT correction data for all frequencies.
- Saved as the data for normal operation when the multi-correction function is OFF and as the data for the multi-correction when it is ON (for the channel that has been selected at execution).

NOTE

When you write correction data, note the following:

- Before writing, recall the settings of the measurement frequency, cable length, and frequency shift (for 1 MHz) specified when the data was read.
- For the LOAD correction data, in addition to the above, you also need to recall the settings of the LOAD correction standard (definition value and parameter type) specified when the data was read.
- Turn ON the correction function. (Unlike when measuring the correction data, writing the correction data does not automatically turn this function ON.)

Sample program

See Example 9-2 to view an example of execution of OPEN/SHORT/LOAD correction.

Avoiding work-related mistakes in measuring correction data

To avoid simple mistakes related to work when measuring correction data (for example, setting up the OPEN state and SHORT state inversely), it is important to confirm that the measured data has a proper value.

If a measured value is not proper during measurement of the correction data, a warning message appears on the display. However, the occurrence of a warning message cannot be detected through GPIB/LAN/USB. Therefore, to detect erroneous correction data through GPIB/LAN/USB, you need to read out the correction data after each measurement and then confirm that the value is appropriate.

Sample program

See Example 9-3 to view an example of recovering the correction data.

Executing Offset Correction

Turning ON/OFF the correction function

When you turn ON the offset correction, assuming that the measured value before correction is $Meas$ and the offset correction data is $Offset$, the measured value is compensated as $Meas - Offset$.

To turn ON/OFF the offset correction, use the following command.

- `[:SENSe]:CORRection:OFFSet[:STATe]` on page 247

You cannot turn ON/OFF the primary parameter and secondary parameter separately. However, if you set the correction value to 0, the state is actually the same as OFF even if the offset correction is ON. Therefore, in practice you can realize separate ON/OFF states by setting the correction value for either parameter to 0.

Setting up the correction data

To set up the offset correction data, use the following command.

- `[:SENSe]:CORRection:OFFSet:DATA` on page 246

The entered value is set as the offset correction data for the measurement frequency at the time of the entry.

Using the Multi-correction Function

Turning ON/OFF the multi-correction function

To turn ON/OFF the multi-correction function, use the following command.

- `[:SENSe]:CORRection:MULTiple[:STATe]` on page 245

Selecting a channel

To select a channel for the multi correction function, use the following command.

- `[:SENSe]:CORRection:MULTiple:CHANnel` on page 243

Measuring correction data

Selecting the definition method of the LOAD correction standard

To select whether to define the LOAD correction standard value (LOAD correction reference value) for each channel individually or for all channels commonly, use the following command.

- `[:SENSe]:CORRection:MULTiple:CKIT:STAN3[:STATe]` on page 244

Measuring correction data

The method to measure the OPEN/SHORT/LOAD correction data for multi-correction is the same as that for basic correction data except that you need to select the proper channel before measurement. For more details, refer to “Measuring correction data” on page 50.

Reading/writing correction data

The method used to read/write the OPEN/SHORT/LOAD correction data for multi-correction is the same as that for basic correction data except that you need to select the proper channel before measurement. For more details, refer to “Reading/writing correction data (saving/recalling correction conditions)” on page 51.

Sample program

See Example 9-4 to view an example of multi-correction.

Using the Cable Correction Function

Turning ON/OFF the cable correction function

To turn ON/OFF the cable correction function, use the following command:

- `:CALibration:CABLe:CORRection:STATe?` on page 170

Executing LOAD measurement in cable correction

To execute LOAD measurement in cable correction, use the following command:

- `:CALibration:CABLe:CORRection:COLLect[:ACQuire]:LOAD` on page 166

Executing OPEN measurement in cable correction

To execute OPEN measurement in cable correction, use the following command:

- `:CALibration:CABLe:CORRection:COLLect[:ACQuire]:OPEN` on page 167

Executing REFerence measurement in cable correction

To execute REFerence measurement in cable correction, use the following command:

- `:CALibration:CABLe:CORRection:COLLect[:ACQuire]:REFerence` on page 168

Saving cable correction data

To save the cable correction data, use the following command:

- `:CALibration:CABLe:CORRection:SAVE` on page 169

Clearing cable correction data

To clear the cable correction data, use the following command:

- `:CALibration:CABLe:CORRection:CLEar` on page 165

5 Starting (Triggering) Measurement and Waiting for Completion of Measurement

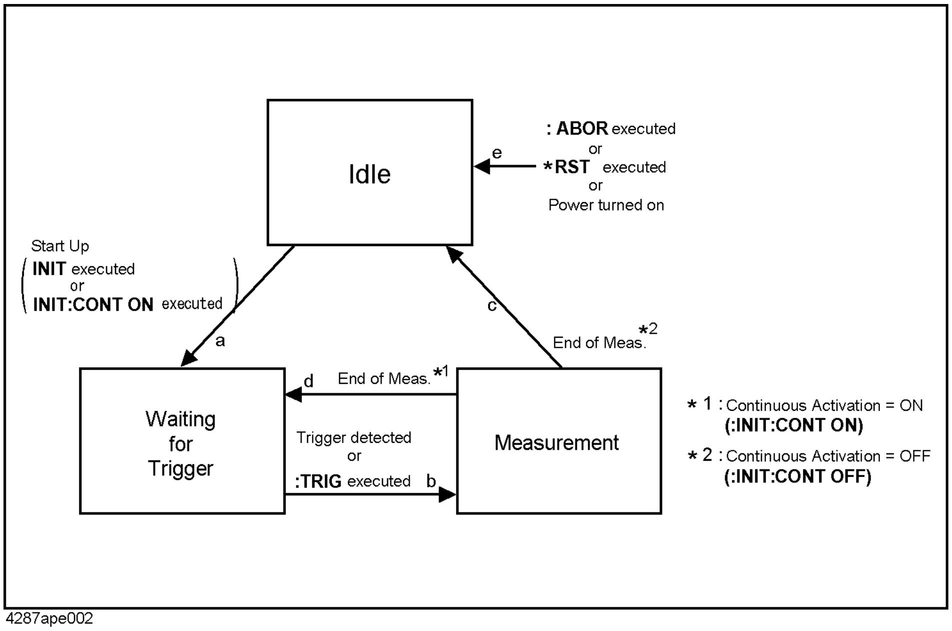
This chapter describes how to trigger the instrument to start measurement and how to detect completion of measurement.

Starting (Triggering) Measurement

Trigger system

The trigger system is responsible for tasks such as detecting the start of a measurement cycle (triggering) and controlling the system state. As Figure 5-1 shows, the trigger system has a system-wide state that can be “Idle,” “Waiting for Trigger,” or “Measurement.”

Figure 5-1 Trigger system



The following subsections describe each state and explain how the trigger system switches among the states.

System-wide states and transitions

Idle state

When the following commands are executed, the trigger system transitions to the idle state (e in Figure 5-1). The idle state is also in effect immediately after power-on. Because the continuous activation of the trigger system and the trigger mode are set to ON and internal trigger, respectively, at power-on, the trigger system immediately transitions to the waiting for trigger state and then repeats the transition between the measurement state and the waiting for trigger state.

- `*RST` on page 160
- `:ABORt` on page 164
- `*CLS` on page 157
- `:INITiate:CONTinuous` on page 227 (when executed with OFF specified)

When the trigger system is started up using the following commands, it transitions to the waiting for trigger state (a in Figure 5-1).

- `:INITiate[:IMMediate]` on page 228
- `:INITiate:CONTinuous` on page 227 (when executed with ON specified)

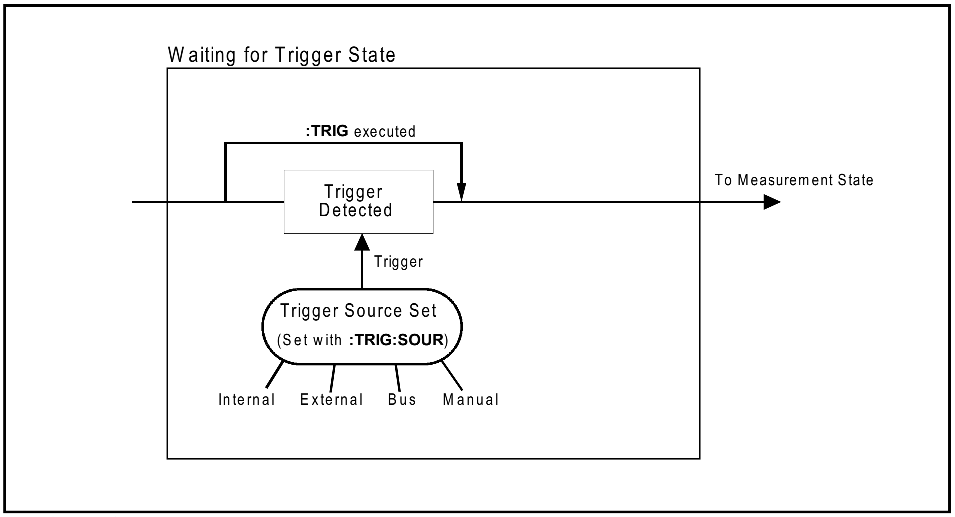
Waiting for trigger state (trigger event detect state)

The waiting for trigger state, when the instrument is triggered (a trigger is detected) or the `:TRIGger[:SEQ1][:IMMediate]` command on page 304 is executed, transit to the measurement state.

As shown in the table below, the instrument triggering method differs depending on which trigger mode is set. To set up the trigger mode, use the `:TRIGger[:SEQ1]:SOURce` command on page 305.

Trigger mode	Instrument triggering method
Internal trigger (Int)	The instrument is automatically triggered within itself.
External trigger (Ext)	The instrument is triggered when a trigger signal is input through the Ext Trig terminal or the handler/scanner interface.
BUS trigger (Bus)	The instrument is triggered when the `*TRG` command on page 163 is executed.
Manual trigger (Man)	The instrument is triggered when the [Trigger] key on the front panel is pressed.

Figure 5-2 Transition flow from waiting for trigger state to measurement state



e4981ape0051

Measurement state (sequence operation state)

Measurement is performed in the measurement state. After transiting from "Waiting for trigger state (trigger event detect state)" to the measurement state, then wait for trigger delay and source delay.When the measurement is finished, the state transitions to a different state depending on the setup of the continuous activation of the trigger system (set with `:INITiate:CONTinuous` on page 227) as shown below.

When the continuous activation is OFF:

Transition to the idle state (c in Figure 5-1)

When the continuous activation is ON:

Transition to the waiting for trigger state (d in Figure 5-1)

Starting (triggering) measurement

Follow the procedure below to perform successive measurements automatically (initial setup).

Step 1. Set the trigger mode to the internal trigger with the `:TRIGger[:SEQ1]:SOURce` command on page 305.

Step 2. If the trigger system has not started up (in the idle state), use the `:INITiate:CONTinuous` command on page 227 to turn ON the continuous activation of the trigger system.

Two methods to perform a measurement at your desired time:

Triggering the instrument at your desired time

Step 1. Use the :TRIGger[:SEQ1]:SOURce command to set the trigger mode to the Bus mode.

Step 2. If the trigger system has not started up (in the idle state), use the :INITiate:CONTinuous command to turn ON the continuous activation of the trigger system.

Step 3. Trigger the instrument at your desired time. An external controller can trigger the instrument with one of the following two commands:

Command	Query response	Applicable trigger mode
`*TRG` on page 163	Yes (The measured result is read out.)	BUS
`:TRIGger[:SEQ1][:IMMediate]` on page 304	No	MANual|BUS

Step 4. To repeat measurement, repeat Step 3.

Starting up the trigger system at your desired time

Step 1. If the trigger system has started up (in a state other than the idle state), use the :INITiate:CONTinuous command to turn OFF the continuous activation of the trigger system and then use the `:ABORt` command on page 164 to stop the trigger system.

Step 2. Set the trigger mode to the internal trigger with the :TRIGger[:SEQ1]:SOURce command.

Step 3. Start up the trigger system with the `:INITiate[:IMMediate]` command on page 228 at your desired time. The instrument will be automatically triggered by the internal trigger and measurement will be performed once.

Step 4. To repeat measurement, repeat Step 3.

Waiting For Completion Of Measurement (Detecting Completion Of Measurement)

You can detect the status of the E4981A by using the status register, as described in this section. For information on the entire status report system (for example, information on each bit of the status register), refer to Appendix C "Status Reporting System".

The measurement state is indicated by the operation status register. An SRQ (service request) is useful for detecting the completion of measurement in your program by using the information indicated by this register.

To detect the completion of measurement using SRQ, use the following commands.

- `*SRE` on page 162
- `:STATus:OPERation:ENABle` on page 270

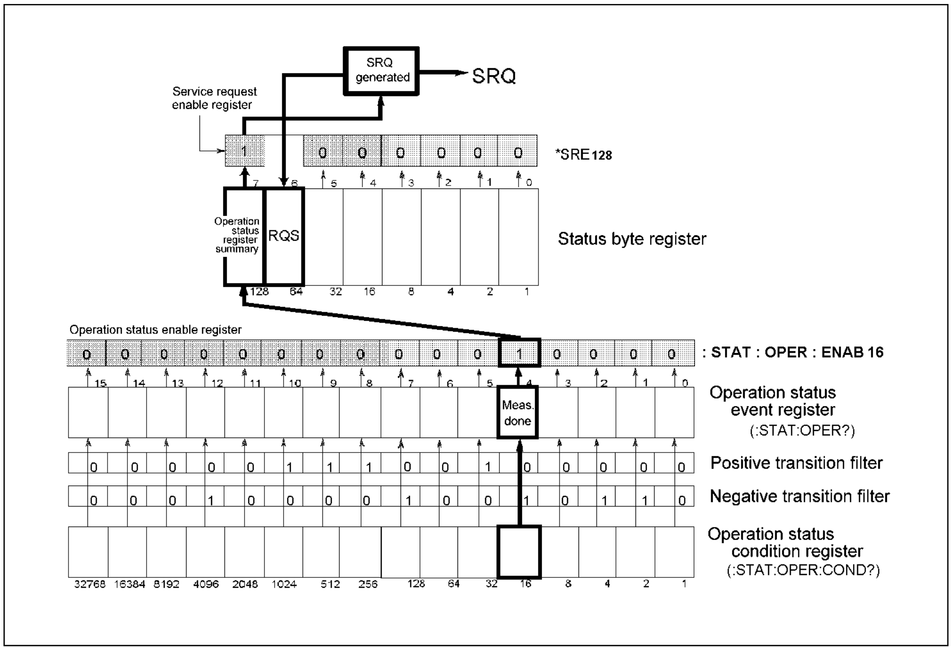
The procedure is given below.

Step 1. Make the setup so that the E4981A generates an SRQ if bit 4 of the operation status event register is set to 1.

Step 2. Trigger the instrument to start a measurement.

Step 3. Perform interrupt handling in the program when the SRQ occurs.

Figure 5-3 SRQ generation sequence (when measurement finishes)



4288ape006

Starting (Triggering) Measurement and Waiting for Completion of Measurement
Waiting For Completion Of Measurement (Detecting Completion Of Measurement)

Sample program

See Example 9-5 to view an example of detection of completion of measurement using SRQ.

6 Reading Out Measured Result

This chapter describes how to read out the measured result and the measurement signal level monitor result.

Data Transfer Format

You can select either the ASCII transfer format or binary transfer format when transferring data with the following commands:

The ASCII transfer format is always used when transferring data with commands other than those listed below:

- `:FETCh?` on page 220
- `:READ?` on page 230
- `*TRG` on page 163
- `:DATA[:DATA]` on page 208
- `[:SENSe]:CORRection:DATA` on page 240
- `[:SENSe][:FIMPedance]:CONTact1:VERify:BUF1:DATA?` on page 253
- `[:SENSe][:FIMPedance]:CONTact1:VERify:BUF2:DATA?` on page 256

To set up the data transfer format, use the following command:

- `:FORMat[:DATA]` on page 225

ASCII transfer format

When data is transferred in the ASCII transfer format, values are transferred as ASCII bytes in one of the following formats. According to the specification of IEEE488.2, values are separated by a comma (,).

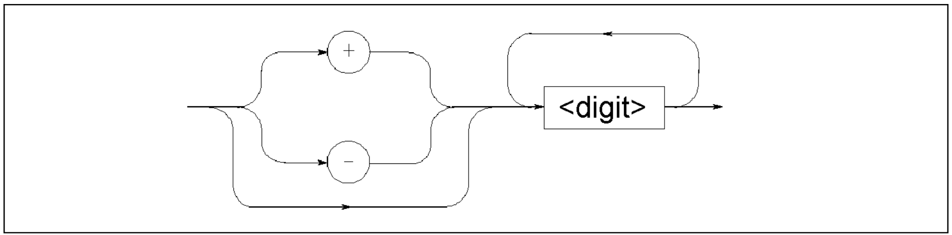
NOTE

The string length of numerical values is not constant. Therefore, the comma does not occur at constant intervals when extracting numerical values from the read out numerical data string.

- Integer format

  The format shown in Figure 6-4. Numerical values are expressed as integers. For example, the value, 11, is expressed as “+11.”

Figure 6-4 Integer format

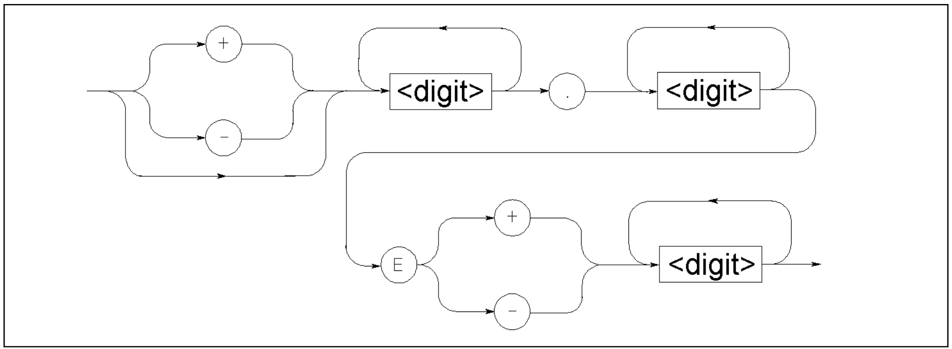


lb005013e

- Floating point format

  The format shown in Figure 6-5. Numerical values are expressed using a floating point. For example, the value, 1000, is expressed as “+1.00000E+03.”

Figure 6-5 Floating point format



lb005015e

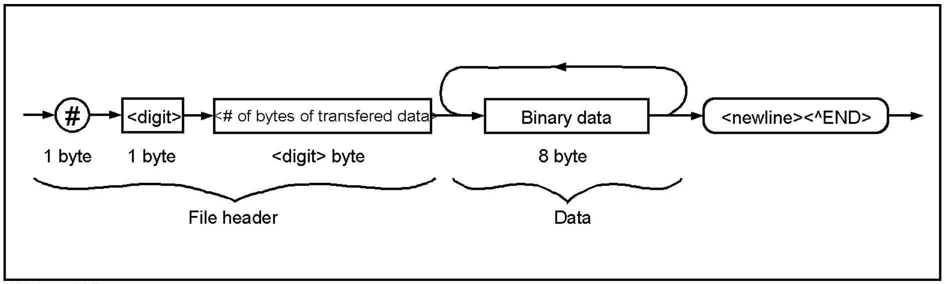
NOTE

The long format is available in floating point format. To set the long floating point format, use the following command: `:FORMat:ASCii:LONG` **on page 222.**

Binary transfer format

When data is transferred in the binary transfer format, values (binary data) are transferred in the format shown in Figure 6-6.

Figure 6-6 Binary transfer format

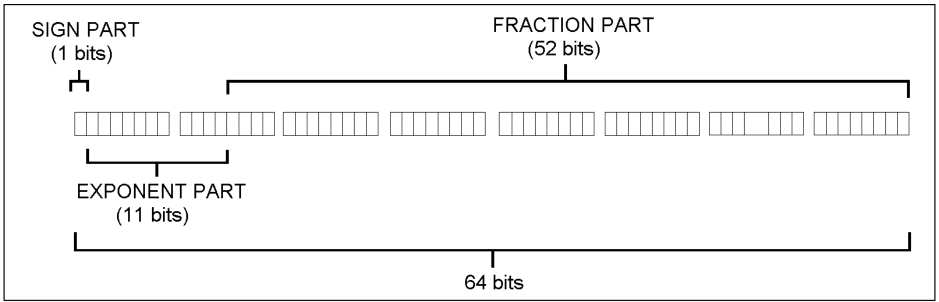


4288ape007

In this data transfer format, a pounds symbol (#) occurs at the beginning. The 2nd byte <digit count> indicates the number of bytes in the <transfer data byte count> part. The <transfer data byte count> indicates the total number of bytes of the binary data. <new line><^END> is the message terminator.

Binary data is in the IEEE 754-1985 standard floating point format consisting of 64 bits as shown in Figure 6-7.

Figure 6-7 64-bit floating-point data



4287ape025

Byte order

In binary transfer, data bytes (8 bytes) are transferred from the byte including the MSB (Most Significant Bit) (the left-most byte in Figure 6-7) through the byte including the LSB (Least Significant Bit) (the right-most byte in Figure 6-7).

To change the byte order, use the following command:

- `:FORMat:BORDer` on page 223

Reading Out Measured Result

Reading Out Measured Result

This section describes how to read out the measured result.

You can read out the measured result in two ways: reading out data for each measurement or reading out data of several measurements in batch.

The three commands shown in the table below can be used to read out the measured data for each measurement.

	Available trigger mode	Readout procedure
Using the `*TRG` command	Bus trigger	Executing `*TRG` ↓ Readout
Using the `:FETCh?` command	All	Triggering the instrument ↓ Executing `:FETCh?` ↓ Readout
Using the `:READ?` command	External trigger (Ext) Internal trigger (Int)	Executing `:READ?` ↓ Triggering the instrument ↓ Readout

When you want to read out the data of several measurements in batch, use the data buffer.

Reading out measured result using `*TRG` command

This command actually performs two tasks: it triggers the instrument and returns the results. It is useful, for example, when you want to retrieve measurement results immediately after triggering the instrument from an external controller.

The readout procedure using the `*TRG` command is described below.

Step 1. Use the `:TRIGger[:SEQ1]:SOURce` command on page 305 to set up the trigger mode to the BUS trigger (BUS).

Step 2. Execute the `*TRG` command.

Step 3. Read out the measured result. To repeat the measurement, repeat Steps 2 and 3.

Sample Program

See Example 9-6 and Example 9-7 to view an example of reading out measured result in ASCII and Binary format respectively using *TRG command.

Reading out measured result using `:FETCh?` command

You can use this readout method when you want to trigger the instrument from any source other than an external controller or when you want to perform a process that is between triggering the instrument and reading out the measured result.

The readout procedure using the `:FETCh?` command is described below.

Step 1. Set up the trigger mode as necessary.

Step 2. Trigger the instrument by using the method for the trigger mode.

NOTE To trigger the instrument from an external controller in this procedure, use the `:TRIGger[:SEQ1][:IMMediate]` command on page 304.

Step 3. Execute the `:FETCh?` command at the completion of the measurement.

Step 4. Read out the measured result. To repeat the measurement, repeat Steps 2 to 4.

Sample program

See Example 9-8 and Example 9-9 to view an example of reading out measured result in ASCII and Binary format respectively using :FETCh? command.

Reading out measured result using `:READ?` command

You can use this readout method when you want to read out the result synchronously with the completion of measurement without detecting the time of trigger in the trigger wait state in your program. Therefore, this is useful, for example, when you want to trigger the instrument from external equipment such as a handler and read out the result using an external controller immediately after the completion of measurement.

The readout procedure using the `:READ?` command is described below.

Step 1. Use the `:TRIGger[:SEQ1]:SOURce` command on page 305 to set the trigger mode to internal trigger (Internal) or external trigger (External).

Step 2. Execute the `:READ?` command.

Step 3. Trigger the instrument using the method for the trigger mode setup.

Step 4. Read out the measured result. To repeat the measurement, repeat Steps 2 to 4.

Sample program

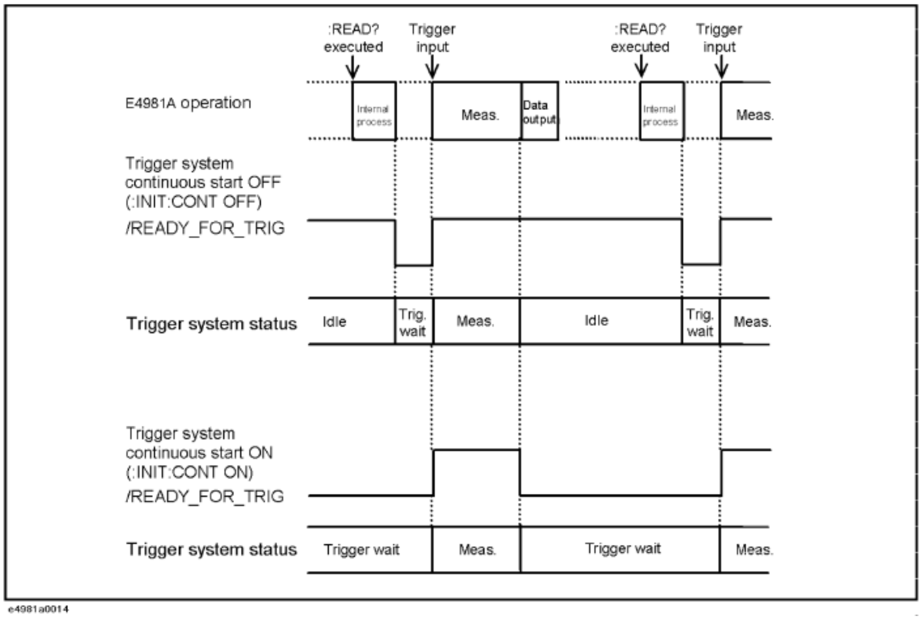
See Example 9-10 and Example 9-11 to view an example of reading out measured result in ASCII and Binary format respectively using :READ? command.

Trigger input timing when using `:READ?` command

You must input a trigger after execution of the `:READ?` command. Therefore, if you use different controllers for outputting the trigger and executing the `:READ?` command, you need to find out whether the `:READ?` command has been executed from the controller for outputting the trigger in order to prepare the proper timing for trigger output (to trigger the instrument after execution of the `:READ?` command).

You can obtain this information from the /READY_FOR_TRIG signal of the handler interface as shown in Figure 6-8.

Figure 6-8 Behavior of /READY_FOR_TRIG signal at execution of `:READ?` command



If the `:READ?` command is executed when the trigger system is in the idle state, the /READY_FOR_TRIG signal of the handler interface changes from HIGH to LOW after completion of the internal processing following the reception of the command as shown in Figure 6-8.

If the trigger system continuous startup is ON, the /READY_FOR_TRIG signal changes from HIGH to LOW (the trigger system transitions to the trigger wait state instead of the idle state) after the completion of measurement and display update as shown in Figure 6-8. As a result, it is not possible to determine whether the `:READ?` command has actually been executed. Therefore, you need to turn OFF the trigger system continuous startup (execute the `:INITiate:CONTinuous` command on page 227 specifying OFF) in advance.

Reading out results of several measurements in batch (using data buffer)

You can use the data buffer to temporarily store the results of several measurements and then later read out these results in batch.

Data buffer types

There are three types of data buffers (buffers 1 - 3). Buffers 1 and 2 have the same function. Buffer 3 has a different function as shown in the below table.

		Buffers 1 and 2	Buffer 3
Maximum number of measurements that can be fed		200	1000
Data to be fed for each measurement	When the comparator function is OFF	3 data items: measurement status, measured value of the primary parameter or secondary parameter[1], and comparator sorting result[2] (Independent of ON/OFF of the comparator function)	3 data items: measurement status, measured primary parameter value, and measured secondary parameter value
	When the comparator function is ON		4 data items: measurement status, measured primary parameter value, measured secondary parameter value, and comparator sorting result

1. Use the **`:DATA:FEED:BUF1`** command on page 195 to select the parameter to store into the buffer from the primary or secondary parameters.
2. When the comparator function is OFF, 11 is read out.

Feed position

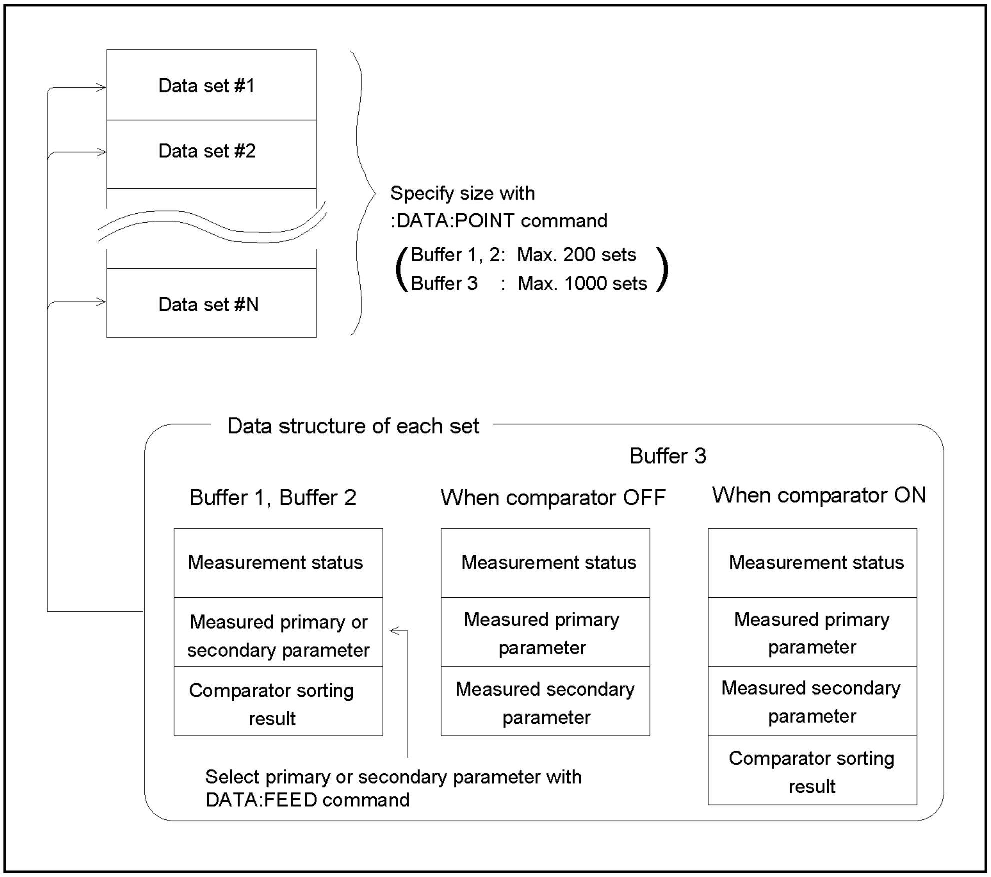
Data fed into the buffer is stored sequentially in the order of measurement. When the data is read out, the feed position returns to the beginning of the buffer and storage newly starts from the position of the first data set (Figure 6-9).

The feed position also returns to the beginning of the buffer in the following case:

- When you set up the number of measurements to be fed into the data buffer.

Reading Out Measured Result

Figure 6-9 Structure of data buffer


Data set #1
Data set #2
Data set #N
Specify size with
:DATA:POINT command
(Buffer 1, 2: Max. 200 sets
Buffer 3 : Max. 1000 sets)
Data structure of each set
Buffer 3
Buffer 1, Buffer 2
When comparator OFF
When comparator ON
Measurement status
Measured primary or secondary parameter
Comparator sorting result
Measurement status
Measured primary parameter
Measured secondary parameter
Measurement status
Measured primary parameter
Measured secondary parameter
Comparator sorting result
Select primary or secondary parameter with
DATA:FEED command

4288ape014

Procedure for use

Step 1. Use the `:DATA:POINts:BUF1` command on page 202 to set up the number of measurements you want to feed into the data buffer.

Step 2. When you use buffer 1 or buffer 2, use the `:DATA:FEED:BUF1` command on page 195 to select whether the parameter is fed from the primary or secondary parameter.

Step 3. Use the `:DATA:FEED:CONTrol[:STATe]` command on page 200 to make the setup feed the measured result into the data buffer.

Step 4. Execute the measurement for the number of times specified in Step 1.

Step 5. Use the `:DATA[:DATA]` command on page 208 to read out the data temporarily stored in the data buffer.

Step 6. Repeat Steps 4 to 5.

Sample program

See Example 9-12 to view an example of reading out measured result using data buffer.

Reading Out Measurement Signal Level Monitor Result

To read out measurement signal level monitor result, use the following command:

- `:DATA[:DATA]` on page 208

Sample program

See Example 9-13 to view an example of reading out the measurement signal level monitor result.

Reading Out Measured Result
Reading Out Measurement Signal Level Monitor Result

7 Sorting Based on Measured Result (Comparator Function)

This chapter describes how to use the comparator function to perform sorting based on the measured result.

Setting Up Comparator Function

Turning ON/OFF comparator function

To enable/disable the comparator function, use the following command:

- `:CALCulate1:COMParator[:STATe]` on page 187

Setting up limit range

Clearing (resetting) limit range

You can clear the ON/OFF state and the lower and upper limit values of all limit ranges (BIN1 to BIN9 and the secondary parameter limit range) and the ON/OFF state of AUX BIN to recover the factory-shipped default values (refer to *Operation Manual, Appendix C “Initial Setting”*).

To clear the limit ranges, use the following command.

- `:CALCulate1:COMParator:CLEar` on page 175

Selecting limit range designation method

You can select the designation method of the primary parameter limit ranges (BIN1 to BIN9) from the following three modes:

Mode	Description	
Absolute mode	Designation using the absolute value	
Absolute tolerance mode	Designation using the relative value (deviation from the reference value)	Designation using the absolute value[1]
Percent tolerance mode		Designation using the percentage of the reference value[2]

[1] Boundary value - reference value

[2] ((Boundary value - reference value)/reference value)×100

NOTE

Only the absolute mode can be used as the designation method of the secondary parameter limit range.

To select the designation method, use the following command.

- `:CALCulate1:COMParator:MODE` on page 181

To set up the reference value for the absolute tolerance mode or percent tolerance mode, use the following command.

- `:CALCulate1:COMParator:PRIMary:NOMinal` on page 184

Turning ON/OFF limit range

To turn ON/OFF the primary parameter limit range (BIN1 to BIN9), use the following command.

- `:CALCulate1:COMParator:PRIMary:BIN{1-9}:STATe` on page 183

Only BINs set to ON with the above command become the targets of sorting judgment. In other words, even if the measured result falls within the limit range of a BIN set to OFF, it will not be sorted to that BIN.

To turn ON/OFF the secondary parameter limit range, use the following command.

- `:CALCulate1:COMParator:SECondary:STATe` on page 186

If you specify OFF with the above command, no sorting judgment is made for the measured result of the secondary parameter. The comparator sorting judgment result is determined only on the basis of the measured primary parameter result.

Setting the range’s lower and upper limit values

To set the lower limit value and upper limit value of the primary parameter limit range (BIN1 to BIN9), use the following command.

- `:CALCulate1:COMParator:PRIMary:BIN{1-9}[:LIMIT]` on page 182

To set the lower limit value and upper limit value of the secondary parameter limit range, use the following command.

- `:CALCulate1:COMParator:SECondary:LIMit` on page 185

NOTE

If you set the upper limit value to a value equal to or less than the lower limit value, no limit range is used. Doing this would be equivalent to setting the limit range to OFF.

As shown in Figure 7-1, “Sorting judgment flow,” on page 79, the sorting judgment is performed in increasing order of BIN number. Therefore, when the limit ranges of two BINs overlap, the result is sorted to the BIN with the smaller number.

For the tolerance mode, the reference value does not need to be within the limit range (between the lower limit value and upper limit value).

Gaps are allowed between the limit ranges.

WARNING

When the low limit value is greater than the upper limit, a warning message “improper high/low limits” is displayed.

Turning ON/OFF AUX BIN function

To turn ON/OFF the AUX BIN function, use the following command.

- `:CALCulate1:COMParator:AUXBin` on page 172

Depending on the ON/OFF state of the AUX BIN function, the sorting result when the value of the secondary parameter exceeds the limit range varies as shown in Table 7-1.

Table 7-1 **Sorting result when measured secondary parameter value exceeds limit range**

Primary parameter sorting result	AUX BIN function	Sorting result
One of BIN1 to BIN9	OFF	OUT_OF_BINS
	ON	AUX_BIN
Not sorted to any BIN	Not applicable	OUT_OF_BINS

Setting up condition to make a beep sound

You can select the condition that produces a beep sound based on the comparator sorting judgment result from the following:

- Beep when the sorting judgment result is OUT_OF_BINS or AUX_BIN
- Beep when the sorting judgment result is BIN1 to BIN9

To set the condition that makes a beep sound, use the following command.

- `:CALCulate1:COMParator:BEEPer:CONDition` on page 173

You can also disable the beep sound. To turn ON/OFF the beep, use either of the following commands (they have the same function).

- `:CALCulate1:COMParator:BEEPer[:STATe]` on page 174
- `:SYSTem:BEEPer:STATe` on page 273

Rejecting Excessively Low Measured Results (Low C Reject Function)

Turning ON/OFF Low C reject function

To turn ON/OFF the Low C reject function, use the following command.

- `[:SENSe][:FIMPedance]:CREJect[:STATe]` on page 264

You can detect an excessively low (equal to or less than the preset boundary value) measured result of the primary parameter value (Cp or Cs) as Low C (abnormal measurement status) by turning ON the Low C reject function.

NOTE

Handler output shares the line for No Contact & Low C Reject and becomes active when either one is detected.

When primary parameter is lower than the border value.

NOTE

When the comparator function is ON, the sorting judgment is performed normally even if Low C is detected. However, the sorting judgment result displayed on the screen is LOWC and, on the handler interface, the /LOWC_OR_NC signal becomes active (LOW) in addition to the sorting judgment signal.

Setting limit (range boundary values) of Low C reject function

To set the limit (boundary values of the range in which Low C is detected) of the Low C reject function, use the following command.

- `[:SENSe][:FIMPedance]:CREJect:LIMit` on page 263

Reading Out Sorting Judgment Result

You can read out the sorting judgment result (readout value) along with the measured result by the comparator function according to the correspondences in Figure 7-1. To do read out sorting judgment results, use the following commands:

- `*TRG` on page 163
- `:FETCh?` on page 220
- `:READ?` on page 230

The comparator sorting result is read out as an integer between 0 and 11, as shown in the table below.

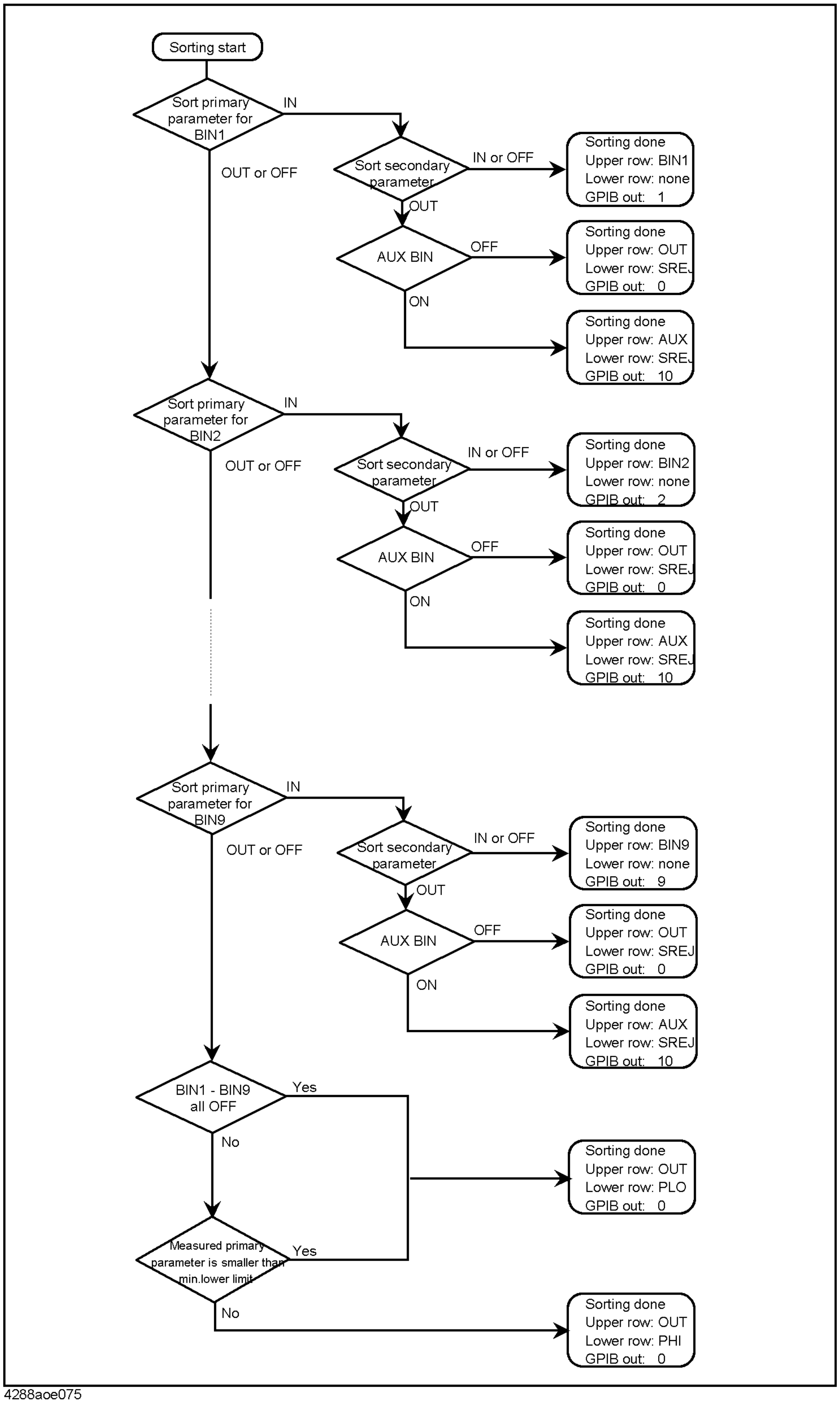
Table 7-2 **Relationship between comparator sorting result and readout value**

Readout value	Comparator sorting result
0	OUT_OF_BINS
1	BIN1
2	BIN2
3	BIN3
4	BIN4
5	BIN5
6	BIN6
7	BIN7
8	BIN8
9	BIN9
10	AUX_BIN
11	Cannot be sorted (OVLD or NC is detected)

To display the comparator sorting result use the following command to select the <BIN No. DISPLAY> page.

- `:DISPlay:PAGE` on page 213

Figure 7-1 Sorting judgment flow



Reading Out Sort Count Of Each Bin (BIN Count Function)

Turning ON/OFF BIN count function

To turn ON/OFF the BIN count function, use the following command.

- `:CALCulate1:COMParator:COUNt[:STATe]` on page 180

You can count the number of DUTs sorted into each BIN by turning ON the BIN count function. The maximum value of the count is 999999. If this value is exceeded, the count does not continue to increase but remains at 999999 (does not return to 0).

When the MULTI correction function is ON (set to ON with `[:SENSe]:CORRection:MULTiple[:STATe]` command on page 245), a channel-by-channel count is performed in addition to the normal count (total of all the channels).

Reading out BIN count value

To read out the BIN count value, use the following commands:

- `:CALCulate1:COMParator:COUNt:DATA?` on page 176
- `:CALCulate1:COMParator:COUNt:OVLD?` on page 179

To read out the BIN count value for each channel when the MULTI correction function is ON, use the following commands:

- `:CALCulate1:COMParator:COUNt:MULTiple:DATA?` on page 177
- `:CALCulate1:COMParator:COUNt:MULTiple:OVLD?` on page 179

Clearing (resetting) the BIN count value

To clear all of the BIN count values (initialize to 0), use the following command.

- `:CALCulate1:COMParator:COUNt:CLEar` on page 175

Sample Program

See Example 9-14 to view an example of sorting using comparator function.

8 Avoiding Mistakes Related to Work and Daily Checks

This chapter describes how to avoid simple mistakes related to work, how to detect the occurrence of an error, and how to execute the self-test.

Avoiding Mistakes Related To Work

Avoiding improper input from the front panel (key lock function)

When you do not need to operate the keys on the front panel, you can disable entry from the front panel keys (key lock function) to avoid improper input due to touching the front panel keys accidentally.

To turn ON/OFF the key lock function, use the following command.

- `:SYSTem:KLOCk` on page 283

Avoiding mistakes related to work when obtaining correction data

By confirming that the data is measured properly, you can avoid careless mistakes related to work when measuring data for OPEN/SHORT/LOAD correction (for example, setting up the OPEN state and SHORT state inversely).

For details, refer to “Avoiding work-related mistakes in measuring correction data” on page 52.

Detecting the occurrence of an error

Using error queue

The error queue contains the error number and error message of an error that has occurred. By reading out the contents of the error queue, you can find out which error has occurred. To read out the contents of the error queue, use the following command.

- `:SYSTem:ERRor[:NEXT]?` on page 280

You can use the error queue in the following ways.

1. You can use it to make a branch for error handling in your program. When you read out the contents of the error queue, 0 and “No error” are read out as the error number and error message if no error has occurred. This result helps you to determine whether an error has occurred and to make the branch of your program’s flow. You can also use it to restrict error handling to the occurrence of a specific error. However, it is difficult to use this method to perform processing during the occurrence of an error.
2. You can use it to investigate the kind of error that has occurred when an error is detected by using SRQ and other means.

Using status report system

You can detect the status of the E4981A by using the status register. This section describes how to detect the occurrence of an error by using the status register. For information on the entire status report system (for example, information on each bit of the status register), refer to Appendix C, “Status Reporting System” on page 363.

The occurrence of an error is indicated by the standard event status register. An SRQ (service request), which is useful for detecting the occurrence of an error in your program, uses the information indicated by this register.

To detect the end of sweep with an SRQ, use the following commands.

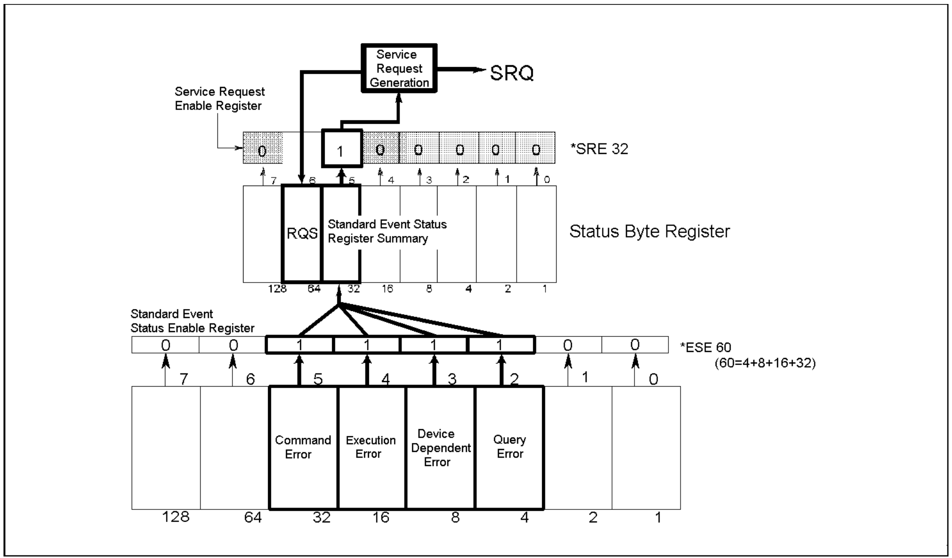
- `*SRE` on page 162
- `*ESE` on page 157

The procedure is given below.

Step 1. **Set the E4981A to generate an SRQ if one of the error occurrence bits of the standard event status register is set to 1.**

Step 2. **Perform interrupt handling in the program when SRQ occurs.**

Figure 8-1 SRQ generation sequence (when an error occurs)



4288ape015

Sample program

Refer to Example 9-14 to detect the occurence of an error using SRQ.

Daily checks (executing the self-test)

For information on daily checks, refer to *User Guide* “Precautions for use and Daily Checks”.

9 Measurement Applications (Sample Programs)

This chapter contains sample programs in Excel VBA format for both basic measurement and measurement using a system integrated with the handler/scanner interface.

Setting Up Measurement Conditions and LCD Display

This application program sets the measurement conditions and the LCD display of the E4981A.

The program of Example 9-1 is described in detail below. Each explanation below is inserted as a comment line in the program. For details on the VISA functions, refer to the I/O Library Suite manual.

Configuration: Define Variables and initialize the program.

Open Instrument: Assigns a USB address or GPIB address to the I/O bus.

Setup Start: Presets the E4981A. Sets the measurement conditions and the LCD display.

Setup End: Closes the I/O bus.

ErrorProc: Displays an error number when an error occurs.

ErrorCheck: A subroutine to check whether an error occurs when the IO Library Suite function is executed.

SelectMode: A subroutine to change assignment of the I/O bus and address setting according to the connection type i.e. USB or GPIB between the E4981A and a PC.

Example 9-1 Setting measurement conditions and screen display

```
Sub Example1()

'====================================
' Configuration
'====================================

Dim defrm As Long 'Session to Default Resource Manager
Dim Agte4981a As Long 'Session to instrument
Dim Result As String * 500
Dim Res As Variant
Dim Nop As Integer, i As Integer
Const TimeOutTime = 30000
On Error GoTo ErrorHandler

'====================================
' Open Instrument
'====================================

ErrorCheck viOpenDefaultRM(defrm)
Call SelectMode(defrm, Agte4981a)
ErrorCheck viSetAttribute(Agte4981a, VI_ATTR_TMO_VALUE, TimeOutTime)

'====================================
' Setup Start
'====================================

    ErrorCheck viVPrintf(Agte4981a, "SYST:PRES;*CLS" + vbLf, 0)
    ErrorCheck viVPrintf(Agte4981a, "CALC1:FORM CS" + vbLf, 0)
    ErrorCheck viVPrintf(Agte4981a, "CALC2:FORM Q" + vbLf, 0)
    ErrorCheck viVPrintf(Agte4981a, "SOUR:FREQ 1E3" + vbLf, 0)
    ErrorCheck viVPrintf(Agte4981a, "SOUR:VOLT 0.5" + vbLf, 0)
    ErrorCheck viVPrintf(Agte4981a, "RANG:AUTO ON" + vbLf, 0)
    ErrorCheck viVPrintf(Agte4981a, "APER:TIME 1" + vbLf, 0)
    ErrorCheck viVPrintf(Agte4981a, "CAL:CABL 0" + vbLf, 0)
    ErrorCheck viVPrintf(Agte4981a, "AVER ON" + vbLf, 0)

    ErrorCheck viVPrintf(Agte4981a, "AVER:COUN 4" + vbLf, 0)

    ErrorCheck viVPrintf(Agte4981a, "TRIG:DEL 0.001" + vbLf, 0)

    ErrorCheck viVPrintf(Agte4981a, "DISP ON" + vbLf, 0)

    ErrorCheck viVPrintf(Agte4981a, "CALC1:MATH:STAT ON" + vbLf, 0)

    ErrorCheck viVPrintf(Agte4981a, "CALC1:MATH:EXPR:NAME PCNT" + vbLf, 0)

    ErrorCheck viVPrintf(Agte4981a, "DATA REF1, 1.0E-8" + vbLf, 0)

    ErrorCheck viVPrintf(Agte4981a, "CALC2:MATH:STAT OFF" + vbLf, 0)

    ErrorCheck viVPrintf(Agte4981a, "CALC:COMP:BEEP ON" + vbLf, 0)

    ErrorCheck viVPrintf(Agte4981a, "CALC:COMP:BEEP:COND FAIL" + vbLf, 0)

    ErrorCheck viVPrintf(Agte4981a, "*SAV 1" + vbLf, 0)


'====================================

' Setup End

'====================================


Call viClose(defrm)


End


'====================================

' ErrorProc

'====================================


ErrorHandler:

' Display the error message

MsgBox "*** Error : " + Error$, MB_ICON_EXCLAMATION

End


End Sub
```

```
'===================================

' ErrorCheck

'===================================


Sub ErrorCheck(ErrorStatus As Long)

Dim strVisaErr As String * 500


' Check if VISA Error


If ErrorStatus <> VI_SUCCESS Then

Call viStatusDesc(defrm, ErrorStatus, strVisaErr)

MsgBox "*** Error : " + strVisaErr

End If


End Sub


'===================================

' Select Connection Mode (GPIB/USB)

'===================================


Sub SelectMode(defrm As Long, Agte4981a As Long)


Dim SelectMode As String

SelectMode = Worksheets("ControlPanel").Range("B3").Value


If SelectMode = "GPIB" Then

ErrorCheck viOpen(defrm, "GPIB0::17::INSTR", 0, 0, Agte4981a)

End If


If SelectMode = "USB" Then

ErrorCheck viOpen(defrm, "USB0::2391::2313::MY12345678::0::INSTR", 0, 0,
Agte4981a)

End If


End Sub
```

Executing OPEN/SHORT/LOAD Correction

This application program executes the OPEN/SHORT/LOAD correction in the E4981A.

The program of Example 9-2 is described in detail below. Each explanation below is inserted as a comment line in the program. For details on the VISA functions, refer to the I/O Library Suite manual.

Configuration Define Variables and initialize the program.

Open Instrument Assigns a USB address or GPIB address to the I/O bus.

Setup Start Presets the E4981A. Sets the measurement conditions and the LCD display.

Setup End Closes the I/O bus.

ErrorProc Displays an error number when an error occurs.

ErrorCheck A subroutine to check whether an error occurs when the IO Library Suite function is executed.

SelectMode A subroutine to change assignment of the I/O bus and address setting according to the connection type i.e. USB or GPIB between the E4981A and a PC.

FnComp A function which performs OPEN/SHORT/LOAD correction and returns 1 on success and 0 on failure.

Save_Corr_File A function which reads the E4981A correction data such as Frequency, Primary and Secondary parameters correction values for Open/Short/Load, Standard values for Open/Short/Load, Cable length and Phase Shift and then saves this data in a text file.

Example 9-2 Executing OPEN/SHORT/LOAD correction

```
Sub Example2()

'====================================

' Configuration

'====================================


Dim defrm As Long 'Session to Default Resource Manager

Dim Agte4981a As Long 'Session to instrument

Dim Result As String * 500

Dim Freq As String

Dim Res As Variant

Dim Nop As Integer, i As Integer

Dim Corr_File_Name As String

Const TimeOutTime = 30000

On Error GoTo ErrorHandler


'====================================

' Open Instrument

'====================================


ErrorCheck viOpenDefaultRM(defrm)

Call SelectMode(defrm, Agte4981a)

ErrorCheck viSetAttribute(Agte4981a, VI_ATTR_TMO_VALUE, TimeOutTime)


'====================================

' Setup Start

'====================================


 ErrorCheck viVPrintf(Agte4981a, "SYST:PRES;*CLS" + vbLf, 0) ' Preset the
E4981A



 Freq = InputBox("Enter the frequency (120/1E3/1E6) you want to measure")




'====================================
' Measurement
'===================================

   Corr_Result = FnComp("OPEN", 0.00002, Freq)

    If Corr_Result <> 0 Then
        MsgBox "OPEN compensation is not successful."
        'End
    End If

    Corr_Result = FnComp("SHORT", 20, Freq)

    If Corr_Result <> 0 Then
        MsgBox "SHORT compensation is not successful."
        'End
    End If

    Corr_Result = FnComp("LOAD", 0.2, Freq)

    If Corr_Result <> 0 Then
        MsgBox "LOAD compensation is not successful."
        'End
    End If

    Corr_File_Name = "C:\E4981A_Corr_Data.txt"
    ' Save Data
    Call Save_Corr_File(Corr_File_Name, Freq)

    MsgBox "Correction data stored at " & Corr_File_Name




'===================================
' Setup End
```

```
'====================================


Call viClose(defrm)


End


'====================================

' ErrorProc

'====================================


ErrorHandler:

' Display the error message

MsgBox "*** Error : " + Error$, MB_ICON_EXCLAMATION

End


End Sub


Function FnComp(Standard As String, Limit As Double, Freq As String)


    Dim Std As String

    Dim i As Integer, j As Integer

    Dim Load1 As Double, Load2 As Double

    Dim Finish_pros As String * 1

    Dim Result As String * 500

    Dim Res As Variant

    Dim Err_Flag As Boolean

    Dim Param1 As Double, Param2 As Double

    Dim Zm As Double, Ym As Double, Gm As Double, Bm As Double

    Dim Cpref As Double, Dref As Double, Zref As Double, Gref As Double,
Bref As Double

    Dim defrm As Long 'Session to Default Resource Manager

    Dim Agte4981a As Long 'Session to instrument

    Const TimeOutTime = 30000
```

```
'===================================
' Open Instrument
'===================================

ErrorCheck viOpenDefaultRM(defrm)
Call SelectMode(defrm, Agte4981a)
ErrorCheck viSetAttribute(Agte4981a, VI_ATTR_TMO_VALUE, TimeOutTime)

    Err_Flag = False
    Const Pi = 3.141592654

    ErrorCheck viVPrintf(Agte4981a, "STAT:OPER:ENAB 16" + vbLf, 0)
    ErrorCheck viVPrintf(Agte4981a, "*SRE 128" + vbLf, 0)

    Select Case Standard

    Case "OPEN"
    Std = "STAN1"

    Case "SHORT"
    Std = "STAN2"

    Case "LOAD"
    Std = "STAN3"

    Load1 = InputBox("Enter the load Cp Value for " & Freq)
    Load2 = InputBox("Enter the load D Value for " & Freq)

   ErrorCheck viVPrintf(Agte4981a, ":CORR:CKIT:STAN3:FORMAT CPD" + vbLf, 0)
' Set Load Type CP and D
    ErrorCheck viVPrintf(Agte4981a, ":SOUR:FREQ " & Freq + vbLf, 0)
    ErrorCheck viVPrintf(Agte4981a, ":CORR:CKIT:STAN3 " & CStr(Load1) & ","
& CStr(Load2) + vbLf, 0) ' Set Primary/Secondary parameter Load Coorection
Values
          End Select
```

```
j= MsgBox("Set " & Standard & " connection and Press OK.", vbOKCancel)

    If j = vbCancel Then
    MsgBox "Operation cancelled. Ending the program."
    Err_Flag = True

    Else

           ErrorCheck viVPrintf(Agte4981a, ":CORR:COLL " & Std + vbLf, 0)

          ' Data Check
            ErrorCheck viVPrintf(Agte4981a, ":CORR:DATA? " & Std + vbLf, 0)
             ErrorCheck viVScanf(Agte4981a, "%t", Result)

            Res = Split(Result, ",")
            Param1 = Val(Res(0))
            Param2 = Val(Res(1))

            Select Case Standard

            Case "OPEN"
                Ym = Sqr(Param1 * Param1 + Param2 * Param2)
              MsgBox "G= " & Param1 & vbNewLine & "B= " & Param2 & vbNewLine
& "|Y| = " & Ym

                If Ym >= Limit Then Err_Flag = True

            Case "SHORT"
                Zm = Sqr(Param1 * Param1 + Param2 * Param2)
              MsgBox "R= " & Param1 & vbNewLine & "X= " & Param2 & vbNewLine
& "|Z| = " & Zm

                If Zm >= Limit Then Err_Flag = True

            Case "LOAD"
```

```
                   ErrorCheck viVPrintf(Agte4981a, (":SOUR:FREQ " & Freq) +
vbLf, 0)
                    ErrorCheck viVPrintf(Agte4981a, "*CLS" + vbLf, 0)


                    While Finish_pros <> "1" ' Loop till processing is
finished. *OPC? retruns 1 when processing of executed commands is finished.
                     ErrorCheck viVPrintf(Agte4981a, "*OPC?" + vbLf, 0)
                     ErrorCheck viVScanf(Agte4981a, "%t", Finish_pros)
                    Wend


                    ErrorCheck viVPrintf(Agte4981a, ":CORR:CKIT:STAN3?" +
vbLf, 0)
                    ErrorCheck viVScanf(Agte4981a, "%t", Result)


                    Res = Split(Result, ",")
                    Cpref = Val(Res(0))
                    Dref = Val(Res(1))


                    Bref = 2 * Pi * Val(Freq) * Cpref
                    Gref = Bref * Dref
                    Zref = 1 / (Sqr(Gref * Gref + Bref * Bref))
                    Bm = 2 * Pi * Val(Freq) * Param1
                    Gm = Bm * Param2
                    Zm = 1 / (Sqr(Gm * Gm + Bm * Bm))



MsgBox "Cpref= " & Cpref & vbNewLine & "Dref= " & Dref & vbNewLine & "|Zref|
= " & Zref
MsgBox "Cp= " & Param1 & vbNewLine & "D= " & Param2 & vbNewLine & "|Z| = " &
Zm


If Abs((Zm - Zref) / Zref) >= Limit Then Err_Flag = 1


            End Select
```

```
   If Err_Flag = False Then

   MsgBox (Standard & " Data measurement completed.")

   Else

   MsgBox (Standard & " Data measurement not completed."), vbExclamation

   End If


End If


'===================================

' Setup End

'===================================


Call viClose(defrm)


FnComp = Err_Flag


End Function


Sub Save_Corr_File(FileName As String, Freq As String)


Dim defrm As Long 'Session to Default Resource Manager

Dim Agte4981a As Long 'Session to instrument

Dim Corr_File_Object As Object

Dim FileS As Object

Dim Result As String * 500

Dim Res As Variant

Dim Nop As Integer, i As Integer

Dim Open1 As Double, Open2 As Double

Dim Short1 As Double, Short2 As Double

Dim Load(1 To 3, 1 To 3) As Double

Dim StdOpen As String

Dim StdLoad As String

Dim StdShort As String

Dim Cab_Len As Integer

Dim Sys_Fsh As Integer

Const TimeOutTime = 30000
```

```
'===================================
' Open Instrument
'===================================

ErrorCheck viOpenDefaultRM(defrm)
Call SelectMode(defrm, Agte4981a)
ErrorCheck viSetAttribute(Agte4981a, VI_ATTR_TMO_VALUE, TimeOutTime)

' Reading and Saving Correction values for Open, Short and Load measurements

      Result = ""
      ' Open Correction Parameter
      ErrorCheck viVPrintf(Agte4981a, ":CORR:CKIT:STAN1:FORMAT?" + vbLf, 0)
' Read Primary and Secondary Load Types CP and D
      ErrorCheck viVScanf(Agte4981a, "%t", Result)

      StdOpen = Mid(Result, 1, 2)
      Result = ""

      ' Open Correction Values
      ErrorCheck viVPrintf(Agte4981a, ":CORR:DATA? STAN1" + vbLf, 0)
      ErrorCheck viVScanf(Agte4981a, "%t", Result)

      Res = Split(Result, ",")

      Open1 = Val(Res(0))
      Open2 = Val(Res(1))

      Result = ""
      ' Short Correction Parameter
      ErrorCheck viVPrintf(Agte4981a, ":CORR:CKIT:STAN2:FORMAT?" + vbLf, 0)
' Read Primary and Secondary Load Types CP and D
      ErrorCheck viVScanf(Agte4981a, "%t", Result)

      StdShort = Mid(Result, 1, 2)
      Result = ""
      ' Short Correction Values
      ErrorCheck viVPrintf(Agte4981a, ":CORR:DATA? STAN2" + vbLf, 0)
      ErrorCheck viVScanf(Agte4981a, "%t", Result)

      Res = Split(Result, ",")

      Short1 = Val(Res(0))
      Short2 = Val(Res(1))

      Result = ""

      ' Load Correction Parameter
      ErrorCheck viVPrintf(Agte4981a, ":CORR:CKIT:STAN3:FORMAT?" + vbLf, 0)
' Read Primary and Secondary Load Types CP and D
      ErrorCheck viVScanf(Agte4981a, "%t", Result)

      StdLoad = Mid(Result, 1, 3)
      Result = ""

        ErrorCheck viVPrintf(Agte4981a, ":CORR:DATA? STAN3" + vbLf, 0)
        ErrorCheck viVScanf(Agte4981a, "%t", Result)

        Res = Split(Result, ",")

        Load1 = Val(Res(0))
        Load2 = Val(Res(1))
        Result = ""

      Result = ""
      ' Cable Length
      ErrorCheck viVPrintf(Agte4981a, "CAL:CABL?" + vbLf, 0)
      ErrorCheck viVScanf(Agte4981a, "%t", Result)
      Cab_Len = Val(Result)
      Result = ""

      ' Frequency Shift

      ErrorCheck viVPrintf(Agte4981a, "SYST:FSH?" + vbLf, 0)

      ErrorCheck viVScanf(Agte4981a, "%t", Result)

      Sys_Fsh = Val(Result)


Set FileS = CreateObject("Scripting.FileSystemObject")

Set Corr_File_Object = FileS.CreateTextFile(FileName, True)


Corr_File_Object.WriteLine (Freq)

Corr_File_Object.WriteLine (Open1)

Corr_File_Object.WriteLine (Open2)

Corr_File_Object.WriteLine (Short1)

Corr_File_Object.WriteLine (Short2)

Corr_File_Object.WriteLine (Load1)

Corr_File_Object.WriteLine (Load2)

Corr_File_Object.WriteLine (StdOpen)

Corr_File_Object.WriteLine (StdShort)

Corr_File_Object.WriteLine (StdLoad)

Corr_File_Object.WriteLine (Cab_Len)

Corr_File_Object.WriteLine (Sys_Fsh)


'===================================

' Setup End

'===================================


Call viClose(defrm)



End Sub
```

Recovering the Correction State

This application program recovers the OPEN/SHORT/LOAD correction state of E4981A from a text file and then updates the E4981A to the saved correction state.

The program of Example 9-3 is described in detail below. Each explanation below is inserted as a comment line in the program. For details on the VISA functions, refer to the I/O Library Suite manual.

Configuration	Define Variables and initialize the program.
Open Instrument	Assigns a USB address or GPIB address to the I/O bus.
Setup Start	Presets the E4981A. Sets the measurement conditions and the LCD display.
Setup End	Presets the E4981A. Reads the E4981A correction data such as Frequency, Primary and Secondary parameters correction values for Open/Short/Load, Standard values for Open/Short/Load, Cable length and Phase Shift stored in a text file.
ErrorProc	Displays an error number when an error occurs.
ErrorCheck	A subroutine to check whether an error occurs when the IO Library Suite function is executed.
SelectMode	A subroutine to change assignment of the I/O bus and address setting according to the connection type i.e. USB or GPIB between the E4981A and a PC.
FnComp	A function which performs OPEN/SHORT/LOAD correction and returns 1 on success and 0 on failure.

Example 9-3 Recovering the Correction State

```
Sub Example3()
'====================================
' Configuration
'====================================

Dim defrm As Long 'Session to Default Resource Manager
Dim Agte4981a As Long 'Session to instrument
Dim FileS As Object
Dim Result As String * 500
Dim Res As Variant
Dim Nop As Integer, i As Integer
Dim Open1 As Double, Open2 As Double
Dim Short1 As Double, Short2 As Double
Dim Load1, Load2 As Double
Dim StdOpen As String, StdLoad As String, StdShort As String
Dim Freq As String
Dim File_Path As String
Dim FileOpen As Object
Dim Corr_File_Object As New FileSystemObject
Dim Corr_Data(1 To 12) As String
Dim Cab_Len As String
Dim Sys_Fsh As String

Const TimeOutTime = 30000
i = 1

'====================================
' Open Instrument
'====================================

ErrorCheck viOpenDefaultRM(defrm)
Call SelectMode(defrm, Agte4981a)
ErrorCheck viSetAttribute(Agte4981a, VI_ATTR_TMO_VALUE, TimeOutTime)
```

```
'===================================

' Setup Start

'===================================


File_Path = "C:\E4981A_Corr_Data.txt"


Set FileOpen = Corr_File_Object.OpenTextFile(File_Path)

Do Until FileOpen.AtEndOfStream

    Corr_Data(i) = FileOpen.ReadLine

    i = i + 1

Loop


Freq = Corr_Data(1)

Open1 = Corr_Data(2)

Open2 = Corr_Data(3)

Short1 = Corr_Data(4)

Short2 = Corr_Data(5)

Load1 = Corr_Data(6)

Load2 = Corr_Data(7)

StdOpen = Corr_Data(8)

StdShort = Corr_Data(9)

StdLoad = Corr_Data(10)

Cab_Len = Corr_Data(11)

Sys_Fsh = Corr_Data(12)


'===================================

' Recover Data

'===================================


 ErrorCheck viVPrintf(Agte4981a, "SYST:PRES;*CLS" + vbLf, 0) ' Preset the
E4981A


 ErrorCheck viVPrintf(Agte4981a, "SOUR:FREQ " & Freq + vbLf, 0)

 ErrorCheck viVPrintf(Agte4981a, "CORR:DATA STAN3," & CStr(Load1) & "," &
CStr(Load2) + vbLf, 0)


 ErrorCheck viVPrintf(Agte4981a, "CAL:CABL " & Cab_Len + vbLf, 0)

 ErrorCheck viVPrintf(Agte4981a, "SYST:FSH " & Sys_Fsh + vbLf, 0)
 ErrorCheck viVPrintf(Agte4981a, "CORR:CKIT:STAN1:FORMAT " & StdOpen + vbLf,
0) ' Write Primary and Secondary Load Type

 ErrorCheck viVPrintf(Agte4981a, "CORR:CKIT:STAN2:FORMAT " & StdShort +
vbLf, 0)

 ErrorCheck viVPrintf(Agte4981a, "CORR:CKIT:STAN3:FORMAT " & StdLoad + vbLf,
0)

 ErrorCheck viVPrintf(Agte4981a, "CORR:DATA STAN1," & CStr(Open1) & "," &
CStr(Open2) + vbLf, 0)

 ErrorCheck viVPrintf(Agte4981a, "CORR:DATA STAN2," & CStr(Short1) & "," &
CStr(Short2) + vbLf, 0)


 ErrorCheck viVPrintf(Agte4981a, "CORR:OPEN ON" + vbLf, 0)

 ErrorCheck viVPrintf(Agte4981a, "CORR:SHORT ON" + vbLf, 0)

 ErrorCheck viVPrintf(Agte4981a, "CORR:LOAD ON" + vbLf, 0)


 MsgBox "Correction Data sucessfully applied to E4981A from " & File_Path


'===================================

' ErrorProc

'===================================

End

ErrorHandler:

' Display the error message

MsgBox "*** Error : " + Error$, MB_ICON_EXCLAMATION

End


End Sub
```

Executing Multi Correction

This application program recovers the OPEN/SHORT/LOAD correction state for multiple channels in the E4981A.

The program of Example 9-4 is described in detail below. Each explanation below is inserted as a comment line in the program. For details on the VISA functions, refer to the I/O Library Suite manual.

Configuration	Define Variables and initialize the program.
Open Instrument	Assigns a USB address or GPIB address to the I/O bus.
Setup Start	Presets the E4981A. Enables Multi correction. Inputs the Measurement Frequency (120/1E3/1E6) of the E4981A.
Measurement	Performs OPEN/SHORT/LOAD correction for four channels respectively.
Setup End	Closes the I/O bus.
ErrorProc	Displays an error number when an error occurs.
ErrorCheck	A subroutine to check whether an error occurs when the IO Library Suite function is executed.
SelectMode	A subroutine to change assignment of the I/O bus and address setting according to the connection type i.e. USB or GPIB between the E4981A and a PC.
FnComp	A function which performs OPEN/SHORT/LOAD correction and returns 1 on success and 0 on failure.

Example 9-4 Executing Multi Correction

```
Sub Example4()


'===================================
' Configuration
'===================================
Dim defrm As Long 'Session to Default Resource Manager
Dim Agte4981a As Long 'Session to instrument
Const TimeOutTime = 30000
Dim Nop As Integer, i As Integer
Dim Corr_Result As Boolean
Dim Freq As String
On Error GoTo ErrorHandler


'===================================
' Open Instrument
'===================================
ErrorCheck viOpenDefaultRM(defrm)
Call SelectMode(defrm, Agte4981a)
ErrorCheck viSetAttribute(Agte4981a, VI_ATTR_TMO_VALUE, TimeOutTime)


'===================================
' Setup Start
'===================================


   ErrorCheck viVPrintf(Agte4981a, "SYST:PRES;*CLS" + vbLf, 0) ' Preset the
E4981A


    ErrorCheck viVPrintf(Agte4981a, ":CORR:MULT ON" + vbLf, 0) ' Enable
Multi Correction
    ErrorCheck viVPrintf(Agte4981a, ":CORR:MULT:CKIT:STAN3 ON" + vbLf, 0) '
Enable Multi Correction channel-by-channel value
    ErrorCheck viVPrintf(Agte4981a, ":DISP:PAGE CSET" + vbLf, 0) ' Display
the Correction Page


   Freq = InputBox("Enter the frequency (120/1E3/1E6) you want to measure")
For i = 0 To 3
```

```
        ErrorCheck viVPrintf(Agte4981a, (":CORR:MULT:CHAN " & i) + vbLf, 0)
' Select Channel No. i
         MsgBox "OPEN correction for Channel: " & i
         Corr_Result = FnComp("OPEN", 0.00002, Freq)

    Next i

    For i = 0 To 3

        ErrorCheck viVPrintf(Agte4981a, (":CORR:MULT:CHAN " & i) + vbLf, 0)
' Select Channel No. i
         MsgBox "SHORT correction for Channel: " & i
         Corr_Result = FnComp("SHORT", 20, Freq)

    Next i

    For i = 0 To 3

        ErrorCheck viVPrintf(Agte4981a, (":CORR:MULT:CHAN " & i) + vbLf, 0)
' Select Channel No. i
         MsgBox "LOAD correction for Channel: " & i
         Corr_Result = FnComp("LOAD", 0.2, Freq)

    Next i

'===================================
' Setup End
'===================================

Call viClose(defrm)

End
```

```
'====================================

' ErrorProc

'====================================


ErrorHandler:

' Display the error message

MsgBox "*** Error: " + Error$, MB_ICON_EXCLAMATION

End


End Sub
```

Detecting the Completion of Measurement using SRQ

This application program detects the end of measurement with the status byte after the triggered.E4981A is triggered

The program of Example 9-5 is described in detail below. Each explanation below is inserted as a comment line in the program. For details on the VISA functions, refer to the I/O Library Suite manual.

Configuration Define Variables and initialize the program.

Open Instrument Assigns a USB address or GPIB address to the I/O bus.

Setup Start Presets the E4981A. Sets the trigger conditions of the E4981A.

STB Check Waits until the status byte becomes 192. Displays the status byte while waiting and again after processing.

Setup End Closes the I/O bus.

ErrorProc Displays an error number when an error occurs.

ErrorCheck A subroutine to check whether an error occurs when the IO Library Suite function is executed.

SelectMode A subroutine to change assignment of the I/O bus and address setting according to the connection type i.e. USB or GPIB between the E4981A and a PC.

FnComp A function which performs OPEN/SHORT/LOAD correction and returns 1 on success and 0 on failure.

Example 9-5 Detecting the Completion of Measurement Using SRQ

```
Sub Example5()
'===================================
' Configuration
'===================================

Dim defrm As Long 'Session to Default Resource Manager
Dim Agte4981a As Long 'Session to instrument
Dim Result As String * 500
Dim Res As Variant
Dim Nop As Integer, i As Integer, StbStatus As Integer
Const TimeOutTime = 30000
On Error GoTo ErrorHandler

'===================================
' Open Instrument
'===================================

ErrorCheck viOpenDefaultRM(defrm)
Call SelectMode(defrm, Agte4981a)
ErrorCheck viSetAttribute(Agte4981a, VI_ATTR_TMO_VALUE, TimeOutTime)

'===================================
' Setup Start
'===================================
Worksheets("5-1").Range("B6").Value = ""

ErrorCheck viVPrintf(Agte4981a, "SYST:PRES;*CLS" + vbLf, 0)
ErrorCheck viVPrintf(Agte4981a, "TRIG:SOUR BUS" + vbLf, 0)

ErrorCheck viVPrintf(Agte4981a, ":AVER ON" + vbLf, 0)
ErrorCheck viVPrintf(Agte4981a, ":AVER:COUN 10" + vbLf, 0)

ErrorCheck viVPrintf(Agte4981a, "STAT:OPER:ENAB 16" + vbLf, 0)
ErrorCheck viVPrintf(Agte4981a, "*SRE 128" + vbLf, 0)
ErrorCheck viVPrintf(Agte4981a, "INIT:CONT OFF" + vbLf, 0)

ErrorCheck viVPrintf(Agte4981a, "ABOR;INIT" + vbLf, 0)
ErrorCheck viVPrintf(Agte4981a, "TRIG:IMM" + vbLf, 0)

'====================================
' STB Check
'====================================

Do
    ErrorCheck viReadSTB(Agte4981a, StbStatus)
    Worksheets("Example5").Range("B5").Value = StbStatus

Loop Until StbStatus = 192

Worksheets("Example5").Range("B6").Value = "Measurement Done"

ErrorCheck viVPrintf(Agte4981a, "*CLS" + vbLf, 0)

'====================================
' Setup End
'====================================

Call viClose(defrm)

End

'====================================
' ErrorProc
'====================================

ErrorHandler:
' Display the error message
MsgBox "*** Error : " + Error$, MB_ICON_EXCLAMATION
End
End Sub
```

Reading Measurement Results in ASCII Format Using *TRG Command

This application program reads measurement data in ASCII format using the *TRG command.

The program of Example 9-6 is described in detail below. Each explanation below is inserted as a comment line in the program. For details on the VISA functions, refer to the I/O Library Suite manual.

Configuration	Define Variables and initialize the program.
Open Instrument	Assigns a USB address or GPIB address to the I/O bus.
Setup Start	Presets the E4981A. Sets the trigger conditions of the E4981A.
Measurement	Reads the measurement data using the *TRG command, separates the returned ASCII format character string data with commas, and displays the Primary and Secondary Measurement data.
Setup End	Closes the I/O bus.
ErrorProc	Displays an error number when an error occurs.
ErrorCheck	A subroutine to check whether an error occurs when the IO Library Suite function is executed.
SelectMode	A subroutine to change assignment of the I/O bus and address setting according to the connection type i.e. USB or GPIB between the E4981A and a PC.

Example 9-6 Reading Measurement Results in Ascii Format using *TRG Command

```
Sub Example6()

'===================================
' Configuration
'===================================

Dim defrm As Long 'Session to Default Resource Manager
Dim Agte4981a As Long 'Session to instrument
Dim Result As String * 500
Dim Res As Variant
Dim Nop As Integer, i As Integer
Const TimeOutTime = 30000
On Error GoTo ErrorHandler

'===================================
' Open Instrument
'===================================

ErrorCheck viOpenDefaultRM(defrm)
Call SelectMode(defrm, Agte4981a)
ErrorCheck viSetAttribute(Agte4981a, VI_ATTR_TMO_VALUE, TimeOutTime)

'===================================
' Setup Start
'===================================

ErrorCheck viVPrintf(Agte4981a, "SYST:PRES;*CLS" + vbLf, 0)
ErrorCheck viVPrintf(Agte4981a, ":FORM ASC" + vbLf, 0)
'
ErrorCheck viVPrintf(Agte4981a, ":INIT:CONT ON" + vbLf, 0)
ErrorCheck viVPrintf(Agte4981a, ":TRIG:SOUR BUS" + vbLf, 0)
```

```
'===================================
' Measurement
'===================================

ErrorCheck viVPrintf(Agte4981a, "*TRG" + vbLf, 0)
ErrorCheck viVScanf(Agte4981a, "%t", Result)

Res = Split(Result, ",")

Worksheets("Example6").Range("B5").Value = Val(Res(1))
Worksheets("Example6").Range("B6").Value = Val(Res(2))

'===================================
' Setup End
'===================================

Call viClose(defrm)

End

'===================================
' ErrorProc
'===================================

ErrorHandler:
' Display the error message
MsgBox "*** Error : " + Error$, MB_ICON_EXCLAMATION

End

End Sub
```

Reading Measurement Results in Binary Format Using *TRG Command

This application program reads measurement data in Binary format using the *TRG command.

The program of Example 9-7 is described in detail below. Each explanation below is inserted as a comment line in the program. For details on the VISA functions, refer to the I/O Library Suite manual.

Configuration	Define Variables and initialize the program.
Open Instrument	Assigns a USB address or GPIB address to the I/O bus.
Setup Start	Presets the E4981A. Sets the trigger conditions of the E4981A.
Measurement	Reads the measurement data using the *TRG command, calls the function to convert data in Binary format to Ascii format, and displays the Primary and Secondary Measurement data.
Setup End	Closes the I/O bus.
ErrorProc	Displays an error number when an error occurs.
Binary Read	A subroutine to convert Binary format data to Ascii format.
ErrorCheck	A subroutine to check whether an error occurs when the IO Library Suite function is executed.
SelectMode	A subroutine to change assignment of the I/O bus and address setting according to the connection type i.e. USB or GPIB between the E4981A and a PC.

Example 9-7 Reading Measurement Results in Binary Format using *TRG Command

```
Sub Example7()

'===================================

' Configuration

'===================================


Dim defrm As Long 'Session to Default Resource Manager

Dim Agte4981a As Long 'Session to instrument

Dim Res() As Double

Dim Nop As Long, i As Integer, j As Integer, k As Integer

Const TimeOutTime = 30000

On Error GoTo ErrorHandler


'===================================

' Open Instrument

'===================================


ErrorCheck viOpenDefaultRM(defrm)

Call SelectMode(defrm, Agte4981a)

ErrorCheck viSetAttribute(Agte4981a, VI_ATTR_TMO_VALUE, TimeOutTime)


'===================================

' Setup Start

'===================================


ErrorCheck viVPrintf(Agte4981a, "SYST:PRES;*CLS" + vbLf, 0)

ErrorCheck viVPrintf(Agte4981a, "FORM REAL" + vbLf, 0)

ErrorCheck viVPrintf(Agte4981a, "APER:TIME 6" + vbLf, 0)

ErrorCheck viVPrintf(Agte4981a, "TRIG:SOUR BUS" + vbLf, 0)

ErrorCheck viVPrintf(Agte4981a, "CALC1:COMP OFF" + vbLf, 0)

ErrorCheck viVPrintf(Agte4981a, "INIT:CONT OFF" + vbLf, 0)

ErrorCheck viVPrintf(Agte4981a, "INIT" + vbLf, 0)
```

```
'===================================
' Measurement
'===================================

ErrorCheck viVPrintf(Agte4981a, "*TRG" + vbLf, 0)

Call Scpi_read_binary_double_array(Agte4981a, Res, Nop)

Worksheets("Example7").Range("B5").Value = Val(Res(1))
Worksheets("Example7").Range("B6").Value = Val(Res(2))

'===================================
' Setup End
'===================================

Call viClose(defrm)
End

'===================================
' ErrorProc
'===================================

ErrorHandler:
' Display the error message
MsgBox "*** Error : " + Error$, MB_ICON_EXCLAMATION
End
End Sub
```

```
'===================================

' Binary Read

'===================================


Sub Scpi_read_binary_double_array(vi As Long, data() As Double, Nop As Long)


Dim dblArray(10000) As Double

Dim paramsArray(3) As Long

Dim err As Long

Dim i As Long

Dim lf_eoi As String * 1


Nop = UBound(dblArray) - LBound(dblArray) + 1

paramsArray(0) = VarPtr(Nop)

paramsArray(1) = VarPtr(dblArray(0))

err = viVScanf(vi, "%#Zb%1t", paramsArray(0))


If err <> 0 Then MsgBox "Binary Error"


ReDim data(Nop - 1)


For i = 0 To Nop - 1

data(i) = dblArray(i)

Next


End Sub
```

Reading Measurement Results in ASCII Format Using :FETCh? Command

This application program reads measurement data in ASCII format using the :FETCh? command.

The program of Example 9-8 is described in detail below. Each explanation below is inserted as a comment line in the program. For details on the VISA functions, refer to the I/O Library Suite manual.

Configuration	Define Variables and initialize the program.
Open Instrument	Assigns a USB address or GPIB address to the I/O bus.
Setup Start	Presets the E4981A. Sets the trigger conditions of the E4981A.
Measurement	Reads the measurement data using the :FETCh? command, separates the returned ASCII format character string data with commas, and displays the Primary and Secondary Measurement data.
Setup End	Closes the I/O bus.
ErrorProc	Displays an error number when an error occurs.
ErrorCheck	A subroutine to check whether an error occurs when the IO Library Suite function is executed.
SelectMode	A subroutine to change assignment of the I/O bus and address setting according to the connection type i.e. USB or GPIB between the E4981A and a PC.

Example 9-8 Reading Measurement Results in Ascii Format using :FETCh? Command

```
Sub Example8()

'===================================

' Configuration

'===================================

Dim defrm As Long 'Session to Default Resource Manager

Dim Agte4981a As Long 'Session to instrument

Dim Result As String * 500

Dim Res As Variant

Const TimeOutTime = 30000

On Error GoTo ErrorHandler

'===================================


' Open Instrument

'===================================


ErrorCheck viOpenDefaultRM(defrm)


Call SelectMode(defrm, Agte4981a)


ErrorCheck viSetAttribute(Agte4981a, VI_ATTR_TMO_VALUE, TimeOutTime)


'===================================

' Setup Start

'===================================


ErrorCheck viVPrintf(Agte4981a, "SYST:PRES;*CLS" + vbLf, 0)

ErrorCheck viVPrintf(Agte4981a, "FORM ASC" + vbLf, 0)

ErrorCheck viVPrintf(Agte4981a, "APER:TIME 6" + vbLf, 0)

ErrorCheck viVPrintf(Agte4981a, "TRIG:SOUR BUS" + vbLf, 0)

ErrorCheck viVPrintf(Agte4981a, "CALC1:COMP OFF" + vbLf, 0)

ErrorCheck viVPrintf(Agte4981a, "INIT:CONT OFF" + vbLf, 0)

ErrorCheck viVPrintf(Agte4981a, "INIT" + vbLf, 0)

ErrorCheck viVPrintf(Agte4981a, "TRIG" + vbLf, 0)
```

```
'===================================

' Measurement

'===================================`


ErrorCheck viVPrintf(Agte4981a, "FETCh?" + vbLf, 0)

ErrorCheck viVScanf(Agte4981a, "%t", Result)


Res = Split(Result, ",")


Worksheets("Example8").Range("B5").Value = Val(Res(1))

Worksheets("Example8").Range("B6").Value = Val(Res(2))


'===================================

' Setup End

'===================================`


Call viClose(defrm)

End


'===================================

' ErrorProc

'===================================`


ErrorHandler:

' Display the error message

MsgBox "*** Error : " + Error$, MB_ICON_EXCLAMATION

End


End Sub
```

Reading Measurement Results in Binary Format Using :FETCh? Command

This application program reads measurement data in Binary format using the :FETCh? command.

The program of Example 9-9 is described in detail below. Each explanation below is inserted as a comment line in the program. For details on the VISA functions, refer to the I/O Library Suite manual.

Configuration — Define Variables and initialize the program.

Open Instrument — Assigns a USB address or GPIB address to the I/O bus.

Setup Start — Presets the E4981A. Sets the trigger conditions of the E4981A.

Measurement — Reads the measurement data using the :FETCh? command, calls the function to convert data in Binary format to Ascii format, and displays the Primary and Secondary Measurement data.

Setup End — Closes the I/O bus.

ErrorProc — Displays an error number when an error occurs.

Binary Read — A subroutine to convert Binary format data to Ascii format.

ErrorCheck — A subroutine to check whether an error occurs when the IO Library Suite function is executed.

SelectMode — A subroutine to change assignment of the I/O bus and address setting according to the connection type i.e. USB or GPIB between the E4981A and a PC.

Example 9-9 Reading Measurement Results in Binary Format using :FETCh? Command

```
Sub Example9()
'===================================
' Configuration
'===================================

Dim defrm As Long 'Session to Default Resource Manager
Dim Agte4981a As Long 'Session to instrument
Dim Res() As Double
Dim Nop As Long, i As Integer, j As Integer, k As Integer
Const TimeOutTime = 30000
On Error GoTo ErrorHandler

'===================================
' Open Instrument
'===================================

ErrorCheck viOpenDefaultRM(defrm)
Call SelectMode(defrm, Agte4981a)
ErrorCheck viSetAttribute(Agte4981a, VI_ATTR_TMO_VALUE, TimeOutTime)


'===================================
' Setup Start
'===================================


ErrorCheck viVPrintf(Agte4981a, "SYST:PRES;*CLS" + vbLf, 0)
ErrorCheck viVPrintf(Agte4981a, "FORM REAL" + vbLf, 0)
ErrorCheck viVPrintf(Agte4981a, "APER:TIME 6" + vbLf, 0)
ErrorCheck viVPrintf(Agte4981a, "TRIG:SOUR BUS" + vbLf, 0)
ErrorCheck viVPrintf(Agte4981a, "CALC1:COMP OFF" + vbLf, 0)
ErrorCheck viVPrintf(Agte4981a, "INIT:CONT OFF" + vbLf, 0)
ErrorCheck viVPrintf(Agte4981a, "INIT" + vbLf, 0)
ErrorCheck viVPrintf(Agte4981a, "TRIG" + vbLf, 0)
```

```
'====================================
' Measurement
'====================================

ErrorCheck viVPrintf(Agte4981a, ":FETC?" + vbLf, 0)

Call Scpi_read_binary_double_array(Agte4981a, Res, Nop)



Worksheets("Example9").Range("B5").Value = Val(Res(1))
Worksheets("Example9").Range("B6").Value = Val(Res(2))

'====================================
' Setup End
'====================================

Call viClose(defrm)
End

'====================================
' ErrorProc
'====================================

ErrorHandler:
' Display the error message
MsgBox "*** Error : " + Error$, MB_ICON_EXCLAMATION
End

End Sub
```

Reading Measurement Results in ASCII Format Using :READ? Command

This application program reads measurement data in ASCII format using the :READ? command.

The program of Example 9-10 is described in detail below. Each explanation below is inserted as a comment line in the program. For details on the VISA functions, refer to the I/O Library Suite manual.

Configuration
Define Variables and initialize the program.

Open Instrument
Assigns a USB address or GPIB address to the I/O bus.

Setup Start
Presets the E4981A. Sets the trigger conditions of the E4981A.

Measurement
Reads the measurement data using the :READ? command, separates the returned ASCII format character string data with commas, and displays the Primary and Secondary Measurement data.

Setup End
Closes the I/O bus.

ErrorProc
Displays an error number when an error occurs.

ErrorCheck
A subroutine to check whether an error occurs when the IO Library Suite function is executed.

SelectMode
A subroutine to change assignment of the I/O bus and address setting according to the connection type i.e. USB or GPIB between the E4981A and a PC.

Example 9-10 Reading Measurement Results in Ascii Format using :READ? Command

```
Sub Example10()

'====================================

' Configuration

'====================================

Dim defrm As Long 'Session to Default Resource Manager

Dim Agte4981a As Long 'Session to instrument

Dim Result As String * 500

Dim Res As Variant

Const TimeOutTime = 10000

On Error GoTo ErrorHandler


'====================================

' Open Instrument

'====================================


ErrorCheck viOpenDefaultRM(defrm)


Call SelectMode(defrm, Agte4981a)


ErrorCheck viSetAttribute(Agte4981a, VI_ATTR_TMO_VALUE, TimeOutTime)


'====================================

' Setup Start

'====================================


ErrorCheck viVPrintf(Agte4981a, "SYST:PRES;*CLS" + vbLf, 0)

ErrorCheck viVPrintf(Agte4981a, "FORM ASC" + vbLf, 0)

ErrorCheck viVPrintf(Agte4981a, "INIT:CONT OFF" + vbLf, 0)

ErrorCheck viVPrintf(Agte4981a, "TRIG:SOUR EXT" + vbLf, 0)
```

```
'====================================
' Meas Read
'====================================`

ErrorCheck viVPrintf(Agte4981a, "READ?" + vbLf, 0)

Worksheets("Example10").Range("B5").Value = "Waiting for"
Worksheets("Example10").Range("B6").Value = "External Trigger"

ErrorCheck viVScanf(Agte4981a, "%t", Result)

Res = Split(Result, ",")

Worksheets("Example10").Range("B5").Value = Val(Res(1))
Worksheets("Example10").Range("B6").Value = Val(Res(2))

'====================================
' Setup End
'====================================`
Call viClose(defrm)
End

'====================================
' ErrorProc
'====================================`

ErrorHandler:
' Display the error message
MsgBox "*** Error : " + Error$, MB_ICON_EXCLAMATION
End

End Sub
```

Reading Measurement Results in Binary Format Using :READ? Command

This application program reads measurement data in Binary format using the :READ? command.

The program of Example 9-11 is described in detail below. Each explanation below is inserted as a comment line in the program. For details on the VISA functions, refer to the I/O Library Suite manual.

Configuration	Define Variables and initialize the program.
Open Instrument	Assigns a USB address or GPIB address to the I/O bus.
Setup Start	Presets the E4981A. Sets the trigger conditions of the E4981A.
Measurement	Reads the measurement data using the :READ? command, calls the function to convert data in Binary format to Ascii format, and displays the Primary and Secondary Measurement data.
Setup End	Closes the I/O bus.
ErrorProc	Displays an error number when an error occurs.
Binary Read	A subroutine to convert Binary format data to Ascii format.
ErrorCheck	A subroutine to check whether an error occurs when the IO Library Suite function is executed.
SelectMode	A subroutine to change assignment of the I/O bus and address setting according to the connection type i.e. USB or GPIB between the E4981A and a PC.

Example 9-11 Reading Measurement Results in Binary Format using :FETCh? Command

```
Sub Example11()
'====================================
' Configuration
'====================================

Dim defrm As Long 'Session to Default Resource Manager
Dim Agte4981a As Long 'Session to instrument
Dim Res() As Double
Dim Nop As Long, i As Integer, j As Integer, k As Integer
Const TimeOutTime = 30000
On Error GoTo ErrorHandler

'====================================
' Open Instrument
'====================================

ErrorCheck viOpenDefaultRM(defrm)
Call SelectMode(defrm, Agte4981a)
ErrorCheck viSetAttribute(Agte4981a, VI_ATTR_TMO_VALUE, TimeOutTime)

'====================================
' Setup Start
'====================================

ErrorCheck viVPrintf(Agte4981a, "SYST:PRES;*CLS" + vbLf, 0)
ErrorCheck viVPrintf(Agte4981a, "FORM REAL" + vbLf, 0)
ErrorCheck viVPrintf(Agte4981a, "APER:TIME 6" + vbLf, 0)
ErrorCheck viVPrintf(Agte4981a, "CALC1:COMP ON" + vbLf, 0)
ErrorCheck viVPrintf(Agte4981a, "INIT:CONT OFF" + vbLf, 0)
ErrorCheck viVPrintf(Agte4981a, "TRIG:SOUR BUS" + vbLf, 0)
ErrorCheck viVPrintf(Agte4981a, "INIT" + vbLf, 0)
ErrorCheck viVPrintf(Agte4981a, "TRIG" + vbLf, 0)
```

```
'===================================
' Meas Read
'===================================

ErrorCheck viVPrintf(Agte4981a, "TRIG:SOUR INT" + vbLf, 0)
ErrorCheck viVPrintf(Agte4981a, ":READ?" + vbLf, 0)

Call Scpi_read_binary_double_array(Agte4981a, Res, Nop)



Worksheets("Example11").Range("B5").Value = Val(Res(1))
Worksheets("Example11").Range("B6").Value = Val(Res(2))

'===================================
' Setup End
'===================================

Call viClose(defrm)
End

'===================================
' ErrorProc
'===================================

ErrorHandler:
' Display the error message
MsgBox "*** Error : " + Error$, MB_ICON_EXCLAMATION
End
End Sub
```

Reading Measurement Results Using Data Buffer

This application program reads measurement data in ASCII format using data buffer.

The program of Example 9-12 is described in detail below. Each explanation below is inserted as a comment line in the program. For details on the VISA functions, refer to the I/O Library Suite manual.

Configuration
Define Variables and initialize the program.

Open Instrument
Assigns a USB address or GPIB address to the I/O bus.

Setup Start
Presets the E4981A. Sets the Data buffer memory to 5 points. Sets the trigger conditions of the E4981A.

Measurement
Reads the measurement data using the :FETCh? command for 5 memory data points, separates the returned ASCII format character string data with commas, and displays the Primary and Secondary Measurement data for all memory points.

Setup End
Closes the I/O bus.

ErrorProc
Displays an error number when an error occurs.

ErrorCheck
A subroutine to check whether an error occurs when the IO Library Suite function is executed.

SelectMode
A subroutine to change assignment of the I/O bus and address setting according to the connection type i.e. USB or GPIB between the E4981A and a PC.

Example 9-12 Reading Measurement Results in Ascii Format using Data Buffer

```
Sub Example12()

'====================================

' Configuration

'====================================

Dim defrm As Long 'Session to Default Resource Manager

Dim Agte4981a As Long 'Session to instrument

Dim Result As String * 500

Dim PSData As Variant

Dim Res(5, 5) As Variant

Dim NoofMeas As Integer, i As Integer, j As Integer, k As Integer

Dim outEventType As Long, outEventContext As Long

Const TimeOutTime = 30000

On Error GoTo ErrorHandler


'====================================

' Open Instrument

'====================================


ErrorCheck viOpenDefaultRM(defrm)

Call SelectMode(defrm, Agte4981a)


ErrorCheck viSetAttribute(Agte4981a, VI_ATTR_TMO_VALUE, TimeOutTime)


'====================================

' Setup Start

'====================================


NoofMeas = 5

ErrorCheck viVPrintf(Agte4981a, "SYST:PRES;*CLS" + vbLf, 0)

ErrorCheck viVPrintf(Agte4981a, "FORM ASC" + vbLf, 0)


ErrorCheck viVPrintf(Agte4981a, "DATA:POIN:BUF3 " + CStr(NoofMeas) + vbLf,
0)

ErrorCheck viVPrintf(Agte4981a, ":DATA:FEED:CONT:BUF3 ALW" + vbLf, 0)

ErrorCheck viVPrintf(Agte4981a, "APER:TIME 6" + vbLf, 0)
ErrorCheck viVPrintf(Agte4981a, "INIT:CONT ON" + vbLf, 0)

ErrorCheck viVPrintf(Agte4981a, "TRIG:SOUR BUS" + vbLf, 0)


'===================================

' Measurement

'===================================


For i = 1 To NoofMeas


ErrorCheck viVPrintf(Agte4981a, "TRIG" + vbLf, 0)


ErrorCheck viVPrintf(Agte4981a, "FETCH?" + vbLf, 0)

ErrorCheck viVScanf(Agte4981a, "%t", Result)


PSData = Split(Result, ",")


Res(i, 1) = Val(PSData(1))

Res(i, 2) = Val(PSData(2))

Next i


Worksheets("Example12").Range("B5").Value = Res(1, 1)

Worksheets("Example12").Range("B6").Value = Res(1, 2)


Worksheets("Example12").Range("B8").Value = Res(2, 1)

Worksheets("Example12").Range("B9").Value = Res(2, 2)


Worksheets("Example12").Range("B11").Value = Res(3, 1)

Worksheets("Example12").Range("B12").Value = Res(3, 2)


Worksheets("Example12").Range("B14").Value = Res(4, 1)

Worksheets("Example12").Range("B15").Value = Res(4, 2)


Worksheets("Example12").Range("B17").Value = Res(5, 1)

Worksheets("Example12").Range("B18").Value = Res(5, 2)
```

```
'===================================

' Setup End

'===================================`

Call viClose(defrm)

End

'===================================

' ErrorProc

'===================================`

ErrorHandler:

' Display the error message

MsgBox "*** Error : " + Error$, MB_ICON_EXCLAMATION

End

End Sub
```

Reading Measurement Signal Level Monitor (IMON & VMON) Results

This application program reads measurement data in ASCII format using data buffer.

The program of Example 9-13 is described in detail below. Each explanation below is inserted as a comment line in the program. For details on the VISA functions, refer to the I/O Library Suite manual.

Configuration
Define Variables and initialize the program.

Open Instrument
Assigns a USB address or GPIB address to the I/O bus.

Setup Start
Presets the E4981A. Sets the trigger conditions of the E4981A.

Measurement
Reads the measurement signal level monitor values (IMON and VMON) and then display the values.

Setup End
Closes the I/O bus.

ErrorProc
Displays an error number when an error occurs.

ErrorCheck
A subroutine to check whether an error occurs when the IO Library Suite function is executed.

SelectMode
A subroutine to change assignment of the I/O bus and address setting according to the connection type i.e. USB or GPIB between the E4981A and a PC.

Example 9-13 Reading Measurement Signal Level Monitor (IMON & VMON) Results

```
Sub Example13()
'===================================
' Configuration
'===================================

Dim defrm As Long 'Session to Default Resource Manager
Dim Agte4981a As Long 'Session to instrument
Dim Imon1 As String * 500, Vmon1 As String * 500
Dim Finish_pros As String * 1
Const TimeOutTime = 30000
On Error GoTo ErrorHandler

'===================================
' Open Instrument
'===================================

ErrorCheck viOpenDefaultRM(defrm)
Call SelectMode(defrm, Agte4981a)

ErrorCheck viSetAttribute(Agte4981a, VI_ATTR_TMO_VALUE, TimeOutTime)

'===================================
' Setup Start
'===================================

ErrorCheck viVPrintf(Agte4981a, "SYST:PRES;*CLS" + vbLf, 0)
ErrorCheck viVPrintf(Agte4981a, "INIT:CONT ON" + vbLf, 0)

ErrorCheck viVPrintf(Agte4981a, "TRIG:SOUR BUS" + vbLf, 0)

ErrorCheck viVPrintf(Agte4981a, "STAT:OPER:ENAB 16" + vbLf, 0)
ErrorCheck viVPrintf(Agte4981a, "*SRE 128" + vbLf, 0)
ErrorCheck viVPrintf(Agte4981a, "*CLS" + vbLf, 0)
ErrorCheck viVPrintf(Agte4981a, "TRIG" + vbLf, 0)
```

```
'===================================
' Measurement
'===================================


While Finish_pros <> "1" ' Loop till processing is finished. *OPC? retruns 1
when processing of all commands is finished.
ErrorCheck viVPrintf(Agte4981a, "*OPC?" + vbLf, 0)
ErrorCheck viVScanf(Agte4981a, "%t", Finish_pros)
Wend


ErrorCheck viVPrintf(Agte4981a, "Data? IMON" + vbLf, 0)
ErrorCheck viVScanf(Agte4981a, "%t", Imon1)


ErrorCheck viVPrintf(Agte4981a, "Data? VMON" + vbLf, 0)
ErrorCheck viVScanf(Agte4981a, "%t", Vmon1)



MsgBox ("Current Monitor Value: " & Val(Imon1) & vbNewLine & "Voltage
Monitor Value: " & Val(Vmon1)), vbInformation


'===================================
' Setup End
'===================================`
Call viClose(defrm)
End


'===================================
' ErrorProc
'===================================`
ErrorHandler:
' Display the error message
MsgBox "*** Error : " + Error$, MB_ICON_EXCLAMATION
End
End Sub
```

Sorting Measurement Results Using Comparator

This application program sorts the measurement results using the comparator function.

The program of Example 9-14 is described in detail below. Each explanation below is inserted as a comment line in the program. For details on the VISA functions, refer to the I/O Library Suite manual.

Configuration
Define Variables and initialize the program.

Open Instrument
Assigns a USB address or GPIB address to the I/O bus.

Setup Start
Presets the E4981A. Sets the comparator limit values for Bin1,Bin2 and Bin3 of the E4981A. Sets the trigger conditions of the E4981A.

Measurement
Enables Bin 1 to 3. Disables Bin 4 to 9. Reads measurement values using *TRG command fifty times. Displays the bin values.

Setup End
Closes the I/O bus.

ErrorProc
Displays an error number when an error occurs.

ErrorCheck
A subroutine to check whether an error occurs when the IO Library Suite function is executed.

SelectMode
A subroutine to change assignment of the I/O bus and address setting according to the connection type i.e. USB or GPIB between the E4981A and a PC.

Example 9-14 Sorting Measurement Results Using Comparator

```
Sub Example14()
'===================================
' Configuration
'===================================

Dim defrm As Long 'Session to Default Resource Manager
Dim Agte4981a As Long 'Session to instrument
Dim TRG_Result As String * 500
Dim BIN_Result As String * 500
Dim OVLD_Result As String * 500
Dim Res As Variant
Dim Finish_pros As String * 1
Dim Nop As Integer, i As Integer
Const TimeOutTime = 30000
On Error GoTo ErrorHandler

'===================================
' Open Instrument
'===================================

ErrorCheck viOpenDefaultRM(defrm)
Call SelectMode(defrm, Agte4981a)
ErrorCheck viSetAttribute(Agte4981a, VI_ATTR_TMO_VALUE, TimeOutTime)

'===================================
' Setup Start
'===================================

   ErrorCheck viVPrintf(Agte4981a, "SYST:PRES;*CLS" + vbLf, 0) ' Preset the
E4981A

    ErrorCheck viVPrintf(Agte4981a, ":CALC1:FORM CP" + vbLf, 0)
    ErrorCheck viVPrintf(Agte4981a, ":CALC2:FORM D" + vbLf, 0)
    ErrorCheck viVPrintf(Agte4981a, ":SOUR:FREQ 1E3" + vbLf, 0)
    ErrorCheck viVPrintf(Agte4981a, ":CALC1:COMP:MODE PCNT" + vbLf, 0)

    ErrorCheck viVPrintf(Agte4981a, ":CALC1:COMP:PRIM:NOM 1E-9" + vbLf, 0)


   ErrorCheck viVPrintf(Agte4981a, ":CALC1:COMP:PRIM:BIN1 -1.0,1.0" + vbLf,
0)

   ErrorCheck viVPrintf(Agte4981a, ":CALC1:COMP:PRIM:BIN2 -2.0,2.0" + vbLf,
0)

   ErrorCheck viVPrintf(Agte4981a, ":CALC1:COMP:PRIM:BIN3 -3.0,3.0" + vbLf,
0)


    ErrorCheck viVPrintf(Agte4981a, ":CALC1:COMP:SEC:LIM 0, 0.1" + vbLf, 0)

    ErrorCheck viVPrintf(Agte4981a, ":CALC1:COMP:SEC:STAT ON" + vbLf, 0)


    ErrorCheck viVPrintf(Agte4981a, ":CALC1:COMP:AUXB ON" + vbLf, 0)

    ErrorCheck viVPrintf(Agte4981a, ":CREJ ON" + vbLf, 0)

    ErrorCheck viVPrintf(Agte4981a, ":CREJ:LIM 10.0" + vbLf, 0)


    ErrorCheck viVPrintf(Agte4981a, ":CALC1:COMP ON" + vbLf, 0)


    ErrorCheck viVPrintf(Agte4981a, "TRIG:SOUR BUS" + vbLf, 0) ' Trigger
Setting


    ErrorCheck viVPrintf(Agte4981a, "CALC1:COMP:COUN ON" + vbLf, 0)

    ErrorCheck viVPrintf(Agte4981a, "CALC1:COMP:COUN:CLE" + vbLf, 0)


'====================================

' Measurement

'====================================


    For i = 1 To 3

    ErrorCheck viVPrintf(Agte4981a, ":CALC1:COMP:PRIM:BIN" & i & ":STAT ON"
+ vbLf, 0)

    Next i


    For i = 4 To 9

   ErrorCheck viVPrintf(Agte4981a, ":CALC1:COMP:PRIM:BIN" & i & ":STAT OFF"
+ vbLf, 0)

    Next i
```

```
        While Finish_pros <> "1" ' Loop till processing is finished. *OPC?
retruns 1 when processing of all commands is finished.

    ErrorCheck viVPrintf(Agte4981a, "*OPC?" + vbLf, 0)

    ErrorCheck viVScanf(Agte4981a, "%t", Finish_pros)

    Wend


    For i = 1 To 50

    ErrorCheck viVPrintf(Agte4981a, "*TRG" + vbLf, 0)

    ErrorCheck viVScanf(Agte4981a, "%t", TRG_Result)


    Res = Split(TRG_Result, ",")


    Worksheets("Example14").Range("A" & (i + 4)).Value = Val(Res(0))

    Worksheets("Example14").Range("B" & (i + 4)).Value = Val(Res(1))

    Worksheets("Example14").Range("C" & (i + 4)).Value = Val(Res(2))

    Worksheets("Example14").Range("D" & (i + 4)).Value = Val(Res(3))

    Next i


    ErrorCheck viVPrintf(Agte4981a, "CALC1:COMP:COUN:DATA?" + vbLf, 0)

    ErrorCheck viVScanf(Agte4981a, "%t", BIN_Result)


    ErrorCheck viVPrintf(Agte4981a, "CALC1:COMP:COUN:OVLD?" + vbLf, 0)

    ErrorCheck viVScanf(Agte4981a, "%t", OVLD_Result)


    Res = Split(BIN_Result, ",")


    For i = 1 To 9


    Worksheets("Example14").Range("K" & (i + 4)).Value = "BIN " & i

    Worksheets("Example14").Range("L" & (i + 4)).Value = Val(Res(i))


    Next i


    Worksheets("Example14").Range("K14").Value = "OUT OF BINS "

    Worksheets("Example14").Range("L14").Value = Val(Res(0))
    Worksheets("Example14").Range("K15").Value = "AUX BINS "
    Worksheets("Example14").Range("L15").Value = Val(Res(10))

    Worksheets("Example14").Range("K16").Value = "OVLD "
    Worksheets("Example14").Range("L16").Value = Val(OVLD_Result)

'====================================
' Setup End
'====================================

Call viClose(defrm)

End

'====================================
' ErrorProc
'====================================

ErrorHandler:
' Display the error message
MsgBox "*** Error : " + Error$, MB_ICON_EXCLAMATION
End

End Sub
```

Capacitor Measurement Using Test Fixture

This application program perfrom OPEN/SHORT/LOAD correction and measures a capacitor using a test fixture in the E4981A.

The program of Example 9-15 is described in detail below. Each explanation below is inserted as a comment line in the program. For details on the VISA functions, refer to the I/O Library Suite manual.

Configuration
Define Variables and initialize the program.

Open Instrument
Assigns a USB address or GPIB address to the I/O bus.

Setup Start
Presets the E4981A. Sets Primary and Secondary parameter format. Inputs measurement frequency and sets E4981A. Sets trigger settings.

Measurement
Performs OPEN/SHORT/LOAD correction. Reads the measurement results using the *TRG command. Displays the Primary and Secondary measurement results.

Setup End
Closes the I/O bus.

ErrorProc
Displays an error number when an error occurs.

ErrorCheck
A subroutine to check whether an error occurs when the IO Library Suite function is executed.

SelectMode
A subroutine to change assignment of the I/O bus and address setting according to the connection type i.e. USB or GPIB between the E4981A and a PC.

FnComp
A function which performs OPEN/SHORT/LOAD correction and returns 1 on success and 0 on failure.

Example 9-15 Capacitor Measurement Using Test Fixture

```
Sub Example15()

'====================================

' Configuration

'====================================


Dim defrm As Long 'Session to Default Resource Manager

Dim Agte4981a As Long 'Session to instrument

Dim Result As String * 500

Dim Res As Variant

Dim Nop As Integer, i As Integer

Dim Corr_File_Name As String

Dim Freq As String

Const TimeOutTime = 30000

On Error GoTo ErrorHandler


'====================================

' Open Instrument

'====================================


ErrorCheck viOpenDefaultRM(defrm)

Call SelectMode(defrm, Agte4981a)

ErrorCheck viSetAttribute(Agte4981a, VI_ATTR_TMO_VALUE, TimeOutTime)


'====================================

' Setup Start

'====================================


MsgBox "Connect the Test Fixture and then press OK", vbOKOnly


ErrorCheck viVPrintf(Agte4981a, "SYST:PRES;*CLS" + vbLf, 0) ' Preset E4981A


Freq = InputBox("Enter the frequency (120/1E3/1E6) you want to measure")


ErrorCheck viVPrintf(Agte4981a, ":CALC1:FORM CS" + vbLf, 0) ' Set Primary
parameter to Cs
```

```
ErrorCheck viVPrintf(Agte4981a, ":CALC2:FORM Q" + vbLf, 0) ' Set Secondary
parameter to Q

ErrorCheck viVPrintf(Agte4981a, ":SOUR:FREQ " & Freq + vbLf, 0) ' Set source
frequency

ErrorCheck viVPrintf(Agte4981a, ":SOUR:VOLT 0.5" + vbLf, 0) ' Set source
voltage as 0.5V


ErrorCheck viVPrintf(Agte4981a, ":FORM ASC" + vbLf, 0) ' Set format to Ascii

ErrorCheck viVPrintf(Agte4981a, ":INIT:CONT ON" + vbLf, 0)

ErrorCheck viVPrintf(Agte4981a, ":TRIG:SOUR BUS" + vbLf, 0) ' Set Trigger
source as GPIB


'====================================

' Measurement

'====================================


Corr_Result = FnComp("OPEN", 0.00002, Freq)


    If Corr_Result <> 0 Then

        MsgBox "OPEN compensation is not successful."

        'End

    End If


    Corr_Result = FnComp("SHORT", 20, Freq)


    If Corr_Result <> 0 Then

        MsgBox "SHORT compensation is not successful."

        'End

    End If


    Corr_Result = FnComp("LOAD", 0.2, Freq)


    If Corr_Result <> 0 Then

        MsgBox "LOAD compensation is not successful."

        'End

    End If
```

```
ErrorCheck viVPrintf(Agte4981a, "*TRG" + vbLf, 0)

ErrorCheck viVScanf(Agte4981a, "%t", Result)


Res = Split(Result, ",")


MsgBox "Value of Primary parameter (Cs) is " & Val(Res(1)) & " and value of
Secondary Parameter (Q) is " & Val(Res(2))


'====================================

' Setup End

'====================================


Call viClose(defrm)


End


'====================================

' ErrorProc

'====================================


ErrorHandler:

' Display the error message

MsgBox "*** Error : " + Error$, MB_ICON_EXCLAMATION

End


End Sub
```

Measuring Results with Contact Check Enabled

This application program measures Primary and Secondary measurement parameters of the E4981A with contact check enabled.

The program of Example 9-16 is described in detail below. Each explanation below is inserted as a comment line in the program. For details on the VISA functions, refer to the I/O Library Suite manual.

Configuration — Define Variables and initialize the program.

Open Instrument — Assigns a USB address or GPIB address to the I/O bus.

Setup Start — Presets the E4981A. Sets trigger, measurement frequency, voltage and range settings.

Measurement — Sets Contact Check Parameters. Executes measurement using *TRG. Separates the returned ASCII format character string data with commas, and displays the Primary and Secondary Measurement data.

Setup End — Closes the I/O bus.

ErrorProc — Displays an error number when an error occurs.

ErrorCheck — A subroutine to check whether an error occurs when the IO Library Suite function is executed.

SelectMode — A subroutine to change assignment of the I/O bus and address setting according to the connection type i.e. USB or GPIB between the E4981A and a PC.

Example 9-16 Measuring Results with Contact Check Enabled

```
Sub Example17()
'====================================
' Configuration
'====================================

Dim defrm As Long
Dim Agte4981a As Long
Dim Result As String * 100
Dim Res As Variant
Dim CCParamHi As Double
Dim CCParamLo As Double
Const TimeOutTime = 30000
On Error GoTo ErrorHandler

'====================================
' Open Instrument
'====================================

ErrorCheck viOpenDefaultRM(defrm)
Call SelectMode(defrm, Agte4981a)
ErrorCheck viSetAttribute(Agte4981a, VI_ATTR_TMO_VALUE, TimeOutTime)

'====================================
' Setup Start
'====================================

ErrorCheck viVPrintf(Agte4981a, "SYST:PRES;*CLS" + vbLf, 0) ' Preset E4981A

ErrorCheck viVPrintf(Agte4981a, ":INIT:CONT ON" + vbLf, 0)
ErrorCheck viVPrintf(Agte4981a, ":TRIG:SOUR BUS" + vbLf, 0)

' Frequency should be 120 or 1000.  The contact check is not available 1MHz.
ErrorCheck viVPrintf(Agte4981a, ":SOUR:FREQ 1000" + vbLf, 0)
```

```
' Write the SCPI commands for your desire setting. This should be the same
as one in the CC_ReadParameter

ErrorCheck viVPrintf(Agte4981a, ":SOUR:VOLT 1" + vbLf, 0)

ErrorCheck viVPrintf(Agte4981a, ":RANG 1E-6" + vbLf, 0)


CCParamHi = Worksheets("Example17").Range("D2").Value

CCParamLo = Worksheets("Example17").Range("D3").Value


'====================================

' Measurement

'====================================


' Setup Contact Check Threshold parameters

ErrorCheck viVPrintf(Agte4981a, ":CONT1:VER ON" + vbLf, 0) 'Contact Check
Function On

ErrorCheck viVPrintf(Agte4981a, ":CONT1:VER:THR1 " + Str(CCParamHi) + vbLf,
0) ' Threshold of Contact Check parameter for Hp/Hc (TH1)

ErrorCheck viVPrintf(Agte4981a, ":CONT1:VER:THR2 " + Str(CCParamLo) + vbLf,
0) ' Threshold of Contact Check parameter for Lp/Lc (TH2)

'

'

' Execute Measurement

ErrorCheck viVPrintf(Agte4981a, "*TRG" + vbLf, 0)

ErrorCheck viVScanf(Agte4981a, "%t", Result)

'


Res = Split(Result, ",")

Worksheets("Example17").Range("B6").Value = Val(Res(0))

Worksheets("Example17").Range("C6").Value = Val(Res(1))

Worksheets("Example17").Range("D6").Value = Val(Res(2))
```

```
'===================================
' Setup End
'===================================

ErrorCheck viClose(Agte4981a)
ErrorCheck viClose(defrm)

End

'===================================
' ErrorProc
'===================================

ErrorHandler:
' Display the error message
MsgBox "*** Error : " + Error$, MB_ICON_EXCLAMATION
End

End Sub
```

Detecting Error Occurrence using SRQ (HT Basic)

Example 9-17 shows a sample program to detect the occurrence of an error using SRQ. This program is stored on the sample program disk under the filename “srq_err.bas”.

This program sets up SRQ, intentionally sends a nonexistent command for the E4981A to generate an error, and performs error handling. In the error handling, the program checks which error has occurred, displays the error number and error message, and displays the “PROGRAM INTERRUPT” message.

Line 40 Sets the GPIB address.

Lines 60 to 70 Enables bit 2, bit 3, bit 4, and bit 5 of the standard event status register and sets bit 5 of the service request enable register to 1.

Lines 80 to 100 Clears the status byte register, standard event status register, and error queue.

Lines 120 to 130 Sets the branch destination of the SRQ interrupt and enables SRQ interrupt.

Lines 140 to 190 Selects the primary parameter and secondary parameter. However, because the command for the secondary parameter is wrong, an error occurs.

Lines 220 to 230 Processing when an error occurs. Reads out the error number and error message of the error that has occurred.

Lines 240 to 260 Displays the “Error occurred!!” message, error number, error message, and “PROGRAM INTERRUPT!!” message.

Line 280 Displays the “PROGRAM DONE” Message. Notice that this message is not displayed unless you correct and execute the secondary parameter selection command.

Example 9-17 Error occurrence detection using SRQ (srq_err.bas)

```
10     DIM Buff$[9],Err_mes$[50]
20     INTEGER Err_no
30     !
40     ASSIGN @Agt4981a TO 717
50     !
60     OUTPUT @Agt4981a;"*ESE 60"
70     OUTPUT @Agt4981a;"*SRE 32"
80     OUTPUT @Agt4981a;"*CLS"
90     OUTPUT @Agt4981a;"*OPC?"
100    ENTER @Agt4981a;Buff$
110    !
120    ON INTR 7 GOTO Err_proc
130    ENABLE INTR 7;2
140    OUTPUT @Agt4981a;":CALC1:FORM CS"
150    PRINT "Primary   Parameter Setting: CS"
160    OUTPUT @Agt4981a;":CALC2:FOR Q"
170    PRINT "Secondary Parameter Setting: Q"
180    OUTPUT @Agt4981a;"*OPC?"
190    ENTER @Agt4981a;Buff$
200    GOTO Skip_err_proc
210 Err_proc: OFF INTR 7
220    OUTPUT @Agt4981a;";:SYST:ERR?"
230    ENTER @Agt4981a;Err_no,Err_mes$
240    PRINT "Error occurred!!"
250    PRINT "  No:";Err_no,"Description: "&Err_mes$
260    PRINT "PROGRAM INTERRUPT!!"
270    GOTO Prog_end
280 Skip_err_proc: PRINT "PROGRAM DONE."
290 Prog_end: END
```

10 Command Reference

This chapter provides the SCPI command reference for the Keysight E4981A. Each command is fully described and ordered alphabetically based on its abbreviated name format. Use the index to look up a SCPI command by its full syntax. To find a command according to its function, refer to the "SCPI Command Table" on page 307.

Notational Conventions in this Command Reference

This section describes the rules to read the description of the commands in this chapter.

Description

A part with the heading "Description" describes how to use the command or the operation when executed.

Syntax

A part with the heading "Syntax" describes the syntax to send a command from the external controller to the E4981A. A syntax con...sists of a command part and a parameter part. A space is used to separate the command part and the parameter part.

If there are several parameters, a comma is used instead as the separator between adjacent parameters. Three periods (...) between commas indicate that parameters in that part are omitted. For example, <numeric 1>,..., <numeric 4> indicates that four parameters,<numeric 1>, <numeric 2>, <numeric 3>, and <numeric 4>, are required.

String-type parameters, <string>, <string 1>, and so on, must be enclosed in double quotation marks ("). Furthermore, <block> indicates block-format data.

You can omit the lowercase letters in syntax. For example, ":CALibration:CABLe[:LENGth] " can be shortened to "CAL:CABL."

The definitions of symbols used in the syntax are as follows.

<>	Characters enclosed in this pair of symbols are necessary parameters when sending a command.
[]	A part enclosed in these parentheses can be omitted.
{}	A part enclosed in these parentheses indicates that you must select one of the items in this part. Individual items are separated by a vertical bar (|).

For example, ":APER SHOR", ":SENS:FIMP:APER:MODE LONG," and so on are valid for the syntax given below.

Syntax [:SENSe][:FIMPedance]:APERture[:MODE] {SHORt|MEDium|LONG}

Parameters

A part with heading "Parameters" describes parameters necessary for sending the command. When a parameter is a value type or a string type enclosed with <>, its description, allowable setup range, Initial (Preset) value, and so on are given; when a parameter is a selection type enclosed by {}, the description of each selection item is given.

Equivalent key

A part with the heading "Equivalent key" shows the operational procedure for the front panel keys. An equivalent key has the same effect as a certain command.

[Key]	Indicates that you press the key named Key.
[Key] – ITEM	Indicates a series of key operations in which you press **[Key]**, select the item called ITEM (softkey or field name) on the displayed menu using the cursor keys, and then press the softkey.

IEEE Common Commands

This section describes the IEEE common commands.

*CLS

Syntax

*CLS

Description

Clears the following: (No query)

- Error Queue
- Status Byte Register
- Standard Event Status Register
- Operation Status Event Register
- Questionable Status Event Register

Equivalent key sequence

No equivalent key is available on the front panel.

*ESE

Syntax

*ESE <numeric>

*ESE?

Description

Sets the value of the Standard Event Status Enable Register.

Parameters

	<numeric>
Description	Setup value of the register
Range	0 to 255
Initial value	0
Resolution	1

Query response

{numeric}<newline><^END>

Related commands

*SRE on page 162

Equivalent key sequence

No equivalent key is available on the front panel.

*ESR?

Syntax	*ESR?
Description	Reads out the value of the Standard Event Status Register. Executing this command clears the register value. (Query only)
Query response	{numeric}<newline><^END>
Equivalent key sequence	No equivalent key is available on the front panel.

*IDN?

Syntax	*IDN?
Description	Reads out the product information (model number, serial number, and firmware version number) of the E4981A. (Query only)
Query response	{string 1},{string 2},{string 3}, {string 4} <newline><^END>

Readout data is as follows:

{string 1}	Keysight Technologies.
{string 2}	Model number. E4981A is always read out.
{string 3}	10-digit serial number (example: JP1KH00101).
{string 4}	Firmware version number (example: A.01.00).

Equivalent key sequence	**[System]**

*LRN?

Syntax	*LRN?
Description	Returns all the necessary commands to set the E4981A at its present state. The response can later be sent back to the E4981A to place it in this state. (Query Only)
Equivalent key sequence	No equivalent key is available on the front panel.

*OPC

Syntax *OPC

Description Sets the OPC bit (bit 0) of the Standard Event Status Register at the completion of all pending operations. (No query)

Equivalent key sequence No equivalent key is available on the front panel.

*OPC?

Syntax *OPC?

Description 1 is read out at the completion of all pending operations. (Query only)

Query response {1}<newline><^END>

Equivalent key sequence No equivalent key is available on the front panel.

*OPT?

Syntax *OPT?

Description Reads out the identification number of an option installed in the E4981A. (Query only)

Query response {numeric}<newline><^END>

NOTE Option 001 or 002 is available for E4981A.

Equivalent key sequence **[System]**

*RCL

Syntax *RCL <numeric>

Description Recalls the instrument setups saved in the register of the specified number on Flash (internal) memory or USB pen drive (external memory). For details of recalled instrument setups, refer to *Operation Manual, Appendix C "Initial Setting"*.(No query)

Parameters

	<numeric>
Description	Specified number
Range	0 to 9 (Internal Flash Memory)
	10 to 19 (External USB Memory)
Resolution	1

If the specified parameter is out of the allowable setup range, an error occurs.

Related commands `*SAV` on page 161

Equivalent key sequence [**Save/Recall**] - No {1-9} - RECALL

*RST

Syntax *RST

Description Resets the instrument to the preset state.

The preset state is different from that when resetting is done using the `:SYSTem:PRESet` command. For details, refer to *Operation Manual, Appendix C "Initial Setting"* (No query)

Related commands `:SYSTem:PRESet` on page 284

`:INITiate:CONTinuous` on page 227

Equivalent key sequence No equivalent key is available on the front panel.

*SAV

Syntax *SAV <numeric>

Description Saves the instrument setups to the register of the specified number on Flash (internal) memory or USB pen drive (external memory). For details of saved instrument setups, refer to *Operation Manual, Appendix C “Initial Setting”* (No query)

Parameters

	<numeric>
Description	Specified number
Range	0 to 9 (Internal Flash Memory)
	10 to 19 (External USB Memory)
Resolution	1

If the specified parameter is out of the allowable setup range, an error occurs.

Related commands `*RCL` on page 160

Equivalent key sequence [**Save/Recall**] - No {1-9} - SAVE

*SRE

Syntax

*SRE <numeric>

*SRE?

Description

Sets the value of the Service Request Enable Register.

Parameters

	<numeric>
Description	Setup value of the register
Range	0 to 255
Initial value	0
Resolution	1

If the specified parameter is out of the allowable setup range, “Out of range” error is generated. Note that bit 6 cannot be set to 1.

Query response

{numeric}<newline><^END>

Related commands

`*ESE` on page 157

`:STATus:OPERation:ENABle` on page 270

Equivalent key sequence

No equivalent key is available on the front panel.

*STB?

Syntax

*STB?

Description

Reads out the value of the Status Byte Register. (Query only)

Query response

{numeric}<newline><^END>

Equivalent key sequence

No equivalent key is available on the front panel.

*TRG

Syntax

*TRG

Description

If the trigger source is set to BUS (set with the `:TRIGger[:SEQ1]:SOURce` command), triggers the E4981A as it waits for a trigger and reads out the measured data after the measurement is completed.

The transfer format of data read out with this command conforms to the setup of the `:FORMat[:DATA]` command.

NOTE

Although this command does not have "?", a query response is given.

Query response

It is the same as that of the `:FETCh?` command. For details, refer to the description of `:FETCh?`.

Related commands

`:FETCh?` on page 220

`:TRIGger[:SEQ1]:SOURce` on page 305

`:FORMat[:DATA]` on page 225

`:READ?` on page 230

`:FORMat:STATus:EXTension` on page 224

Equivalent key sequence

[**Trigger**]

*TST?

Syntax

*TST?

Description

Does nothing. The self-test is not executed by this command in the case of the E4981A. Always returns 0. (Query Only)

Equivalent key sequence

No equivalent key is available on the front panel.

*WAI

Syntax

*WAI

Description

Waits until all commands sent before this command are executed. (No query)

Equivalent key sequence

No equivalent key is available on the front panel.

E4981A SCPI Commands

This section describes the SCPI commands of the E4981A.

:ABORt

Syntax

:ABORt

Description

Resets the trigger system and places the trigger sequence in the idle state.

If the trigger system is set to start up successively (ON is specified with the `:INITiate:CONTinuous` command), the trigger system starts up immediately after the transition to the idle state.

For details on the trigger system, refer to "Trigger system" on page 57. (No query)

Related commands

`:INITiate[:IMMediate]` on page 228

`:INITiate:CONTinuous` on page 227

Equivalent key sequence

No equivalent key is available on the front panel.

:CALibration:CABLe:CORRection:CLEar

Syntax :CALibration:CABLe:CORRection:CLEar <numeric>

Description Clears the cable correction data. (No query)

Parameters

	<numeric>
Description	Cable Length (in meters)
Range	1 or 2

Related commands

`:CALibration:CABLe:CORRection:COLLect[:ACQuire]:LOAD` on page 166

`:CALibration:CABLe:CORRection:COLLect[:ACQuire]:OPEN` on page 167

`:CALibration:CABLe:CORRection:COLLect[:ACQuire]:REFerence` on page 168

`:CALibration:CABLe:CORRection:SAVE` on page 169

`:CALibration:CABLe:CORRection:STATe?` on page 170

Equivalent key sequence [**System**] - CABLE CORR - 1m|2m - CLEAR

:CALibration:CABLe:CORRection:COLLect[:ACQuire]:LOAD

Syntax :CALibration:CABLe:CORRection:COLLect[:ACQuire]:LOAD <numeric>

Description Measures the cable correction LOAD. (No query)

Parameters

	<numeric>
Description	Cable Length (in meters)
Range	1 or 2

Related commands

`:CALibration:CABLe:CORRection:CLEar` on page 165

`:CALibration:CABLe:CORRection:COLLect[:ACQuire]:OPEN` on page 167

`:CALibration:CABLe:CORRection:COLLect[:ACQuire]:REFerence` on page 168

`:CALibration:CABLe:CORRection:SAVE` on page 169

`:CALibration:CABLe:CORRection:STATe?` on page 170

Equivalent key sequence [**System**] - CABLE CORR - 1m|2m - MEAS LOAD

:CALibration:CABLe:CORRection:COLLect[:ACQuire]:OPEN

Syntax

:CALibration:CABLe:CORRection:COLLect[:ACQuire]:OPEN <numeric>

Description

Measures the cable correction OPEN. (No query)

Parameters

	<numeric>
Description	Cable Length (in meters)
Range	1 or 2

Related commands

`:CALibration:CABLe:CORRection:CLEar` on page 165

`:CALibration:CABLe:CORRection:COLLect[:ACQuire]:LOAD` on page 166

`:CALibration:CABLe:CORRection:COLLect[:ACQuire]:REFerence` on page 168

`:CALibration:CABLe:CORRection:SAVE` on page 169

`:CALibration:CABLe:CORRection:STATe?` on page 170

Equivalent key sequence

[**System**] - CABLE CORR - 1m|2m - MEAS OPEN

:CALibration:CABLe:CORRection:COLLect[:ACQuire]:REFerence

Syntax :CALibration:CABLe:CORRection:COLLect[:ACQuire]:REFerence <numeric>

Description Measures the cable correction reference. (No query)

Parameters

	<numeric>
Description	Cable Length (in meters)
Range	1 or 2

Related commands

`:CALibration:CABLe:CORRection:CLEar` on page 165

`:CALibration:CABLe:CORRection:COLLect[:ACQuire]:LOAD` on page 166

`:CALibration:CABLe:CORRection:COLLect[:ACQuire]:OPEN` on page 167

`:CALibration:CABLe:CORRection:SAVE` on page 169

`:CALibration:CABLe:CORRection:STATe?` on page 170

Equivalent key sequence [**System**] - CABLE CORR - 1m|2m - MEAS REF

:CALibration:CABLe:CORRection:SAVE

Syntax :CALibration:CABLe:CORRection:SAVE <numeric>

Description Saves the cable correction data. (No query)

Parameters

	<numeric>
Description	Cable Length (in meters)
Range	1 or 2

Related commands

`:CALibration:CABLe:CORRection:CLEar` on page 165

`:CALibration:CABLe:CORRection:COLLect[:ACQuire]:LOAD` on page 166

`:CALibration:CABLe:CORRection:COLLect[:ACQuire]:OPEN` on page 167

`:CALibration:CABLe:CORRection:COLLect[:ACQuire]:REFerence` on page 168

`:CALibration:CABLe:CORRection:STATe?` on page 170

Equivalent key sequence [**System**] - CABLE CORR - 1m|2m - SAVE

:CALibration:CABLe:CORRection:STATe?

Syntax :CALibration:CABLe:CORRection:STATe? <numeric>

Description Gets cable correction ON/OFF. (Query Only)

Parameters

	<numeric>
Description	Cable Length (in meters)
Range	1or 2

NOTE The value of this command is not changed with `*RST` and `:SYSTem:PRESet` and is only changed with Factory Default reset (available only through Front Panel).

Query Response {1|0}<newline><^END>

Related commands `:CALibration:CABLe:CORRection:CLEar` on page 165

`:CALibration:CABLe:CORRection:COLLect[:ACQuire]:LOAD` on page 166

`:CALibration:CABLe:CORRection:COLLect[:ACQuire]:OPEN` on page 167

`:CALibration:CABLe:CORRection:COLLect[:ACQuire]:REFerence` on page 168

`:CALibration:CABLe:CORRection:SAVE` on page 169

Equivalent key sequence No equivalent key is available on the front panel.

:CALibration:CABLe[:LENGth]

Syntax

```
:CALibration:CABLe[:LENGth] <numeric>
:CALibration:CABLe[:LENGth]?
```

Description

Sets the measurement cable length.

Parameters

	<numeric>
Description	Desired cable length
Range	0 to 2
Initial value	0
Unit	m (meter)
Resolution	1

If the specified parameter is out of the allowable setup range, the minimum value (if the lower limit of the range is not reached) or the maximum value (if the upper limit of the range is exceeded) is set.

MAX or MIN can be used to specify the parameters.

Query response

{numeric}<newline><^END>

Equivalent key sequence

[**Meas Setup**] - CORRECTION - CABLE

:CALCulate1:COMParator:AUXBin

Syntax

```
:CALCulate1:COMParator:AUXBin {ON|OFF|1|0}
:CALCulate1:COMParator:AUXBin?
```

Description

Turns ON/OFF the AUX_BIN function for sorting of the comparator function.

Depending on the ON/OFF state of the AUX_BIN function, the following difference occurs in the sorting result when the measurement result of the secondary parameter exceeds the limit range.

ON:	Sorted into AUX_BIN if the measurement result of the primary parameter is within the limit range. Otherwise, sorted into OUT_OF_BINS.
OFF:	Always sorted into OUT_OF_BINS.

Parameters

	Description
ON or 1	Turns ON the AUX_BIN function.
OFF or 0 (initial value)	Turns OFF the AUX_BIN function.

Query response

{1|0}<newline><^END>

Equivalent key sequence

[**Meas Setup**] - LIMIT TABLE - AUX

:CALCulate1:COMParator:BEEPer:CONDition

Syntax

:CALCulate1:COMParator:BEEPer:CONDition {FAIL|PASS}

:CALCulate1:COMParator:BEEPer:CONDition?

Description

Sets the condition for producing a beep sound: when sorting with the comparator fails (sorts into any BIN other than BIN1 to BIN9) or passes (sorts into BIN1 to BIN9).

Parameters

	Description
FAIL (initial value)	Instructs the instrument to beep when the sorting fails.
PASS	Instructs the instrument to beep when the sorting pass.

Query response

{FAIL|PASS}<newline><^END>

Related commands

`:SYSTem:BEEPer:STATe` on page 273

`:CALCulate1:COMParator:BEEPer[:STATe]` on page 174

Equivalent key sequence

[**Meas Setup**] - LIMIT TABLE - BEEP

:CALCulate1:COMParator:BEEPer[:STATe]

Syntax

:CALCulate1:COMParator:BEEPer[:STATe] {ON|OFF|1|0}

:CALCulate1:COMParator:BEEPer[:STATe]?

Description

Turns ON/OFF the beep output.

If you turn off the beep output, the beep sound is not produced, regardless of the sorting result of the comparator.

This command has the same function as the `:SYSTem:BEEPer:STATe` command.

Parameters

	Description
ON or 1 (initial value)	Turns ON the beep output.
OFF or 0	Turns OFF the beep output.

Query response

{1|0}<newline><^END>

Related commands

`:SYSTem:BEEPer:STATe` on page 273

`:CALCulate1:COMParator:BEEPer:CONDition` on page 173

Equivalent key sequence

[**Meas Setup**] - LIMIT TABLE - BEEP

:CALCulate1:COMParator:CLEar

Syntax :CALCulate1:COMParator:CLEar

Description Clears the ON/OFF state and range of every limit range (BIN1 to BIN9, the limit range for the secondary parameter), the limit range designation method, and the reference value for tolerance mode. (No query)

Equivalent key sequence [**Meas Setup**] - LIMIT TABLE - BIN - CLEAR TABLE

:CALCulate1:COMParator:COUNt:CLEar

Syntax :CALCulate1:COMParator:COUNt:CLEar

Description Clears the count value of each BIN (resets it to 0) for the BIN counter function of the comparator function. (No query)

Equivalent key sequence [**Display Format**] - BIN COUNT - COUNT - RESET COUNT

:CALCulate1:COMParator:COUNt:DATA?

Syntax

:CALCulate1:COMParator:COUNt:DATA?

Description

Reads out each count value of BIN1 to BIN9, OUT_OF_BINS, and AUX_BIN of the BIN counter function.

Reads out all BIN count values regardless of the ON/OFF state of each BIN (set with the `:CALCulate1:COMParator:PRIMary:BIN{1-9}:STATe` command). (Query only)

Parameters

	Description
{numeric 1}	The count value of BIN1.
{numeric 2}	The count value of BIN2.
{numeric 3}	The count value of BIN3.
{numeric 4}	The count value of BIN4.
{numeric 5}	The count value of BIN5.
{numeric 6}	The count value of BIN6.
{numeric 7}	The count value of BIN7.
{numeric 8}	The count value of BIN8.
{numeric 9}	The count value of BIN9.
{numeric 10}	The count value of OUT_OF_BINS.
{numeric 11}	The count value of AUX_BIN.

Query response

{numeric 1},...,{numeric 11}<newline><^END>

Related commands

`:CALCulate1:COMParator:COUNt:OVLD?` on page 179

`:CALCulate1:COMParator:COUNt[:STATe]` on page 180

`:CALCulate1:COMParator:COUNt:CLEar` on page 175

`:CALCulate1:COMParator:COUNt:MULTiple:DATA?` on page 177

`:CALCulate1:COMParator[:STATe]` on page 187

`:CALCulate1:COMParator:PRIMary:BIN{1-9}:STATe` on page 183

Equivalent key sequence

No equivalent key is available on the front panel.

:CALCulate1:COMParator:COUNt:MULTiple:DATA?

Syntax :CALCulate1:COMParator:COUNt:MULTiple:DATA?

Description Reads out each count value of BIN1 to BIN9, OUT_OF_BINS, and AUX_BIN of the selected channel.

When the multi-correction function is turned off, reads out normal count values (common value for all channels). In other words, this command has the same function as the `:CALCulate1:COMParator:COUNt:DATA?` command.

Reads out all BIN count values regardless of the ON/OFF state of each BIN (set with the `:CALCulate1:COMParator:PRIMary:BIN{1-9}:STATe` command). (Query only)

Parameters

	Description
{numeric 1}	The count value of BIN1 of selected channel.
{numeric 2}	The count value of BIN2 of selected channel.
{numeric 3}	The count value of BIN3 of selected channel.
{numeric 4}	The count value of BIN4 of selected channel.
{numeric 5}	The count value of BIN5 of selected channel.
{numeric 6}	The count value of BIN6 of selected channel.
{numeric 7}	The count value of BIN7 of selected channel.
{numeric 8}	The count value of BIN8 of selected channel.
{numeric 9}	The count value of BIN9 of selected channel.
{numeric 10}	The count value of OUT_OF_BINS of selected channel.
{numeric 11}	The count value of AUX_BIN of selected channel.

Query response {numeric 1},...,{numeric 11}<newline><^END>

Related commands

`:CALCulate1:COMParator:COUNt:MULTiple:OVLD?` on page 179

`[:SENSe]:CORRection:MULTiple[:STATe]` on page 245

`[:SENSe]:CORRection:MULTiple:CHANnel` on page 243

`:CALCulate1:COMParator:COUNt[:STATe]` on page 180

`:CALCulate1:COMParator:COUNt:CLEar` on page 175

`:CALCulate1:COMParator:COUNt:DATA?` on page 176

`:CALCulate1:COMParator[:STATe]` on page 187

`:CALCulate1:COMParator:PRIMary:BIN{1-9}:STATe` on page 183

Equivalent key sequence

No equivalent key is available on the front panel.

:CALCulate1:COMParator:COUNt:MULTiple:OVLD?

Syntax :CALCulate1:COMParator:COUNt:MULTiple:OVLD?

Description Reads out the overload count value of the selected channel.

When the multi-correction function is turned off, reads out normal count value (common value for all channels). In other words, this command has the same function as the `:CALCulate1:COMParator:COUNt:OVLD?` command. (Query only)

Query response {numeric}<newline><^END>

Related commands `:CALCulate1:COMParator:COUNt:MULTiple:DATA?` on page 177

`[:SENSe]:CORRection:MULTiple[:STATe]` on page 245

`:CALCulate1:COMParator:COUNt[:STATe]` on page 180

`:CALCulate1:COMParator:COUNt:CLEar` on page 175

`:CALCulate1:COMParator[:STATe]` on page 187

Equivalent key sequence No equivalent key is available on the front panel.

:CALCulate1:COMParator:COUNt:OVLD?

Syntax :CALCulate1:COMParator:COUNt:OVLD?

Description Reads out each overload count value. (Query only)

Query response {numeric}<newline><^END>

Related commands `:CALCulate1:COMParator:COUNt:DATA?` on page 176

`:CALCulate1:COMParator:COUNt[:STATe]` on page 180

`:CALCulate1:COMParator:COUNt:CLEar` on page 175

`:CALCulate1:COMParator[:STATe]` on page 187

Equivalent key sequence No equivalent key is available on the front panel.

:CALCulate1:COMParator:COUNt[:STATe]

Syntax

:CALCulate1:COMParator:COUNt[:STATe] {ON|OFF|1|0}

:CALCulate1:COMParator:COUNt[:STATe]?

Description

Turns ON/OFF the BIN counter function of the comparator function.

If you turn on this function, the number of DUTs sorted into each BIN based on the comparator sorting result is counted. The maximum count is 999999. If this is exceeded, the count value will not be updated but remain at 999999.

Parameters

	Description
ON or 1	Turns ON the BIN counter function.
OFF or 0 (initial value)	Turns OFF the BIN counter function.

Query response

{1|0}<newline><^END>

Related commands

`:CALCulate1:COMParator:COUNt:CLEar` on page 175

`:CALCulate1:COMParator:COUNt:DATA?` on page 176

`:CALCulate1:COMParator:COUNt:OVLD?` on page 179

`:CALCulate1:COMParator:COUNt:MULTiple:DATA?` on page 177

`:CALCulate1:COMParator:COUNt:MULTiple:OVLD?` on page 179

Equivalent key sequence

[**Display Format**] - BIN COUNT - COUNT - COUNT ON|COUNT OFF

:CALCulate1:COMParator:MODE

Syntax

:CALCulate1:COMParator:MODE {ABS|DEV|PCNT}

:CALCulate1:COMParator:MODE?

Description

Determines how to specify the limit range of the primary parameter for the comparator function.

Parameters

	Description
ABS (initial value)	Specifies the limit border value in an absolute value (absolute mode).
DEV	Specifies the border value relative to the reference value[1] as an absolute value (absolute tolerance mode).
PCNT	Specifies the border value relative to the reference value as a percentage of the reference value[*1] (percent tolerance mode).

[1] Use the `:CALCulate1:COMParator:PRIMary:NOMinal` command to set the reference value.

Query response

{ABS|DEV|PCNT}<newline><^END>

Related commands

`:CALCulate1:COMParator:PRIMary:NOMinal` on page 184

`:CALCulate1:COMParator:PRIMary:BIN{1-9}[:LIMIT]` on page 182

Equivalent key sequence

[**Meas Setup**] - LIMIT TABLE - Delta Mode

:CALCulate1:COMParator:PRIMary:BIN{1-9}[:LIMIT]

Syntax

:CALCulate1:COMParator:PRIMary:BIN{1|2|3|4|5|6|7|8|9}[:LIMIT] <numeric 1>,<numeric 2>

:CALCulate1:COMParator:PRIMary:BIN{1|2|3|4|5|6|7|8|9}[:LIMIT]?

Description

Sets the low and high limit ranges of BIN1 to BIN9 as the primary parameter used in the comparator function, depending on the designation method set using the :CALCulate1:COMParator:MODE command.

This command only sets the limit range. To activate it, use the :CALCulate1:COMParator:PRIMary:BIN{1-9}:STATe command to turn on the BIN for which the limit range has been set.

Parameters

	<numeric 1>	<numeric 2>
Description	The lower limit value of the limit range.	The upper limit value of the limit range.
Range	-999.999 to 999.999	-999.999 to 999.999
Initial value	0	0
Unit	F (farad) or % (percent)	F (farad) or % (percent)

If the specified parameter is out of the allowable setup range, the minimum value (if the lower limit of the range is not reached) or the maximum value (if the upper limit of the range is exceeded) is set.

NOTE The unit of the parameter may change depending on the limit range designation method.

Query response

{numeric 1},{numeric 2}<newline><^END>

Related commands

:CALCulate1:COMParator:MODE on page 181

:CALCulate1:COMParator:PRIMary:BIN{1-9}:STATe on page 183

:CALCulate1:COMParator:SECondary:LIMit on page 185

:CALCulate1:COMParator[:STATe] on page 187

Equivalent key sequence

[**Meas Setup**] - LIMIT TABLE - BIN{1|2|3|4|5|6|7|8|9}- LOW, HIGH

:CALCulate1:COMParator:PRIMary:BIN{1-9}:STATe

Syntax

:CALCulate1:COMParator:PRIMary:BIN{1|2|3|4|5|6|7|8|9}:STATe {ON|OFF|1|0}

:CALCulate1:COMParator:PRIMary:BIN{1|2|3|4|5|6|7|8|9}:STATe?

Description

Turns ON/OFF BIN1 to BIN9 of the comparator function.

Only BINs that you turn on using this command are used for the sorting judgment of the comparator function.

Parameters

	Description
ON or 1 (initial value of BIN1)	Turns ON BIN.
OFF or 0 (initial value of BIN2 to BIN9)	Turns OFF BIN.

Query response

{1|0}<newline><^END>

Related commands

`:CALCulate1:COMParator:PRIMary:BIN{1-9}[:LIMIT]` on page 182

`:CALCulate1:COMParator:SECondary:STATe` on page 186

`:CALCulate1:COMParator[:STATe]` on page 187

Equivalent key sequence

[**Meas Setup**] - LIMIT TABLE - BIN{1|2|3|4|5|6|7|8|9}- ON|OFF

:CALCulate1:COMParator:PRIMary:NOMinal

Syntax

:CALCulate1:COMParator:PRIMary:NOMinal <numeric>

:CALCulate1:COMParator:PRIMary:NOMinal?

Description

Specifies the reference value used when specifying the primary parameter limit range for the comparator function. This value can be used when the limit range designation method is set to either absolute tolerance mode or percent tolerance mode.

Parameters

	<numeric>
Description	The reference value used when specifying the primary parameter limit range.
Range	-999.999 to 999.999
Initial value	0
Unit	F (farad)

If the specified parameter is out of the allowable setup range, the minimum value (if the lower limit of the range is not reached) or the maximum value (if the upper limit of the range is exceeded) is set.

MAX or MIN can be used to specify the parameters.

Query response

{numeric}<newline><^END>

Related commands

`:CALCulate1:COMParator:MODE` on page 181

`:CALCulate1:COMParator:PRIMary:BIN{1-9}[:LIMIT]` on page 182

Equivalent key sequence

[**Meas Setup**] - LIMIT TABLE - NOM

:CALCulate1:COMParator:SECondary:LIMit

Syntax

:CALCulate1:COMParator:SECondary:LIMit <numeric 1>,<numeric 2>

:CALCulate1:COMParator:SECondary:LIMit?

Description

Sets the limit range for the secondary parameter used in the comparator function.

This command only sets the limit range. To activate the set limit range, use the `:CALCulate1:COMParator:SECondary:STATe` command to enable sorting judgment for the measurement result of the secondary parameter.

Parameters

	<numeric 1>	**<numeric 2>**
Description	The lower limit value of the limit range.	The upper limit value of the limit range.
Range	-99.9999E9 to 99.9999E9	-99.9999E9 to 99.9999E9
Initial value	0	0
Unit	Depends on the type of parameter.	Depends on the type of parameter.

If the specified parameter is out of the allowable setup range, the minimum value (if the lower limit of the range is not reached) or the maximum value (if the upper limit of the range is exceeded) is set.

Query response

{numeric 1},{numeric 2}<newline><^END>

Related commands

`:CALCulate1:COMParator:SECondary:STATe` on page 186

`:CALCulate1:COMParator:PRIMary:BIN{1-9}[:LIMIT]` on page 182

`:CALCulate1:COMParator[:STATe]` on page 187

Equivalent key sequence

[**Meas Setup**] - LIMIT TABLE - BIN|2nd - LOW, HIGH

:CALCulate1:COMParator:SECondary:STATe

Syntax

:CALCulate1:COMParator:SECondary:STATe {ON|OFF|1|0}

:CALCulate1:COMParator:SECondary:STATe?

Description

Sets whether to enable sorting judgment for the measurement result of the secondary parameter when using the comparator function.

Parameters

	Description
ON or 1 (initial value)	Enables sorting judgment for the measurement result of the secondary parameter.
OFF or 0	Disables sorting judgment for the measurement result of the secondary parameter.

Query response

{1|0}<newline><^END>

Related commands

`:CALCulate1:COMParator:SECondary:LIMit` on page 185

`:CALCulate1:COMParator:PRIMary:BIN{1-9}:STATe` on page 183

`:CALCulate1:COMParator[:STATe]` on page 187

Equivalent key sequence

[**Meas Setup**] - LIMIT TABLE - BIN|2nd - ON|OFF

:CALCulate1:COMParator[:STATe]

Syntax

:CALCulate1:COMParator[:STATe] {ON|OFF|1|0}

:CALCulate1:COMParator[:STATe]?

Description

Turns ON/OFF the comparator function.

This setup is interlocked with the ON/OFF state of the signal output of the handler interface.

NOTE

Changing the measurement parameter will automatically turn off this function. To avoid this, you have to write a program so that this command is executed after the execution of the measurement parameter setup command (`:CALCulate1:FORMat` and `:CALCulate2:FORMat`).

Parameters

	Description
ON or 1	Turns ON the comparator function.
OFF or 0 (initial value)	Turns OFF the comparator function.

Query response

{1|0}<newline><^END>

Related commands

`:CALCulate1:FORMat` on page 188

`:CALCulate2:FORMat` on page 191

Equivalent key sequence

[**Meas Setup**] - LIMIT TABLE - COMP

:CALCulate1:FORMat

Syntax

:CALCulate1:FORMat {CP|CS}

:CALCulate1:FORMat?

Description

Specifies the primary parameter to be measured.

When the secondary parameter has been set to G or RP and the primary parameter is set to CS, the secondary parameter is automatically changed to D. Also, when the secondary parameter has been set to RS and the primary parameter is set to CP, the secondary parameter is automatically changed to D.

Parameters

	Description
CP (initial value)	Specifies the capacitance value measured using the parallel equivalent circuit model for the primary parameter.
CS	Specifies the capacitance value measured using the series equivalent circuit model for the primary parameter.

Query response

{CP|CS}<newline><^END>

Related commands

`:CALCulate2:FORMat` on page 191

Equivalent key sequence

[**Meas Setup**]- FUNC

:CALCulate1:MATH:EXPRession:CATalog?

Syntax :CALCulate1:MATH:EXPRession:CATalog?

Description Reads out available parameters used when executing the command to specify the expression of the primary parameter in the deviation measurement mode (the (`:CALCulate1:MATH:EXPRession:NAME` command). The query response is always DEV,PCNT. (Query only)

Query response {DEV,PCNT}<newline><^END> (fixed)

Related commands `:CALCulate1:MATH:EXPRession:NAME` on page 189

Equivalent key sequence No equivalent key is available on the front panel.

:CALCulate1:MATH:EXPRession:NAME

Syntax :CALCulate1:MATH:EXPRession:NAME {DEV|PCNT}

:CALCulate1:MATH:EXPRession:NAME?

Description Specifies the expression of the primary parameter used when displaying the measurement result in the deviation measurement mode.

Parameters

	Description
DEV (initial value)	Displays the result in the difference between the measurement value and the reference value[1] (measurement value - reference value).
PCNT	Displays the difference between the measurement value and the reference value*1 in a percentage[2] to the reference value.

1. Use the `:DATA[:DATA]` command to set the reference value.
2. (Measurement value - reference value) / reference value×100

Query response {DEV|PCNT}<newline><^END>

Related commands `:CALCulate1:MATH:STATe` on page 190

`:CALCulate2:MATH:EXPRession:NAME` on page 192

`:DATA[:DATA]` on page 208

Equivalent key sequence [**Meas Setup**] - DEV A - delta ABS|delta %

:CALCulate1:MATH:STATe

Syntax

:CALCulate1:MATH:STATe {ON|OFF|1|0}

:CALCulate1:MATH:STATe?

Description

Determines whether to use the function (deviation measurement mode) that displays the primary parameter measurement result in deviation from the reference value (set using the `:DATA[:DATA]` command).

NOTE

Changing any of the measurement parameters will automatically disable this function. To avoid this, you have to write a program so that this command is executed after the execution of the measurement parameter setup command (`:CALCulate1:FORMat` and `:CALCulate2:FORMat`).

Parameters

	Description
ON or 1	Enables the deviation measurement mode.
OFF or 0 (initial value)	Disables the deviation measurement mode (in other words, displays the measurement result in an absolute value).

Query response

{1|0}<newline><^END>

Related commands

`:CALCulate1:MATH:EXPRession:NAME` on page 189

`:CALCulate1:FORMat` on page 188

`:CALCulate2:FORMat` on page 191

`:CALCulate2:MATH:STATe` on page 193

Equivalent key sequence

[**Meas Setup**] - DEV A - OFF

:CALCulate2:FORMat

Syntax

:CALCulate2:FORMat {D|Q|G|RP|RS}

:CALCulate2:FORMat?

Description

Specifies the secondary parameter to be measured.

If the primary parameter has been set to CP and the secondary parameter is set to RS, the primary parameter is automatically changed to CS. Also, if the primary parameter has been set to CS and the secondary parameter is set to G or RP, the primary parameter is automatically set to CP.

Parameters

	Description
D (initial value)	Specifies the dissipation factor as the secondary parameter.
Q	Specifies the quality factor (inverse value of D) as the secondary parameter.
G	Specifies the equivalent parallel conductance measured using the parallel equivalent circuit model as the secondary parameter.
RP	Specifies the equivalent parallel resistance measured using the parallel equivalent circuit model as the secondary parameter.
RS	Specifies the equivalent series resistance measured using the series equivalent circuit model as the secondary parameter.

Query response

{D|Q|G|RP|RS}<newline><^END>

Related commands

`:CALCulate1:FORMat` on page 188

Equivalent key sequence

[**Meas Setup**] - FUNC - Cp{D|Q|G|Rp}|Cs{D|Q|Rs}

:CALCulate2:MATH:EXPRession:CATalog?

Syntax

:CALCulate2:MATH:EXPRession:CATalog?

Description

Reads out available parameters used when executing the command to specify the expression of the secondary parameter in the deviation measurement mode (the `:CALCulate1:MATH:EXPRession:NAME` command). The query response is always DEV,PCNT. (Query only)

Query response

DEV,PCNT<newline><^END> (fixed)

Related commands

`:CALCulate2:MATH:EXPRession:NAME` on page 192

Equivalent key sequence

No equivalent key is available on the front panel.

:CALCulate2:MATH:EXPRession:NAME

Syntax

:CALCulate2:MATH:EXPRession:NAME {DEV|PCNT}

:CALCulate2:MATH:EXPRession:NAME?

Description

Specifies the expression of the secondary parameter used when displaying the measurement result in the deviation measurement mode.

Parameters

	Description
DEV (initial value)	Displays the result as the difference between the measurement value and the reference value[1] (measurement value - reference value).
PCNT	Displays the difference between the measurement value and the reference value*1 as a percentage[2] of the reference value.

1. Use the `:DATA[:DATA]` command to set the reference value.
2. (Measurement value - reference value) / reference value×100

Query response

{DEV|PCNT}<newline><^END>

Related commands

`:CALCulate2:MATH:STATe` on page 193

`:CALCulate1:MATH:EXPRession:NAME` on page 189

`:DATA[:DATA]` on page 208

Equivalent key sequence

[**Meas Setup**] - DEV B - delta ABS|delta %

:CALCulate2:MATH:STATe

Syntax

:CALCulate2:MATH:STATe {ON|OFF|1|0}

:CALCulate2:MATH:STATe?

Description

Determines whether to use the function (deviation measurement mode) that displays the secondary parameter measurement result as deviation from the reference value (set using the `:DATA[:DATA]` command).

NOTE

Changing any of the measurement parameters will automatically disable this function. To avoid this, you have to write a program so that this command is executed after the execution of the measurement parameter setup command (`:CALCulate1:FORMat` and `:CALCulate2:FORMat`).

Parameters

	Description
ON or 1	Enables the deviation measurement mode.
OFF or 0 (initial value)	Disables the deviation measurement mode (i.e., displays the measurement result as an absolute value).

Query response

{1|0}<newline><^END>

Related commands

`:CALCulate2:MATH:EXPRession:NAME` on page 192

`:CALCulate1:FORMat` on page 188

`:CALCulate2:FORMat` on page 191

`:CALCulate1:MATH:STATe` on page 190

Equivalent key sequence

[**Meas Setup**] - DEV B - OFF

:CALCulate3:MATH:STATe

Syntax

:CALCulate3:MATH:STATe {ON|OFF|1|0}

:CALCulate3:MATH:STATe?

Description

This is a dummy command for 4268A/4288A compatibility. The current monitor function is always set to ON in E4981A. The query response is always 1.

Query response

{1}<newline><^END>

Related commands

`:CALCulate4:MATH:STATe` on page 194

Equivalent key sequence

No equivalent key is available on the front panel.

:CALCulate4:MATH:STATe

Syntax

:CALCulate4:MATH:STATe {ON|OFF|1|0}

:CALCulate4:MATH:STATe?

Description

This is a dummy command for 4268A/4288A compatibility. The voltage monitor function is always set to ON in E4981A. The query response is always 1.

Query response

{1}<newline><^END>

Related commands

`:CALCulate3:MATH:STATe` on page 194

Equivalent key sequence

No equivalent key is available on the front panel.

:DATA:FEED:BUF1

Syntax

:DATA:FEED:BUF1 {“CALCulate1”|”CALCulate2”|””}

:DATA:FEED:BUF1?

Description

Selects the measurement data fed into data buffer 1 from the following: primary parameter, secondary parameter, or none to be fed. The query response is a string with double quotation marks (“).

NOTE

The functionality of this command is same as `:DATA:FEED[:SOURce]` with value of BUF1.

Parameters

	Description
“CALCulate1”	Specifies the primary parameter as the measurement data fed into the data buffer 1.
“CALCulate2”	Specifies the secondary parameter as the measurement data fed into the data buffer 1.
““ (initial value)	Does not feed the measurement data into the data buffer 1.

Query response

{"CALCulate1"|"CALCulate2"|""}<newline><^END>

Related commands

`:DATA:FEED:CONTrol[:STATe]` on page 200

`:DATA:POINts:BUF1` on page 202

Equivalent key sequence

No equivalent key is available on the front panel.

:DATA:FEED:BUF2

Syntax

:DATA:FEED:BUF2 {“CALCulate1”|”CALCulate2”|””}

:DATA:FEED:BUF2?

Description

Selects the measurement data fed into data buffer 2from the following: primary parameter, secondary parameter, or none to be fed. The query response is a string with double quotation marks (“).

NOTE

The functionality of this command is same as `:DATA:FEED[:SOURce]` with value of BUF2.

Parameters

	Description
“CALCulate1”	Specifies the primary parameter as the measurement data fed into the data buffer 2.
“CALCulate2”	Specifies the secondary parameter as the measurement data fed into the data buffer 2.
““ (initial value)	Does not feed the measurement data into the data buffer 2.

Query response

{"CALCulate1"|"CALCulate2"|""}<newline><^END>

Related commands

`:DATA:FEED:CONTrol[:STATe]` on page 200

`:DATA:POINTs:BUF1` on page 202

Equivalent key sequence

No equivalent key is available on the front panel.

:DATA:FEED:CONTrol:BUF1[:STATe]

Syntax

:DATA:FEED:CONTrol:BUF1[:STATe] {NEVer|ALWays}

:DATA:FEED:CONTrol:BUF1[:STATe]?

Description

Determines whether the measurement data is fed into data buffer 1 never or always.

Parameters

	Description
ALWays	Feeds the measurement data into the data buffer 1 specified by BUF1 each time a measurement is performed.
NEVer (initial value)	Does not feed the measurement data into the data buffer 1 specified with BUF1.

Query response

{NEV|ALW}<newline><^END>

Related commands

`:DATA:FEED:BUF1` on page 195

`:DATA:POINts:BUF1` on page 202

Equivalent key sequence

[**Save/Recall**] - CATALOG - SAVE DATA - START LOG

:DATA:FEED:CONTrol:BUF2[:STATe]

Syntax

:DATA:FEED:CONTrol:BUF2[:STATe] {NEVer|ALWays}

:DATA:FEED:CONTrol:BUF2[:STATe]?

Description

Determines whether the measurement data is fed into data buffer 2 never or always.

Parameters

	Description
ALWays	Feeds the measurement data into the data buffer 2 specified with BUF2 each time a measurement is performed.
NEVer (initial value)	Does not feed the measurement data into the data buffer 2 specified with BUF2.

Query response

{ALW|NEV}<newline><^END>

Related commands

`:DATA:FEED:BUF1` on page 195

`:DATA:POINts:BUF1` on page 202

Equivalent key sequence

[**Save/Recall**] - CATALOG - SAVE DATA - START LOG

:DATA:FEED:CONTrol:BUF3[:STATe]

Syntax

:DATA:FEED:CONTrol:BUF3[:STATe] {NEVer|ALWays}

:DATA:FEED:CONTrol:BUF3[:STATe]?

Description

Determines whether the measurement data is fed into data buffer 3 never or always.

Parameters

	Description
ALWays	Feeds the measurement data into the data buffer 3 specified with BUF3 each time a measurement is performed.
NEVer (initial value)	Does not feed the measurement data into the data buffer specified with BUF3.

Query response

{NEV|ALW}<newline><^END>

Related commands

`:DATA:FEED:BUF1` on page 195

`:DATA:POINts:BUF1` on page 202

Equivalent key sequence

[**Save/Recall**] - CATALOG - SAVE DATA - START LOG

:DATA:FEED:CONTrol[:STATe]

Syntax :DATA:FEED:CONTrol[:STATe] {BUF1|BUF2|BUF3}, {NEVer|ALWays}

:DATA:FEED:CONTrol[:STATe]? {BUF1|BUF2|BUF3}

Description Determines whether to feed the measurement data into data buffer 1, data buffer 2, or data buffer 3.

Parameters

	Description
ALWays	Feeds the measurement data into the data buffer specified with BUF1/BUF2/BUF3 each time a measurement is performed.
NEVer	Does not feed the measurement data into the data buffer specified with BUF1/BUF2/BUF3.

	Description
BUF1	Specifies data buffer 1 as the buffer to which the setup is applied or reads out the setup of data buffer 1.
BUF2	Specifies data buffer 2 as the buffer to which the setup is applied or reads out the setup of data buffer 2.
BUF3	Specifies data buffer 3 as the buffer to which the setup is applied or reads out the setup of data buffer 3.

Query response {NEV|ALW}<newline><^END>

Related commands `:DATA:FEED:BUF1` on page 195

`:DATA:POINts:BUF1` on page 202

Equivalent key sequence No equivalent key is available on the front panel.

:DATA:FEED[:SOURce]

Syntax :DATA:FEED[:SOURce] {BUF1|BUF2},{"CALCulate1"|"CALCulate2"|""}

:DATA:FEED[:SOURce]? {BUF1|BUF2}

Description Selects the measurement data fed into data buffer 1 or data buffer 2 from the following: primary parameter, secondary parameter, or none to be fed. The query response is a string with double quotation marks (“).

Parameters

	Description
BUF1	Specifies data buffer 1, to which the {“CALCulate1”|”CALCulate2”|””} setting is applied or reads out the setup of data buffer 1.
BUF2	Specifies data buffer 2, to which the {“CALCulate1”|”CALCulate2“|””} setting is applied or reads out the setup of data buffer 2.

	Description
“CALCulate1”	Specifies the primary parameter as the measurement data fed into the data buffer specified with {BUF1|BUF2}.
“CALCulate2”	Specifies the secondary parameter as the measurement data fed into the data buffer specified with {BUF1|BUF2}.
““ (initial value)	Does not feed the measurement data into the data buffer specified with {BUF1|BUF2}.

Query response {"CALCulate1"|"CALCulate2"|""}<newline><^END>

Related commands `:DATA:FEED:BUF1` on page 195

`:DATA:POINts:BUF2` on page 203

Equivalent key sequence No equivalent key is available on the front panel.

:DATA:POINts:BUF1

Syntax

:DATA:POINts:BUF1 <numeric>

:DATA:POINts:BUF1?

Description

Specifies the size of data buffer 1 in number of measurements. Executing this command returns the pointer to the specified data buffer (the location to feed measurement data) back to the start.

NOTE

The functionality of this command is same as `:DATA:POINts[:DATA]` with value of BUF1.

Parameters

	<numeric>
Description	Number of measurements for the data buffer specified with BUF1
Range	1 to 200
Initial value	200
Resolution	1

If the specified parameter is out of the allowable setup range, the minimum value (if the lower limit of the range is not reached) or the maximum value (if the upper limit of the range is exceeded) is set.

MAX or MIN can be used to specify the parameters.

Query response

{numeric}<newline><^END>

Related commands

`:DATA:FEED:BUF1` on page 195

`:DATA:FEED:CONTrol[:STATe]` on page 200

Equivalent key sequence

No equivalent key is available on the front panel.

:DATA:POINts:BUF2

Syntax

:DATA:POINts:BUF2 <numeric>

:DATA:POINts:BUF2?

Description

Specifies the size of data buffer 2 in number of measurements. Executing this command returns the pointer to the specified data buffer (the location to feed measurement data) back to the start.

NOTE

The functionality of this command is same as `:DATA:POINts[:DATA]` with value of BUF2.

Parameters

	<numeric>
Description	Number of measurements for the data buffer specified with BUF2
Range	1 to 200
Initial value	200
Resolution	1

If the specified parameter is out of the allowable setup range, the minimum value (if the lower limit of the range is not reached) or the maximum value (if the upper limit of the range is exceeded) is set.

MAX or MIN can be used to specify the parameters.

Query response

{numeric} <newline><^END>

Related commands

`:DATA:FEED:BUF2` on page 196

`:DATA:FEED:CONTrol[:STATe]` on page 200

Equivalent key sequence

No equivalent key is available on the front panel.

:DATA:POINts:BUF3

Syntax

:DATA:POINts:BUF3 <numeric>

:DATA:POINts:BUF3?

Description

Specifies the size of data buffer 3 in number of measurements. Executing this command returns the pointer to the specified data buffer (the location to feed measurement data) back to the start.

NOTE The functionality of this command is same as `:DATA:POINts[:DATA]` with value of BUF3.

Parameters

	<numeric>
Description	Number of measurements for the data buffer specified with BUF3
Range	1 to 1000
Initial value	1000
Resolution	1

If the specified parameter is out of the allowable setup range, the minimum value (if the lower limit of the range is not reached) or the maximum value (if the upper limit of the range is exceeded) is set.

MAX or MIN can be used to specify the parameters.

Query response

{numeric}<newline><^END>

Related commands

`:DATA:FEED:CONTrol:BUF3[:STATe]` on page 199

`:DATA:FEED:CONTrol[:STATe]` on page 200

Equivalent key sequence

No equivalent key is available on the front panel.

:DATA:POINts[:DATA]

Syntax

:DATA:POINts[:DATA] {BUF1|BUF2|BUF3},<numeric>

:DATA:POINts[:DATA]? {BUF1|BUF2|BUF3}

Description

Specifies the size of data buffer 1, data buffer 2, or data buffer 3 in number of measurements. Executing this command returns the pointer to the specified data buffer (the location to feed measurement data) back to the start.

Parameters

	Description
BUF1	Specifies data buffer 1 as the buffer to which the <numeric> setup is applied or reads out the setup of data buffer 1.
BUF2	Specifies data buffer 2 as the buffer to which the <numeric> setup is applied or reads out the setup of data buffer 2.
BUF3	Specifies data buffer 3 as the buffer to which the <numeric> setup is applied or reads out the setup of data buffer 3.

	<numeric>
Description	Number of measurements for the data buffer specified with {BUF1|BUF2|BUF3}
Range	Data buffer 1: 1 to 200 Data buffer 2: 1 to 200 Data buffer 3: 1 to 1000
Initial value	Data buffer 1: 200 Data buffer 2: 200 Data buffer 3: 1000
Resolution	1

If the specified parameter is out of the allowable setup range, the minimum value (if the lower limit of the range is not reached) or the maximum value (if the upper limit of the range is exceeded) is set.

Query response

{numeric}<newline><^END>

Related commands

`:DATA:FEED:BUF1` on page 195

`:DATA:FEED:CONTrol[:STATe]` on page 200

Equivalent key sequence

No equivalent key is available on the front panel.

:DATA:REFerence1:DATA

Syntax

```
:DATA:REFerence1:DATA <numeric>
:DATA:REFerence1:DATA?
```

Description

Sets or reads out the reference value for the primary parameter used in the deviation measurement mode.

NOTE The functionality of this command is same as `:DATA[:DATA]` with value of REF1.

Parameters

	<numeric>
Range	-99.9999E9 to 99.9999E9
Initial value	0

If the specified parameter is out of the allowable setup range, the minimum value (if the lower limit of the range is not reached) or the maximum value (if the upper limit of the range is exceeded) is set.

Query response

{numeric}<newline><^END>

Related commands

`:DATA:REFerence2:DATA` on page 207

Equivalent key sequence

[**Meas Setup**] - REF A or B

:DATA:REFerence1:FILL

Syntax

:DATA:REFerence1:FILL

Description

Execute a single measurement and set the measured value into the primary and secondary reference values for deviation. This command is same as **:DATA:REFerence2:FILL** on page 208. (No Query)

Related commands

:DATA:REFerence2:FILL on page 208

Equivalent key sequence

[**Meas Setup**] - REF A - MEASURE

:DATA:REFerence2:DATA

Syntax

:DATA:REFerence2:DATA <numeric>

:DATA:REFerence2:DATA?

Description

Sets or reads out the reference value for the secondary parameter used in the deviation measurement mode.

NOTE

The functionality of this command is same as :DATA[:DATA] with value of REF2.

Parameters

	<numeric>
Range	-99.9999G to 99.9999G
Initial value	0

If the specified parameter is out of the allowable setup range, the minimum value (if the lower limit of the range is not reached) or the maximum value (if the upper limit of the range is exceeded) is set.

Query response

{numeric}<newline><^END>

Related commands

:DATA:REFerence1:DATA on page 206

Equivalent key sequence

[**Meas Setup**] - REF A or B

:DATA:REFerence2:FILL

Syntax :DATA:REFerence2:FILL

Description Execute a single measurement and set two measured value into each of the reference values for deviation. This command is same as `:DATA:REFerence1:FILL` on page 207. (No Query)

Related commands `:DATA:REFerence1:FILL` on page 207

Equivalent key sequence [**Meas Setup**] - REF B- MEASURE

:DATA[:DATA]

Syntax :DATA[:DATA] {VMON|IMON|BUF[1-3]|REF1|REF2}

:DATA[:DATA]?

Description Reads out the current monitor value or voltage monitor value of the measured signal or the data in data buffer 1, data buffer 2 or data buffer 3. Executing this command rewinds the pointer to the specified data buffer (the location to feed measurement data) to the start.

The transfer format of data read out with this command conforms to the setup made with the `:FORMat[:DATA]` command.

Parameters

	Description
IMON	Reads out the current monitor.
VMON	Reads out the voltage monitor.
BUF1	Reads out data in data buffer 1.
BUF2	Reads out data in data buffer 2.
BUF3	Reads out data in data buffer 3.

	Description
REF1	Sets or reads out the reference value for the primary parameter used in the deviation measurement mode.
REF2	Sets or reads out the reference value for the secondary parameter used in the deviation measurement mode.

	<numeric>
Description	Reference value used in the deviation measurement mode
Range	-99.9999E9 to 99.9999E9 (for REF1) -99.9999E9 to 99.9999E9 (for REF2)
Initial value	0
Unit	Depends on the type of parameter.

If the specified parameter is out of the allowable setup range, the minimum value (if the lower limit of the range is not reached) or the maximum value (if the upper limit of the range is exceeded) is set.

Query response

When specifying BUF1 or BUF2 as parameter

{numeric 1},{numeric 2},{numeric 3},...,{numeric (N×3)}<newline><^END>

The N sets of measurement data (a data set consists of the measurement status, measured value, and comparator sorting result) fed into the data buffer are read out in the order of measurement.

Where N is the number of measurement points specified with the `:DATA:POINTs:BUF1`or `:DATA:POINTs:BUF2` command and n is an integer between 1and N:

{numeric (n–1)×3+1}	The measurement status in the n-th measurement (an integer between 0 and 3 as shown below). 0: No error 1: Detection of measurement impossibility (overload) 2: Detection of Low C reject or No Contact
{numeric (n–1)×3+2}	The measured value of the primary or secondary parameter[1] in the n-th measurement. If overload is detected, 9.9E37 is outputted.

{numeric (n−1)×3+3}	The comparator sorting result in the n-th measurement (an integer between 0 and 11 as shown below). The output (output value is 11) is produced even if the comparator is off. 0: Sorted into OUT_OF_BINS. 1: Sorted into BIN1. 2: Sorted into BIN2. 3: Sorted into BIN3. 4: Sorted into BIN4. 5: Sorted into BIN5. 6: Sorted into BIN6. 7: Sorted into BIN7. 8: Sorted into BIN8. 9: Sorted into BIN9. 10: AUX_BIN 11: OVLD (Overload) or NC (No contact) [sorting not possible]

[1] You need to select primary parameter or secondary parameter by using the `:DATA:FEED:BUF1` or `:DATA:POINts:BUF2` command in advance.

When specifying BUF3 as parameter (Comparator: OFF)

{numeric 1},{numeric 2},{numeric 3},...,{numeric (N×3)}<newline><^END>

The N sets of measurement data (a data set consists of the measurement status, measured value, and comparator sorting result) fed into the data buffer are read out in the order of measurement.

Where N is the number of measurement points specified with the `:DATA:POINts:BUF3` command and n is an integer between 1 and N:

{numeric (n−1)×3+1}	The measurement status in the n-th measurement (an integer between 0 and 2).
{numeric (n−1)×3+2}	The measured value of the primary parameter in the n-th measurement. If overload is detected, 9.9E37 is outputted.
{numeric (n−1)×3+3}	The measured value of the secondary parameter in the n-th measurement. If overload is detected, 9.9E37 is outputted.

When specifying BUF3 as parameter (Comparator: ON)

{numeric 1},{numeric 2},{numeric 3},{numeric 4},...,{numeric (N×4)}<newline><^END>

The N sets of measurement data (a data set consists of the measurement status, measured value, and comparator sorting result) fed into the data buffer are read out in the order of measurement.

Where N is the number of measurement points specified with the `:DATA:POINts:BUF3` command and n is an integer between 1 and N:

{numeric (n−1)×3+1}	The measurement status in the n-th measurement (an integer between 0 and 2).
{numeric (n−1)×3+2}	The measured value of the primary parameter in the n-th measurement. If overload is detected, 9.9E37 is outputted.
{numeric (n−1)×3+3}	The measured value of the secondary parameter in the n-th measurement. If overload is detected, 9.9E37 is outputted.
{numeric (n−1)×3+4}	The comparator sorting result in the n-th measurement (an integer between 0 and 11). The output (output value is 11) is produced even if the comparator is off.

Related commands

`:DATA:FEED:BUF1` on page 195

`:DATA:FEED:CONTrol[:STATe]` on page 200

`:DATA:POINts:BUF1` on page 202

`:DATA:POINts:BUF2` on page 203

`:DATA:POINts:BUF3` on page 204

`:CALCulate3:MATH:STATe` on page 194

`:CALCulate4:MATH:STATe` on page 194

`:FETCh?` on page 220

`:FORMat[:DATA]` on page 225

Equivalent key sequence

No equivalent key is available on the front panel.

:DISPlay:CCLear

Syntax :DISPlay:CCLear

Description Clears errors or caution messages from the display. (No Query)

Equivalent key No equivalent key is available on the front panel.

:DISPlay:LINE

Syntax :DISPlay:LINE <String>

:DISPlay:LINE?

Description Enters arbitrary comments containing up to 30 ASCII characters in the comment field. The string “USER COMMENT” is displayed if this is empty.

Parameter

	<String>
Preset value	“” (NULL)

Query response {“string”}<newline><^END>

Equivalent key [**Meas Setup**] - USER COMMENT - ENTER

:DISPlay:PAGE

Syntax :DISPlay:PAGE {MEASurement|BNUMber|BCOunt|MSETup|CSETup|LTABle|CATAlog|SYSTem|SELF|MLARge|SCONfig|SERVice|CCORrection|CCHeck}

:DISPlay:PAGE?

Description Sets/Gets the page of the display area.

Parameters

Page	Description of displayed item
MEASurement	Measurement Display.
BNUMber	Bin No. Display.
BCOunt	Bin Count Display.
MSETup	Measurement Setup.
CSETup	Correction.
LTABle	Limit Table Setup.
CATAlog	Catalog.
SYSTem	System Info.
SELF	Self Test.
MLARge	Measurement Data in Large font size.
SCONfig	System Config.
SERVice	Service.
CCORrection	Cable Correction.
CCHeck	Contact Check.

NOTE CCORrection is available in Option 001 only.

Query response {MEAS|BNUM|BCO|MSET|CSET|LTABl|CATA|SYST|SELF|MLAR|SCON|SERV|CCOR|CCH}<newline><^END>

Equivalent key sequence

[**Display Format**]

[**Display Format**] – BIN No.

[**Display Format**] – BIN COUNT

[**Meas Setup**]

[**Meas Setup**] – CORRECTION

[**Meas Setup**] – LIMIT TABLE

[**Save/Recall**]

[**System**]

[**System**] – SELF TEST

[**Display Format**] – [**Display Format**]

[**System**] – SYSTEM CONFIG

[**System**] – SERVICE

[**System**] – CABLE CORR

[**Meas Setup**] – CONT CHECK

:DISPlay[:WINDow][:STATe]

Syntax

:DISPlay[:WINDow][:STATe] {ON|OFF|1|0}

:DISPlay[:WINDow][:STATe]?

Description

Enables/disables the display of the measurement result.

If you disables the display, the screen is blanked and “DISPLAY NORMAL” is always displayed on the softkey number 5.

Parameters

	Description
ON or 1 (initial value)	Enables the display.
OFF or 0	Disables the display.

Query response

{1|0}<newline><^END>

Equivalent key sequence

[**Display Format**] - DISPLAY BLANK

:DISPlay[:WINDow]:TEXT1[:DATA]:FMSD:DATA

Syntax

:DISPlay[:WINDow]:TEXT1[:DATA]:FMSD:DATA <numeric>

:DISPlay[:WINDow]:TEXT1[:DATA]:FMSD:DATA?

Description

Sets the value of the highest digit of the primary parameters in the fixed point display. The parameter is always fixed to the same value, regardless of this setting, if the following conditions are met:

- Deviation is displayed as a percentage in the deviation measurement mode.
- The resulting D, Q and % of the secondary parameter is displayed.

Parameters

	<numeric>
Description	The value of the highest digit of the primary parameter.
Range	1a|10a|100a|1f|10f|100f|1p|10p|100p|1n|10n|100n|1u|10u|100u|1m|10m|100m|1|10|100|1k|10k|100k|1M|10M|100M|1G|10G|100G|1T|10T|100T|1P|10P|100P|1E
Initial value	1n

If one of the settable values is not specified for the parameter, the minimum possible value, which is larger than the specified parameter value, is set. If the specified parameter exceeds the maximum value, the maximum value is set.

MAX or MIN can be used to specify the parameters.

Query response

{numeric}<newline><^END>

Related commands

`:DISPlay[:WINDow]:TEXT1[:DATA]:FMSD[:STATe]` on page 217

`:CALCulate1:MATH:EXPRession:NAME` on page 189

`:CALCulate1:MATH:STATe` on page 190

`:CALCulate2:MATH:EXPRession:NAME` on page 192

`:CALCulate2:MATH:STATe` on page 193

Equivalent key sequence

[**Display Format**] - <Primary Measurement Parameter i.e Cp/Cs> - D.P. POS INCR+|D.P. POS DECL-

:DISPlay[:WINDow]:TEXT1[:DATA]:FMSD[:STATe]

Syntax

:DISPlay[:WINDow]:TEXT1[:DATA]:FMSD[:STATe] {ON|OFF|1|0}

:DISPlay[:WINDow]:TEXT1[:DATA]:FMSD[:STATe]?

Description

Determines whether to use the fixed point display when displaying both the primary and secondary parameters. In the following cases, the fixed point display is always used regardless of this setting (the value of the highest digit is also fixed).

- When displaying deviation as a percentage in the deviation measurement mode.
- When displaying the results of D, Q and % of the secondary parameter.

Parameters

	Description
ON or 1	Specifies the fixed point display.
OFF or 0 (initial value)	Specifies the floating point display.

Query response

{1|0}<newline><^END>

Related commands

:DISPlay[:WINDow]:TEXT1[:DATA]:FMSD:DATA on page 216

:CALCulate1:MATH:EXPRession:NAME on page 189

:CALCulate1:MATH:STATe on page 190

:CALCulate2:MATH:EXPRession:NAME on page 192

:CALCulate2:MATH:STATe on page 193

Equivalent key sequence

[**Display Format**] - <Primary Measurement Parameter i.e Cp/Cs> - D.P. AUTO|D.P. FIX

:DISPlay[:WINDow]:TEXT2[:DATA]:FMSD:DATA

Syntax

:DISPlay[:WINDow]:TEXT2[:DATA]:FMSD:DATA <numeric>

:DISPlay[:WINDow]:TEXT2[:DATA]:FMSD:DATA?

Description

Sets the value of the highest digit of the secondary parameters in the fixed point display. The parameter is always fixed to the same value, regardless of this setting, if the following conditions are met:

- Deviation is displayed as a percentage in the deviation measurement mode.
- The resulting D, Q and % of the secondary parameter is displayed.

Parameters

	<numeric>
Description	The value of the highest digit of the secondary parameter.
Range	1a|10a|100a|1f|10f|100f|1p|10p|100p|1n|10n|100n|1u|10u|100u|1m|10m|100m|1|10|100|1k|10k|100k|1M|10M|100M|1G|10G|100G|1T|10T|100T|1P|10P|100P|1E
Initial value	1n

If one of the settable values is not specified for the parameter, the minimum possible value, which is larger than the specified parameter value, is set. If the specified parameter exceeds the maximum value, the maximum value is set.

MAX or MIN can be used to specify the parameters.

Query response

{numeric}<newline><^END>

Related commands

`:DISPlay[:WINDow]:TEXT1[:DATA]:FMSD[:STATe]` on page 217

`:CALCulate1:MATH:EXPRession:NAME` on page 189

`:CALCulate1:MATH:STATe` on page 190

`:CALCulate2:MATH:EXPRession:NAME` on page 192

`:CALCulate2:MATH:STATe` on page 193

Equivalent key sequence

[**Display Format**] - <Secondary Measurement Parameter i.e D/Q/G/Rp/Rs> - D.P. POS INCR+|D.P. POS DECL-

:DISPlay[:WINDow]:TEXT2[:DATA]:FMSD[:STATe]

Syntax

:DISPlay[:WINDow]:TEXT2[:DATA]:FMSD[:STATe] {ON|OFF|1|0}

:DISPlay[:WINDow]:TEXT2[:DATA]:FMSD[:STATe]?

Description

Determines whether to use the fixed point display when displaying both the primary and secondary parameters. In the following cases, the fixed point display is always used regardless of this setting (the value of the highest digit is also fixed).

- When displaying deviation as a percentage in the deviation measurement mode.
- When displaying the results of D, Q and % of the secondary parameter.

Parameters

	Description
ON or 1	Specifies the fixed point display.
OFF or 0 (initial value)	Specifies the floating point display.

Query response

{1|0}<newline><^END>

Related commands

`:DISPlay[:WINDow]:TEXT1[:DATA]:FMSD:DATA` on page 216

`:CALCulate1:MATH:EXPRession:NAME` on page 189

`:CALCulate1:MATH:STATe` on page 190

`:CALCulate2:MATH:EXPRession:NAME` on page 192

`:CALCulate2:MATH:STATe` on page 193

Equivalent key sequence

[**Display Format**] - <Secondary Measurement Parameter i.e D/Q/G/Rp/Rs> - D.P. AUTO|D.P. FIX

:FETCh?

Syntax :FETCh?

Description Reads out the measurement result. The target measurement to read out depends on the state of the E4981A. (Query Only).

• During measurement:	Waits for end of the measurement and then reads out its result.
• Other states:	Reads out the result of the immediately preceding measurement.

If overload or no contact is detected (that is, the measurement status is 1), the measured values of the primary parameter and secondary parameter are 9.9E37 and the comparator sorting result is 11.

The transfer format of data read out with this command conforms to the setup made with the `:FORMat[:DATA]` command. (Query only)

NOTE

If the comparator function is disabled (OFF has been specified with the `:CALCulate1:COMParator[:STATe]` command), only three data items, {numeric1}, {numeric2}, and {numeric3}, are read out.

{numeric 1}, {numeric 2}, {numeric 3}, and {numeric 4} are as follows:

{numeric 1}: Measurement status (one of the integers between 0 and 3 listed below)

0: No error
1: Detection of overload (OVLD)
2: Detection of Low C or No Contact (See `:FORMat:STATus:EXTension` on page 224)

{numeric 2}: Measured value of the primary parameter

{numeric 3}: Measured value of the secondary parameter

{numeric 4}: Comparator sorting result (an integer between 0 and 11 listed below)

0: Sorted into OUT_OF_BINS.
1: Sorted into BIN1.
2: Sorted into BIN2.
3: Sorted into BIN3.
4: Sorted into BIN4.
5: Sorted into BIN5.
6: Sorted into BIN6.
7: Sorted into BIN7.
8: Sorted into BIN8.
9: Sorted into BIN9.
10: Sorted into AUX_BIN.
11: OVLD (Overload) or NC (No contact) [sorting not possible]

Query response

{numeric 1},{numeric 2},{numeric 3},{numeric 4}<newline><^END>

Related commands

`:READ?` on page 230
`*TRG` on page 163
`:FORMat[:DATA]` on page 225
`:CALCulate1:COMParator[:STATe]` on page 187

Equivalent key sequence

No equivalent key is available on the front panel.

:FORMat:ASCii:LONG

Syntax

:FORMat:ASCii:LONG {ON|OFF|1|0}

:FORMat:ASCii:LONG?

Description

Enables the long and short format. The value read in long format is (+0.000000000E+00) and the value read as short format is (0.00000E+00).

Parameter

	Description
ON or 1	Long format
OFF or 0 (Preset value)	Short format

Query response

{1|0}<newline><^END>

Related commands

`:FETCh?` on page 220

`:READ?` on page 230

`:DATA[:DATA]` on page 208

`*TRG` on page 163

`[:SENSe]:CORRection:DATA` on page 240

`[:SENSe][:FIMPedance]:CONTact1:VERify:BUF1:DATA?` on page 253

`[:SENSe][:FIMPedance]:CONTact1:VERify:BUF2:DATA?` on page 256

Equivalent key

No equivalent key is available on the front panel.

:FORMat:BORDer

Syntax

:FORMat:BORDer {NORMal|SWAPped}

:FORMat:BORDer?

Description

When the data transfer format is set to the binary type, this command sets the transfer order of each byte in data (byte order).

Parameter

	Description
NORMal (Preset value)	Specifies the byte order in which transfer starts from the byte that includes the MSB (Most Significant Bit)
SWAPped	Specifies the byte order in which transfer starts from the byte that includes the LSB (Least Significant Bit)

Query response

{NORM|SWAP}<newline><^END>

Related commands

:FETCh? on page 220

:READ? on page 230

:DATA[:DATA] on page 208

*TRG on page 163

[:SENSe]:CORRection:DATA on page 240

[:SENSe][:FIMPedance]:CONTact1:VERify:BUF1:DATA? on page 253

[:SENSe][:FIMPedance]:CONTact1:VERify:BUF2:DATA? on page 256

Equivalent key

No equivalent key is available on the front panel.

:FORMat:STATus:EXTension

Syntax

:FORMat:STATus:EXTension {ON|OFF|1|0}

:FORMat:STATus:EXTension?

Description

This command enables/disables the status extension reporting for Contact check failure.

If the status reported (numeric1) from `*TRG`, `:FETCh?` and `:READ?` is 2 (Detection of LowC or No Contact), then `:FORMat:STATus:EXTension` can be enabled to find the source of Contact Check failure. The return value of status determines the source of contact check failure and are as follows:

- +8: Threshold1 Failure
- +16: Threshold2 Failure
- +128: Signal Level Check failure when LVL COMP = ON and Contact Check = ON.

Parameter

	Description
ON or 1	Enables the extended reporting of Contact Check Failure.
OFF or 0	Disables the extended reporting of Contact Check Failure.

Query response

{1|0}<newline><^END>

Related commands

`*TRG` on page 163

`:FETCh?` on page 220

`:READ?` on page 230

`[:SENSe][:FIMPedance]:CONTact1:VERify[:STATe]` on page 260

Equivalent key

No equivalent key is available on the front panel.

:FORMat[:DATA]

Syntax

:FORMat[:DATA] {ASCii|REAL}

:FORMat[:DATA]?

Description

Sets the transfer format of data read out using the following commands:

- `:FETCh?` on page 220
- `:READ?` on page 230
- `*TRG` on page 163
- `:DATA[:DATA]` on page 208
- `[:SENSe]:CORRection:DATA` on page 240
- `[:SENSe][:FIMPedance]:CONTact1:VERify:BUF1:DATA?` on page 253
- `[:SENSe][:FIMPedance]:CONTact1:VERify:BUF2:DATA?` on page 256

For details on the data transfer formats, refer to "Data Transfer Format" on page 63.

Parameters

	Description
ASCii (initial value)	Specifies the ASCII data transfer format.
REAL	Specifies the 64-bit real number data transfer format.

Query response

{ASC|REAL}<newline><^END>

Equivalent key sequence

No equivalent key is available on the front panel.

:HCOPy:SDUMp:DATA?

Syntax :HCOPy:SDUMp:DATA?

Description Outputs screen image display in GIF format. (Query Only)

Equivalent key No equivalent key is available on the front panel.

Example of Use

```
Dim Nop As Long
Dim GifData(10000) As Byte
Dim paramsArray(2) As Long
Dim i As Integer

Nop = UBound(GifData) - LBound(GifData) + 1
paramsArray(0) = VarPtr(Nop)
paramsArray(1) = VarPtr(GifData(0))

Call viVPrintf(Agte4981a, ":HCOPY:SDUMP:DATA?" + vbLf, 0)
Call viVScanf(Agte4981a, "%#b", paramsArray(0))

Open "C:\TEST.gif" For Binary As #1
For i = 0 To Nop - 1
    Put #1, , GifData(i)
Next i
Close
```

NOTE Pressing [**Save/Recall**] - SAVE DISPLAY key saves the current screen image into the USB memory.

:INITiate:CONTinuous

Syntax

:INITiate:CONTinuous {ON|OFF|1|0}

:INITiate:CONTinuous?

Description

This command changes the trigger state from “idle” state to the “wait for trigger” automatically or stay in “idle” state. After executing `:SYSTem:PRESet`, this value changes to ON, after `*RST` this value changes to OFF .

For details on the trigger system, refer to “Trigger system” on page 57.

NOTE

This command is turned ON when the trigger source is changed from the front panel of the E4981A.

Parameters

	Description
ON or 1	Enables successive startup.
OFF or 0 (initial value)	Disables successive startup.

This setup is initialized to ON when using the `:SYSTem:PRESet` command and to OFF when using the `*RST` command.

Query response

{1|0}<newline><^END>

Related commands

`*RST` on page 160

`:SYSTem:PRESet` on page 284

Equivalent key sequence

No equivalent key is available on the front panel.

:INITiate[:IMMediate]

Syntax

:INITiate[:IMMediate]

Description

Changes the trigger state from “idle” state to “wait for trigger” state for one trigger sequence. (No Query)

For details on the trigger system, refer to “Trigger system” on page 57. (No query)

Related commands

`:INITiate:CONTinuous` on page 227

Equivalent key sequence

No equivalent key is available on the front panel.

:MMEMory:DELete[:REGister]

Syntax

:MMEMory:DELete[:REGister] <numeric>

Description

Deletes the state from the memories. Numbers 0 to 9 are located in the internal memory, while No. 10 to 19 are situated in the external USB memory. (No Query)

Parameter

	<Numeric>
Range	0 to 19
Resolution	1

Equivalent key

[**Save/Recall**] - No {1-9} - DELETE

:MMEMory:LOAD:STATe[:REGister]

Syntax :MMEMory:LOAD:STATe[:REGister] <numeric>

Description Loads the state from the memories. Numbers 0 to 9 are located in the internal memory, while No. 10 to 19 are situated in the external USB memory. (No Query)

Parameter

<Numeric>	
Range	0 to 19
Resolution	1

Equivalent key [**Save/Recall**] - No {1-9} - RECALL

:MMEMory:STORe:STATe[:REGister]

Syntax :MMEMory:STORe:STATe[:REGister] <numeric>

Description Stores the state to the memories. Numbers 0 to 9 are located in the internal memory, while Nos. 10 to 19 are situated in the external USB memory. Number 9 is automatically recalled. (No Query)

Parameter

<Numeric>	
Range	0 to 19
Resolution	1

Equivalent key [**Save/Recall**] - No {1-9} - SAVE

:READ?

Syntax :READ?

Description Waits for the end of measurement and reads out the measurement result.

Executing this command brings the E4981A into the trigger wait state. When the trigger system is in the idle state, this command invokes the trigger system (the `:INITiate[:IMMediate]` command) once and then brings the instrument into the trigger wait state. After that, when the instrument is triggered and the measurement finishes, this command reads the measurement result and exits.

This command can be executed when the trigger mode has been set to either internal (Int) or external (Ext) (set to INT or EXT with the `:TRIGger[:SEQ1]:SOURce` command).

If this command is executed with the trigger mode set to manual (Man) or Bus (specified as MAN or BUS with the `:TRIGger[:SEQ1]:SOURce` command), an error occurs because there is no way to trigger and the command is ignored.

The transfer format of data read out with this command conforms to the setup made with the `:FORMat[:DATA]` command. (Query only)

NOTE If the trigger mode is External (EXT), no subsequent command can be accepted until an external trigger is supplied. To release this state without entering an external trigger, send Device Clear (the “CLEAR” instruction in HTBasic) to the GPIB/USB/LAN port to abort the query operation.

NOTE

If the comparator function is disabled (OFF has been specified with the `:CALCulate1:COMParator[:STATe]` command), only three data items, {numeric1}, {numeric2}, and {numeric3}, are read out.

{numeric 1}, {numeric 2}, {numeric 3}, and {numeric 4} are as follows:

{numeric 1}: Measurement status (one of the integers between 0 and 3 listed below)

0: No error
1: Detection of overload (OVLD)
2: Detection of Low C or No Contact (See `:FORMat:STATus:EXTension` on page 224)

{numeric 2}: Measured value of the primary parameter

{numeric 3}: Measured value of the secondary parameter

{numeric 4}: Comparator sorting result (an integer between 0 and 11 listed below)

0: Sorted into OUT_OF_BINS.
1: Sorted into BIN1.
2: Sorted into BIN2.
3: Sorted into BIN3.
4: Sorted into BIN4.
5: Sorted into BIN5.
6: Sorted into BIN6.
7: Sorted into BIN7.
8: Sorted into BIN8.
9: Sorted into BIN9.
10: Sorted into AUX_BIN.
11: OVLD (Overload) or NC (No Contact) [sorting not possible]

Query response

{numeric 1},{numeric 2},{numeric 3},{numeric 4}<newline><^END>

The query response is the same as that of the `:FETCh?` command. For details, refer to the description of `:FETCh?`.

Related commands

`:FETCh?` on page 220

`*TRG` on page 163

`:INITiate[:IMMediate]` on page 228

`:TRIGger[:SEQ1]:SOURce` on page 305

`:FORMat[:DATA]` on page 225

Equivalent key sequence

No equivalent key is available on the front panel.

[:SENSe]:AVERage:COUNt

Syntax

[:SENSe]:AVERage:COUNt <numeric>

[:SENSe]:AVERage:COUNt?

Description

Sets the averaging count of the measured value for the averaging function.

Unlike setting the averaging count with the front panel keys, using this command to set the averaging count does not automatically turn on the averaging function. Therefore, if the averaging function has been set to off, you have to turn it on by using the `[:SENSe]:AVERage[:STATe]` command.

Parameters

	<numeric>
Description	Averaging count
Range	1 to 256
Initial value	1
Resolution	1

If the specified parameter is out of the allowable setup range, the minimum value (if the lower limit of the range is not reached) or the maximum value (if the upper limit of the range is exceeded) is set.

MAX or MIN can be used to specify the parameters.

Query response

{numeric}<newline><^END>

Related commands

`[:SENSe]:AVERage[:STATe]` on page 233

Equivalent key sequence

[**Meas Setup**] - AVG - INCR + / DECR-

[:SENSe]:AVERage[:STATe]

Syntax

[:SENSe]:AVERage[:STATe] {ON|OFF|1|0}

[:SENSe]:AVERage[:STATe]?

Description

Turns ON/OFF the averaging function.

Parameters

	Description
ON or 1 (initial value)	Turns ON the averaging function.
OFF or 0	Turns OFF the averaging function.

Query response

{1|0}<newline><^END>

Related commands

`[:SENSe]:AVERage:COUNt` on page 232

Equivalent key sequence

[**Meas Setup**] - AVG - ON/OFF

NOTE

Setting the averaging count with the front panel keys will automatically turn on the function.

[:SENSe]:CORRection:CKIT:STAN1:FORMat

Syntax

[:SENSe]:CORRection:CKIT:STAN1:FORMat {GB|CPG}

[:SENSe]:CORRection:CKIT:STAN1:FORMat?

Description

Sets the parameter types of the OPEN correction data.

Parameters

	Description
GB (initial value)	Specifies G as the primary parameter and B as the secondary parameter.
CPG	Specifies Cp as the primary parameter and G as the secondary parameter.

Query response

{GB|CPG}<newline><^END>

Related commands

`[:SENSe]:CORRection:DATA` on page 240

Equivalent key sequence

[**Meas Setup**] - CORRECTION - OPEN - G-B/Cp-G

[:SENSe]:CORRection:CKIT:STAN2:FORMat

Syntax [:SENSe]:CORRection:CKIT:STAN2:FORMat {RX|LSRS}

[:SENSe]:CORRection:CKIT:STAN2:FORMat?

Description Sets the parameter types of the SHORT correction data.

Parameters

	Description
RX (initial value)	Specifies R as the primary parameter and X as the secondary parameter.
LSRS	Specifies Ls as the primary parameter and Rs as the secondary parameter.

Query response {RX|LSRS}<newline><^END>

Related commands `[:SENSe]:CORRection:DATA` on page 240

Equivalent key sequence [**Meas Setup**] - CORRECTION - SHORT- R-X/Ls-Rs

[:SENSe]:CORRection:CKIT:STAN3[:DATA]

Syntax

[:SENSe]:CORRection:CKIT:STAN3[:DATA] <numeric 1>,<numeric 2>

[:SENSe]:CORRection:CKIT:STAN3[:DATA]?

Description

Defines the values of the LOAD correction standard for the parameters you specify by using the `[:SENSe]:CORRection:CKIT:STAN3:FORMat` command.

These are set as the standard values for the measurement frequency when executing the command (set with the `:SOURce:FREQuency[:CW]` command).

When using the multi-correction function (set to ON with the `[:SENSe]:CORRection:MULTiple[:STATe]` command) with the channel-by-channel definition of the standard values enabled (set to ON with the `[:SENSe]:CORRection:MULTiple:CKIT:STAN3[:STATe]` command), the standard values for the selected channel (selected with the `[:SENSe]:CORRection:MULTiple:CHANnel` command) are set at execution of the command.

Parameters

	<numeric 1>	<numeric 2>
Description	Value of the primary parameter.	Value of the secondary parameter.
Range	-999.999 to 999.999	-99.9999E9 to 99.9999E9
Initial value	100E-9	0
Unit	F (farad)	Depends on the type of parameter.

If the specified parameter is out of the allowable setup range, the minimum value (if the lower limit of the range is not reached) or the maximum value (if the upper limit of the range is exceeded) is set.

Query response

{numeric 1},{numeric 2}<newline><^END>

Related commands

`[:SENSe]:CORRection:LOAD[:STATe]` on page 242

`[:SENSe]:CORRection:CKIT:STAN3:FORMat` on page 236

`[:SENSe]:CORRection:MULTiple[:STATe]` on page 245

`[:SENSe]:CORRection:MULTiple:CHANnel` on page 243

`[:SENSe]:CORRection:MULTiple:CKIT:STAN3[:STATe]` on page 244

Equivalent key sequence

[**Meas Setup**] - CORRECTION - REF- A|B

[:SENSe]:CORRection:CKIT:STAN3:FORMat

Syntax

[:SENSe]:CORRection:CKIT:STAN3:FORMat {CPD|CPQ|CPG|CPRP|CSD|CSQ|CSRS}

[:SENSe]:CORRection:CKIT:STAN3:FORMat?

Description

Sets the types of parameters used to define the standard for LOAD correction.

Parameters

	Description
CPD (initial value)	Specifies Cp as the primary parameter, D as the secondary parameter.
CPQ	Specifies Cp as the primary parameter, Q as the secondary parameter.
CPG	Specifies Cp as the primary parameter, G as the secondary parameter.
CPRP	Specifies Cp as the primary parameter, Rp as the secondary parameter.
CSD	Specifies Cs as the primary parameter, D as the secondary parameter.
CSQ	Specifies Cs as the primary parameter, Q as the secondary parameter.
CSRS	Specifies Cs as the primary parameter, Rs as the secondary parameter.

For details on Cp and Cs, refer to `:CALCulate1:FORMat` on page 188. For details on D, Q, Rp, and Rs, refer to `:CALCulate2:FORMat` on page 191.

Query response

{CPD|CPQ|CPG|CPRP|CSD|CSQ|CSRS}<newline><^END>

Related commands

`[:SENSe]:CORRection:CKIT:STAN3[:DATA]` on page 235

Equivalent key sequence

[**Meas Setup**] - CORRECTION - LOAD- Cp{CPD|CPQ|CPG|CPRP}/ Cs{CSD|CSQ|CSRS}

[:SENSe]:CORRection:COLLect[:ACQuire]

Syntax

[:SENSe]:CORRection:COLLect[:ACQuire] {STAN1|STAN2|STAN3}

Description

Measures the correction data for OPEN/SHORT/LOAD correction and turns on the correction function.

For Open and Short correction, the correction data is measured for all measurement frequencies. For load correction, the correction data is measured for the measurement frequency used in executing the command (set with the `:SOURce:FREQuency[:CW]` command).

When using the multi-correction function (set to ON with the `[:SENSe]:CORRection:MULTiple[:STATe]` command), this is measured as the correction data for the selected channel when executing the command (selected with the `[:SENSe]:CORRection:MULTiple:CHANnel` command). (No query)

Parameters

	Description
STAN1	Specifies the OPEN correction.
STAN2	Specifies the SHORT correction.
STAN3	Specifies the LOAD correction.

Related commands

`[:SENSe]:CORRection:OPEN[:STATe]` on page 248

`[:SENSe]:CORRection:SHORt[:STATe]` on page 249

`[:SENSe]:CORRection:LOAD[:STATe]` on page 242

`[:SENSe]:CORRection:MULTiple[:STATe]` on page 245

`[:SENSe]:CORRection:MULTiple:CHANnel` on page 243

Equivalent key sequence

[**Meas Setup**] - CORRECTION - OPEN - MEAS OPEN

[**Meas Setup**] - CORRECTION - SHORT - MEAS SHORT

[**Meas Setup**] - CORRECTION - LOAD - MEAS LOAD

[:SENSe]:CORRection:COLLect:LOAD:RANGe:AUTO

Syntax

[:SENSe]:CORRection:COLLect:LOAD:RANGe:AUTO {ON|OFF|1|0}

[:SENSe]:CORRection:COLLect:LOAD:RANGe:AUTO?

Description

Sets on/off of auto ranging when the load correction is executed. When this function is set at off, the range which is selected by `[:SENSe][:FIMPedance]:RANGe[:UPPer]` is used for the load correction.

For auto ranging when a measurement is executed, use `[:SENSe][:FIMPedance]:RANGe:AUTO`

NOTE

The functionality of this command is same as `[:SENSe]:CORRection:COLLect:LOAD:RANGE:AUTO`.

Parameter

	Description
ON or 1 (initial setting)	Set auto ranging at on. The load correction is performed with auto ranging. (AUTO).
OFF or 0	Set auto ranging at off. The load correction is performed at the range which is selected by "[:SENSe][:FIMPedance]:RANGe[:UPPer]".

Query response

{1|0}<newline><^END>

Related Commands

`[:SENSe]:CORRection:COLLect[:ACQuire]` on page 237

`[:SENSe][:FIMPedance]:RANGe[:UPPer]` on page 266

`:SYSTem:PRESet` on page 284

Equivalent key sequence

[**Meas Setup**] - CORRECTION - LOAD RNG- AUTO|FIX

[:SENSe]:CORRection:COLLect:STAN3:RANGe:AUTO

Syntax

[:SENSe]:CORRection:COLLect:STAN3:RANGe:AUTO {ON|OFF|1|0}

[:SENSe]:CORRection:COLLect:STAN3:RANGe:AUTO?

Description

Sets on/off of auto ranging when the load correction is executed. When this function is set at off, the range which is selected by `[:SENSe][:FIMPedance]:RANGe[:UPPer]` is used for the load correction.

For auto ranging when a measurement is executed, use `[:SENSe][:FIMPedance]:RANGe:AUTO`

NOTE

The functionality of this command is same as `[:SENSe]:CORRection:COLLect:LOAD:RANGe:AUTO`.

Parameter

	Description
ON or 1 (initial setting)	Measures the load standard with auto range.
OFF or 0	Measures the load standard measurement range which has been set just before the load standard measurement is performed (auto range function is invalid).

Query response

{1|0}<newline><^END>

Related Commands

`[:SENSe]:CORRection:COLLect[:ACQuire]` on page 237

`[:SENSe][:FIMPedance]:RANGe[:UPPer]` on page 266

`:SYSTem:PRESet` on page 284

Equivalent key sequence

[**Meas Setup**] - CORRECTION - LOAD RNG- AUTO/FIX.

[:SENSe]:CORRection:DATA

Syntax

[:SENSe]:CORRection:DATA {STAN1|STAN2|STAN3},<numeric 1>,<numeric 2>

[:SENSe]:CORRection:DATA? {STAN1|STAN2|STAN3}

Description

Sets the correction data for OPEN/SHORT/LOAD correction.

These correction data are measured for the measurement frequency used in executing the command (set with the `:SOURce:FREQuency[:CW]` command).

When using the multi-correction function (set to ON with the `[:SENSe]:CORRection:MULTiple[:STATe]` command), this is set as the correction data for the selected channel when executing the command (selected with the `[:SENSe]:CORRection:MULTiple:CHANnel` command).

To activate the set correction data, use the `[:SENSe]:CORRection:OPEN[:STATe]`, `[:SENSe]:CORRection:SHORt[:STATe]`, and `[:SENSe]:CORRection:LOAD[:STATe]` commands to turn on the OPEN/SHORT/LOAD correction function.

The transfer format of data read out with this command conforms to the setup of the `:FORMat[:DATA]` command.

Parameters

	Parameter 1: {STAN1|STAN2|STAN3}
STAN1	Sets or reads out the data for OPEN correction.
STAN2	Sets or reads out the data for SHORT correction.
STAN3	Sets or reads out the data for LOAD correction.

- When specifying STAN1 as parameter 1:

	Parameter 2: <numeric 1>	Parameter 3: <numeric 2>
Description	Value of the primary parameter[1].	Value of the secondary parameter*[1].
Range	-99.9999E9 to 99.9999E9	-99.9999E9 to 99.9999E9
Initial value	0	0
Unit	Depends on the type of parameter.	Depends on the type of parameter.

[1] Use the `[:SENSe]:CORRection:CKIT:STAN1:FORMat` command to specify the type of parameter.

- When specifying STAN2 as parameter 1:

	Parameter 2: <numeric 1>	Parameter 3: <numeric 2>
Description	Value of the primary parameter[1].	Value of the secondary parameter*[1].
Range	-99.9999E9 to 99.9999E9	-99.9999E9 to 99.9999E9
Initial value	0	0
Unit	Depends on the type of parameter.	Depends on the type of parameter.

1. Use the `[:SENSe]:CORRection:CKIT:STAN2:FORMat` command to specify the type of parameter.

- When specifying STAN3 as parameter 1:

	Parameter 2: <numeric 1>	Parameter 3: <numeric 2>
Description	Value of the primary parameter[1].	Value of the secondary parameter*[1].
Range	-999.999 to 999.999	-99.9999E9 to 99.9999E9
Initial value	100E-9	0
Unit	F (farad)	Depends on the type of parameter.

1. Use the `[:SENSe]:CORRection:CKIT:STAN3:FORMat` command to specify the type of parameter.

In any case, if the specified parameter is out of the allowable setup range, the minimum value (if the lower limit of the range is not reached) or the maximum value (if the upper limit of the range is exceeded) is set.

Query response {numeric 1},{numeric 2}<newline><^END>

Related commands

`[:SENSe]:CORRection:OPEN[:STATe]` on page 248

`[:SENSe]:CORRection:SHORt[:STATe]` on page 249

`[:SENSe]:CORRection:LOAD[:STATe]` on page 242

`[:SENSe]:CORRection:MULTiple[:STATe]` on page 245

`[:SENSe]:CORRection:MULTiple:CHANnel` on page 243

`[:SENSe]:CORRection:CKIT:STAN1:FORMat` on page 233

`[:SENSe]:CORRection:CKIT:STAN2:FORMat` on page 234

`[:SENSe]:CORRection:CKIT:STAN3:FORMat` on page 236

`:FORMat[:DATA]` on page 225

Equivalent key sequence

[**Meas Setup**] - CORRECTION - OPEN - A|B

[**Meas Setup**] - CORRECTION - SHORT- A|B

[**Meas Setup**] - CORRECTION - LOAD- A|B

[:SENSe]:CORRection:LOAD[:STATe]

Syntax

[:SENSe]:CORRection:LOAD[:STATe] {ON|OFF|1|0}

[:SENSe]:CORRection:LOAD[:STATe]?

Description

Turns ON/OFF the LOAD correction function.

With the LOAD correction set to ON, if you change the cable length (set with the `:CALibration:CABLe[:LENGth]` command) or frequency shift (set with the `:SYSTem:FSHift` command), the LOAD correction is automatically changed to OFF.

Parameters

	Description
ON or 1	Turns ON the LOAD correction.
OFF or 0 (initial value)	Turns OFF the LOAD correction.

Query response

{1|0}<newline><^END>

Related commands

`:CALibration:CABLe[:LENGth]` on page 171

`:SYSTem:FSHift` on page 281

`[:SENSe]:CORRection:COLLect[:ACQuire]` on page 237

Equivalent key sequence

[**Meas Setup**] - CORRECTION - LOAD - ON/OFF

[:SENSe]:CORRection:MULTiple:CHANnel

Syntax

[:SENSe]:CORRection:MULTiple:CHANnel <numeric>

[:SENSe]:CORRection:MULTiple:CHANnel?

Description

Specifies a channel number used in the multi-correction function.

You can also specify the channel number via the scanner interface.

NOTE

A channel number specified through the interface overrides a channel number selected with this command.

Parameters

	<numeric>
Description	The desired channel number.
Range	0 to 255
Initial value	0
Resolution	1

If the specified parameter is out of the allowable setup range, an error occurs.

MAX or MIN can be used to specify the parameters.

Query response

{numeric}<newline><^END>

Equivalent key sequence

[**Meas Setup**] - CORRECTION - CH

[:SENSe]:CORRection:MULTiple:CKIT:STAN3[:STATe]

Syntax

[:SENSe]:CORRection:MULTiple:CKIT:STAN3[:STATe] {ON|OFF|1|0}

[:SENSe]:CORRection:MULTiple:CKIT:STAN3[:STATe]?

Description

Determines whether to define the standard values for LOAD correction for each channel individually when using the multi-correction function (set to ON with the `[:SENSe]:CORRection:MULTiple[:STATe]` command).

Parameters

	Description
ON or 1	Enables channel-by-channel value definition.
OFF or 0 (initial value)	Disables channel-by-channel value definition (defines the same values for all channels).

Query response

{1|0}<newline><^END>

Related commands

`[:SENSe]:CORRection:MULTiple[:STATe]` on page 245

`[:SENSe]:CORRection:CKIT:STAN3[:DATA]` on page 235

Equivalent key sequence

[**Meas Setup**] - CORRECTION - LOAD REF - MULTI|SINGLE

[:SENSe]:CORRection:MULTiple[:STATe]

Syntax

[:SENSe]:CORRection:MULTiple[:STATe] {ON|OFF|1|0}

[:SENSe]:CORRection:MULTiple[:STATe]?

Description

Turns ON/OFF the multi-correction function.

This setup is interlocked with the ON/OFF state of the signal output of the scanner interface.

Parameters

	Description
ON or 1	Turns ON the multi-correction function.
OFF or 0 (initial value)	Turns OFF the multi-correction function.

Query response

{1|0}<newline><^END>

Equivalent key sequence

[**Meas Setup**] - CORRECTION - MULTI - ON|OFF

[:SENSe]:CORRection:OFFSet:DATA

Syntax

[:SENSe]:CORRection:OFFSet:DATA <numeric 1>,<numeric 2>

[:SENSe]:CORRection:OFFSet:DATA?

Description

Sets the correction values for the primary parameter and secondary parameter used in the offset correction function.

These correction data are measured for the measurement frequency used in executing the command (set with the `:SOURce:FREQuency[:CW]` command).

Parameters

	<numeric 1>	<numeric 2>
Description	The offset correction value for the primary parameter.	The offset correction value for the secondary parameter.
Range	-999.999 to 999.999	-99.9999E9 to 99.9999E9
Initial value	0	0
Unit	F (farad)	Depends on the setup of the secondary parameter.

If the specified parameter is out of the allowable setup range, the minimum value (if the lower limit of the range is not reached) or the maximum value (if the upper limit of the range is exceeded) is set.

NOTE

The unit of parameter changes depending on the limit range designation method.

Query response

{numeric 1},{numeric 2}<newline><^END>

Related commands

`[:SENSe]:CORRection:OFFSet[:STATe]` on page 247

Equivalent key sequence

[**Meas Setup**] - CORRECTION - OFFSET - A|B

[:SENSe]:CORRection:OFFSet[:STATe]

Syntax

[:SENSe]:CORRection:OFFSet[:STATe] {ON|OFF|1|0}

[:SENSe]:CORRection:OFFSet[:STATe]?

Description

Turns ON/OFF the offset correction function.

NOTE

Changing any of the measurement parameter will automatically turn off this function. To avoid this, you have to write a program so that this command is executed after the execution of the measurement parameter setup command (`:CALCulate1:FORMat` and `:CALCulate2:FORMat`).

Parameters

	Description
ON or 1	Turns ON the offset correction function.
OFF or 0 (initial value)	Turns OFF the offset correction function.

Query response

{1|0}<newline><^END>

Related commands

`[:SENSe]:CORRection:OFFSet:DATA` on page 246

`:CALCulate1:FORMat` on page 188

`:CALCulate2:FORMat` on page 191

Equivalent key sequence

[**Meas Setup**] - CORRECTION - OFFSET -ON|OFF

[:SENSe]:CORRection:OPEN[:STATe]

Syntax

[:SENSe]:CORRection:OPEN[:STATe] {ON|OFF|1|0}

[:SENSe]:CORRection:OPEN[:STATe]?

Description

Turns ON/OFF the OPEN correction.

With the OPEN correction set to ON, if you change the cable length (set with the `:CALibration:CABLe[:LENGth]` command) or frequency shift (set with the `:SYSTem:FSHift` command), the OPEN correction is automatically changed to OFF.

Parameters

	Description
ON or 1	Turns ON the OPEN correction.
OFF or 0 (initial value)	Turns OFF the OPEN correction.

Query response

{1|0}<newline><^END>

Related commands

`:CALibration:CABLe[:LENGth]` on page 171

`:SYSTem:FSHift` on page 281

`[:SENSe]:CORRection:COLLect[:ACQuire]` on page 237

Equivalent key sequence

[**Meas Setup**] - CORRECTION - OPEN - ON|OFF

[:SENSe]:CORRection:SHORt[:STATe]

Syntax

[:SENSe]:CORRection:SHORt[:STATe] {ON|OFF|1|0}

[:SENSe]:CORRection:SHORt[:STATe]?

Description

Turns ON/OFF the SHORT correction.

With the SHORT correction set to ON, if you change the cable length (set with the :CALibration:CABLe[:LENGth] command) or frequency shift (set with the :SYSTem:FSHift command), the SHORT correction is automatically changed to OFF.

Parameters

	Description
ON or 1	Turns ON the SHORT correction.
OFF or 0 (initial value)	Turns OFF the SHORT correction.

Query response

{1|0}<newline><^END>

Related commands

:CALibration:CABLe[:LENGth] on page 171

:SYSTem:FSHift on page 281

[:SENSe]:CORRection:COLLect[:ACQuire] on page 237

Equivalent key sequence

[**Meas Setup**] - CORRECTION - SHORT - ON|OFF

[:SENSe]:DETector:DELay1

Syntax [:SENSe]:DETector:DELay1 <numeric>

[:SENSe]:DETector:DELay1?

Description Sets/Gets the waiting time for analog measurement for 120 Hz Frequency measurement.

Parameters

	<numeric 1>
Description	Waiting time for 120 Hz measurement frequency
Range	0 to 100m
Initial value	1.67m
Resolution	1u

Query response {numeric}<newline><^END>

Related commands `[:SENSe]:DETector:DELay2` on page 250

`[:SENSe]:DETector:DELay3` on page 251

Equivalent key sequence No equivalent key can be used on front panel.

[:SENSe]:DETector:DELay2

Syntax [:SENSe]:DETector:DELay2 <numeric>

[:SENSe]:DETector:DELay2?

Description Sets/Gets the waiting time for analog measurement for 1 kHz Frequency measurement.

Parameters

	<numeric 1>
Description	Waiting time for 1 kHz measurement frequency
Range	0 to 100m
Initial value	1m
Resolution	1u

Query response {numeric}<newline><^END>

Related commands

`[:SENSe]:DETector:DELay1` on page 250

`[:SENSe]:DETector:DELay3` on page 251

Equivalent key sequence

No equivalent key can be used on front panel.

[:SENSe]:DETector:DELay3

Syntax

[:SENSe]:DETector:DELay3 <numeric>

[:SENSe]:DETector:DELay3?

Description

Sets/Gets the waiting time for analog measurement for 1 MHz Frequency measurement.

Parameters

	<numeric>
Description	Waiting time for 1 MHz measurement frequency
Range	0 to 100m
Initial value	270u
Resolution	1u

Query response

{numeric}<newline><^END>

Related commands

`[:SENSe]:DETector:DELay1` on page 250

`[:SENSe]:DETector:DELay2` on page 250

Equivalent key sequence

No equivalent key can be used on front panel.

[:SENSe][:FIMPedance]:APERture[:MODE]

Syntax

[:SENSe][:FIMPedance]:APERture[:MODE] {SHORt|MEDium|LONG}

[:SENSe][:FIMPedance]:APERture[:MODE]?

Description

Selects the measurement time (integral time) mode from SHORt, MEDium or LONG. This command is provided to support 4268A/4288A commands.

When Short is selected, the `[:SENSe][:FIMPedance]:APERture:TIME` is set at 1, MED is set at 4, LONG is set at 8.

The Query command returns SHORt when `[:SENSe][:FIMPedance]:APERture:TIME` command is set to 1 or 2, returns MED when APER:TIME is set at 4, and returns LONG when APER:TIME is set at 6 or 8.

For information on the specific measurement time of each mode, see “Specification and Supplemental Performance Characteristics” in the *Operation Manual*.

Parameters

	Description
SHORt (initial value)	Specifies the short mode (Aperture Time=1 or 2).
MEDium	Specifies the medium mode (Aperture Time=4).
LONG	Specifies the long mode (Aperture Time=6 or 8)

Query response

{SHOR|MED|LONG}<newline><^END>

Equivalent key sequence

No equivalent key can be used on front panel.

[:SENSe][:FIMPedance]:APERture:TIME

Syntax

[:SENSe][:FIMPedance]:APERture:TIME {1|2|4|6|8}

[:SENSe][:FIMPedance]:APERture:TIME?

Description

Specifies the measurement time.

For information on the specific measurement time of each mode, see "Specification and Supplemental Performance Characteristics" in the *Operation Manual*.

Parameters

	Description
1, 2, 4, 6, 8	Specifies measurement speed (time).

Query response

{1|2|4|6|8}<newline><^END>

Equivalent key sequence

[**Meas Setup**] - MEAS TIME - INCR+|DECR-

[:SENSe][:FIMPedance]:CONTact1:VERify:BUF1:DATA?

Syntax

[:SENSe][:FIMPedance]:CONTact1:VERify:BUF1:DATA?

Description

Gets the data from buffer1 Contact1. Maximum buffer size is 1000. (Query Only)

NOTE

The transfer format is based on the setting of `:FORMat[:DATA]`.

Related commands

`[:SENSe][:FIMPedance]:CONTact1:VERify:BUF1:FEED:CONTrol:INTerval` on page 254

`[:SENSe][:FIMPedance]:CONTact1:VERify:BUF1:FEED:CONTrol[:STATe]` on page 255

`[:SENSe][:FIMPedance]:CONTact1:VERify:BUF1:POINts` on page 256

Equivalent key sequence

No equivalent key is available on the front panel.

[:SENSe][:FIMPedance]:CONTact1:VERify:BUF1:FEED:CONTrol:INTerval

Syntax

[:SENSe][:FIMPedance]:CONTact1:VERify:BUF1:FEED:CONTrol:INTerval <numeric>

[:SENSe][:FIMPedance]:CONTact:VERify:BUF1:FEED:CONTrol:INTerval?

Description

Sets/Gets the data buffer1 interval for Contact 1.

Parameters

	<numeric>
Description	Interval for Data buffer1.
Range	1 to 100k
Initial value	1
Resolution	1

Query response

<numeric><newline><^END>

Related commands

`[:SENSe][:FIMPedance]:CONTact1:VERify:BUF1:DATA?` on page 253

`[:SENSe][:FIMPedance]:CONTact1:VERify:BUF1:FEED:CONTrol[:STATe]` on page 255

`[:SENSe][:FIMPedance]:CONTact1:VERify:BUF1:POINts` on page 256

Equivalent key sequence

No equivalent key is available on the front panel.

[:SENSe][:FIMPedance]:CONTact1:VERify:BUF1:FEED:CONTrol[:STATe]

Syntax

[:SENSe][:FIMPedance]:CONTact1:VERify:BUF1:FEED:CONTrol[:STATe] {NEVer|ALWays}

[:SENSe][:FIMPedance]:CONTact:VERify:BUF1:FEED:CONTrol[:STATe]?

Description

Determines whether to get data buffer1 for Contact 1.

Parameters

	Description
ALWays	Feeds the contact check measurement data into the data buffer1
NEVer (Initial value)	Does not feeds the contact check measurement data into the data buffer1

Query response

{NEV|ALW}<newline><^END>

Related commands

`[:SENSe][:FIMPedance]:CONTact1:VERify:BUF1:DATA?` on page 253

`[:SENSe][:FIMPedance]:CONTact1:VERify:BUF1:FEED:CONTrol:INTerval` on page 254

`[:SENSe][:FIMPedance]:CONTact1:VERify:BUF1:POINts` on page 256

Equivalent key sequence

No equivalent key is available on the front panel.

[:SENSe][:FIMPedance]:CONTact1:VERify:BUF1:POINts

Syntax

[:SENSe][:FIMPedance]:CONTact1:VERify:BUF1:POINts <numeric>

[:SENSe][:FIMPedance]:CONTact:VERify:BUF1:POINts?

Description

Sets/Gets the number of measurement points of data buffer1.

Parameters

	<numeric>
Description	Number of measurement points for Data buffer1.
Range	1 to 1000
Initial value	1000
Resolution	1

Query response

<numeric><newline><^END>

Related commands

`[:SENSe][:FIMPedance]:CONTact1:VERify:BUF1:DATA?` on page 253

`[:SENSe][:FIMPedance]:CONTact1:VERify:BUF1:FEED:CONTrol:INTerval` on page 254

`[:SENSe][:FIMPedance]:CONTact1:VERify:BUF1:FEED:CONTrol[:STATe]` on page 255

Equivalent key sequence

No equivalent key is available on the front panel.

[:SENSe][:FIMPedance]:CONTact1:VERify:BUF2:DATA?

Syntax

[:SENSe][:FIMPedance]:CONTact1:VERify:BUF2:DATA?

Description

Gets the data from buffer2 of Contact1. Maximum buffer size is 1000. (Query Only)

Related commands

`[:SENSe][:FIMPedance]:CONTact1:VERify:BUF2:FEED:CONTrol:INTerval` on page 257

`[:SENSe][:FIMPedance]:CONTact1:VERify:BUF2:FEED:CONTrol[:STATe]` on page 258

`[:SENSe][:FIMPedance]:CONTact1:VERify:BUF2:POINts` on page 259

Equivalent key sequence

No equivalent key is available on the front panel.

[:SENSe][:FIMPedance]:CONTact1:VERify:BUF2:FEED:CONTrol:INTerval

Syntax

[:SENSe][:FIMPedance]:CONTact1:VERify:BUF2:FEED:CONTrol:INTerval <numeric>

[:SENSe][:FIMPedance]:CONTact:VERify:BUF2:FEED:CONTrol:INTerval?

Description

Sets/Gets the data buffer2 interval for Contact 1.

Parameters

	<numeric>
Description	Interval for Data buffer2.
Range	1 to 100k
Initial value	1
Resolution	1

Query response

<numeric><newline><^END>

Related commands

`[:SENSe][:FIMPedance]:CONTact1:VERify:BUF2:DATA?` on page 256

`[:SENSe][:FIMPedance]:CONTact1:VERify:BUF2:FEED:CONTrol[:STATe]` on page 258

`[:SENSe][:FIMPedance]:CONTact1:VERify:BUF2:POINts` on page 259

Equivalent key sequence

No equivalent key is available on the front panel.

[:SENSe][:FIMPedance]:CONTact1:VERify:BUF2:FEED:CONTrol[:STATe]

Syntax

[:SENSe][:FIMPedance]:CONTact1:VERify:BUF2:FEED:CONTrol[:STATe] {NEVer|ALWays}

[:SENSe][:FIMPedance]:CONTact:VERify:BUF2:FEED:CONTrol[:STATe]?

Description

Determines whether to get data buffer2 for Contact 1.

Parameters

	Description
ALWays	Feeds the contact check measurement data into the data buffer2
NEVer (Initial value)	Does not feeds the contact check measurement data into the data buffer2

Query response

{NEV|ALW}<newline><^END>

Related commands

`[:SENSe][:FIMPedance]:CONTact1:VERify:BUF2:DATA?` on page 256

`[:SENSe][:FIMPedance]:CONTact1:VERify:BUF2:FEED:CONTrol:INTerval` on page 257

`[:SENSe][:FIMPedance]:CONTact1:VERify:BUF2:POINts` on page 259

Equivalent key sequence

No equivalent key is available on the front panel.

[:SENSe][:FIMPedance]:CONTact1:VERify:BUF2:POINts

Syntax

[:SENSe][:FIMPedance]:CONTact1:VERify:BUF2:POINts <numeric>

[:SENSe][:FIMPedance]:CONTact:VERify:BUF2:POINts?

Description

Sets/Gets the number of measurement points of data buffer2.

Parameters

	<numeric>
Description	Number of measurement points for Data buffer 2.
Range	1 to 1000
Initial value	1000
Resolution	1

Query response

<numeric><newline><^END>

Related commands

`[:SENSe][:FIMPedance]:CONTact1:VERify:BUF2:DATA?` on page 256

`[:SENSe][:FIMPedance]:CONTact1:VERify:BUF2:FEED:CONTrol:INTerval` on page 257

`[:SENSe][:FIMPedance]:CONTact1:VERify:BUF2:FEED:CONTrol[:STATe]` on page 258

Equivalent key sequence

No equivalent key is available on the front panel.

[:SENSe][:FIMPedance]:CONTact1:VERify[:STATe]

Syntax

[:SENSe][:FIMPedance]:CONTact1:VERify[:STATE] {ON|OFF|1|0}

[:SENSe][:FIMPedance]:CONTact1:VERify[:STATe]?

Description

Determines whether to use the contact check function.

NOTE

Contact check function is not available at 1 MHz measurement frequency.

Parameters

	Description
ON or 1	Enables the contact check function.
OFF or 0 (initial value)	Disables the contact check function.

Query response

{1|0}<newline><^END>

Related commands

`[:SENSe][:FIMPedance]:CONTact1:VERify:THReshold1` on page 261

`[:SENSe][:FIMPedance]:CONTact1:VERify:THReshold2` on page 262

Equivalent key sequence

[**Meas Setup**] - CONT CHK1- ON|OFF

[:SENSe][:FIMPedance]:CONTact1:VERify:THReshold1

Syntax

[:SENSe][:FIMPedance]:CONTact1:VERify:THReshold1 <numeric>

[:SENSe][:FIMPedance]:CONTact1:VERify:THReshold1?

Description

Sets/Gets the Threshold1 value for Contact check. Threshold1 is a parameter which is proportional contact resistance of Hp or Hc. It is recommended to use the default (initial) value.

NOTE

The available range for Hp, Hc and Lc is 220 µF to 1 mF at 120 Hz and 22 µF to 100 µF at 1 kHz. The contact resistance for Lp cannot be detected by contact check feature for this range.

Parameters

	<numeric>
Description	Threshold1 value for Contact1.
Range	0 to 1
Initial value	0.1
Resolution	0.01

Query response

<numeric><newline><^END>

Related commands

`[:SENSe][:FIMPedance]:CONTact1:VERify[:STATe]` on page 260

`[:SENSe][:FIMPedance]:CONTact1:VERify:THReshold2` on page 262

Equivalent key sequence

[**Meas Setup**] - CONT CHK- CC1 TH1

[:SENSe][:FIMPedance]:CONTact1:VERify:THReshold2

Syntax

```
[:SENSe][:FIMPedance]:CONTact1:VERify:THReshold2 <numeric>
[:SENSe][:FIMPedance]:CONTact1:VERify:THReshold2?
```

Description

Sets/Gets the Threshold2 value for Contact check. Threshold2 is a parameter which is proportional contact resistance of Lp or Lc. The recommended setting value is described in the *Operation Manual*.

NOTE The contact failure at Lp can not be detected at the range of 220 uF to 1 mF at 120 Hz and 22 uF to 100 uF at 1 kHz. At the same range, the threshold1 is used instead of this for Lc. (Therefore, the threshold2 is not used at the range).

Parameters

	<numeric>
Description	Threshold2 value for contact 1
Range	0 to 1
Initial value	1
Resolution	0.01

Query response

<numeric><newline><^END>

Related commands

`[:SENSe][:FIMPedance]:CONTact1:VERify[:STATe]` on page 260

`[:SENSe][:FIMPedance]:CONTact1:VERify:THReshold1` on page 261

Equivalent key sequence

[**Meas Setup**] - CONT CHK- CC1 TH2

[:SENSe][:FIMPedance]:CREJect:LIMit

Syntax

[:SENSe][:FIMPedance]:CREJect:LIMit <numeric>

[:SENSe][:FIMPedance]:CREJect:LIMit

Description

Sets the boundary value (percentage of the measurement range) within the detection range of Low C when you turn on the Low C reject function. The measurement range to which the set value is applied differs depending on the setup of the measurement range mode as shown below:

- When in auto-range mode (ON has been specified with the `[:SENSe][:FIMPedance]:RANGe:AUTO` command):

  When the measurement frequency is in the 120 Hz: 10E-9 F (100 pF) range

  When the measurement frequency is in the 1 kHz: 100E-12 F (100 pF) range

  When the measurement frequency is in the 1 MHz: 1E-12 F (1 pF) range

- When in the fixed range mode (OFF has been specified with the `[:SENSe][:FIMPedance]:RANGe:AUTO` command):

For example, if you make a measurement with the range fixed to the 1 μF range and specify 1%, Low C is detected if the measured value of the primary parameter (Cs or Cp) is 10 nF or less.

For information on the screen display, GPIB/USB/LAN output, and so on when Low C is detected, refer to Appendix D, “Operations when Overload, No Contact, or Low C is Detected,”.

Parameters

	<numeric>
Description	Boundary value
Range	0 to 10
Initial value	0
Unit	% (percent)

If the specified parameter is out of the allowable setup range, the minimum value (if the lower limit of the range is not reached) or the maximum value (if the upper limit of the range is exceeded) is set.

MAX or MIN can be used to specify the parameters.

Query response

{numeric}<newline><^END>

Related commands `[:SENSe][:FIMPedance]:CREJect[:STATe]` on page 264

`[:SENSe][:FIMPedance]:RANGe[:UPPer]` on page 266

`[:SENSe][:FIMPedance]:RANGe:AUTO` on page 265

Equivalent key sequence [**Meas Setup**] - LOW C REJ - INCR+|DECR-

[:SENSe][:FIMPedance]:CREJect[:STATe]

Syntax [:SENSe][:FIMPedance]:CREJect[:STATe] {ON|OFF|1|0}

[:SENSe][:FIMPedance]:CREJect[:STATe]?

Description Enables/disables the Low C reject function.

When you enable the Low C reject function, if the measured value of the primary parameter (Cp or Cs) is too small (equal to or less than the boundary value specified with the `[:SENSe][:FIMPedance]:CREJect:LIMit` command), Low C is detected. For information on the screen display, GPIB/USB/LAN output, and so on when Low C is detected, refer to Appendix D, “Operations when Overload, No Contact, or Low C is Detected,”.

Parameters

	Description
ON or 1	Enables the Low C reject function.
OFF or 0 (initial value)	Disables the Low C reject function.

Query response {1|0}<newline><^END>

Related commands `[:SENSe][:FIMPedance]:CREJect:LIMit` on page 263

Equivalent key sequence [**Meas Setup**] - LOW C REJ - ON|OFF

[:SENSe][:FIMPedance]:RANGe:AUTO

Syntax

[:SENSe][:FIMPedance]:RANGe:AUTO {ON|OFF|1|0}

[:SENSe][:FIMPedance]:RANGe:AUTO?

Description

Selects the measurement range mode from the auto range (automatic range switching) or hold range (fixed range).

Setting the measurement range (set with the `[:SENSe][:FIMPedance]:RANGe[:UPPer]` command) automatically selects the hold range mode.

Parameters

	Description
ON or 1 (initial value)	Specifies the auto range mode.
OFF or 0	Specifies the hold range mode.

Query response

{1|0}<newline><^END>

Related commands

`[:SENSe][:FIMPedance]:RANGe[:UPPer]` on page 266

Equivalent key sequence

[**Meas Setup**] - RANGE - AUTO|HOLD

[:SENSe][:FIMPedance]:RANGe[:UPPer]

Syntax

[:SENSe][:FIMPedance]:RANGe[:UPPer] {1p|2.2p|4.7p|10p|22p|47p|100p|220p|470p|1n|2.2n|4.7n|10n|22n|47n|100n|220n|470n|1µ|2.2µ|4.7µ|10µ|22µ|47µ|100µ|220µ|470µ|1m}

[:SENSe][:FIMPedance]:RANGe[:UPPer]?

Description

Sets the measurement range.

When you set the measurement range with this command, the measurement range mode is automatically set to the hold range (specified to OFF with the `[:SENSe][:FIMPedance]:RANGe:AUTO` command).

NOTE

The measurement range varies depending upon the frequency setting.

Parameters

	<numeric>
Description	Measurement Range
Range	120 Hz: 10 nF - 1 mF 1 kHz: 100 pF - 100 µF 1 MHz: 1 pF-1 nF
Initial Value	100 µF

If one of the settable values is not specified for the parameter, a suitable measurement range whose recommended range includes the specified parameter (for example, 10E-9 if the specified parameter is 5E-9) is set.

MAX or MIN can be used to specify the parameters.

Query response

{1pF|2.2pF|4.7pF|10pF|22pF|47pF|100pF|220pF|470pF|1nF|2.2nF|4.7nF|10nF|22nF|47nF|100nF|220nF|470nF|1µF|2.2µF|4.7µF|10µF|22uF|47µF|100µF|220µF|470µF|1mF} <newline><^END>

If the measurement range mode is Auto, the measurement range used in the immediately preceding measurement is read out as the query response.

Related commands

`[:SENSe][:FIMPedance]:RANGe:AUTO` on page 265

Equivalent key sequence

[**Meas Setup**] - RANGE - AUTO|HOLD

:SOURce:FREQuency[:CW]

Syntax

:SOURce:FREQuency[:CW] <numeric> [Hz|kHz|MHz]

:SOURce:FREQuency[:CW]?

Description

Sets the measurement frequency.

Depending on this setting, the available measurement range varies (set with the `[:SENSe][:FIMPedance]:RANGe[:UPPer]` command). If changing the measurement frequency results in a conflict with the measurement range setting, an acceptable range is automatically selected.

Parameters

	Description
Range	Option 001 = 120 Hz, 1kHz and 1MHz
	Option 002 = 120 Hz and 1 kHz
Initial value	1 kHz

Query response

{120|1E3|1E6}<newline><^END>

Related commands

`[:SENSe][:FIMPedance]:RANGe[:UPPer]` on page 266

Equivalent key sequence

[**Meas Setup**] - FREQ - 120 Hz|1 kHz|1 MHz

:SOURce:VOLTage:ALC[:STATe]

Syntax

:SOURce:VOLTage:ALC[:STATe] {ON|OFF|1|0}

:SOURce:VOLTage:ALC[:STATe]?

Description

Determines whether to use the signal level compensation function (LVL COMP).

Parameters

	Description
ON or 1	Enables the signal level compensation function.
OFF or 0 (Initial value)	Disables the signal level compensation function.

Query Response

{1|0}<newline><^END>

Equivalent key sequence

[**Meas Setup**] - LVL COMP- ON|OFF

:SOURce:VOLTage[:LEVel][:IMMediate][:AMPLitude]

Syntax

```
:SOURce:VOLTage[:LEVel][:IMMediate][:AMPLitude] <numeric>[mV|V]
:SOURce:VOLTage[:LEVel][:IMMediate][:AMPLitude]?
```

Description

Sets the measurement signal level.

Parameters

	<numeric>
Description	Measurement signal level
Range	100m to 1
Initial value	1
Unit	V
Resolution	10m

If the specified parameter is out of the allowable setup range, the minimum value (if the lower limit of the range is not reached) or the maximum value (if the upper limit of the range is exceeded) is set.

NOTE

A fraction below the resolution is rounded off. MAX or MIN can be used to specify the parameters.

Query response

{numeric}<newline><^END>

Equivalent key sequence

[**Meas Setup**] - LEVEL

:SOURce:VOLTage:MODE

Syntax

:SOURce:VOLTage:MODE {CONTinuous|SYNChronous}

:SOURce:VOLTage:MODE?

Description

Determines whether to use the synchronous source function (to output the measurement signal only during measurement) or not (to always output the measurement signal). The synchronous source function lets you set the source delay time with the ":TRIGger[:SEQ1]:DELay" on page 303, to suspend the signal output during the waiting time after a trigger is generated.

Parameters

	Description
CONTinuous	Always outputs the measurement signal.
SYNChronous	Outputs the measurement signal only during measurement.

Query Response

{CONT|SYNC}<newline><^END>

Equivalent key sequence

[**Meas Setup**] - SYNC SRC- ON|OFF

:STATus:OPERation:CONDition?

Syntax

:STATus:OPERation:CONDition?

Description

Reads out the value of the Operation Status Condition register. (Query only)

Query response

{numeric}<newline><^END>

Equivalent key sequence

No equivalent key is available on the front panel.

:STATus:OPERation:ENABle

Syntax :STATus:OPERation:ENABle <numeric>

:STATus:OPERation:ENABle?

Description Sets the value of the Operation Status Enable register.

Parameters

	<numeric>
Description	The value of the Enable register.
Range	0 to 32767
Initial value	0
Resolution	1

Query response {numeric}<newline><^END>

Related commands `*SRE` on page 162

`:STATus:PRESet` on page 271

Equivalent key sequence No equivalent key is available on the front panel.

:STATus:OPERation[:EVENt]?

Syntax :STATus:OPERation[:EVENt]?

Description Reads out the value of the Operation Status Event register. (Query only)

Query response {numeric}<newline><^END>

Related commands `*CLS` on page 157

Equivalent key sequence No equivalent key is available on the front panel.

:STATus:OPERation:UPDate

Syntax :STATus:OPERation:UPDate {ON|OFF|1|0}

:STATus:OPERation:UPDate?

Description Enables/Disables update of the Operation Status Event register. The disabling of status register can shorten the EOM time.

Parameters

	Description
ON or 1	Enables the update of the Operation Status Event register.
OFF or 0 (initial value)	Disables the update of the Operation Status Event register.

Query response {1|0}<newline><^END>

Related commands `:STATus:OPERation:ENABle` on page 270

Equivalent key sequence No equivalent key is available on the front panel.

:STATus:PRESet

Syntax :STATus:PRESet

Description Initializes the Operation Status register and the Questionable Status register. (No query)

Equivalent key sequence No equivalent key is available on the front panel.

:STATus:QUEStionable:CONDition?

Syntax :STATus:QUEStionable:CONDition?

Description Reads out the value of the Questionable Status Condition register. However, the E4981A does not support the Questionable Status register. Therefore, executing this command has no effect.(Query only)

Query response {numeric}<newline><^END>

The E4981A does not support the Questionable Status register. Therefore, the query response is always 0.

Equivalent key sequence No equivalent key is available on the front panel.

:STATus:QUEStionable:ENABle

Syntax :STATus:QUEStionable:ENABle <numeric>

:STATus:QUEStionable:ENABle?

Description Sets the value of the Questionable Status Enable register. However, the E4981A does not support the Questionable Status register. Therefore, executing this command has no effect.

Query response {numeric}<newline><^END>

Equivalent key sequence No equivalent key is available on the front panel.

:STATus:QUEStionable[:EVENt]?

Syntax :STATus:QUEStionable[:EVENt]?

Description Reads out the value of the Questionable Status Event register. However, the E4981A does not support the Questionable Status register. Therefore, executing this command has no effect.(Query only)

Query response {numeric}<newline><^END>

The E4981A does not support the Questionable Status register. Therefore, the query response is always 0.

Equivalent key sequence No equivalent key is available on the front panel.

:SYSTem:BEEPer[:IMMediate]

Syntax

:SYSTem:BEEPer[:IMMediate]

Description

Produces a beep sound.

If the beep sound is disabled (OFF has been specified with the `:SYSTem:BEEPer:STATe` command), no beep sound is produced even if you execute this command. (No query)

Related commands

`:SYSTem:BEEPer:STATe` on page 273

Equivalent key sequence

No equivalent key is available on the front panel.

:SYSTem:BEEPer:STATe

Syntax

:SYSTem:BEEPer:STATe {ON|OFF|1|0}

:SYSTem:BEEPer:STATe?

Description

Turns ON/OFF the beep output.

This command has the same function as the `:CALCulate1:COMParator:BEEPer[:STATe]` command.

Parameters

	Description
ON or 1 (initial value)	Enables the beep sound.
OFF or 0	Disables the beep sound.

Query response

{1|0}<newline><^END>

Related commands

`:CALCulate1:COMParator:BEEPer[:STATe]` on page 174

Equivalent key sequence

[**System**] - SYSTEM CONFIG - BEEPER ENABLED - ON|OFF

:SYSTem:BEEPer:TONE

Syntax

```
:SYSTem:BEEPer:TONE <numeric>
:SYSTem:BEEPer:TONE?
```

Description

Selects the beep sound tone.

Parameter

	<Numeric>
Range	1 to 5
Initial (Factory set) value	3
Resolution	1

NOTE

The value of this command is not changed with `*RST` and `:SYSTem:PRESet` and is only changed with Factory Default reset (available only through Front Panel).

Query response

{1|2|3|4|5}<newline><^END>

Equivalent key

[**System**] - SYSTEM CONFIG - BEEPER TONE - TONE1|TONE2|TONE3|TONE4|TONE5

:SYSTem:COMMunicate:GPIB[:SELF]:ADDRess

Syntax

:SYSTem:COMMunicate:GPIB[:SELF]:ADDRess <numeric>

:SYSTem:COMMunicate:GPIB[:SELF]:ADDRess?

Description

Sets the GPIB address.

Parameter

<Numeric>	
Range	0 to 30
Initial (Factory set) value	17
Resolution	1

NOTE

The value of this command is not changed with *RST and :SYSTem:PRESet and is only changed with Factory Default reset (available only through Front Panel).

Query response

{numeric}<newline><^END>

Equivalent key

[**System**] - SYSTEM CONFIG - GPIB ADDR

:SYSTem:COMMunicate:LAN[:SELF]:ADDRess

Syntax

:SYSTem:COMMunicate:LAN[:SELF]:ADDRess <String>

:SYSTem:COMMunicate:LAN[:SELF]:ADDRess?

Description

Sets the static IP address.

Parameter

<String>	
Initial (Factory set) value	"192.168.1.101"

NOTE

The value of this command is not changed with *RST and :SYSTem:PRESet and is only changed with Factory Default reset (available only through Front Panel).

Query response

{“string”}<newline><^END>

Equivalent key

[**System**] - SYSTEM CONFIG - MANUAL IP ADDR - ENTER

:SYSTem:COMMunicate:LAN[:SELF]:CONFigure

Syntax

:SYSTem:COMMunicate:LAN[:SELF]:CONFigure {AUTO|MANual}

:SYSTem:COMMunicate:LAN[:SELF]:CONFigure?

Description

Sets the IP configuration setup method i.e. Auto/Manual.

Parameters

	Description
AUTO (Initial [Factory set] value)	Sets the IP configuration setup to auto mode.
MANual	Sets the IP configuration setup to manual mode.

NOTE

The value of this command is not changed with *RST and :SYSTem:PRESet and is only changed with Factory Default reset (available only through Front Panel).

Query response

{AUTO|MAN}<newline><^END>

Equivalent key sequence

[**System**] - SYSTEM CONFIG - IP CONFIG - AUTO|MANUAL

:SYSTem:COMMunicate:LAN[:SELF]:CONTrol?

Syntax

:SYSTem:COMMunicate:LAN[:SELF]:CONTrol?

Description

Returns the SOCKET control port number. If the parser is a SOCKET, it will return a number from 5000 to 5100. Otherwise, 0. (Query Only)

Query response

{numeric}<newline><^END>

Equivalent key

No equivalent key is available on the front panel.

:SYSTem:COMMunicate:LAN[:SELF]:CURRent:ADDRess?

Syntax

:SYSTem:COMMunicate:LAN[:SELF]:CURRent:ADDRess?

Description

Returns the current IP address. (Query Only)

Query response

{“string”}<newline><^END>

Equivalent key

No equivalent key is available on the front panel.

:SYSTem:COMMunicate:LAN[:SELF]:CURRent:DGATeway?

Syntax :SYSTem:COMMunicate:LAN[:SELF]:CURRent:DGATeway?

Description Returns the current Gateway address. (Query Only)

Query response {“string”}<newline><^END>

Equivalent key No equivalent key is available on the front panel.

:SYSTem:COMMunicate:LAN[:SELF]:CURRent:SMASk?

Syntax :SYSTem:COMMunicate:LAN[:SELF]:CURRent:SMASk?

Description Returns the current Subnet Mask. (Query Only)

Query response {“string”}<newline><^END>

Equivalent key No equivalent key is available on the front panel.

:SYSTem:COMMunicate:LAN[:SELF]:DGATeway

Syntax :SYSTem:COMMunicate:LAN[:SELF]:DGATeway <String>

:SYSTem:COMMunicate:LAN[:SELF]:DGATeway?

Description Sets the static Gateway address.

Parameter

	<String>
Initial (Factory set) value	"0.0.0.0"

NOTE The value of this command is not changed with `*RST` and `:SYSTem:PRESet` and is only changed with Factory Default reset (available only through Front Panel).

Query response {“string”}<newline><^END>

Equivalent key [**System**] - SYSTEM CONFIG - MANUAL GATEWAY - ENTER

:SYSTem:COMMunicate:LAN[:SELF]:MAC?

Syntax :SYSTem:COMMunicate:LAN[:SELF]:MAC?

Description Returns the MAC address. (Query Only)

Query response {“string”}<newline><^END>

Equivalent key No equivalent key is available on the front panel.

:SYSTem:COMMunicate:LAN[:SELF]:PRESet

Syntax :SYSTem:COMMunicate:LAN[:SELF]:PRESet

Description Presets the network settings and restarts the network. (No Query)

Equivalent key [**Preset**] - LAN RESET- OK

:SYSTem:COMMunicate:LAN[:SELF]:RESTart

Syntax :SYSTem:COMMunicate:LAN[:SELF]:RESTart

Description Restarts the network (No Query)

Equivalent key [**System**] - SYSTEM CONFIG - IP CONFIG - RESTART NETWORK

:SYSTem:COMMunicate:LAN[:SELF]:SMASk

Syntax :SYSTem:COMMunicate:LAN[:SELF]:SMASk <String>

:SYSTem:COMMunicate:LAN[:SELF]:SMASk?

Description Sets the static Subnet Mask.

Parameter

	<String>
Initial (Factory set) value	"255.255.255.0"

NOTE The value of this command is not changed with `*RST` and `:SYSTem:PRESet` and is only changed with Factory Default reset (available only through Front Panel).

Query response {“string”}<newline><^END>

Equivalent key [**System**] - SYSTEM CONFIG - MANUAL SUBNET MASK- ENTER

:SYSTem:DATE

Syntax :SYSTem:DATE <year>,<month>,<day>

:SYSTem:DATE?

Description Sets the date in the internal clock.

Parameter

	<year>
Range	2000 to 2098
Unit	years
Resolution	1

	<month>
Range	1 to 12
Unit	months
Resolution	1

	<day>
Range	1 to 31 (maximum number of days depends upon month).
Unit	days
Resolution	1

Query response {“string”}<newline><^END>

string: {year, month, day}

Equivalent key [**System**] - SYSTEM CONFIG - DATE/TIME - DATE - YEAR|MONTH|DAY

:SYSTem:ERRor[:NEXT]?

Syntax	:SYSTem:ERRor[:NEXT]?
Description	Reads out the oldest error remaining in the E4981A's error queue. The size of the error queue is 100. Executing the `*CLS` command clears errors stored in the error queue. (Query only)
Query response	{numeric},{string}<newline><^END> {numeric}: Error number {string}: Error message (a string within double quotation marks (“)) If no error is stored in the error queue, 0 is read out as the error number and “No error” as the error message.
Related commands	`*CLS` on page 157
Equivalent key sequence	No equivalent key is available on the front panel.

:SYSTem:FSHift

Syntax

:SYSTem:FSHift <numeric>

:SYSTem:FSHift?

Description

When you make a measurement with the measurement frequency set to 1 MHz, specifies the shift of the signal frequency (frequency shift value) actually applied to the DUT relative to 1 MHz as a percentage of 1 MHz.

Parameters

	<numeric>
Description	The value of the frequency shift.
Range	-2 to 2
Initial (Factory set) value	0
Unit	% (percent)
Resolution	1

If the specified parameter is out of the allowable setup range, the minimum value or the maximum value is set.

NOTE

A fraction below the resolution is rounded off. MAX or MIN can be used to specify the parameters.

This command is not available in Option 002.

The value of this command is not changed with `*RST` and `:SYSTem:PRESet` and is only changed with Factory Default reset (available only through Front Panel).

Query response

{numeric}<newline><^END>

Equivalent key sequence

[**Meas Setup**] - FREQ SHFT- 0%|1%|-1%|2%|-2%

:SYSTem:HANDler:TRIGger:VOLTage

Syntax

:SYSTem:HANDler:TRIGger:VOLTage <numeric>

:SYSTem:HANDler:TRIGger:VOLTage?

Description

Sets the handler trigger input voltage.

Parameters

	<numeric>
Description	The value of the handler trigger input voltage
Range	5 to 24
Initial (Factory set) value	24
Unit	V (voltage)
Resolution	100m

NOTE

The value of this command is not changed with *RST and :SYSTem:PRESet and is only changed with Factory Default reset (available only through Front Panel).

CAUTION

The product may get damaged if incorrect voltage is set to Handler Trigger. Take caution while changing the Handler Trigger voltage.

Query response

{numeric}<newline><^END>

Equivalent key

No equivalent key is available on the front panel.

:SYSTem:KLOCk

Syntax

```
:SYSTem:KLOCk {ON|OFF|1|0}
:SYSTem:KLOCk?
```

Description

Locks or unlocks the front panel keys.

Parameters

	Description
ON or 1	Locks the keys.
OFF or 0 (initial value)	Unlocks the keys.

Query response

{1|0}<newline><^END>

Equivalent key sequence

[**Local/Lock**]

:SYSTem:PRESet

Syntax :SYSTem:PRESet

Description Resets the instrument to the preset state.

The preset state is different from that when resetting is done by using the *RST command. For details, refer to *Operation Manual, Appendix C "Initial Setting"* (No query)

Related commands *RST on page 160

Equivalent key sequence [**Preset**] - CLEAR SETTING - OK

:SYSTem:RESTart

Syntax :SYSTem:RESTart

Description Reboots the instrument immediately. (No Query)

Equivalent key No equivalent key is available on the front panel.

:SYSTem:SCANner:TRIGger:VOLTage

Syntax

:SYSTem:SCANner:TRIGger:VOLTage <numeric>

:SYSTem:SCANner:TRIGger:VOLTage?

Description

Sets the scanner trigger input voltage.

Parameters

	<numeric>
Description	The value of the scanner trigger input voltage
Range	5 to 15
Initial (factory set) value	15
Unit	V (voltage)
Resolution	100m

NOTE

The value of this command is not changed with `*RST` and `:SYSTem:PRESet` and is only changed with Factory Default reset (available only through Front Panel).

CAUTION

The product may get damaged if incorrect voltage is set to Scanner Trigger. Take caution while changing the Scanner Trigger voltage.

Query response

{numeric}<newline><^END>

Equivalent key

No equivalent key is available on the front panel.

:SYSTem:TIME

Syntax

```
:SYSTem:TIME <hour>,<minute>,<second>
:SYSTem:TIME?
```

Description Sets the time in the internal clock.

Parameter

	<hour>
Range	0 to 23
Unit	hours
Resolution	1

	<minute>
Range	0 to 59
Unit	minutes
Resolution	1

	<second>
Range	0 to 59
Unit	seconds
Resolution	1

Query response {“string”}<newline><^END>

string: {hour, minute, second}

Equivalent key [**System**] - SYSTEM CONFIG - DATE/TIME - TIME - HOUR|MINUTE|SECOND

:SYSTem:TZONe

Syntax

:SYSTem:TZONe <hour>[,<minute>]

:SYSTem:TZONe?

Description

Configures the time zone. Sets the time difference from Greenwich mean time (GMT).

Parameter

	<hour>
Initial (Factory set) value	0
Range	-12 to 15
Unit	hours
Resolution	1

	<minute>
Initial (Factory set) value	0
Range	-45 to 45
Unit	minutes
Resolution	15

NOTE

The value of this command is not changed with `*RST` and `:SYSTem:PRESet` and is only changed with Factory Default reset (available only through Front Panel).

Query response

{“string”}<newline><^END>

string: {hour, minute}

Equivalent key

[**System**] - SYSTEM CONFIG - TIME ZONE - HOUR INCR++|MINUTE INCR+|MINUTE DECR-|HOUR DECR--

:TEST:HANDler:BIN

Syntax :TEST:HANDler:BIN <numeric>

Description This command sets the Handler Bin No for test purpose and can be used to check/troubleshoot the Test pin signal of Handler interface. (No query)

NOTE When `:TEST:HANDler:MODE` is set to ON, the interface pin signal value can be controlled/read by: TEST:HAND:xxxx commands. After completing the test, `:TEST:HANDler:MODE` should set to OFF so that handler interface pin signal is generated according to the actual measurement result.

Parameters

Numeric Value	Description
0	Out of Bins
1-9	BIN 1 to 9
10	AUX BIN
11	All Off

Related Commands `:TEST:HANDler:COMP` on page 289

`:TEST:HANDler:KEYLock?` on page 289

`:TEST:HANDler:MODE` on page 290

Equivalent key sequence No equivalent key is available on the front panel.

:TEST:HANDler:COMP

Syntax :TEST:HANDler:COMP {PHI|PLO|SREJ|OFF}

Description This command sets the Handler Comparator Function value for test purpose and can be used to check/troubleshoot the Test pin signal of Handler interface. (No query)

NOTE When `:TEST:HANDler:MODE` is set to ON, the interface pin signal value can be controlled/read by: TEST:HAND:xxxx commands. After completing the test, `:TEST:HANDler:MODE` should set to OFF so that handler interface pin signal is generated according to the actual measurement result.

Related commands `:TEST:HANDler:BIN` on page 288

`:TEST:HANDler:KEYLock?` on page 289

`:TEST:HANDler:MODE` on page 290

Equivalent key sequence No equivalent key is available on the front panel.

:TEST:HANDler:KEYLock?

Syntax :TEST:HANDler:KEYLock?

Description This command gets the Handler/Key_Lock signal level for test purpose and can be used to check/troubleshoot the Test pin signal of Handler interface. (Query only)

NOTE When `:TEST:HANDler:MODE` is set to ON, the interface pin signal value can be controlled/read by: TEST:HAND:xxxx commands. After completing the test, `:TEST:HANDler:MODE` should set to OFF so that handler interface pin signal is generated according to the actual measurement result.

Query Response {1|0}<newline><^END>

Related commands `:TEST:HANDler:BIN` on page 288

`:TEST:HANDler:COMP` on page 289

`:TEST:HANDler:MODE` on page 290

Equivalent key sequence No equivalent key is available on the front panel.

:TEST:HANDler:MODE

Syntax

:TEST:HANDler:MODE {ON|OFF|1|0}

:TEST:HANDler:MODE?

Description

This command sets the Handler Comparator Function value for test purpose and can be used to check/troubleshoot the Test pin signal of Handler interface.

NOTE

When `:TEST:HANDler:MODE` is set to ON, the interface pin signal value can be controlled/read by: TEST:HAND:xxxx commands. After completing the test, `:TEST:HANDler:MODE` should set to OFF so that handler interface pin signal is generated according to the actual measurement result.

Parameters

	Description
ON or 1	Sets Handler mode to ON.
OFF or 0 (initial value)	Sets Handler mode to OFF.

Query response

{1|0}<newline><^END>

Related commands

`:TEST:HANDler:BIN` on page 288

`:TEST:HANDler:COMP` on page 289

`:TEST:HANDler:KEYLock?` on page 289

Equivalent key sequence

No equivalent key is available on the front panel.

:TEST:HANDler:STATus:ALARm

Syntax

:TEST:HANDler:STATus:ALARm {ON|OFF|1|0}

Description

This command sets the Handler Alarm signal for test purpose and can be used to check/troubleshoot the Test pin signal of Handler interface. (No Query)

NOTE

When `:TEST:HANDler:MODE` is set to ON, the interface pin signal value can be controlled/read by: TEST:HAND:xxxx commands. After completing the test, `:TEST:HANDler:MODE` should set to OFF so that handler interface pin signal is generated according to the actual measurement result.

Parameters

	Description
ON	Sets Handler alarm signal to ON.
OFF	Sets Handler alarm signal to OFF.

Related commands

`:TEST:HANDler:STATus:INDex` on page 293

`:TEST:HANDler:STATus:NC` on page 294

`:TEST:HANDler:STATus:OVLD` on page 295

`:TEST:HANDler:STATus:RDYTrig` on page 296

Equivalent key sequence

No equivalent key is available on the front panel.

:TEST:HANDler:STATus:EOM

Syntax :TEST:HANDler:STATus:EOM {ON|OFF|1|0}

Description This command sets the Handler End of Measurement (EOM) status for test purpose and can be used to check/troubleshoot the Test pin signal of Handler interface. (No Query)

NOTE When `:TEST:HANDler:MODE` is set to ON, the interface pin signal value can be controlled/read by: TEST:HAND:xxxx commands. After completing the test, `:TEST:HANDler:MODE` should set to OFF so that handler interface pin signal is generated according to the actual measurement result.

Parameters

	Description
ON	Sets Handler (End of Measurement) EOM status to ON.
OFF	Sets Handler (End of Measurement) EOM status to OFF.

Related commands

`:TEST:HANDler:STATus:ALARm` on page 291

`:TEST:HANDler:STATus:INDex` on page 293

`:TEST:HANDler:STATus:NC` on page 294

`:TEST:HANDler:STATus:OVLD` on page 295

`:TEST:HANDler:STATus:RDYTrig` on page 296

Equivalent key sequence No equivalent key is available on the front panel.

:TEST:HANDler:STATus:INDex

Syntax :TEST:HANDler:STATus:INDex {ON|OFF|1|0}

Description This command sets the Handler Index value for test purpose and can be used to check/troubleshoot the Test pin signal of Handler interface. (No Query)

NOTE When `:TEST:HANDler:MODE` is set to ON, the interface pin signal value can be controlled/read by: TEST:HAND:xxxx commands. After completing the test, `:TEST:HANDler:MODE` should set to OFF so that handler interface pin signal is generated according to the actual measurement result.

Parameters

	Description
ON	Sets Handler index signal to ON.
OFF	Sets Handler index signal to OFF.

Related commands

`:TEST:HANDler:STATus:ALARm` on page 291

`:TEST:HANDler:STATus:NC` on page 294

`:TEST:HANDler:STATus:OVLD` on page 295

`:TEST:HANDler:STATus:RDYTrig` on page 296

Equivalent key sequence No equivalent key is available on the front panel.

:TEST:HANDler:STATus:NC

Syntax :TEST:HANDler:STATus:NC {ON|OFF|1|0}

Description This command sets the Handler No Contact/Low C Reject signal for test purpose and can be used to check/troubleshoot the Test pin signal of Handler interface. (No Query)

NOTE When `:TEST:HANDler:MODE` is set to ON, the interface pin signal value can be controlled/read by: TEST:HAND:xxxx commands. After completing the test, `:TEST:HANDler:MODE` should set to OFF so that handler interface pin signal is generated according to the actual measurement result.

Parameters

	Description
ON	Sets No Contact/Low C Reject status to ON.
OFF	Sets No Contact/Low C Reject status to OFF.

Related commands

`:TEST:HANDler:STATus:ALARm` on page 291

`:TEST:HANDler:STATus:INDex` on page 293

`:TEST:HANDler:STATus:OVLD` on page 295

`:TEST:HANDler:STATus:RDYTrig` on page 296

Equivalent key sequence No equivalent key is available on the front panel.

:TEST:HANDler:STATus:OVLD

Syntax

:TEST:HANDler:STATus:OVLD {ON|OFF|1|0}

Description

This command sets the Handler Overload signal status for test purpose and can be used to check/troubleshoot the Test pin signal of Handler interface. (No Query)

NOTE

When `:TEST:HANDler:MODE` is set to ON, the interface pin signal value can be controlled/read by: TEST:HAND:xxxx commands. After completing the test, `:TEST:HANDler:MODE` should set to OFF so that handler interface pin signal is generated according to the actual measurement result.

Parameters

	Description
ON	Sets Handler Overload signal status to ON.
OFF	Sets Handler Overload signal status to OFF.

Related commands

`:TEST:HANDler:STATus:ALARm` on page 291

`:TEST:HANDler:STATus:INDex` on page 293

`:TEST:HANDler:STATus:NC` on page 294

`:TEST:HANDler:STATus:RDYTrig` on page 296

Equivalent key sequence

No equivalent key is available on the front panel.

:TEST:HANDler:STATus:RDYTrig

Syntax :TEST:HANDler:STATus:RDYTrig {ON|OFF|1|0}

Description This command sets the Handler Ready for Trigger signal for test purpose and can be used to check/troubleshoot the Test pin signal of Handler interface. (No Query)

NOTE

When :TEST:HANDler:MODE is set to ON, the interface pin signal value can be controlled/read by: TEST:HAND:xxxx commands. After completing the test, :TEST:HANDler:MODE should set to OFF so that handler interface pin signal is generated according to the actual measurement result.

Parameters

	Description
ON	Sets Handler Ready for Trigger signal status to ON.
OFF	Sets Handler Ready for Trigger signal status to OFF.

Related commands

`:TEST:HANDler:STATus:ALARm` on page 291

`:TEST:HANDler:STATus:INDex` on page 293

`:TEST:HANDler:STATus:NC` on page 294

`:TEST:HANDler:STATus:OVLD` on page 295

Equivalent key sequence No equivalent key is available on the front panel.

:TEST:HANDler:TRIGger?

Syntax :TEST:HANDler:TRIGger?

Description This command gets the Handler Trigger signal for test purpose and can be used to check/troubleshoot the Test pin signal of Handler interface. (Query Only)

NOTE When `:TEST:HANDler:MODE` is set to ON, the interface pin signal value can be controlled/read by: TEST:HAND:xxxx commands. After completing the test, `:TEST:HANDler:MODE` should set to OFF so that handler interface pin signal is generated according to the actual measurement result.

Query response {1|0}<newline><^END>

Related commands `:TEST:REAR:TRIGger?` on page 297

Equivalent key sequence No equivalent key is available on the front panel.

:TEST:REAR:TRIGger?

Syntax :TEST:REAR:TRIGger?

Description This command gets the BNC Trigger signal for test purpose and can be used to check/troubleshoot the Test pin signal of Handler interface. (Query Only)

NOTE When `:TEST:HANDler:MODE` is set to ON, the interface pin signal value can be controlled/read by: TEST:HAND:xxxx commands. After completing the test, `:TEST:HANDler:MODE` should set to OFF so that handler interface pin signal is generated according to the actual measurement result.

Query response {1|0}<newline><^END>

Related commands `:TEST:HANDler:TRIGger?` on page 297

Equivalent key sequence No equivalent key is available on the front panel.

:TEST:SCANner:CH?

Syntax

:TEST:SCANner:CH?

Description

This command gets the Scanner channel number for test purpose and can be used to check/troubleshoot the Test pin signal of Scanner interface. (Query Only)

NOTE

When `:TEST:SCANner:MODE` is set to ON, the interface pin signal value can be controlled/read by: TEST:SCAN:xxxx commands. After completing the test, `:TEST:SCANner:MODE` should set to OFF so that scanner interface pin signal is generated according to the actual measurement result.

Query response

<numeric><newline><^END>

Related commands

`:TEST:SCANner:EOM` on page 299

`:TEST:SCANner:INDex` on page 300

`:TEST:SCANner:MODE` on page 301

`:TEST:SCANner:TRIGger?` on page 302

`:TEST:SCANner:VALid?` on page 302

Equivalent key sequence

No equivalent key is available on the front panel.

:TEST:SCANner:EOM

Syntax :TEST:SCANner:EOM {ON|OFF|1|0}

Description This command sets the Scanner End of Measurement (EOM) signal for test purpose and can be used to check/troubleshoot the Test pin signal of Scanner interface. (No Query)

NOTE

When `:TEST:SCANner:MODE` is set to ON, the interface pin signal value can be controlled/read by: TEST:SCAN:xxxx commands. After completing the test, `:TEST:SCANner:MODE` should set to OFF so that scanner interface pin signal is generated according to the actual measurement result.

Parameters

	Description
ON	Sets Scanner EOM status to ON.
OFF	Sets Scanner EOM status to OFF.

Related commands

`:TEST:SCANner:CH?` on page 298

`:TEST:SCANner:INDex` on page 300

`:TEST:SCANner:MODE` on page 301

`:TEST:SCANner:TRIGger?` on page 302

`:TEST:SCANner:VALid?` on page 302

Equivalent key sequence No equivalent key is available on the front panel.

:TEST:SCANner:INDex

Syntax

:TEST:SCANner:INDex {ON|OFF|1|0}

Description

This command sets the Scanner Index signal for test purpose and can be used to check/troubleshoot the Test pin signal of Scanner interface. (No Query)

NOTE

When `:TEST:SCANner:MODE` is set to ON, the interface pin signal value can be controlled/read by: TEST:SCAN:xxxx commands. After completing the test, `:TEST:SCANner:MODE` should set to OFF so that scanner interface pin signal is generated according to the actual measurement result.

Parameters

	Description
ON	Sets Scanner Index signal status to ON.
OFF	Sets Scanner Index status to OFF.

Related commands

`:TEST:SCANner:CH?` on page 298

`:TEST:SCANner:EOM` on page 299

`:TEST:SCANner:MODE` on page 301

`:TEST:SCANner:TRIGger?` on page 302

`:TEST:SCANner:VALid?` on page 302

Equivalent key sequence

No equivalent key is available on the front panel.

:TEST:SCANner:MODE

Syntax

`:TEST:SCANner:MODE {ON|OFF|1|0}`

`:TEST:SCANner:MODE?`

Description

This command sets the Scanner Test Mode ON/OFF for test purpose and can be used to check/troubleshoot the Test pin signal of Scanner interface.

NOTE

When `:TEST:SCANner:MODE` is set to ON, the interface pin signal value can be controlled/read by: TEST:SCAN:xxxx commands. After completing the test, `:TEST:SCANner:MODE` should set to OFF so that scanner interface pin signal is generated according to the actual measurement result.

Parameters

	Description
ON	Sets Scanner mode to ON
OFF	Sets Scanner mode to OFF.

Query response

{1|0}<newline><^END>

Related commands

`:TEST:SCANner:CH?` on page 298

`:TEST:SCANner:EOM` on page 299

`:TEST:SCANner:INDex` on page 300

`:TEST:SCANner:TRIGger?` on page 302

`:TEST:SCANner:VALid?` on page 302

Equivalent key sequence

No equivalent key is available on the front panel.

:TEST:SCANner:TRIGger?

Syntax

:TEST:SCANner:TRIGger?

Description

This command gets the Scanner Trigger signal for test purpose and can be used to check/troubleshoot the Test pin signal of Scanner interface. (Query Only)

NOTE

When `:TEST:SCANner:MODE` is set to ON, the interface pin signal value can be controlled/read by: TEST:SCAN:xxxx commands. After completing the test, `:TEST:SCANner:MODE` should set to OFF so that scanner interface pin signal is generated according to the actual measurement result.

Query response

{1|0}<newline><^END>

Related commands

`:TEST:SCANner:CH?` on page 298

`:TEST:SCANner:EOM` on page 299

`:TEST:SCANner:INDex` on page 300

`:TEST:SCANner:MODE` on page 301

`:TEST:SCANner:VALid?` on page 302

Equivalent key sequence

No equivalent key is available on the front panel.

:TEST:SCANner:VALid?

Syntax

:TEST:SCANner:VALid?

Description

Gets scanner/CH_VALID signal.(Query only)

Query response

{1|0}<newline><^END>

Related commands

`:TEST:SCANner:CH?` on page 298

`:TEST:SCANner:EOM` on page 299

`:TEST:SCANner:INDex` on page 300

`:TEST:SCANner:MODE` on page 301

`:TEST:SCANner:TRIGger?` on page 302

Equivalent key sequence

No equivalent key is available on the front panel.

:TRIGger[:SEQ1]:DELay

Syntax

```
:TRIGger[:SEQ1]:DELay <numeric>[mS|S]
:TRIGger[:SEQ1]:DELay?
```

Description

Specifies the waiting time between when a trigger is inputted and when the measurement signal is outputted (source delay time). The source delay time is valid only when the synchronous source function is enabled (SYNC, with the `:SOURce:VOLTage:MODE` on page 269).

Parameters

	<numeric>
Description	Source delay time
Range	0 to 1
Initial value	0
Unit	s (second)
Resolution	100u

If the specified parameter is out of the allowable setup range, the minimum value (if the lower limit of the range is not reached) or the maximum value (if the upper limit of the range is exceeded) is set.

NOTE A fraction below the resolution is rounded off. MAX or MIN can be used to specify the parameters.

Query response {numeric}<newline><^END>

Equivalent key sequence [**Meas Setup**] - SRC DLY

:TRIGger[:SEQ1][:IMMediate]

Syntax

:TRIGger[:SEQ1][:IMMediate]

Description

Immediately generates a trigger and executes a measurement. This command is valid for trigger source values of MANual or BUS trigger specified by `:TRIGger[:SEQ1]:SOURce` command. On setting the trigger source value to INTernal or EXTernal, this command generates an error.

If the trigger system is not in the trigger wait state (trigger event detection state), executing this command causes an error and the command is ignored.

For details on the trigger system, refer to “Trigger system” on page 57. (No query)

Equivalent key sequence

[**Trigger**]

:TRIGger[:SEQ1]:SLOPe

Syntax

:TRIGger[:SEQ1]:SLOPe {POSitive|NEGative}

:TRIGger[:SEQ1]:SLOPe?

Description

Determines the trigger polarity for BNC external trigger on the rear panel. This does not affects the trigger signal on handler and scanner interface.

Related commands

`*TRG` on page 163

Parameters

	Description
POSitive (Initial [Factory set] value)	Detects positive edge trigger signal.
NEGative	Detects negative edge trigger signal.

NOTE

The value of this command is not changed with `*RST` and `:SYSTem:PRESet` and is only changed with Factory Default reset (available only through Front Panel).

Query response

{POS|NEG}<newline><^END>

Equivalent key sequence

[**System**] - EXT TRIG POL - POS|NEG

:TRIGger[:SEQ1]:SOURce

Syntax

```
:TRIGger[:SEQ1]:SOURce {INTernal|MANual|EXTernal|BUS}
:TRIGger[:SEQ1]:SOURce?
```

Description

Selects the trigger mode from the following four types.

Internal	Configures the instrument to use its internal trigger source so that is it automatically and continuously triggered.
Manual	Configures the instrument to be triggered when you press the [Trig] key on the front panel.
External	Configures the instrument to be triggered when a trigger signal is inputted through the Ext TRIG terminal or handler/scanner interface.
BUS	Configures the instrument to be triggered when the *TRG command is executed through GPIB/LAN/USB.

Parameters

	Description
INTernal (initial value)	Specifies Internal.
MANual	Specifies Manual.
EXTernal	Specifies External.
BUS	Specifies GPIB/USB/LAN (Bus).

Query response

{INT|MAN|EXT|BUS}<newline><^END>

Related commands

*TRG on page 163

:READ? on page 230

Equivalent key sequence

[**Meas Setup**] - TRIG - INT|MAN|EXT|BUS

NOTE

You cannot set the trigger mode to GPIB/LAN/USB (Bus) using the front panel keys.

:TRIGger:SEQ2:DELay

Syntax

```
:TRIGger:SEQ2:DELay <numeric>[mS|S]
:TRIGger:SEQ2:DELay?
```

Description

Specifies the waiting time between when a trigger is inputted and when the measurement starts (trigger delay time).

Parameters

	<numeric>
Description	Trigger delay time
Range	0 to 1
Initial value	0
Unit	s (second)
Resolution	100u

If the specified parameter is out of the allowable setup range, the minimum value (if the lower limit of the range is not reached) or the maximum value (if the upper limit of the range is exceeded) is set.

NOTE

A fraction below the resolution is rounded off. MAX or MIN can be used to specify the parameters.

Query response

{numeric}<newline><^END>

Equivalent key sequence

[**Meas Setup**] - TRIG DLY

SCPI Command Table

Table 10-1 lists the E4981A SCPI commands sorted according to function

Table 10-1 SCPI command table

Function	Item to Be Set Up/Executed		SCPI Command
Measurement condition	Reset		`:SYSTem:PRESet` on page 284,`*RST` on page 160
	Measurement parameter setup	Primary parameter	`:CALCulate1:FORMat` on page 188
		Secondary parameter	`:CALCulate2:FORMat` on page 191
	Measurement signal setup	Frequency	`:SOURce:FREQuency[:CW]` on page 267
		1 MHz frequency shift	`:SYSTem:FSHift` on page 281
		Level	`:SOURce:VOLTage[:LEVel][:IMMediate][:AMPLitude]` on page 268
		Signal level compensation (LVL COMP)	`:SOURce:VOLTage:ALC[:STATe]` on page 267
		Output mode	`:SOURce:VOLTage:MODE` on page 269
	Measurement range setup	AUTO	`[:SENSe][:FIMPedance]:RANGe:AUTO` on page 265
		Range	`[:SENSe][:FIMPedance]:RANGe[:UPPer]` on page 266
	Measurement time mode setup	Conventional commands for 4268A/4288A compatibility)	`[:SENSe][:FIMPedance]:APERture[:MODE]` on page 252
		N	`[:SENSe][:FIMPedance]:APERture:TIME` on page 253
	Averaging setup	ON/OFF	`[:SENSe]:AVERage[:STATe]` on page 233
		Number of Averaging	`[:SENSe]:AVERage:COUNt` on page 232
	Cable length setup		`:CALibration:CABLe[:LENGth]` on page 171
	Source delay setup		`:TRIGger[:SEQ1]:DELay` on page 303
	Trigger delay setup		`:TRIGger:SEQ2:DELay` on page 306
	Analog convergence waiting time setup		`[:SENSe]:DETector:DELay1` on page 250 `[:SENSe]:DETector:DELay2` on page 250 `[:SENSe]:DETector:DELay3` on page 251

Table 10-1 SCPI command table

Function	Item to Be Set Up/Executed		SCPI Command
Correction	OPEN correction ON/OFF		[:SENSe]:CORRection:OPEN[:STATe] on page 248
	SHORT correction ON/OFF		[:SENSe]:CORRection:SHORt[:STATe] on page 249
	LOAD correction ON/OFF		[:SENSe]:CORRection:LOAD[:STATe] on page 242
	OPEN correction data parameter format		[:SENSe]:CORRection:CKIT:STAN1:FORMat on page 233
	SHORT correction data parameter format		[:SENSe]:CORRection:CKIT:STAN2:FORMat on page 234
	LOAD standard definition	Definition value	[:SENSe]:CORRection:CKIT:STAN3[:DATA] on page 235
		Parameter type	[:SENSe]:CORRection:CKIT:STAN3:FORMat on page 236
	Measurement Range on Load correction		[:SENSe]:CORRection:COLLect:LOAD:RANGe:AUTO on page 238, [:SENSe]:CORRection:COLLect:STAN3:RANGe:AUTO on page 239
	Correction data	Measurement	[:SENSe]:CORRection:COLLect[:ACQuire] on page 237
		setup and read out	[:SENSe]:CORRection:DATA on page 240
	Offset correction ON/OFF		[:SENSe]:CORRection:OFFSet[:STATe] on page 247
	Offset correction data setup		[:SENSe]:CORRection:OFFSet:DATA on page 246
Scanner (multi correction)	ON/OFF		[:SENSe]:CORRection:MULTiple[:STATe] on page 245
	Channel setup		[:SENSe]:CORRection:MULTiple:CHANnel on page 243
	LOAD standard definition method setup		[:SENSe]:CORRection:MULTiple:CKIT:STAN3[:STATe] on page 244
Cable Correction	Correction Data	OPEN	:CALibration:CABLe:CORRection:COLLect[:ACQuire]:OPEN on page 167
		LOAD	:CALibration:CABLe:CORRection:COLLect[:ACQuire]:LOAD on page 166
		0m standard	:CALibration:CABLe:CORRection:COLLect[:ACQuire]:REFerence on page 168
	Correction coefficient clear		:CALibration:CABLe:CORRection:CLEar on page 165
	Correction coefficient calculation and save		:CALibration:CABLe:CORRection:SAVE on page 169
	Read ON/OFF on correction function		:CALibration:CABLe:CORRection:STATe? on page 170

Table 10-1 SCPI command table

Function	Item to Be Set Up/Executed		SCPI Command
Trigger	Triggering		`*TRG` on page 163, `:TRIGger[:SEQ1][:IMMediate]` on page 304
	Trigger mode setup		`:TRIGger[:SEQ1]:SLOPe` on page 304
	Trigger delay time setup		`:TRIGger:SEQ2:DELay` on page 306
	Trigger system	Resets	`:ABORt` on page 164
		Initiates	`:INITiate[:IMMediate]` on page 228
		Continuous activation ON/OFF	`:INITiate:CONTinuous` on page 227
	BNC External Trigger slope		`:TRIGger[:SEQ1]:SLOPe` on page 304
Measurement Data	Data transfer format setup	Binary/ASCII	`:FORMat[:DATA]` on page 225
		Binary data byte order	`:FORMat:BORDer` on page 223
		ASCII long format	`:FORMat:ASCii:LONG` on page 222
	Data readout	Measurement result	`:FETCh?` on page 220, `:READ?` on page 230
		Data buffer	`:DATA[:DATA]` on page 208
		Measurement signal monitor result	`:DATA[:DATA]` on page 208
	Data buffer setup	Feeding target parameter	`:DATA:FEED[:SOURce]` on page 201, `:DATA:FEED:BUF1` on page 195, `:DATA:FEED:BUF2` on page 196
		Control (feed/not feed)	`:DATA:FEED:CONTrol[:STATe]` on page 200, `:DATA:FEED:CONTrol:BUF1[:STATe]` on page 197, `:DATA:FEED:CONTrol:BUF2[:STATe]` on page 198, `:DATA:FEED:CONTrol:BUF3[:STATe]` on page 199
		Buffer size	`:DATA:POINts[:DATA]` on page 205, `:DATA:POINts:BUF1` on page 202, `:DATA:POINts:BUF2` on page 203, `:DATA:POINts:BUF3` on page 204

Table 10-1 SCPI command table

Function	Item to Be Set Up/Executed		SCPI Command
Comparator	ON/OFF		`:CALCulate1:COMParator[:STATe]` on page 187
	Limit range reset		`:CALCulate1:COMParator:CLEar` on page 175
	Primary parameter limit range setup	ON/OFF	`:CALCulate1:COMParator:PRIMary:BIN{1-9}:STATe` on page 183
		Range setup	`:CALCulate1:COMParator:PRIMary:BIN{1-9}[:LIMIT]` on page 182
		Limit range designation method (mode selection)	`:CALCulate1:COMParator:MODE` on page 181
		Reference (nominal) value	`:CALCulate1:COMParator:PRIMary:NOMinal` on page 184
	Secondary parameter limit range setup	ON/OFF	`:CALCulate1:COMParator:SECondary:STATe` on page 186
		Range setup	`:CALCulate1:COMParator:SECondary:LIMit` on page 185
	AUX BIN function ON/OFF		`:CALCulate1:COMParator:AUXBin` on page 172
	Low C reject function	ON/OFF	`[:SENSe][:FIMPedance]:CREJect[:STATe]` on page 264
		Limit value setup	`[:SENSe][:FIMPedance]:CREJect:LIMit` on page 263
	BIN count function	ON/OFF	`:CALCulate1:COMParator:COUNt[:STATe]` on page 180
		Resets count values	`:CALCulate1:COMParator:COUNt:CLEar` on page 175
		Readout of count values	`:CALCulate1:COMParator:COUNt:DATA?` on page 176
		Readout of count value of overload	`:CALCulate1:COMParator:COUNt:OVLD?` on page 179
		Readout of count values for each channel	`:CALCulate1:COMParator:COUNt:MULTiple:DATA?` on page 177
		Readout of count value of overload for each channel	`:CALCulate1:COMParator:COUNt:MULTiple:OVLD?` on page 179
Measurement signal monitor	Current monitor	ON/OFF (Dummy command for 4268A/4288A compatibility)	`:CALCulate3:MATH:STATe` on page 194
		Monitor value readout	`:DATA[:DATA]` on page 208
	Voltage monitor	ON/OFF (Dummy command for 4268A/4288A compatibility)	`:CALCulate4:MATH:STATe` on page 194
		Monitor value readout	`:DATA[:DATA]` on page 208
Save/Recall	Save		`:MMEMory:STORe:STATe[:REGister]` on page 229, `*SAV` on page 161
	Recall		`:MMEMory:LOAD:STATe[:REGister]` on page 229, `*RCL` on page 160
	Delete		`:MMEMory:DELete[:REGister]` on page 228

Table 10-1 SCPI command table

Function	Item to Be Set Up/Executed			SCPI Command
Display	ON/OFF			:DISPlay[:WINDow][:STATe] on page 215
	Fixed point display setup	ON/OFF		:DISPlay[:WINDow]:TEXT1[:DATA]:FMSD[:STATe] on page 217, :DISPlay[:WINDow]:TEXT2[:DATA]:FMSD[:STATe] on page 219
		value of the highest digit		:DISPlay[:WINDow]:TEXT1[:DATA]:FMSD:DATA on page 216, :DISPlay[:WINDow]:TEXT2[:DATA]:FMSD:DATA on page 218
	Deviation measurement mode setup	Primary parameter	ON/OFF	:CALCulate1:MATH:STATe on page 190
			Mode	:CALCulate1:MATH:EXPRession:NAME on page 189
			Setup	:CALCulate1:MATH:EXPRession:CATalog? on page 189
		Secondary parameter	ON/OFF	:CALCulate2:MATH:STATe on page 193
			Mode	:CALCulate2:MATH:EXPRession:NAME on page 192
			Setup	:CALCulate2:MATH:EXPRession:CATalog? on page 192
		Reference value		:DATA:REFerence1:DATA on page 206, :DATA:REFerence2:DATA on page 207, :DATA[:DATA] on page 208, :DATA:REFerence1:FILL on page 207, :DATA:REFerence2:FILL on page 208
	Setup of displayed page			:DISPlay:PAGE on page 213
	Reset of displayed error/message			:DISPlay:CCLear on page 212
	Input the comment lines			:DISPlay:LINE on page 212
	Output the displayed image to controller			:HCOPy:SDUMp:DATA? on page 226
Contact Check	ON/OFF			[:SENSe][:FIMPedance]:CONTact1:VERify[:STATe] on page 260
	Threshold	TH1		[:SENSe][:FIMPedance]:CONTact1:VERify:THReshold1 on page 261
		TH2		[:SENSe][:FIMPedance]:CONTact1:VERify:THReshold2 on page 262
	Data buffer	Data readout		[:SENSe][:FIMPedance]:CONTact1:VERify:BUF1:DATA? on page 253, [:SENSe][:FIMPedance]:CONTact1:VERify:BUF2:DATA? on page 256, :FORMat:STATus:EXTension on page 224

Table 10-1 SCPI command table

Function	Item to Be Set Up/Executed		SCPI Command
Contact Check	Data buffer setup	Feeding target parameter	[:SENSe][:FIMPedance]:CONTact1:VERify:BUF1:FEED:CONTrol:INTerval on page 254, [:SENSe][:FIMPedance]:CONTact1:VERify:BUF2:FEED:CONTrol:INTerval on page 257
		Control (feed/not feed)	[:SENSe][:FIMPedance]:CONTact1:VERify:BUF1:FEED:CONTrol[:STATe] on page 255, [:SENSe][:FIMPedance]:CONTact1:VERify:BUF2:FEED:CONTrol[:STATe] on page 258
		Buffer size	[:SENSe][:FIMPedance]:CONTact1:VERify:BUF1:POINts on page 256, [:SENSe][:FIMPedance]:CONTact1:VERify:BUF2:POINts on page 259
Key lock	ON/OFF		:SYSTem:KLOCk on page 283
Beeper	ON/OFF		:CALCulate1:COMParator:BEEPer[:STATe] on page 174, :SYSTem:BEEPer:STATe on page 273
	Beep mode setup		:SYSTem:BEEPer:TONE on page 274
	Beep ON		:SYSTem:BEEPer[:IMMediate] on page 273
	Comparator Beep condition		:CALCulate1:COMParator:BEEPer:CONDition on page 173
Status report structure	Clear		*CLS on page 157
	Status byte register value readout		*STB? on page 162
	Service request enable register setup		*SRE on page 162
	Standard event status register	Register value readout	*ESR? on page 158
		OPC bit setup	*OPC on page 159
		Enable register setup	*ESE on page 157
	Operation status register	Clear	:STATus:PRESet on page 271
		Condition register value readout	:STATus:OPERation:CONDition? on page 269
		Enable register setup	:STATus:OPERation:ENABle on page 270
		Event register value readout	:STATus:OPERation[:EVENt]? on page 270

Table 10-1 SCPI command table

Function	Item to Be Set Up/Executed		SCPI Command
External Connector	GPIB Address		`:SYSTem:COMMunicate:GPIB[:SELF]:ADDRess` on page 275
	LAN setup	Fixed IP address	`:SYSTem:COMMunicate:LAN[:SELF]:ADDRess` on page 275
		Fixed Gateway	`:SYSTem:COMMunicate:LAN[:SELF]:DGATeway` on page 277
		Fixed Subnet Mask	`:SYSTem:COMMunicate:LAN[:SELF]:SMASk` on page 278
		Auto IP	`:SYSTem:COMMunicate:LAN[:SELF]:CONFigure` on page 276
	LAN status	Address	`:SYSTem:COMMunicate:LAN[:SELF]:CURRent:ADDRess?` on page 276
		Gateway	`:SYSTem:COMMunicate:LAN[:SELF]:CURRent:DGATeway?` on page 277
		Subnet Mask	`:SYSTem:COMMunicate:LAN[:SELF]:CURRent:SMASk?` on page 277
	MAC Address		`:SYSTem:COMMunicate:LAN[:SELF]:MAC?` on page 278
	Reconnect after reset to factory state		`:SYSTem:COMMunicate:LAN[:SELF]:PRESet` on page 278
	Reconnection		`:SYSTem:COMMunicate:LAN[:SELF]:RESTart` on page 278
	Socket Control Port number		`:SYSTem:COMMunicate:LAN[:SELF]:CONTrol?` on page 276
Internal Clock	Date		`:SYSTem:DATE` on page 279
	Time		`:SYSTem:TIME` on page 286
	Zone		`:SYSTem:TZONe` on page 287

Table 10-1 SCPI command table

Function	Item to Be Set Up/Executed		SCPI Command
Others	Executes self-test (Dummy command for 4268A/4288A compatibility)		`*TST?` on page 163
	Readout of the model name and firmware version		`*IDN?` on page 158
	Readout of the installed option number		`*OPT?` on page 159
	Reads 1 when operation completes		`*OPC?` on page 159
	Readout of the occurred error information		`:SYSTem:ERRor[:NEXT]?` on page 280
	Waits for completion of operation		`*WAI` on page 163
	Reset		`:SYSTem:RESTart` on page 284
	LEARN		`*LRN?` on page 158
	Handler interface signal control	BIN	`:TEST:HANDler:BIN` on page 288
		COMP	`:TEST:HANDler:COMP` on page 289
		OVLD	`:TEST:HANDler:STATus:OVLD` on page 295
		No_Cont/Low_C	`:TEST:HANDler:STATus:NC` on page 294
		Alarm	`:TEST:HANDler:STATus:ALARm` on page 291
		Index	`:TEST:HANDler:STATus:INDex` on page 293
		EOM	`:TEST:HANDler:STATus:EOM` on page 292
		Ready for trigger	`:TEST:HANDler:STATus:RDYTrig` on page 296
		Trigger	`:TEST:HANDler:TRIGger?` on page 297
		Key Lock	`:TEST:HANDler:KEYLock?` on page 289
		Voltage setting for input trigger	`:SYSTem:HANDler:TRIGger:VOLTage` on page 282
	Scanner interface signal control	EOM	`:TEST:SCANner:EOM` on page 299
		Index	`:TEST:SCANner:INDex` on page 300
		Channel	`:TEST:SCANner:CH?` on page 298
		Channel Valid	`:TEST:SCANner:VALid?` on page 302
		Trigger	`:TEST:SCANner:TRIGger?` on page 302
		Voltage setting for input trigger	`:SYSTem:SCANner:TRIGger:VOLTage` on page 285
	Rear Trigger	Trigger	`:TEST:REAR:TRIGger?` on page 297

Front Panel Key Tree vs. SCPI Command

Table 10-2 shows the commands that correspond to operation of the front panel keys.

Table 10-2 Front panel key tree vs. SCPI command

Key Operation				SCPI Command
[Display Format]				
	BIN COUNT DISPLAY			
		COUNT		
			COUNT OFF	`:CALCulate1:COMParator:COUNt[:STATe]` on page 180
			COUNT ON	`:CALCulate1:COMParator:COUNt[:STATe]` on page 180
			RESET COUNT	`:CALCulate1:COMParator:COUNt:CLEar` on page 175
	BIN No.			
		COMP		`:CALCulate1:COMParator[:STATe]` on page 187
			OFF	`:CALCulate1:COMParator[:STATe]` on page 187
			ON	`:CALCulate1:COMParator[:STATe]` on page 187
	DISPLAY BLANK			`:DISPlay[:WINDow][:STATe]` on page 215
	MEAS DISPLAY			
		DISPLAY BLANK		`:DISPlay[:WINDow][:STATe]` on page 215
		Fixed Decimal Point Menu		
			D.P. AUTO	`:DISPlay[:WINDow]:TEXT1[:DATA]:FMSD[:STATe]` on page 217
			D.P. FIX	`:DISPlay[:WINDow]:TEXT1[:DATA]:FMSD[:STATe]` on page 217
			D. P. POS DECL-	`:DISPlay[:WINDow]:TEXT1[:DATA]:FMSD:DATA` on page 216
			D.P. INCR+	`:DISPlay[:WINDow]:TEXT1[:DATA]:FMSD:DATA` on page 216
			D.P. AUTO	`:DISPlay[:WINDow]:TEXT2[:DATA]:FMSD[:STATe]` on page 219
			D.P. FIX	`:DISPlay[:WINDow]:TEXT2[:DATA]:FMSD[:STATe]` on page 219
			D. P. POS DECL-	`:DISPlay[:WINDow]:TEXT2[:DATA]:FMSD:DATA` on page 218
			D.P. INCR+	`:DISPlay[:WINDow]:TEXT2[:DATA]:FMSD:DATA` on page 218
		FREQ		
			120 Hz	`:SOURce:FREQuency[:CW]` on page 267
			1 kHz	`:SOURce:FREQuency[:CW]` on page 267
			1 MHz	`:SOURce:FREQuency[:CW]` on page 267

Table 10-2 Front panel key tree vs. SCPI command

Key Operation	SCPI Command
FUNC	
Cp- ...	`:CALCulate1:FORMat` on page 188
Cp-D	`:CALCulate2:FORMat` on page 191
Cp-G	`:CALCulate2:FORMat` on page 191
Cp-Q	`:CALCulate2:FORMat` on page 191
Cp-Rp	`:CALCulate2:FORMat` on page 191
RETURN	
Cs- ...	`:CALCulate1:FORMat` on page 188
Cs-D	`:CALCulate2:FORMat` on page 191
Cs-Q	`:CALCulate2:FORMat` on page 191
Cs-Rs	`:CALCulate2:FORMat` on page 191
RETURN	
LEVEL	
INCR++	`:SOURce:VOLTage[:LEVel][:IMMediate][:AMPLitude]` on page 268
INCR+	`:SOURce:VOLTage[:LEVel][:IMMediate][:AMPLitude]` on page 268
DECR-	`:SOURce:VOLTage[:LEVel][:IMMediate][:AMPLitude]` on page 268
DECR--	`:SOURce:VOLTage[:LEVel][:IMMediate][:AMPLitude]` on page 268
MEAS TIME	
INCR+	`[:SENSe][:FIMPedance]:APERture:TIME` on page 253
DECR-	`[:SENSe][:FIMPedance]:APERture:TIME` on page 253
RANGE	
AUTO	`[:SENSe][:FIMPedance]:RANGe:AUTO` on page 265
HOLD	`[:SENSe][:FIMPedance]:RANGe:AUTO` on page 265
INCR+	`[:SENSe][:FIMPedance]:RANGe[:UPPer]` on page 266
DECR-	`[:SENSe][:FIMPedance]:RANGe[:UPPer]` on page 266
[Local/Lock]	`:SYSTem:KLOCk` on page 283

Table 10-2 Front panel key tree vs. SCPI command

Key Operation	SCPI Command
[Meas Setup]	
LVL COMP	
ON	:SOURce:VOLTage:ALC[:STATe] on page 267
OFF	:SOURce:VOLTage:ALC[:STATe] on page 267
AVG	
ON	[:SENSe]:AVERage[:STATe] on page 233
OFF	[:SENSe]:AVERage[:STATe] on page 233
INCR+	[:SENSe]:AVERage:COUNt on page 232
DECR-	[:SENSe]:AVERage:COUNt on page 232
DEV A	
ABS	:CALCulate1:MATH:EXPRession:NAME on page 189
%	:CALCulate1:MATH:EXPRession:NAME on page 189
OFF	:CALCulate1:MATH:STATe on page 190
DEV B	
ABS	:CALCulate2:MATH:EXPRession:NAME on page 192
%	:CALCulate2:MATH:EXPRession:NAME on page 192
OFF	:CALCulate2:MATH:STATe on page 193
FREQ	
120 Hz	:SOURce:FREQuency[:CW] on page 267
1 kHz	:SOURce:FREQuency[:CW] on page 267
1 MHz	:SOURce:FREQuency[:CW] on page 267
FREQ SHFT	
0%	:SYSTem:FSHift on page 281
1%	:SYSTem:FSHift on page 281
-1%	:SYSTem:FSHift on page 281
2%	:SYSTem:FSHift on page 281
-2%	:SYSTem:FSHift on page 281

Table 10-2 Front panel key tree vs. SCPI command

Key Operation			SCPI Command
FUNC			
	Cp-...		`:CALCulate1:FORMat` on page 188
		Cp-D	`:CALCulate2:FORMat` on page 191
		Cp-G	`:CALCulate2:FORMat` on page 191
		Cp-Rp	`:CALCulate2:FORMat` on page 191
		Cp-Q	`:CALCulate2:FORMat` on page 191
		RETURN	
	Cs-...		`:CALCulate1:FORMat` on page 188
		Cs-D	`:CALCulate2:FORMat` on page 191
		Cs-Rs	`:CALCulate2:FORMat` on page 191
		Cp-Q	`:CALCulate2:FORMat` on page 191
		RETURN	
LEVEL			
	INCR++		`:SOURce:VOLTage[:LEVel][:IMMediate][:AMPLitude]` on page 268
	INCR+		`:SOURce:VOLTage[:LEVel][:IMMediate][:AMPLitude]` on page 268
	DECR-		`:SOURce:VOLTage[:LEVel][:IMMediate][:AMPLitude]` on page 268
	DECR--		`:SOURce:VOLTage[:LEVel][:IMMediate][:AMPLitude]` on page 268
LOW C REJECT			
	ON		`[:SENSe][:FIMPedance]:CREJect[:STATe]` on page 264
	OFF		`[:SENSe][:FIMPedance]:CREJect[:STATe]` on page 264
	INCR+		`[:SENSe][:FIMPedance]:CREJect:LIMit` on page 263
	DECR-		`[:SENSe][:FIMPedance]:CREJect:LIMit` on page 263
MEAS TIME			
	INCR+		`[:SENSe][:FIMPedance]:APERture:TIME` on page 253
	DECR-		`[:SENSe][:FIMPedance]:APERture:TIME` on page 253
RANGE			
	AUTO		`[:SENSe][:FIMPedance]:RANGe:AUTO` on page 265
	HOLD		`[:SENSe][:FIMPedance]:RANGe:AUTO` on page 265
	INCR+		`[:SENSe][:FIMPedance]:RANGe[:UPPer]` on page 266
	DECR-		`[:SENSe][:FIMPedance]:RANGe[:UPPer]` on page 266
REF A			`:DATA:REFerence1:DATA` on page 206
	Measure		`:DATA:REFerence1:FILL` on page 207

Table 10-2 Front panel key tree vs. SCPI command

Key Operation	SCPI Command
REF B	:DATA:REFerence2:DATA on page 207
Measure	:DATA:REFerence2:FILL on page 208
SRC DLY	
INCR++	:TRIGger[:SEQ1]:DELay on page 303
INCR+	:TRIGger[:SEQ1]:DELay on page 303
DECR-	:TRIGger[:SEQ1]:DELay on page 303
DECR--	:TRIGger[:SEQ1]:DELay on page 303
SYNC SRC	
ON	:SOURce:VOLTage:MODE on page 269
OFF	:SOURce:VOLTage:MODE on page 269
TRIG	
INT	:TRIGger[:SEQ1]:SLOPe on page 304
MAN	:TRIGger[:SEQ1]:SLOPe on page 304
EXT	:TRIGger[:SEQ1]:SLOPe on page 304
BUS	:TRIGger[:SEQ1]:SLOPe on page 304
TRIG DLY	
INCR++	:TRIGger:SEQ2:DELay on page 306
INCR+	:TRIGger:SEQ2:DELay on page 306
DECR-	:TRIGger:SEQ2:DELay on page 306
DECR--	:TRIGger:SEQ2:DELay on page 306
CONT CHECK	
CONT CHK1	
ON	[:SENSe][:FIMPedance]:CONTact1:VERify[:STATe] on page 260
OFF	[:SENSe][:FIMPedance]:CONTact1:VERify[:STATe] on page 260
CC1 TH1	
INCR++	[:SENSe][:FIMPedance]:CONTact1:VERify:THReshold1 on page 261
INCR+	[:SENSe][:FIMPedance]:CONTact1:VERify:THReshold1 on page 261
DECR-	[:SENSe][:FIMPedance]:CONTact1:VERify:THReshold1 on page 261
DECR--	[:SENSe][:FIMPedance]:CONTact1:VERify:THReshold1 on page 261

Table 10-2 Front panel key tree vs. SCPI command

Key Operation	SCPI Command
CC1 TH2	
INCR++	[:SENSe][:FIMPedance]:CONTact1:VERify:THReshold2 on page 262
INCR+	[:SENSe][:FIMPedance]:CONTact1:VERify:THReshold2 on page 262
DECR-	[:SENSe][:FIMPedance]:CONTact1:VERify:THReshold2 on page 262
DECR--	[:SENSe][:FIMPedance]:CONTact1:VERify:THReshold2 on page 262
CORRECTION	
CABLE	
0 m	:CALibration:CABLe[:LENGth] on page 171
1 m	:CALibration:CABLe[:LENGth] on page 171
2 m	:CALibration:CABLe[:LENGth] on page 171
CH	
INCR++	[:SENSe]:CORRection:MULTiple:CHANnel on page 243
INCR+	[:SENSe]:CORRection:MULTiple:CHANnel on page 243
DECR-	[:SENSe]:CORRection:MULTiple:CHANnel on page 243
DECR--	[:SENSe]:CORRection:MULTiple:CHANnel on page 243
LOAD	
Cp- ...	
Cp-D	[:SENSe]:CORRection:CKIT:STAN3:FORMat on page 236
Cp-G	[:SENSe]:CORRection:CKIT:STAN3:FORMat on page 236
Cp-Q	[:SENSe]:CORRection:CKIT:STAN3:FORMat on page 236
Cp-Rp	[:SENSe]:CORRection:CKIT:STAN3:FORMat on page 236
RETURN	
Cs- ...	
Cs-D	[:SENSe]:CORRection:CKIT:STAN3:FORMat on page 236
Cs-Q	[:SENSe]:CORRection:CKIT:STAN3:FORMat on page 236
Cs-Rs	[:SENSe]:CORRection:CKIT:STAN3:FORMat on page 236
RETURN	
LOAD CORRECTION	
MEAS LOAD	[:SENSe]:CORRection:COLLect[:ACQuire] on page 237
ABORT	:ABORt on page 164
OFF	[:SENSe]:CORRection:LOAD[:STATe] on page 242
ON	[:SENSe]:CORRection:LOAD[:STATe] on page 242

Table 10-2 Front panel key tree vs. SCPI command

Key Operation	SCPI Command
LOAD REF	
MULTI	[:SENSe]:CORRection:MULTiple:CKIT:STAN3[:STATe] on page 244
SINGLE	[:SENSe]:CORRection:MULTiple:CKIT:STAN3[:STATe] on page 244
LOAD RNG	
AUTO	[:SENSe]:CORRection:COLLect:LOAD:RANGe:AUTO on page 238
FIX	[:SENSe]:CORRection:COLLect:LOAD:RANGe:AUTO on page 238
MULTI	
OFF	[:SENSe]:CORRection:MULTiple[:STATe] on page 245
ON	[:SENSe]:CORRection:MULTiple[:STATe] on page 245
OFFSET	
OFF	[:SENSe]:CORRection:OFFSet[:STATe] on page 247
ON	[:SENSe]:CORRection:OFFSet[:STATe] on page 247
OFFSET	
A	[:SENSe]:CORRection:OFFSet:DATA on page 246
B	[:SENSe]:CORRection:OFFSet:DATA on page 246
OPEN	
G-B	[:SENSe]:CORRection:CKIT:STAN1:FORMat on page 233
Cp-G	[:SENSe]:CORRection:CKIT:STAN1:FORMat on page 233
OPEN CORRECTION	
MEAS OPEN	[:SENSe]:CORRection:COLLect[:ACQuire] on page 237
ABORT	:ABORt on page 164
OFF	[:SENSe]:CORRection:OPEN[:STATe] on page 248
ON	[:SENSe]:CORRection:OPEN[:STATe] on page 248
REF	
A	:DATA:REFerence1:DATA on page 206
B	:DATA:REFerence2:DATA on page 207
SHORT	
R-X	[:SENSe]:CORRection:CKIT:STAN2:FORMat on page 234
Ls-Rs	[:SENSe]:CORRection:CKIT:STAN2:FORMat on page 234

Table 10-2 Front panel key tree vs. SCPI command

Key Operation	SCPI Command
SHORT CORRECTION	
MEAS SHORT	`[:SENSe]:CORRection:COLLect[:ACQuire]` on page 237
ABORT	`:ABORt` on page 164
OFF	`[:SENSe]:CORRection:SHORt[:STATe]` on page 249
ON	`[:SENSe]:CORRection:SHORt[:STATe]` on page 249
LIMIT TABLE	
AUX	`:CALCulate1:COMParator:AUXBin` on page 172
OFF	`:CALCulate1:COMParator:AUXBin` on page 172
ON	`:CALCulate1:COMParator:AUXBin` on page 172
BEEP	
FAIL	`:CALCulate1:COMParator:BEEPer:CONDition` on page 173
OFF	`:CALCulate1:COMParator:BEEPer[:STATe]` on page 174
PASS	`:CALCulate1:COMParator:BEEPer:CONDition` on page 173
BIN	
CLEAR TABLE	`:CALCulate1:COMParator:CLEar` on page 175
BIN No. {1-9}	
CLEAR LINE	`:CALCulate1:COMParator:PRIMary:BIN{1-9}[:LIMIT]` on page 182
ON	`:CALCulate1:COMParator:PRIMary:BIN{1-9}:STATe` on page 183
HIGH	`:CALCulate1:COMParator:PRIMary:BIN{1-9}[:LIMIT]` on page 182
CLEAR	`:CALCulate1:COMParator:PRIMary:BIN{1-9}[:LIMIT]` on page 182
CLEAR LINE	`:CALCulate1:COMParator:PRIMary:BIN{1-9}[:LIMIT]` on page 182
LOW x(-1)	`:CALCulate1:COMParator:PRIMary:BIN{1-9}[:LIMIT]` on page 182
LOW	`:CALCulate1:COMParator:PRIMary:BIN{1-9}[:LIMIT]` on page 182
CLEAR	`:CALCulate1:COMParator:PRIMary:BIN{1-9}[:LIMIT]` on page 182
CLEAR LINE	`:CALCulate1:COMParator:PRIMary:BIN{1-9}[:LIMIT]` on page 182
HIGH x(-1)	`:CALCulate1:COMParator:PRIMary:BIN{1-9}[:LIMIT]` on page 182
OFF	`:CALCulate1:COMParator:PRIMary:BIN{1-9}:STATe` on page 183
HIGH	`:CALCulate1:COMParator:PRIMary:BIN{1-9}[:LIMIT]` on page 182
CLEAR	`:CALCulate1:COMParator:PRIMary:BIN{1-9}[:LIMIT]` on page 182
CLEAR LINE	`:CALCulate1:COMParator:PRIMary:BIN{1-9}[:LIMIT]` on page 182
LOW x(-1)	`:CALCulate1:COMParator:PRIMary:BIN{1-9}[:LIMIT]` on page 182

Table 10-2 Front panel key tree vs. SCPI command

Key Operation	SCPI Command
LOW	:CALCulate1:COMParator:PRIMary:BIN{1-9}[:LIMIT] on page 182
CLEAR	:CALCulate1:COMParator:PRIMary:BIN{1-9}[:LIMIT] on page 182
CLEAR LINE	:CALCulate1:COMParator:PRIMary:BIN{1-9}[:LIMIT] on page 182
HIGH x(-1)	:CALCulate1:COMParator:PRIMary:BIN{1-9}[:LIMIT] on page 182
BIN No. 2nd	
CLEAR LINE	:CALCulate1:COMParator:SECondary:STATe on page 186
ON	:CALCulate1:COMParator:SECondary:STATe on page 186
HIGH	:CALCulate1:COMParator:SECondary:LIMit on page 185
CLEAR	:CALCulate1:COMParator:SECondary:LIMit on page 185
CLEAR LINE	:CALCulate1:COMParator:SECondary:LIMit on page 185
LOW x(-1)	:CALCulate1:COMParator:PRIMary:BIN{1-9}[:LIMIT] on page 182
LOW	:CALCulate1:COMParator:SECondary:LIMit on page 185
CLEAR	:CALCulate1:COMParator:SECondary:LIMit on page 185
CLEAR LINE	:CALCulate1:COMParator:SECondary:LIMit on page 185
HIGH x(-1)	:CALCulate1:COMParator:SECondary:LIMit on page 185
OFF	:CALCulate1:COMParator:SECondary:STATe on page 186
HIGH	:CALCulate1:COMParator:SECondary:LIMit on page 185
CLEAR	:CALCulate1:COMParator:SECondary:LIMit on page 185
CLEAR LINE	:CALCulate1:COMParator:SECondary:LIMit on page 185
LOW x(-1)	:CALCulate1:COMParator:SECondary:LIMit on page 185
LOW	:CALCulate1:COMParator:SECondary:LIMit on page 185
CLEAR	:CALCulate1:COMParator:SECondary:LIMit on page 185
CLEAR LINE	:CALCulate1:COMParator:SECondary:LIMit on page 185
HIGH x(-1)	:CALCulate1:COMParator:SECondary:LIMit on page 185
COMP	
OFF	:CALCulate1:COMParator[:STATe] on page 187
ON	:CALCulate1:COMParator[:STATe] on page 187
MODE	
%	:CALCulate1:COMParator:MODE on page 181
ABS	:CALCulate1:COMParator:MODE on page 181
OFF	:CALCulate1:COMParator:MODE on page 181

Table 10-2 Front panel key tree vs. SCPI command

Key Operation	SCPI Command
NOM	
INCR++	`:CALCulate1:COMParator:PRIMary:NOMinal` on page 184
INCR+	`:CALCulate1:COMParator:PRIMary:NOMinal` on page 184
DECR-	`:CALCulate1:COMParator:PRIMary:NOMinal` on page 184
DECR--	`:CALCulate1:COMParator:PRIMary:NOMinal` on page 184
USER COMMENT[1]	
ADD CHAR	
ENTER	`:DISPlay:LINE` on page 212
NEXT	
PREV	
[Preset]	
CLEAR SET & CORR	
CANCEL	
OK	`*RST` on page 160
RETURN	
CLEAR SETTING	
CANCEL	
OK	`:SYSTem:PRESet` on page 284
RETURN	
FACTORY DEFAULT	
CANCEL	
OK	
RETURN	
LAN RESET	
CANCEL	
OK	`:SYSTem:COMMunicate:LAN[:SELF]:PRESet` on page 278
RETURN	
[Recall A]	`:MMEMory:LOAD:STATe[:REGister]` on page 229
[Recall B]	`:MMEMory:LOAD:STATe[:REGister]` on page 229

Table 10-2 Front panel key tree vs. SCPI command

Key Operation	SCPI Command
[Save/Recall]	
CATALOG	
MEDIA	
EXT	
INT	
No.	
DELETE	:MMEMory:DELete[:REGister] on page 228
RECALL	:MMEMory:LOAD:STATe[:REGister] on page 229
SAVE	:MMEMory:STORe:STATe[:REGister] on page 229
SAVE DATA	
START LOG	:DATA:FEED:CONTrol:BUF3[:STATe] on page 199
SAVE & STOP	:MMEMory:STORe:STATe[:REGister] on page 229
SAVE DISPLAY	:MMEMory:STORe:STATe[:REGister] on page 229
[System]	
CABLE CORR	
1m	
CLEAR	:CALibration:CABLe:CORRection:CLEar on page 165
MEAS LOAD	:CALibration:CABLe:CORRection:COLLect[:ACQuire]:LOAD on page 166
ABORT	:ABORt on page 164
MEAS OPEN	:CALibration:CABLe:CORRection:COLLect[:ACQuire]:OPEN on page 167
ABORT	:ABORt on page 164
MEAS REF	:CALibration:CABLe:CORRection:COLLect[:ACQuire]:REFerence on page 168
ABORT	:ABORt on page 164
SAVE	:CALibration:CABLe:CORRection:SAVE on page 169
2m	
CLEAR	:CALibration:CABLe:CORRection:CLEar on page 165
MEAS LOAD	:CALibration:CABLe:CORRection:COLLect[:ACQuire]:LOAD on page 166
ABORT	:ABORt on page 164

Table 10-2 Front panel key tree vs. SCPI command

Key Operation	SCPI Command
MEAS OPEN	:CALibration:CABLe:CORRection:COLLect[:ACQuire]:OPEN on page 167
ABORT	:ABORt on page 164
MEAS REF	:CALibration:CABLe:CORRection:COLLect[:ACQuire]:REFerence on page 168
ABORT	:ABORt on page 164
SAVE	:CALibration:CABLe:CORRection:SAVE on page 169
SELF TEST	
TEST NO.	
EXECUTE	
RETURN	
TEST START	
TEST STOP	
INCR+	
DECR-	
SERVICE	
MORE	
RETURN	
SAVE SYS INFO	
SYSTEM CONFIG	
BEEPER ENABLED	
OFF	:SYSTem:BEEPer:STATe on page 273
ON	:SYSTem:BEEPer:STATe on page 273
BEEPER TONE	
TONE 1	:SYSTem:BEEPer:TONE on page 274
TONE 2	:SYSTem:BEEPer:TONE on page 274
TONE 3	:SYSTem:BEEPer:TONE on page 274
TONE 4	:SYSTem:BEEPer:TONE on page 274
TONE 5	:SYSTem:BEEPer:TONE on page 274

Table 10-2 Front panel key tree vs. SCPI command

Key Operation					SCPI Command
		DATE/TIME			
			DATE		
				DAY	`:SYSTem:DATE` on page 279
				MONTH	`:SYSTem:DATE` on page 279
				RETURN	
				YEAR	`:SYSTem:DATE` on page 279
			TIME		
				HOUR	`:SYSTem:TIME` on page 286
				MINUTE	`:SYSTem:TIME` on page 286
				RETURN	
				SECOND	`:SYSTem:TIME` on page 286
		GPIB ADDR			`:SYSTem:COMMunicate:GPIB[:SELF]:ADDRess` on page 275
		IP CONFIG			
			AUTO		`:SYSTem:COMMunicate:LAN[:SELF]:CONFigure` on page 276
			MANUAL		`:SYSTem:COMMunicate:LAN[:SELF]:CONFigure` on page 276
			RESTART NETWORK		`:SYSTem:COMMunicate:LAN[:SELF]:RESTart` on page 278
		MANUAL GATEWAY			
			ENTER		`:SYSTem:COMMunicate:LAN[:SELF]:DGATeway` on page 277
			RESTART NETWORK		`:SYSTem:COMMunicate:LAN[:SELF]:RESTart` on page 278
		MANUAL IP ADDR			
			ENTER		`:SYSTem:COMMunicate:LAN[:SELF]:ADDRess` on page 275
			RESTART NETWORK		`:SYSTem:COMMunicate:LAN[:SELF]:RESTart` on page 278
		MANUAL SUBNET MASK			
			ENTER		`:SYSTem:COMMunicate:LAN[:SELF]:SMASk` on page 278
			RESTART NETWORK		`:SYSTem:COMMunicate:LAN[:SELF]:RESTart` on page 278
		TIME ZONE			
			HOUR INCR++		`:SYSTem:TZONe` on page 287
			HOUR DECR--		`:SYSTem:TZONe` on page 287
			MINUTE INCR+		`:SYSTem:TZONe` on page 287
			MINUTE DECR-		`:SYSTem:TZONe` on page 287

Table 10-2 Front panel key tree vs. SCPI command

Key Operation				SCPI Command
	SYSTEM INFO			
		EXT TRIG POL		
			POS	`:TRIGger[:SEQ1]:SLOPe` on page 304
			NEG	`:TRIGger[:SEQ1]:SLOPe` on page 304
[Trigger]				`:TRIGger[:SEQ1][:IMMediate]` on page 304

1. The field name changes when entering a comment.

SCPI Command Tree

Table 10-3 shows the E4981A SCPI command tree.

Table 10-3 E4981A SCPI command tree

Command	Parameter	Note
*CLS		[No query]
*ESE		
*ESR		[Query only]
*IDN		[Query only]
*LRN		[Query only]
*OPC		[No query]
*OPC		[Query only]
*OPT		[Query only]
*RCL		[No query]
*RST		[No query]
*SAV		[No query]
*SRE		
*STB		[Query only]
*TRG		[No query]
*TST		[Query only]
*WAI		[No query]
:ABORt		[No query]
:CALibration		
:CABLe		
:CORRection		
:CLEar	<numeric>	[No query]
:COLLect		
[:ACQuire]		
:LOAD	<numeric>	[No query]
:OPEN	<numeric>	[No query]
:REFerence	<numeric>	[No query]
:SAVE	<numeric>	[No query]
:STATe		[Query only]
[:LENGth]	{0|1|2}	

Table 10-3 E4981A SCPI command tree

Command	Parameter	Note
:CALCulate1		
:COMParator		
:AUXBin	{ON|OFF|1|0}	
:BEEPer		
:CONDition	{PASS|FAIL}	
[:STATe]	{ON|OFF|1|0}	
:CLEar		[No query]
:COUNt		
:CLEar		[No query]
:DATA?		[Query only]
:MUTLi		
:DATA?		[Query only]
:OVLD?		[Query only]
:OVLD?		[Query only]
[:STATe]	{ON|OFF|1|0}	
:MODE	{ABS|DEV|PCNT}	
:PRIMary		
:BIN{1|2|3|4|5|6|7|8|9}		
[:LIMit]	<numeric>,<numeric>	
:STATe	{ON|OFF|1|0}	
:NOMinal	<numeric>	
:SECondary		
:LIMit	<numeric>,<numeric>	
:STATe	{ON|OFF|1|0}	
[:STATe]	{ON|OFF|1|0}	
:FORMat	{CP|CS}	
:MATH		
:EXPRession		
:CATalog?		[Query only]
:NAME	{DEV|PCNT}	
:STATe	{ON|OFF|1|0}	

Table 10-3 E4981A SCPI command tree

Command	Parameter	Note
:CALCulate2		
:FORMat	{D|Q|G|RP|RS}	
:MATH		
:EXPRession		
:CATalog?		[Query only]
:NAME	{DEV|PCNT}	
:STATe	{ON|OFF|1|0}	
:CALCulate3		
:MATH		
:STATe	{ON|OFF|1|0}	
:CALCulate4		
:MATH		
:STATe	{ON|OFF|1|0}	
:DATA		
:FEED		
:BUF1	{“CALCulate1”|“CALCulate2”|“ ”}	
:BUF2	{“CALCulate1”|“CALCulate2”|“ ”}	
:CONTrol		
:BUF1		
[:STATe]	{NEVer|ALWays}	
:BUF2		
[:STATe]	{NEVer|ALWays}	
:BUF3		
[:STATe]	{NEVer|ALWays}	
[:STATe]	{BUF1|BUF2|BUF3},{NEVer|ALWays}	
[:SOURCe]	{BUF1|BUF2},{“CALCulate1”|“CALCuate2”|“ ”}	
:POINts		
:BUF1	<numeric>	
:BUF2	<numeric>	
:BUF3	<numeric>	
[:DATA]	{BUF1|BUF2|BUF3},<numeric>	
:REFerence1		
:DATA	<numeric>	

Table 10-3 E4981A SCPI command tree

Command	Parameter	Note
:FILL		[No query]
:REFerence2		
:DATA	<numeric>	
:FILL		[No query]
[:DATA]	{BUF1|BUF2|BUF3|IMON|VMON|REF1|REF2}	
:DISPlay		
:CCLear		[No query]
:LINE	<string>	
:PAGE	{MEAS|BNUM|BCO|MSET|CSET|LTAB|CATA|SYST|SELF|MLAR|SCON|SERV|CCOR}	
[:WINDow]		
[:STATe]	{ON|OFF|1|0}	
:TEXT1		
[:DATA]		
:FMSD		
:DATA		
[:STATe]	{ON|OFF|1|0}	
:TEXT2		
[:DATA]		
:FMSD		
:DATA		
[:STATe]	{ON|OFF|1|0}	
:FETCh		[Query only]
:FORMat		
[:DATA]	{ASCii|REAL[,64]}	
:ASCii		
:LONG	{ON|OFF|1|0}	
:STATus		
:EXTension	{ON|OFF|1|0}	
:BORDer	{NORMal|SWAPped}	
:HCOPy		
:SDUMp		
:DATA		[Query only]

Table 10-3 E4981A SCPI command tree

Command	Parameter	Note
:INITiate		
:CONTinuous	{ON|OFF|1|0}	
[:IMMediate]		[No query]
:MMEMory		
:DELete		
[:REGister]	<numeric>	[No query]
:LOAD		
:STATe		
[:REGister]	<numeric>	[No query]
:STORe		
:STATe		
[:REGister]	<numeric>	[No query]
:READ		[Query only]
[:SENSe]		
:AVErage		
:COUNt	<numeric>	
[:STATe]	{ON|OFF|1|0}	
[:FIMPedance]		
:APERture		
[:MODE]	{SHORt|MEDium|LONG}	
:TIME	{1|2|4|6|8}	
:CONTact1		
:VERify		
:BUF1		
:DATA		[Query only]
:FEED		
:CONTrol		
:INTerval	<numeric>	
[:STATe]	{NEVer|ALWays}	
:POINts	<numeric>	
:BUF2		
:DATA		[Query only]
:FEED		

Table 10-3 E4981A SCPI command tree

Command	Parameter	Note
:CONTrol		
:INTerval	<numeric>	
[:STATe]	{NEVer|ALWays}	
:POINts	<numeric>	
[:STATe]	{ON|OFF|1|0}	
:THReshold1	<numeric>	
:THReshold2	<numeric>	
[:FIMPedance]		
:CREJect		
:LIMit	<numeric>	
[:STATe]	{ON|OFF|1|0}	
:RANGe		
:AUTO	{ON|OFF|1|0}	
[:UPPer]	<numeric>[PF|NF|UF|MF|F]	
:CORRection		
:CKIT		
:STAN1		
:FORMat	{GB|CPG}	
:STAN2		
:FORMat	{RX|LSRS}	
:STAN3		
[:DATA]	<numeric>,<numeric>	
:FORMat	{CPD|CPQ|CPG|CPRP|CSD|CSD|CSRS}	
:COLLect		
[:ACQuire]	{STAN1|STAN2|STAN3}	[No query]
:LOAD		
:RANGe		
:AUTO	{ON|OFF|1|0}	
:STAN3		
:RANGe		
:AUTO	{ON|OFF|1|0}	
:DATA	{STAN1|STAN2|STAN3},<numeric>,<numeric>	
:LOAD		

Table 10-3 E4981A SCPI command tree

Command						Parameter	Note
			[:STATe]			{ON|OFF|1|0}	
		:MULTiple					
			:CHANnel			<numeric>	
			:CKIT				
				:STAN3			
					[:STATe]	{ON|OFF|1|0}	
			[:STATe]			{ON|OFF|1|0}	
		:OFFSet					
			:DATA			<numeric>,<numeric>	
			[:STATe]			{ON|OFF|1|0}	
		:OPEN					
			[:STATe]			{ON|OFF|1|0}	
		:SHORT					
			[:STATe]			{ON|OFF|1|0}	
	:DETector						
		:DELay1				<numeric>	
		:DELay2				<numeric>	
		:DELay3				<numeric>	
:SOURce							
	:FREQuency						
		[:CW]				<numeric>[Hz|kHz|MHz]	
	:VOLTage						
		:ALC					
			[:STATe]			{ON|OFF|1|0}	
		[:LEVel]					
			[:IMMediate]				
				[:AMPLitude]		<numeric>	
		:MODE				{SYNChronous|CONTinuous}	

Table 10-3 E4981A SCPI command tree

Command	Parameter	Note
:STATus		
:OPERation		
:CONDition		[Query only]
:ENABle	<numeric>	
[:EVENt]		[Query only]
:UPDate	{ON|OFF|1|0}	
:PRESet		[No query]
:QUEStionable		
:CONDition		[Query only]
:ENABle	<numeric>	
[:EVENt]		[Query only]
:SYSTem		
:BEEPer		
[:IMMediate]		[No query]
:STATe	{ON|OFF|1|0}	
:TONE	<numeric>	
:COMMunicate		
:GPIB		
[:SELF]		
:ADDRess	<numeric>	
:LAN		
[:SELF]		
:ADDRess		
:CONFigure	{AUTO|MANual}	
:CONTrol		[Query only]
:CURRent		
:ADDRess		[Query only]
:DGATeway		[Query only]
:SMASk		[Query only]
:DGATeway		
:MAC		[Query only]
:PRESet		[No query]
:RESTart		[No query]

Table 10-3 E4981A SCPI command tree

	Command				Parameter	Note
				:SMASk		
	:DATE				<numeric>,<numeric>,<numeric>	
	:ERRor					
		[:NEXT]				[Query only]
	:FSHift				<numeric>	
	:HANDler					
		:TRIGger				
			:VOLTage		<numeric>	
	:KLOCk				{ON|OFF|1|0}	
	:PRESet					[No query]
	:RESTart					[No query]
	:TIME				<numeric>,<numeric>,<numeric>	
	:TZONe				<numeric>,<numeric>	
	:SCANner					
		:TRIGger				
			:VOLTage		<numeric>	
:TEST						
	:HANDler					
		:BIN			<numeric>	[No query]
		:COMP			{PHI|PLO|SREJ|OFF}	[No query]
		:KEYLock				[Query only]
		:MODE			{ON|OFF|1|0}	
		:STATus				
			:ALARm		{ON|OFF|1|0}	[No query]
			:EOM		{ON|OFF|1|0}	[No query]
			:INDex		{ON|OFF|1|0}	[No query]
			:NC		{ON|OFF|1|0}	[No query]
			:OVLD		{ON|OFF|1|0}	[No query]
			:RDYTrig		{ON|OFF|1|0}	[No query]
		:TRIGger				[Query only]
	:REAR					
		:TRIGger				[Query only]
	:SCANner					

Table 10-3 E4981A SCPI command tree

Command	Parameter	Note
:CH		[Query only]
:EOM	{ON|OFF|1|0}	[No query]
:INDEX	{ON|OFF|1|0}	[No query]
:MODE	{ON|OFF|1|0}	
:TRIGger		[Query only]
:VALID		[Query only]
:TRIGger		
[:SEQ1]		
:DELay	<numeric>	
[:IMMediate]		[No query]
:SLOPe	{POSitive|NEGative}	
:SOURce	{INTernal|MANual|EXTernal|BUS}	
:SEQ2		
:DELay	<numeric>	

A Manual Changes

This appendix contains the information required to adapt this manual to earlier versions or configurations of the Keysight E4981A than that indicated by the current printing date of this manual. The information in this manual applies directly to the E4981A model that has the serial number prefix listed on the title page of this manual.

Manual Changes

If your E4981A has firmware or serial number shown in Table A-1 and Table A-2, see the corresponding manual changes.

Table A-1 Manual Changes by Serial Number

Serial Prefix or Number	Make Manual Changes

Table A-2 Manual Changes by Firmware Version

Version	Make Manual Changes

The ten-character serial number is stamped on the serial number plate on the rear panel.

Execute the `*IDN?` command on page 158 to check the firmware version.

B Information for Replacing 4268A, 4288A with E4981A

This appendix describes information that is applicable when replacing the Keysight 4268A, 4288A with the Keysight E4981A. See the 4268A, 4288A *Operation Manual* for detailed information on the 4268A, 4288A. See the E4981A *Operation Manual* and other chapters of this manual (*Programming Manual*) for more detailed information on the E4981A.

4268, 4288A vs. E4981A SCPI Command Correspondence Table

Table B-1 (by function) and Table B-2 (by alphabetical order) provides at-a-glance lists of SCPI commands correspondence between 4268A, 4288A and E4981A.

Table B-1 At-a-glance SCPI command correspondence between 4268A, 4288A, and E4981A (by function)

Function	Item to Be Set Up/Executed		SCPI Command			Note
			E4981A	4268A	4288A	
Measurement condition	Reset		**:SYSTem:PRESet**	♦	♦	Correction date is not initialized for 4288A
			***RST**	♦	♦	
	Measurement parameter setup	Primary parameter	**:CALCulate1:FORMat**	♦	♦	
		Secondary parameter	**:CALCulate2:FORMat**	♦	♦	
	Measurement signal setup	Frequency	**:SOURce:FREQuency[:CW]**	♦	♦	
		1 MHz frequency shift	**:SYSTem:FSHift**	NA	♦	
		Level	**:SOURce:VOLTage[:LEVel][:IMMediate][:AMPLitude]**	♦	♦	
		Signal level compensation (SLC)	**:SOURce:VOLTage:ALC[:STATe]**	♦	NA	ALC function is not installed to 4288A
		Output mode (Synchronous source)	**:SOURce:VOLTage:MODE**	♦	NA	Signal synchronization is not on 4288A
	Measurement range setup	AUTO	**[:SENSe][:FIMPedance]:RANGe:AUTO**	♦	♦	
		Range	**[:SENSe][:FIMPedance]:RANGe[:UPPer]**	♦	♦	
	Measurement time mode setup	Conventional	**[:SENSe][:FIMPedance]:APERture[:MODE]**	♦	♦	
		N	**[:SENSe][:FIMPedance]:APERture:TIME**	NA	NA	
	Averaging setup	ON/OFF	**[:SENSe]:AVERage[:STATe]**	♦	♦	
		Number of Averaging	**[:SENSe]:AVERage:COUNt**	♦	♦	
	Cable length setup		**:CALibration:CABLe**	♦	♦	

Table B-1 At-a-glance SCPI command correspondence between 4268A, 4288A, and E4981A (by function)

Function	Item to Be Set Up/Executed		SCPI Command			Note
			E4981A	4268A	4288A	
Measurement condition	Source delay setup		**:TRIG[SEQ1]:DEL**	♦	NA	Trigger delay only for 4288A
	Trigger delay setup		**:TRIG:SEQ2:DEL**	♦	:TRIG:DEL	
	Analog convergence waiting time setup		**[:SENSe]:DETector:DELay[1-3]**	NA	NA	
Correction	Entire correction ON/OFF		**NA**	**[:SENSe]:CORRection[:STATe]**	NA	4268A: Separately set ON/OFF for only LOAD 4288A: ON/OFF for each calibration type.
	OPEN correction ON/OFF		**[:SENSe]:CORRection:OPEN[:STATe]**	NA	♦	
	SHORT correction ON/OFF		**[:SENSe]:CORRection:SHORt[:STATe]**	NA	♦	
	LOAD correction ON/OFF		**[:SENSe]:CORRection:LOAD[:STATe]**	**:CORR:COLL:METH**	♦	
	OPEN correction data parameter format		**[:SENSe]:CORRection:CKIT:STAN1:FORMat**	NA	♦	
	SHORT correction data parameter format		**[:SENSe]:CORRection:CKIT:STAN2:FORMat**	NA	♦	
	LOAD standard definition	Definition value	**[:SENSe]:CORRection:CKIT:STAN3:[DATA]**	♦	♦	
		Parameter type	**[:SENSe]:CORRection:CKIT:STAN3:FORMat**	♦	♦	
	Measurement Range on Load correction		**[:SENSe]:CORRection:COLLect:LOAD:RANGE:AUTO**	NA	♦	
			[:SENSe]:CORRection:COLLect:STAN3:RANGE:AUTO	NA	NA	
	Correction data	Measurement	**[:SENSe]:CORRection:COLLect[:ACQuire]**	♦	♦	
		Setup and read out	**[:SENSe]:CORRection:DATA**	♦	♦	
	Offset correction ON/OFF		**[:SENSe]:CORRection:OFFSet[:STATe]**	NA	♦	
Correction	Offset correction data setup		**[:SENSe]:CORRection:OFFSet:DATA**	NA	♦	

Table B-1 At-a-glance SCPI command correspondence between 4268A, 4288A, and E4981A (by function)

Function	Item to Be Set Up/Executed		SCPI Command			Note
			E4981A	4268A	4288A	
Scanner (multi correction)	ON/OFF		**[:SENSe]:CORRection:MULTiple:[:STATe]**	♦	♦	
	Channel setup		**[:SENSe]:CORRection:MULTiple:CHANnel**	♦	♦	
	LOAD standard correction method setup		**[:SENSe]:CORRection:MULTiple:CKIT:STAN3[:STATe]**	♦	♦	
Cable correction	Correction Data	OPEN	**:CALibration:CABLe:CORRection:COLLect[:ACQuire]:OPEN**	NA	NA	
		LOAD	**:CALibration:CABLe:CORRection:COLLect[:ACQuire]:LOAD**	NA	NA	
		0m standard	**:CALibration:CABLe:CORRection:COLLect[:ACQuire]:REFerence**	NA	NA	
	Correction coefficient clear		**:CALibration:CABLe:CORRection:CLEar**	NA	NA	
	Correction coefficient calculation and save		**:CALibration:CABLe:CORRection:SAVE**	NA	NA	
	Read ON/OFF on correction function		**:CALibration:CABLe:CORRection:STATe?**	NA	NA	
Trigger	Triggering		***TRG**	♦	♦	
			:TRIGger[:SEQ1][:IMMediate]	**:TRIGger[:SEQuence1][:IMMediate]**	**:TRIGger[:SEQuence1][:IMMedaite]**	
	Trigger mode setup		**:TRIGger[:SEQ1]:SOURce**	**:TRIGger[:SEQuence1]:SOURce**	**:TRIGger[:SEQuence1]:SOURce**	
	Trigger delay time setup		**:TRIGger:SEQ2:DELay**	**:TRIGger:SEQuence2:DELay**	**:TRIGger[:SEQuence1]:DELay**	
	Trigger system	Resets	**:ABORt**	♦	♦	
		Initiates	**:INITiate[:IMMediate]**	♦	♦	
		Continuous activation ON/OFF	**:INITiate:CONTinuous**	♦	♦	
	BNC External Trigger slope		**:TRIGger[:SEQ1]:SLOPe**	NA	NA	

Table B-1 At-a-glance SCPI command correspondence between 4268A, 4288A, and E4981A (by function)

Function	Item to Be Set Up/Executed		SCPI Command			Note
			E4981A	4268A	4288A	
Measurement Data	Data transfer format setup	Binary/ASCII	:FORMat[:DATA]	♦	♦	4288A: These functions are affected to input/ouput of correction data(:CORR:DATA)
		Binary data byte order	:FORMat:BORDe	NA	NA	
		ASCII long format	:FORMat:ASCii :LONG	NA	NA	
	Data readout	Measurement result	:FETCh?	♦	♦	
			:READ?	♦	♦	
		Data buffer	:DATA[:DATA]? {BUF1|BUF2|BUF3}	♦	♦	BUF3 is not for 4268A
		Measurement signal monitor result	:DATA[:DATA]? {IMON|VMON}	♦	♦	
Measurement data	Data buffer setup	Feeding target parameter	:DATA:FEED[:SOURce] {BUF1|BUF2},{CALC1|CALC2}	:DATA:FEED	:DATA:FEED	
			:DATA:FEED:BUF[1-2] {CALCulate1, CALCulate2}			
		Control (feed/not feed)	:DATA:FEED:CONTrol [:STATe] {BUF1|BUF2|BUF3},[ALWays|NEVer]	:DATA:FEED:CONTrol	♦	
			:DATA:FEED:CONTrol:BUF[1-3][:STATe] [ALWays|NEVer]			
		Buffer size	:DATA:POINts[:DATA] {BUF1|BUF2|BUf3}	:DATA:POINts	♦	
			:DATA:POINts:BUF[1-3]			

Table B-1 At-a-glance SCPI command correspondence between 4268A, 4288A, and E4981A (by function)

Function	Item to Be Set Up/Executed		SCPI Command			Note
			E4981A	4268A	4288A	
Comparator	ON/OFF		:CALCulate:COMParator[:STATe]	♦	♦	
	Limit range reset		:CALCulate:COMParator:CLEar	NA	♦	
	Primary parameter limit range setup	ON/OFF	:CALCulate:COMParator:PRIMary:BIN{1-9}:STATe	♦	♦	
		Range setup	:CALCulate:COMParator:PRIMary:BIN{1-9}	♦	♦	
		Limit range designation method (mode selection)	:CALCulate:COMParator:MODE	♦	♦	ABS|DEV|PCNT
		Reference (nominal) value	:CALCulate:COMParator:PRIMary:NOMinal	♦	♦	
	Secondary parameter limit range setup	ON/OFF	:CALCulate:COMParator:SECondary:STATe	♦	♦	
		Range setup	:CALCulate:COMParator:SECondary:LIMit	♦	♦	
Comparator	AUX BIN function ON/OFF		:CALCulate:COMParator:AUXBin	♦	♦	
	Low C reject function	ON/OFF	[:SENSe][:FIMPedance]:CREJect[:STATe]	NA	♦	
		Limit value setup	[:SENSe][:FIMPedance]:CREJect:LIMit	NA	♦	
	BIN count function	ON/OFF	:CALCulate:COMParator:COUNt[:STATe]	♦	♦	
		Resets count values	:CALCulate:COMParator:COUNt:CLEar	♦	♦	
		Readout of count values	:CALCulate:COMParator:COUNt:DATA?	♦	♦	
		Readout of count value of overload	:CALCulate:COMParator:COUNt:OVLD?	NA	♦	
		Readout of count values for each channel	:CALCulate:COMParator:COUNt:MULTiple:DATA?	NA	♦	
		Readout of count value of overload for each channel	:CALC:COMP:COUN:MULTIple:OVLD?	NA	♦	

Table B-1 At-a-glance SCPI command correspondence between 4268A, 4288A, and E4981A (by function)

Function	Item to Be Set Up/Executed		SCPI Command			Note
			E4981A	4268A	4288A	
Measurement signal monitor	Current monitor	ON/OFF (Dummy command for 4268A/4288A compatibility)	**:CALCulate3:MATH:STATe**	♦	♦	
		Monitor value readout	**:DATA[:DATA]?**	♦	♦	
	Voltage monitor	ON/OFF (Dummy command for 4268A/4288A compatibility)	**:CALCulate4:MATH:STATe**	♦	♦	
		Monitor value readout	**:DATA[:DATA]?**	♦	♦	
Save/Recall	Save		**:MMEMory:STORe:STATe:[REGister]**	***SAV**	***SAV**	
			***SAV**	♦	♦	
	Recall		**:MMEMory:LOAD:STATe:[REGister]**	***RCL**	***RCL**	
			***RCL**	♦	♦	
	Delete		**:MMEMory:DELete:[:REGister]**	NA	NA	

Table B-1 At-a-glance SCPI command correspondence between 4268A, 4288A, and E4981A (by function)

Function	Item to Be Set Up/Executed			SCPI Command			Note
				E4981A	4268A	4288A	
Display	ON/OFF			**:DISPlay[:WINDow][:STATe]**	♦	♦	
	Number of Digits setup			NA	**:DISPlay[:WINDow]:TEXT1[:DATA]:DIGit**	**:DISPlay[:WINDow]:TEXT1[:DATA]:DIGit**	This fnction is not available in E4981A.
	Fixed point display setup	ON/OFF		**:DISPlay[:WINDow]:TEXT[1-2][:DATA]:FMSD[:STATe]**	NA	**:DISP:WIN:TEXT1:FMSD**	
		value of the highest digit		**:DISPlay[:WINDow]:TEXT[1-2][:DATA]:FMSD:DATA**	NA	**:DISP:WIN:TEXT1:FMSD:DATA**	
	Deviation measurement mode setup	Primary parameter	ON/OFF	**:CALCulate1:MATH:STATe**	♦	♦	
			Mode	**:CALCulate1:MATH:EXPRession:NAME**	♦	♦	
			Setup	**:CALCulate1:MATH:EXPRession:CATalog?**	♦	♦	
		Secondary parameter	ON/OFF	**:CALCulate2:MATH:STATe**	♦	♦	
			Mode	**:CALCulate1:MATH:EXPRession:NAME**	♦	♦	
			Setup	**:CALCulate1:MATH:EXPRession:CATalog?**	♦	♦	
		Reference value		**:DATA:REFerence[1-2]:DATA**	NA	NA	
				:DATA[:DATA]? ?REF1|REF2}	♦	♦	
				:DATA:REFerence[1-2]:FILL	NA	NA	
	Setup of displayed page			**:DISPlay:PAGE**	**:DISPlay[:WINdow]:TEXT2:PAGE**	♦	
	Reset of displayed error/message			**:DISPlay:CCLear**	NA	NA	
	Input the comment lines			**:DISPlay:LINE**	NA	NA	
	Output the displayed image to controller			**:HCOPy:SDUMp:DATA?**	NA	NA	
Contact Check	ON/OFF			**[:SENSe][:FIMPedance]:CONTact1:VERify[:STATe]**	♦	NA	

Table B-1 At-a-glance SCPI command correspondence between 4268A, 4288A, and E4981A (by function)

Function	Item to Be Set Up/Executed		SCPI Command			Note
			E4981A	4268A	4288A	
Contact Check	Threshold	TH1	**[:SENSe][:FIMPedance]:CONTact1:VERify:THReshold1**	NA	NA	
		TH2	**[:SENSe][:FIMPedance]:CONTact1:VERify:THReshold2**	NA	NA	
	Data buffer	Data readout	**[:SENSe][:FIMPedance]:CONTact1:VERify:BUF1:DATA?**	NA	NA	
			[:SENSe][:FIMPedance]:CONTact1:VERify:BUF2:DATA?	NA	NA	
			:FORMat:STATus:EXTension	NA	NA	
	Data buffer setup	Feeding target parameter	**[:SENSe][:FIMPedance]:CONTact1:VERify:BUF1:FEED:CONTrol:INTerval**	NA	NA	
			[:SENSe][:FIMPedance]:CONTact1:VERify:BUF2:FEED:CONTrol:INTerval	NA	NA	
		Control (feed/not feed)	**[:SENSe][:FIMPedance]:CONTact1:VERify:BUF1:FEED:CONTrol[:STATe]**	NA	NA	
			[:SENSe][:FIMPedance]:CONTact1:VERify:BUF1:FEED:CONTrol[:STATe]	NA	NA	
		Buffer size	**[:SENSe][:FIMPedance]:CONTact1:VERify:BUF1:POINts**	NA	NA	
Contact Check	Data buffer setup	Buffer size	**[:SENSe][:FIMPedance]:CONTact1:VERify:BUF2:POINts**	NA	NA	
Key lock	ON/OFF		**:SYSTem:KLOCk**	♦	♦	

Table B-1 At-a-glance SCPI command correspondence between 4268A, 4288A, and E4981A (by function)

Function	Item to Be Set Up/Executed		SCPI Command			Note
			E4981A	4268A	4288A	
Beeper	ON/OFF		**:CALCulate:COMParator:BEEPer[:STATe]**	♦	♦	
			:SYSTem:BEEPer:STATe	♦	♦	
	Beep mode setup		**:SYSTem:BEEPer:TONE**	NA	NA	
	Beep ON		**:SYSTem:BEEPer:IMMediate**	NA	NA	
	Comparator Beep condition		**:CALCulate:COMParator:BEEPer:CONDition**	♦	♦	
Status Report Structure	Clear		***CLS**	♦	♦	
	Status byte register value readout		***STB?**	♦	♦	
	Service request enable register setup		***SRE**	♦	♦	
	Enable/disable status register		**:STATus:OPERation:UPDate**	NA	NA	
	Standard event status register	Register value readout	***ESR?**	♦	♦	
		OPC bit setup	***OPC**	♦	♦	
		Enable register setup	***ESE**	♦	♦	
	Operation status register	Clear	**:STATus:PRESet**	♦	♦	
		Condition register value readout	**STATus:OPERation:CONDition?**	♦	♦	
		Enable register setup	**:STATus:OPERation:ENABle**	♦	♦	
		Event register value readout	**:STATus:OPERation[:EVENt]?**	♦	♦	

Table B-1 At-a-glance SCPI command correspondence between 4268A, 4288A, and E4981A (by function)

Function	Item to Be Set Up/Executed		SCPI Command			Note
			E4981A	4268A	4288A	
External Connector	GPIB Address		**:SYSTem:COMMunicate:GPIB[:SELF];ADDRess**	NA	NA	
	LAN setup	Fixed IP address	**:SYSTem:COMMunicate:LAN[:SELF]:ADDRess**	NA	NA	
		Fixed Gateway	**:SYSTem:COMMunicate:LAN[:SELF]:DGATeway**	NA	NA	
		Fixed Subnet Mask	**:SYSTem:COMMunicate:LAN[:SELF]:SMAS?**	NA	NA	
		AUTO IP	**:SYSTem:COMMunicate:LAN[:SELF]:CONFigure**	NA	NA	
	LAN status	Address	**:SYSTem:COMMunicate:LAN[:SELF]:CURRent:ADDRess?**	NA	NA	
		Gateway	**:SYSTem:COMMunicate:LAN[:SELF]:CURRent:DGATeway?**	NA	NA	
		Subnet Mask	**:SYSTem:COMMunicate:LAN[:SELF]:CURRent:SMASk?**	NA	NA	
	MAC Address		**:SYSTem:COMMunicate:LAN[:SELF]:MAC?**	NA	NA	
	Reconnect after reset to facorty state		**:SYSTem:COMMunicate:LAN[:SELF]:PRESet**	NA	NA	
	Reconnection		**:SYSTem:COMMunicate:LAN[:SELF]:RESTart**	NA	NA	
	Socket Control Port number		**:SYSTem:COMMunicate:LAN[:SELF]:CONTrol**	NA	NA	

Table B-1 At-a-glance SCPI command correspondence between 4268A, 4288A, and E4981A (by function)

Function	Item to Be Set Up/Executed	SCPI Command			Note
		E4981A	4268A	4288A	
Internal Clock	Date	**:SYSTem:DATE**	NA	NA	
	Time	**:SYSTem:TIME**	NA	NA	
	Zone	**:SYSTem:TZONe**	NA	NA	
Others	Executes self-test	***TST?**	♦	♦	E4981A *TST? does not perform test. This is a dummy command for 4268A/4288Ac ompatibility.
	Readout of the model name and firmware version	***IDN?**	♦	♦	
	Readout of the installed option number	***OPT?**	♦	♦	
	Reads 1 when operation completes	***OPC?**	♦	♦	
	Readout of the occurred error information	**:SYSTem:ERRor?**	♦	♦	
	Readout of SCPI version	NA	**SYST:VERS?**	**SYST:VERS?**	
	Waits for completion of operation	***WAI**	♦	♦	
	Reset	**:SYSTem:RESTart**	NA	NA	
	LEARN	***LRN?**	NA	NA	

Table B-1 At-a-glance SCPI command correspondence between 4268A, 4288A, and E4981A (by function)

Function	Item to Be Set Up/Executed		SCPI Command			Note
			E4981A	4268A	4288A	
Others	Handler interface signal control	BIN	:TEST:HANDler:BIN	NA	NA	
		COMP	:TEST:HANDler:COMP	NA	NA	
		OVLD	:TEST:HANDler:STATus:OVLD	NA	NA	
		No_Cont/Low_C	:TEST:HANDler:STATus:NC	NA	NA	
		Alarm	:TEST:HANDler:STATus:ALARm	NA	NA	
		Index	:TEST:HANDler:STATus:INDex	NA	NA	
		EOM	:TEST:HANDler:STATus:EOM	NA	NA	
		Ready for trigger	:TEST:HANDler:STATus:RDYTrig	NA	NA	
		Trigger	:TEST:HANDler:TRIGger?	NA	NA	
		Key Lock	:TEST:HANDler:KEYLock?	NA	NA	
		Volatge setting for input trigger	:SYSTem:HANDler:INPut:RANGe	NA	NA	
	Scanner interface signal control	EOM	:TEST:SCANner:EOM	NA	NA	
		Index	:TEST:SCANner:INDex	NA	NA	
		Channel	:TEST:SCANner:CH?	NA	NA	
		Channel Valid	::TEST:SCANner:VALID?	NA	NA	
		Trigger	:TEST:SCANner:TRIGger?	NA	NA	
		Volatge setting for input trigger	:SYSTem:SCANner:INPut:RANGe	NA	NA	
	Rear Trigger	Trigger	:TEST:REAR:TRIGger	NA	NA	

The “♦” symbol in Table B-1 indicates that the value is the same as that of the E4981A.

Table B-2 At-a-glance SCPI command correspondence between 4268A, 4288A and E4981A (by alphabetical order)

E4981A	4268A	4288A
[:A]		
:ABORt	♦	♦
[:C]		
:CALCulate1:COMParator:AUXBin	♦	♦
:CALCulate1:COMParator:BEEPer:CONDition	♦	♦
:CALCulate1:COMParator:BEEPer[:STATe]	♦	♦
:CALCulate1:COMParator:CLEar	N/A	♦
:CALCulate1:COMParator:COUNt:CLEar	♦	♦
:CALCulate1:COMParator:COUNt:DATA?	♦	♦
:CALCulate1:COMParator:COUNt:MULTiple:DATA?	N/A	♦
:CALCulate1:COMParator:COUNt:MULTiple:OVLD?	N/A	♦
:CALCulate1:COMParator:COUNt:OVLD?	N/A	♦
:CALCulate1:COMParator:COUNt[:STATe]	♦	♦
:CALCulate1:COMParator:MODE	♦	♦
:CALCulate1:COMParator:PRIMary:BIN{1-9}[:LIMit]	♦	♦
:CALCulate1:COMParator:PRIMary:BIN{1-9}:STATe	♦	♦
:CALCulate1:COMParator:PRIMary:NOMinal	♦	♦
:CALCulate1:COMParator:SECondary:LIMit	♦	♦
:CALCulate1:COMParator:SECondary:STATe	♦	♦
:CALCulate1:COMParator[:STATe]	♦	♦
:CALCulate1:FORMat	♦	♦
:CALCulate1:MATH:EXPRession:CATalog?	♦	♦
:CALCulate1:MATH:EXPRession:NAME	♦	♦
:CALCulate1:MATH:STATe	♦	♦
:CALCulate2:FORMat	♦	♦
:CALCulate2:MATH:EXPRession:CATalog?	♦	♦
:CALCulate2:MATH:EXPRession:NAME	♦	♦
:CALCulate2:MATH:STATe	♦	♦
:CALCulate3:MATH:STATe	♦	♦
:CALCulate4:MATH:STATe	♦	♦
:CALibration:CABLe:CORRection:CLEar	N/A	N/A

Table B-2 At-a-glance SCPI command correspondence between 4268A, 4288A and E4981A (by alphabetical order)

E4981A	4268A	4288A
:CALibration:CABLe:CORRection:COLLect[:ACQuire]:LOAD	N/A	N/A
:CALibration:CABLe:CORRection:COLLect[:ACQuire]:OPEN	N/A	N/A
:CALibration:CABLe:CORRection:COLLect[:ACQuire]:REFerence	N/A	N/A
:CALibration:CABLe:CORRection:SAVE	N/A	N/A
:CALibration:CABLe:CORRection:STATe?	N/A	N/A
:CALibration:CABLe[:LENGth]	♦	♦
[:D]		
:DATA[:DATA]?	♦	♦
:DATA:FEED:BUF1	:DATA:FEED	:DATA:FEED
:DATA:FEED:BUF2	:DATA:FEED	:DATA:FEED
:DATA:FEED:CONTrol:BUF1[:STATe]	:DATA:FEED:CONTrol	♦
:DATA:FEED:CONTrol:BUF2[:STATe]	:DATA:FEED:CONTrol	♦
:DATA:FEED:CONTrol:BUF3[:STATe]	:DATA:FEED:CONTrol	♦
:DATA:FEED:CONTrol[:STATe]	:DATA:FEED:CONTrol	♦
:DATA:FEED[:SOURce]	:DATA:FEED	:DATA:FEED
:DATA:POINts:BUF1	:DATA:POINts	♦
:DATA:POINts:BUF2	:DATA:POINts	♦
:DATA:POINts:BUF3	:DATA:POINts	♦
:DATA:POINts[:DATA]	:DATA:POINts	♦
:DATA:REFerence1:DATA	N/A	N/A
:DATA:REFerence1:FILL	N/A	N/A
:DATA:REFerence2:DATA	N/A	N/A
:DATA:REFerence2:FILL	N/A	N/A
:DISPlay:CCLear	N/A	N/A
:DISPlay:LINE	N/A	N/A
:DISPlay:PAGE	:DISPlay[:WINDow]:TEXT2:PAGE	♦
:DISPlay[:WINDow][:STATe]	♦	♦
:DISPlay[:WINDow]:TEXT1[:DATA]:FMSD:DATA	N/A	:DISPlay:WINDow:TEXT1:DATA:FMSD:DATA
:DISPlay[:WINDow]:TEXT1[:DATA]:FMSD[:STATe]	N/A	:DISPlay:WINDow:TEXT1:DATA:FMSD:DATA
:DISPlay[:WINDow]:TEXT2[:DATA]:FMSD:DATA	N/A	:DISPlay:WINDow:TEXT1:DATA:FMSD:DATA

Table B-2 At-a-glance SCPI command correspondence between 4268A, 4288A and E4981A (by alphabetical order)

E4981A	4268A	4288A
:DISPlay[:WINDow]:TEXT2[:DATA]:FMSD[:STATe]	N/A	**:DISPlay:WINDow:TEXT1:DATA:FMSD:DATA**
[:F]		
:FETCh?	♦	♦
:FORMat:ASCii:LONG	N/A	N/A
:FORMat:BORDer	N/A	N/A
:FORMat:STATus:EXTension	N/A	N/A
:FORMat[:DATA]	♦	♦
[:H]		
:HCOPy:SDUMp:DATA?	N/A	N/A
[:I]		
:INITiate:CONTinuous	♦	♦
:INITiate[:IMMed iate]	♦	♦
[:M]		
:MMEMory:DELete[:REGister]	N/A	N/A
:MMEMory:LOAD:STATe[:REGister]	***RCL**	***RCL**
:MMEMory:STORe:STATe[:REGister]	***SAV**	***SAV**
[:R]		
:READ?	♦	♦
[:S]		
[:SENSe]:AVERage:COUNt	♦	♦
[:SENSe]:AVERage[:STATe]	♦	♦
[:SENSe]:CORRection:CKIT:STAN1:FORMat	N/A	♦
[:SENSe]:CORRection:CKIT:STAN2:FORMat	N/A	♦
[:SENSe]:CORRection:CKIT:STAN3[:DATA]	♦	♦
[:SENSe]:CORRection:CKIT:STAN3:FORMat	♦	♦
[:SENSe]:CORRection:COLLect[:ACQuire]	♦	♦
[:SENSe]:CORRection:COLLect:LOAD:RANGe:AUTO	♦	♦
[:SENSe]:CORRection:COLLect:STAN3:RANGe:AUTO	♦	N/A
[:SENSe]:CORRection:DATA	♦	♦
[:SENSe]:CORRection:LOAD[:STATe]	**[:SENSe]:CORRection:COLLect[:METHod]**	♦
[:SENSe]:CORRection:MULTiple:CHANnel	♦	♦

Table B-2 At-a-glance SCPI command correspondence between 4268A, 4288A and E4981A (by alphabetical order)

E4981A	4268A	4288A
[:SENSe]:CORRection:MULTiple:CKIT:STAN3[:STATe]	♦	♦
[:SENSe]:CORRection:MULTiple[:STATe]	♦	♦
[:SENSe]:CORRection:OFFSet:DATA	N/A	♦
[:SENSe]:CORRection:OFFSet[:STATe]	N/A	♦
[:SENSe]:CORRection:OPEN[:STATe]	N/A	♦
[:SENSe]:CORRection:SHORt[:STATe]	N/A	♦
[:SENSe]:DETector:DELay1	N/A	N/A
[:SENSe]:DETector:DELay2	N/A	N/A
[:SENSe]:DETector:DELay3	N/A	N/A
[:SENSe][:FIMPedance]:APERture[:MODE]	♦	♦
[:SENSe][:FIMPedance]:APERture:TIME	N/A	N/A
[:SENSe][:FIMPedance]:CONTact1:VERify:BUF1:DATA?	N/A	N/A
[:SENSe][:FIMPedance]:CONTact1:VERify:BUF1:FEED:CONTrol:INTerval	N/A	N/A
[:SENSe][:FIMPedance]:CONTact1:VERify:BUF1:FEED:CONTrol[:STATe]	N/A	N/A
[:SENSe][:FIMPedance]:CONTact1:VERify:BUF1:POINts	N/A	N/A
[:SENSe][:FIMPedance]:CONTact1:VERify:BUF2:DATA?	N/A	N/A
[:SENSe][:FIMPedance]:CONTact1:VERify:BUF2:FEED:CONTrol:INTerval	N/A	N/A
[:SENSe][:FIMPedance]:CONTact1:VERify:BUF2:FEED:CONTrol[:STATe]	N/A	N/A
[:SENSe][:FIMPedance]:CONTact1:VERify:BUF2:POINts	N/A	N/A
[:SENSe][:FIMPedance]:CONTact1:VERify[:STATe]	N/A	N/A
[:SENSe][:FIMPedance]:CONTact1:VERify:THReshold1	N/A	N/A
[:SENSe][:FIMPedance]:CONTact1:VERify:THReshold2	N/A	N/A
[:SENSe][:FIMPedance]:CREJect[:STATe]	N/A	♦
[:SENSe][:FIMPedance]:CREJect:LIMit	N/A	♦
[:SENSe][:FIMPedance]:RANGe:AUTO	♦	♦
[:SENSe][:FIMPedance]:RANGe[:UPPer]	♦	♦
:SOURce:FREQuency[:CW]	♦	♦
:SOURce:VOLTage:ALC[:STATe]	♦	N/A
:SOURce:VOLTage[:LEVel][:IMMediate][:AMPLitude]	♦	♦
:SOURce:VOLTage:MODE	♦	N/A

Table B-2 At-a-glance SCPI command correspondence between 4268A, 4288A and E4981A (by alphabetical order)

E4981A	4268A	4288A
:STATus:OPERation:CONDition?	♦	♦
:STATus:OPERation:ENABle	♦	♦
:STATus:OPERation[:EVENt]?	♦	♦
:STATus:OPERation:UPDate	N/A	N/A
:STATus:PRESet	♦	♦
:STATus:QUEStionable:CONDition?	♦	♦
:STATus:QUEStionable:ENABle	♦	♦
:STATus:QUEStionable[:EVENt]?	♦	♦
:SYSTem:BEEPer[:IMMed iate]	N/A	N/A
:SYSTem:BEEPer:STATe	♦	♦
:SYSTem:BEEPer:TONE	N/A	N/A
:SYSTem:COMMunicate:GPIB[:SELF]:ADDRess	N/A	N/A
:SYSTem:COMMunicate:LAN[:SELF]:ADDRess	N/A	N/A
:SYSTem:COMMunicate:LAN[:SELF]:CONFigure	N/A	N/A
:SYSTem:COMMunicate:LAN[:SELF]:CONTrol	N/A	N/A
:SYSTem:COMMunicate:LAN[:SELF]:CURRent:ADDRess?	N/A	N/A
:SYSTem:COMMunicate:LAN[:SELF]:CURRent:DGATeway?	N/A	N/A
:SYSTem:COMMunicate:LAN[:SELF]:CURRent:SMASk?	N/A	N/A
:SYSTem:COMMunicate:LAN[:SELF]:DGATeway	N/A	N/A
:SYSTem:COMMunicate:LAN[:SELF]:MAC?	N/A	N/A
:SYSTem:COMMunicate:LAN[:SELF]:PRESet	N/A	N/A
:SYSTem:COMMunicate:LAN[:SELF]:RESTart	N/A	N/A
:SYSTem:COMMunicate:LAN[:SELF]:SMASk	N/A	N/A
:SYSTem:DATE	N/A	N/A
:SYSTem:ERRor[:NEXT]?	♦	♦
:SYSTem:FSHift	N/A	♦
:SYSTem:HANDler:TRIGger:VOLTage	N/A	N/A
:SYSTem:KLOCk	♦	♦
:SYSTem:PRESet	♦	♦
:SYSTem:RESTart	N/A	N/A
:SYSTem:SCANner:TRIGger:VOLTage	N/A	N/A
:SYSTem:TIME	N/A	N/A

Table B-2 At-a-glance SCPI command correspondence between 4268A, 4288A and E4981A (by alphabetical order)

E4981A	4268A	4288A
:SYSTem:TZONe	N/A	N/A
[:T]		
:TEST:HANDler:BIN	:TEST:A1:HANDler:BIN	N/A
:TEST:HANDler:COMP	:TEST:A1:HANDler:COMP	N/A
:TEST:HANDler:KEYLock	:TEST:A1:HANDler:KEYLock	N/A
:TEST:HANDler:MODE	:TEST:A1:HANDler:MODE	N/A
:TEST:HANDler:STATus:ALARm	:TEST:A1:HANDler:ALARm	N/A
:TEST:HANDler:STATus:EOM	:TEST:A1:HANDler:EOM	N/A
:TEST:HANDler:STATus:INDex	:TEST:A1:HANDler:INDex	N/A
:TEST:HANDler:STATus:NC	:TEST:A1:HANDler:STATus	N/A
:TEST:HANDler:STATus:OVLD	:TEST:A1:HANDler:STATus:OVLD	N/A
:TEST:HANDler:STATus:RDYTrig	N/A	N/A
:TEST:HANDler:TRIGger?	:TEST:A1:HANDler:TRIGger?	N/A
:TEST:REAR:TRIGger	:TEST:A1:REAR:TRIGger?	N/A
:TEST:SCANner:CH?	:TEST:A1:SCANner:CH?	N/A
:TEST:SCANner:EOM	:TEST:A1:SCANner:EOM	N/A
:TEST:SCANner:INDex	:TEST:A1:SCANner:INDex	N/A
:TEST:SCANner:MODE	:TEST:A1:SCANner:MODE	N/A
:TEST:SCANner:TRIGger?	:TEST:A1:SCANner:TRIGger?	N/A
:TEST:SCANner:VALID?	:TEST:A1:SCANner:VALID?	N/A
:TRIGger[:SEQ1]:DELay	♦	N/A
:TRIGger[:SEQ1]:SLOPe	N/A	N/A
:TRIGger[:SEQ1]:SOURce	:TRIGger[:SEQuence1]:SOURce	:TRIGger[:SEQuence1]:SOURce
:TRIGger[:SEQ1][:IMMediate]	:TRIGger[:SEQuence1]:IMMediate	:TRIGger[:SEQuence1]:IMMediate
:TRIGger:SEQ2:DELay	♦	:TRIGger[:SEQuence1]:DELay
[*]		
*CLS	♦	♦
*ESE	♦	♦
*ESR?	♦	♦
*IDN?	♦	♦
*LRN?	N/A	N/A
*OPC	♦	♦

Table B-2 At-a-glance SCPI command correspondence between 4268A, 4288A and E4981A (by alphabetical order)

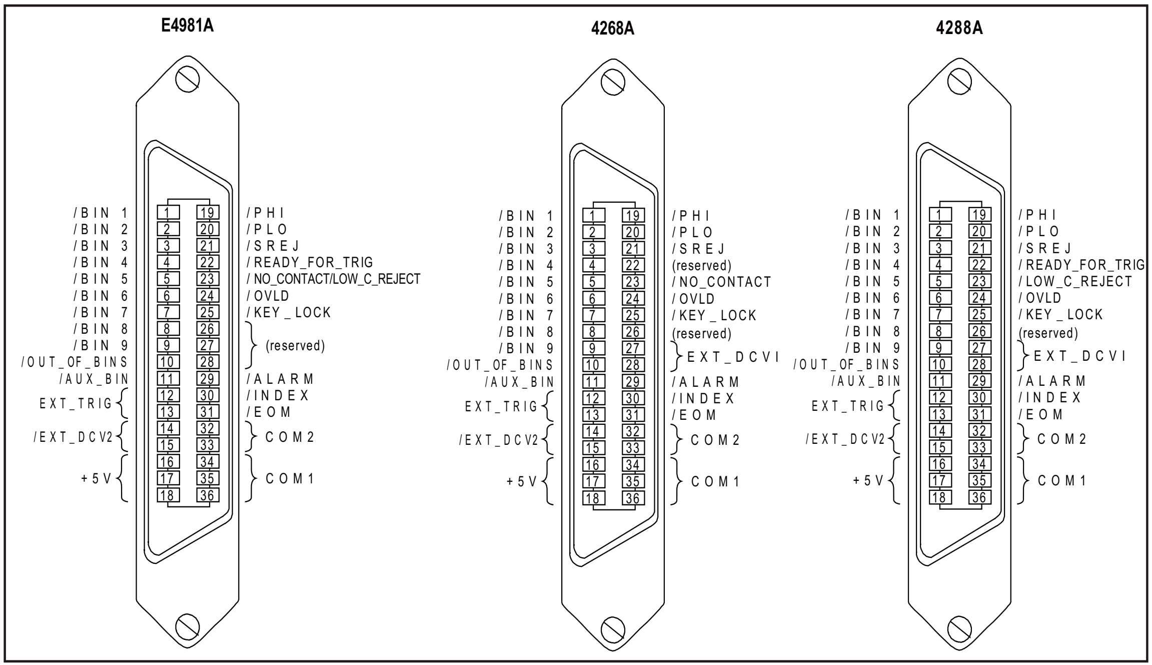
E4981A	4268A	4288A
*OPC?	♦	♦
*OPT?	♦	♦
*RCL	♦	♦
*RST	♦	♦
*SAV	♦	♦
*SRE	♦	♦
*STB?	♦	♦
*TRG	♦	♦
*TST?	♦	♦
*WAI	♦	♦

The “♦” symbol in Table B-2 indicates that the value is the same as that of the E4981A.

Comparison of Interfaces

Handler interface

Figure B-1 Handler interface pin assignment



e4981aue0024

Table B-3 Handler interface factory setting

	E4981A	4268A	4288A
Selection of judgment output signal pull-up power supply	N/A	External power supply (5 V to 24 V)	External power supply (5 V to 24 V)
Selection of operation output signal pull-up power supply	External power supply (5 V to 24 V)	External power supply (5 V to 15 V)	External power supply (5 V to 24 V)
Voltage range of input signal drive power supply	5-24 V	5-15 V	5-24 V

Scanner interface

Figure B-2 Scanner interface pin assignment

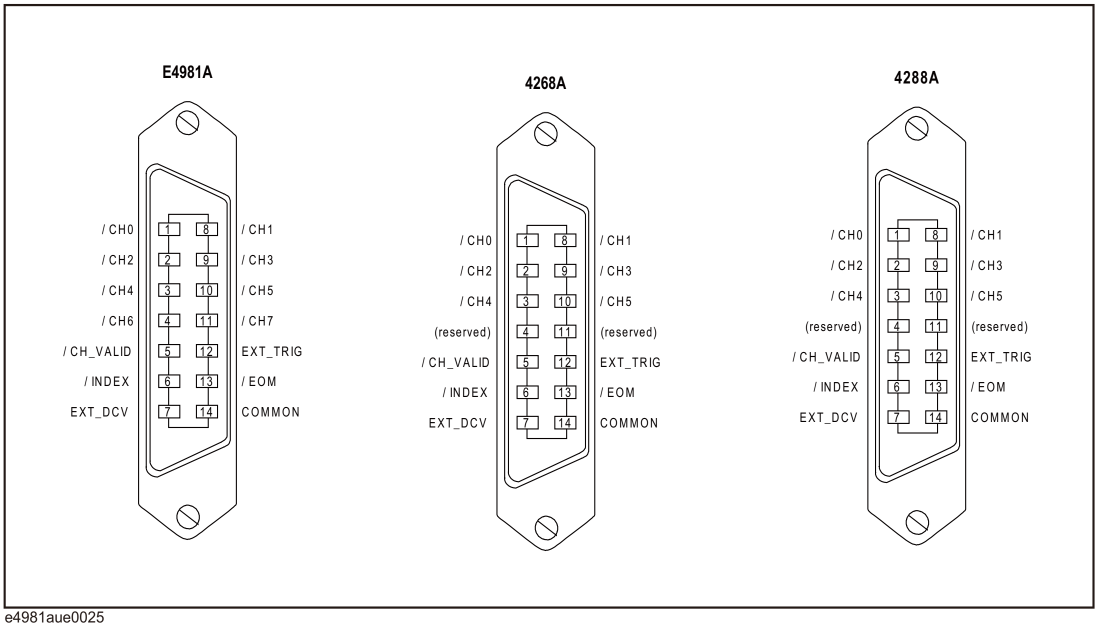


Table B-4 Scanner interface factory setting

	E4981A	4268A	4288A
Voltage range of input signal drive power supply	5-15 V	5-15 V	5-15 V

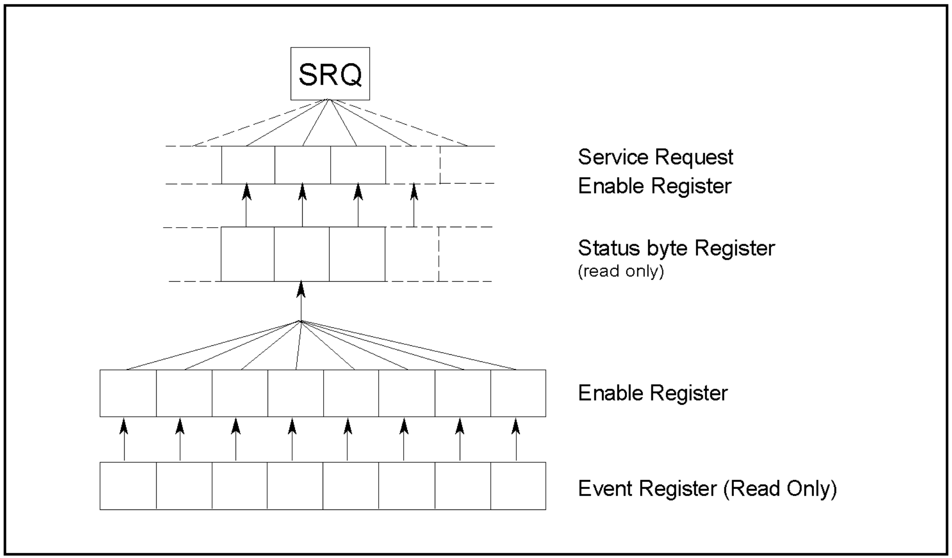
C Status Reporting System

This appendix describes the status reporting system of the Keysight E4981A.

General Status Register Model

The Keysight E4981A has a status reporting system for monitoring the instrument's condition.

Figure C-1 **General status register model**



4294ape021

The status reporting system has the hierarchical structure shown in Figure C-1.When the instrument satisfies a particular condition, the corresponding bit of the event register is set to "1." Therefore, you can check the instrument status by reading the event register.

When the event register bit is set to "1" and a corresponding enable register bit (a bit marked with an arrow in Figure C-1) is also "1," the summary bit of the status byte register is set to "1." You can read the status byte register by using the serial poll.

If the bit of the service request enable register is "1," a service request (SRQ) is generated by the positive transition of the corresponding status byte register bit. By generating an SRQ, you can notify the controller that the E4981A is requesting service. In other words, you can program interruption by using an SRQ. For more information on using SRQ, see "Waiting For Completion Of Measurement (Detecting Completion Of Measurement)" on page 61 in Chapter 5 or "Detecting the occurrence of an error" on page 84 in Chapter 8.

Event register

The event register reflects the corresponding condition of the E4981A (e.g., occurrence of an event) as a bit status. These bits continuously monitor changes in the E4981A's state and change the bit status when the condition (e.g., change bit status to "1" if a specific event occurs) for each bit is met. You cannot change the bit status by issuing a SCPI command.

The Keysight E4981A has the following event registers:

- Standard Event Status Register (see Table C-2 for details.)
- Operation Status Event Register (see Table C-3 for details.)

Enable register

Setting the enable register allows you to specify event register bits that can set "1" to the summary bit of the status byte register when an event occurs. The register bits work as mask bits; setting "1" to an enable register will enable a corresponding bit in the event register.

For example, when you want to set "1" as the summary bit in the status byte register by a specific register condition, set the corresponding enable register to "1."

Status byte register

If the enabled event register is set to "1," a corresponding bit of the status byte register is also set to "1." This register also indicates the output queue and SRQ status.

The value of the status byte register can be read by using the `*STB?` command on page 162 or the serial poll (SPOLL statement in HTBasic) from the controller. The `*STB?` command sets the analyzer to remote mode. On the other hand, the SPOLL statement in HTBasic reads the status byte register value directly without the instrument being set to remote. Therefore, you can continue to operate the front panel keys while a controller is reading the status byte register.

Reading the status byte register by using the `*STB?` command does not affect the contents of the status byte register. However, reading it with the SPOLL statement of HTBasic will clear the RQS bit in the status byte register.

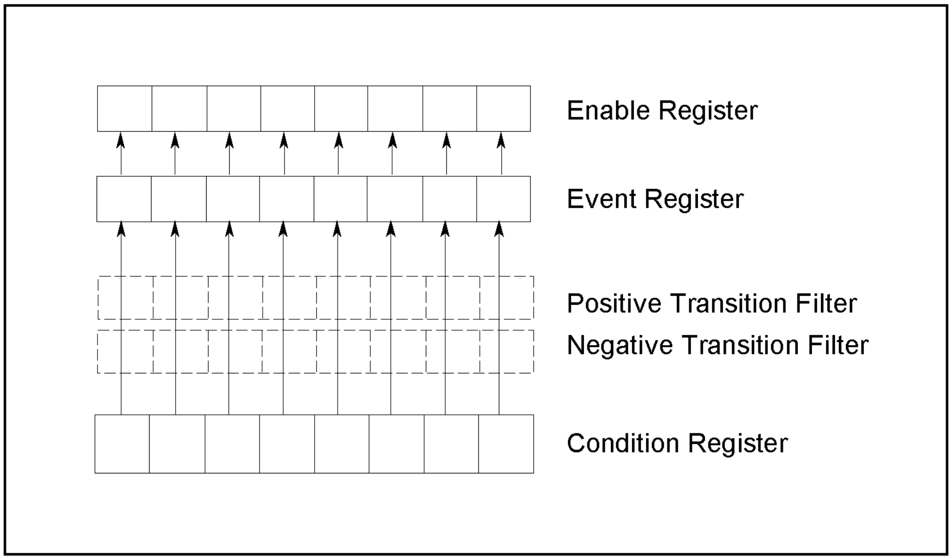
Setting the service request enable register using the `*SRE` on page 162 can generate a service request synchronously with the status byte register.

Condition register and transition filter

When the status register has a transition filter, there is a lower register called a condition register under the event register. The transition filter is between the event register and the condition register.

The transition filter enables you to select a positive and/or negative transition of the condition register bit in order to set a bit in the corresponding event register. For example, using the negative transition filter to set bit 3 to “1” causes bit 3 of the event register to be set to “1” when bit 3 of the condition register makes a negative transition, that is, changes from 1 to 0.

Figure C-2 Transition filter and condition register



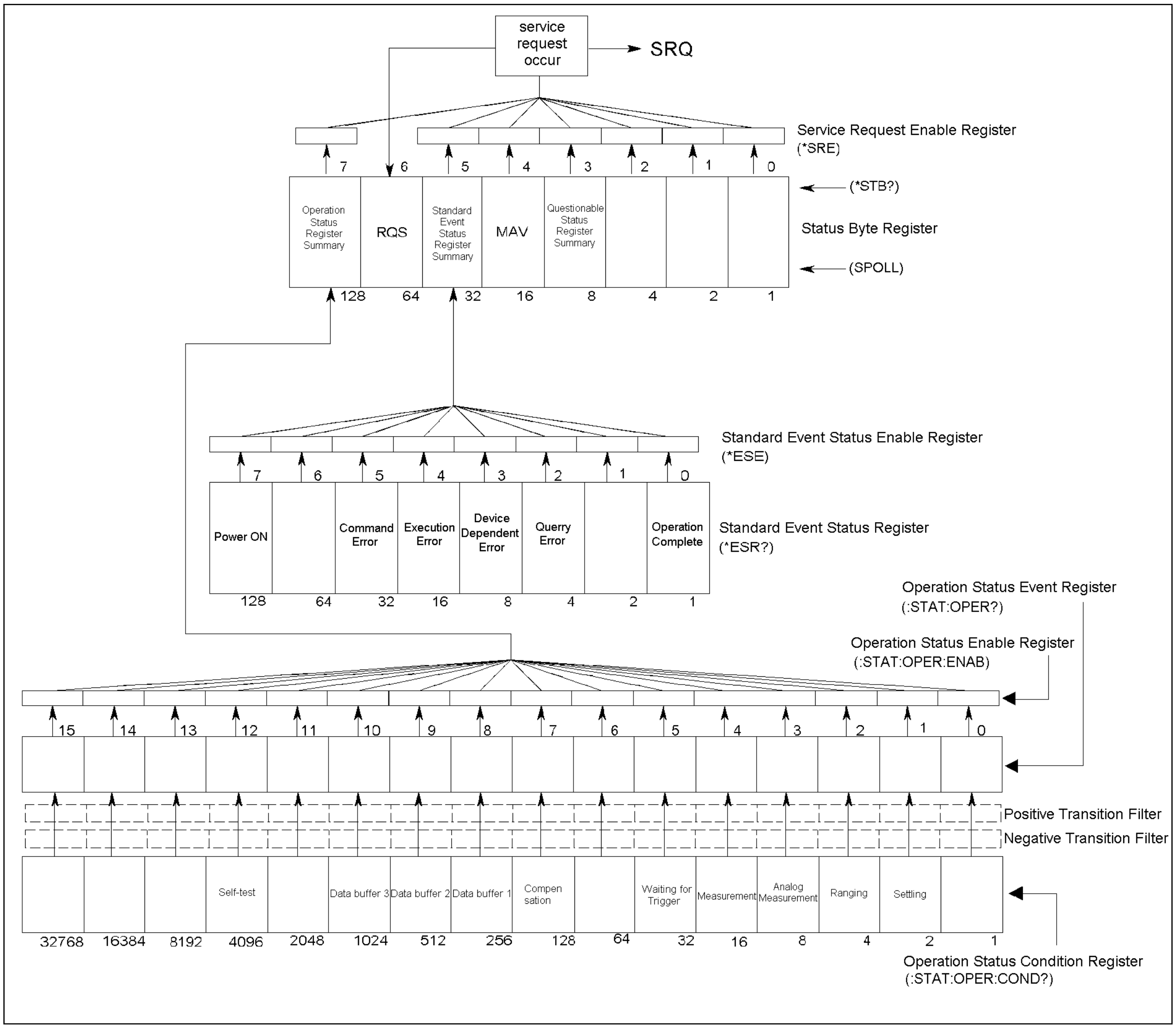
4294ape022

The E4981A's condition register and transition filter work only with the operation status register. However, the E4981A's transition filter’s setting is fixed so that bits 5, 8, 9, 10 of the event register are set to “1” when the condition register makes a positive transition (i.e., changes from 0 to 1) and bits 1, 2, 3, 4, 7, 12 of the event register are set to “1” when the condition register makes a negative transition (i.e., changes from 1 to 0).

Status Register Structure

The status reporting system has the hierarchical structure shown in Figure C-3. The status byte register is a summary of registers in the lower level. This section describes the status registers in each hierarchy. Each bit of the status register is described in Table C-1 through Table C-3.

Figure C-3 Status register structure


service request occur
SRQ
Service Request Enable Register (*SRE)
7
6
5
4
3
2
1
0
(*STB?)
Operation Status Register Summary
RQS
Standard Event Status Register Summary
MAV
Questionable Status Register Summary
Status Byte Register
(SPOLL)
128
64
32
16
8
4
2
1
Standard Event Status Enable Register (*ESE)
Power ON
Command Error
Execution Error
Device Dependent Error
Querry Error
Operation Complete
Standard Event Status Register (*ESR?)
Operation Status Event Register (:STAT:OPER?)
Operation Status Enable Register (:STAT:OPER:ENAB)
15
14
13
12
11
10
9
8
Positive Transition Filter
Negative Transition Filter
Self-test
Data buffer 3
Data buffer 2
Data buffer 1
Compen sation
Waiting for Trigger
Measurement
Analog Measurement
Ranging
Settling
32768
16384
8192
4096
2048
1024
512
256
Operation Status Condition Register (:STAT:OPER:COND?)

4288ape013

Table C-1 Status Bit Definitions of Status Byte (STB)

Bit Position	Name	Description
0 to 3	Not used	Always 0
4	MAV (Message Available)	Set to “1” when the output queue contains data; reset to “0” when all of the data has been retrieved.
5	Standard Event Status Register Summary	Set to “1” when one of the enabled bits in the status event status register is set to “1.”
6	RQS (Request Status) bit for SPOLL MSS (Master Summary Status) bit for *STB?	Set to “1” when any of the status byte register bits enabled by the service request enable register is set to “1”; reset to “0” when all of the data has been retrieved through serial polling.
7	Operation Status Register Summary	Set to “1” when one of the enabled bits in the operational status register is set to “1.”

Issuing the *CLS command will clear all bits from the status byte register.

Table C-2 Status bit definitions of the Event Status Register (ESR)

Bit Position	Name	Description
0	Operation Complete	Set to “1” upon completion of all operations done by commands that precede the *OPC command on page 159.
1	Not used	Always 0
2	Query Error	1. Set to “1” when the E4981A receives a data output request but there is no data to output. 2. Set to “1” when the data of the E4981A's output queue has been cleared because of a new message received before the completion of data output.
3	Device Specific Error	Set to “1” when an error has occurred and the error is not a command, query, or execution error.
4	Execution Error	1. Set to “1” when any parameter in a SCPI command exceeds its input range or is inconsistent with the E4981A’s capabilities. 2. Set to “1” when a SCPI command cannot be properly executed due to some condition of the E4981A.
5	Command Error	1. Set to “1” when an IEEE 488.2 syntax error occurs (a command sent to the E4981A does not follow the IEEE 488.2 syntax). Possible violations include the command parameter violating the E4981A listening formats or being otherwise unacceptable. 2. Set to “1” when a semantic error occurs. Possible errors include a command containing misspellings or an IEEE 488.2 command that is not supported by the E4981A. 3. Set to “1” when GET (Group Execution Trigger) is input while a program message is being received.
6	Not used	Always 0
7	Power ON	Set to “1” when the E4981A is powered ON.

Issuing the *CLS command will clear all bits from the standard event status register.

Table C-3 Status bit definitions of the Operation Status Register

Bit Position	Name	Description: Condition Register	Description: Event Register
0	Not used	Always 0	Always 0
1	Settling	Set to “1” during the waiting time to stabilize the measurement signal.	Set to “1” after the waiting time needed to stabilize the measurement signal has elapsed.
2	Ranging	Set to “1” while switching the measurement range.	Set to “1” when measurement range switching is completed.
3	Analog Measurement	Set to “1” during analog measurement[1].	Set to “1” when analog measurement is completed.
4	Measurement	Set to “1” during measurement[2].	Set to “1” when measurement is completed.
5	Waiting for Trigger	Set to “1” while the instrument is waiting for a trigger[3].	Set to “1” when the instrument starts waiting for a trigger.
6	Not used	Always 0	Always 0
7	Correction	Set to “1” during correction data measurement.	Set to “1” when the correction data measurement is completed.
8	Data buffer 1	Set to “1” while data buffer 1 fills.	Set to “1” when data buffer 1 is full.
9	Data buffer 2	Set to “1” while data buffer 2 fills.	Set to “1” when data buffer 2 is full.
10	Data buffer 3	Set to “1” while data buffer 3 fills.	Set to “1” when data buffer 3 is full.
11	Not used	Always 0	Always 0
12	Self-test	Set to “1” during self-test.	Set to “1” when self-test is completed.
13 to 15	Not used	Always 0	Always 0

[1] This is when the handler interface’s /INDEX signal is active.

[2] This is when the handler interface’s /EOM signal is active.

[3] This is when the trigger system is in trigger wait state. For more information on the trigger system, refer to “Trigger system” on page 57.

Issuing the *CLS command will clear all bits from the operation status event register.

The E4981A does not support events of the questionable status register. Therefore, all of the bits in this register are always 0.

Table C-4 Status bit definitions of the Questionable Status Register

Bit	Name	Description	
Position		Condition Register	Event Register
0 to 15	Not used	Always 0	Always 0

Using the Status Reporting System

You can manage the status report system by using the following commands in any combination:

- `*CLS` on page 157
- `*SRE` on page 162
- `*STB?` on page 162
- `*ESE` on page 157
- `*ESR?` on page 158
- `:STATus:PRESet` on page 271
- `:STATus:OPERation:ENABle` on page 270
- `:STATus:OPERation:CONDition?` on page 269
- `:STATus:OPERation[:EVENt]?` on page 270

For sample programs that demonstrate the use of the commands listed above, refer to "Waiting For Completion Of Measurement (Detecting Completion Of Measurement)" on page 61 in Chapter 5 or "Detecting the occurrence of an error" on page 84 in Chapter 8.

Turning Off the Status Reporting System

The E4981A status reporting system can be disabled. In a disabled condition, the data in status reporting system is not updated. The benefit of disabling Status Reporting System is that it shortens the End of Measurement (EOM) time. See measurement time in Specification and Supplemental Performance Characteristics in the E4981A User Guide for more details.

The Status Reporting System can be turned off using the `:STATus:OPERation:UPDate` on page 271.

D Operations when Overload, No Contact, or Low C is Detected

This appendix describes display output, GPIB/LAN/USB output, and handler interface output when an Overload, No contact, or Low C is detected.

Operations when Overload/No Contact/Low C is Detected

Table D-5 shows operations of the E4981A when one of the following items is detected:

- Overload
  - When the available measurement range is exceeded more than about 10% (refer to “Specifications and Supplemental Performance Characteristics” in *Operation Manual*)
  - When connection between the UNKNOWN terminals to DUT is not correct
  - When SLC is ON, if the applied signal level for DUT is dropped more than 10% from the setting level
- No contact:
  - When the contact check parameter is over the specified threshold
- Low C:
  - When measured primary parameter result is equal to or less than the boundary value specified for the Low C reject function
- Out of display range:
  - When measured result exceeds the allowable display range (refer to “Specifications and Supplemental Performance Characteristics” in *Operation Manual*) (regardless of fixed or floating point display)
  - When measured result exceeds the allowable display range for the fixed point display

Table D-5 At-a-glance table of operations when Overload/No contact/Low C reject is detected

	Display output			GPIB/LAN/USB output			Handler output (handler signal that becomes active)
	Measured value	Voltage/ current monitor value	Comparator sorting result	Measurement status	Measured value	Comparator sorting result	
Overload	OVLD	---	---	1	9.9E37	11	/OVLD
No contact	N.C.	---	---	2	9.9E37	11	/LOWC_OR_NC
Low C	Normal	Normal	LOWC[1]	2	Normal	Normal	/LOWC_OR_NC[2]
Out of display range[3]	-------	---	Normal	Normal	Normal	Normal	Normal

1. When Low C is detected, LOWC is displayed in BIN No. DISPLAY page.
2. /LOWC_OR_NC becomes active together with the signal that corresponds to the result of normal sorting judgment (judgment result when no error occurred).
3. When an out-of-display-range state is detected, measurement is performed normally but the measured values are not displayed.

E Error Messages

The Keysight E4981A provides error messages to indicate its operating status. This appendix describes the error messages of the E4981A in order of error number. To search for error messages alphabetically, refer to the *Operation Manual*.

Error Messages (Order of Error Number)

Error messages are displayed in the lower row of the E4981A's display. You can read them out by using the SCPI command. This section provides a description of each error message and its remedy.

NOTE

Errors with a negative error number are basically general errors defined by IEEE488.2 for GPIB/LAN/USB instruments. On the other hand, errors with a positive error number are defined specifically for the E4981A.

16 **Reference Measurement Aborted**

This error occurs when REF data measurement is aborted.

21 **1 MHz opt. not installed**

This error occurs when cable correction commands are sent to E4981A with Option 002, via GPIB/LAN/USB. This error is not generated with front panel operation.

41 **Correction Measurement Aborted**

This error occurs when the correction data measurement is aborted.

43 **Measurement failed**

A measurement failure has occurred during measuring the correction data.

46 **LOAD measurement incomplete**

This error occurs when the cable correction LOAD measurement is incomplete.

47 **OPEN measurement incomplete**

This error occurs when the cable correction OPEN measurement is incomplete.

48 **REF measurement incomplete**

This error occurs when the cable correction REF measurement is incomplete.

82 **Store failed**

This error occurs when external mass storage device fails or internal FLASH ROM hardware fails.

Contact Keysight Technology’s Sales and Service Office or the company from which you purchased the device.

83 **No data to load**

There is no setup data for the selected number or no external mass storage device is connected.

1070 **Fan failed**

Cooling fan hardware failure is detected.

Contact Keysight Technology’s Sales and Service Office or the company from which you purchased the device.

1080 **Power failed**

Power unit hardware failure is detected.

Contact Keysight Technology’s Sales and Service Office or the company from which you purchased the device.

1103 **A1 EEPROM write error**

An error is generated while writing data to A1 EEPROM.

Contact Keysight Technology’s Sales and Service Office or the company from which you purchased the device.

1200 **CPU bd FLASH ROM write error**

An error is generated while writing data to FLASH.

Contact Keysight Technology’s Sales and Service Office or the company from which you purchased the device.

1201 **CPU bd EEPROM write error**

An error is generated while writing data to EEPROM.

Contact Keysight Technology’s Sales and Service Office or the company from which you purchased the device.

-100 **Command error**

A comprehensive syntax error has occurred for which the E4981A cannot detect further details of the error. This error code simply indicates the occurrence of a command error that is defined in IEEE488.2,11.5.1.1.4.

-101 **Invalid character**

Invalid characters have been found in the program message string. For example, in a correct program message "`:CALC1:FORM CP`", an ampersand (&) is inserted by mistake to give "`:CALC1:FORM&CP`".

-102 **Syntax error**

There is a command or data type that cannot be recognized. For example, in the program message "`:SYST:PRES`", a colon (:) is inserted by mistake to give "`:SYST::PRES`".

-103 **Invalid separator**

The parser (syntax analysis program) expects a separator, but a character other than a separator has been sent. For example, although the correct way is to use ";" to separate two sent program messages such as "`:CALC1:FORM CP;*OPC?`", the semicolon (;) needed to separate the program messages is missing to give "`:CALC1:FORM CP *OPC?`".

-104 **Data type error**

The parser has recognized impossible data elements. For example, numeric value or string data is expected, but block data is sent.

-105 **GET not allowed**

A group execution trigger (GET) has been received in a program message. (Refer to IEEE488.2,7.7.)

-108 **Parameter not allowed**

The number of parameters is larger than required by the command. For example, although the `:CREJ:LIM` command requires one parameter such as "`:CREJ:LIM 3`", two parameters are added to give "`:CREJ:LIM 0,3`".

-109 **Missing parameter**

The number of parameters is less than required by the command. For example, although the `:CREJ:LIM` command requires one parameter such as "`:CREJ:LIM 3`", no parameter is added to give "`:CREJ:LIM`".

-112 **Program mnemonic too long**

The length of the header exceeds 12 characters. (Refer to IEEE488.2,7.6.1.4.1.)

-113 **Undefined header**

A header not defined for the E4981A has been received. For example, "`*XYZ`", which is not defined for the E4981A, is received.

-114 **Header suffix out of range**

The header suffix is out of range.

-120 **Numeric data error**

Numeric data is improper.

-121 **Invalid character in number**

An invalid character for the data type of the syntax analysis target has been received. For example, alphabetical characters exist in a decimal value or “9” exists in octal data.

-123 **Exponent too large**

The absolute value of the exponent has exceeded 32,000. (Refer to IEEE488.2,7.7.2.4.1.)

-124 **Too many digits**

The number of digits of the mantissa of the decimal value data element exceeds 255 except for preceding 0s. (Refer to IEEE488.27.7.2.4.1.)

-128 **Numeric data not allowed**

A numeric value data element (that does not violate the standard) has been received where the E4981A does not accept any numeric value data element.

-131 **Invalid suffix**

The suffix does not meet the syntax defined in IEEE488.2,7.7.3.2 or is inappropriate for the E4981A.

-134 **Suffix too long**

The length of suffix is long.

-138 **Suffix not allowed**

A suffix is added to a numeric value element that does not permit a suffix.

-140 **Character data error**

An error not included in the error numbers between -141 and -149 has occurred during the syntax analysis of a character data element.

-141 **Invalid character data**

A character data element has been recieved where E4981A does not accepts any character data element.

-148 **Character data not allowed**

Character data not allowed for this operation.

-150 **String data error**

An error not included in the error numbers between -151 and -159 has occurred during the syntax analysis of a string data element.

-151 **Invalid string data**

Character string data are expected, but the string data received are invalid for some reason. (Refer to IEEE488.2,7.7.5.2.) For example, the END message is received before the end quotation mark character appears.

-158 **String data not allowed**

A string data element has been received where the E4981A does not accept any string data element. For example, a parameter must be enclosed with double quotation marks ("...") but they are missing.

-161 **Invalid block data**

Block data are expected, but the block data received are invalid for some reason. (Refer to IEEE488.2,7.7.6.2.) For example, the END message is received before the length of the block data is reached.

-168 **Block data not allowed**

A block data element has been received where the E4981A does not accept any block data element.

-170 **Expression error**

An error not included in the error numbers between -171 and -179 has occurred during the syntax analysis of equation data.

-171 **Invalid expression**

The equation data element is invalid. (Refer to IEEE488.2,7.7.7.2.) For example, parentheses are not paired or a character violates the standard.

-178 **Expression data not allowed**

An equation data element has been received where the E4981A does not accept any equation data element.

-200 **Execution error**

A comprehensive execution error has occurred for which the E4981A cannot detect further details. This error code simply indicates the occurrence of an execution error that is defined in IEEE488.2,11.5.1.1.5.

-211 **Trigger ignored**

A trigger command or trigger signal has been received and recognized by the E4981A, but it is ignored due to the timing relationship with the E4981A. For example, this happens when the E4981A's trigger system is not in the Waiting for Trigger state.

-213 **Init ignored**

Another measurement has been being executed and the measurement start request (`:INITiate[:IMMediate]` command on page 228) has been ignored.

-214 **Trigger deadlock**

Indicates that the `:READ?` command on page 230 was ignored because the trigger source setting was MAN or BUS.

-222 **Data out of range**

A data element (that does not violate the standard) has been received out of the range defined for the E4981A.

-223 **Too much data**

The received block, equation, or string type program data complies with the standard, but the amount of data exceeds the limit that the E4981A can handle due to memory or device-specific conditions related to memory.

-224 **Illegal parameter value**

The parameter recieved is not correct. For example, though a correct program message was "**:CALC1:FORM CP**," a wrong program message, "**:CALC1:FORM RP**," was received.

-230 **Data corrupt or stale**

The data is invalid or a newly initiated read operation has not been completed since the latest access.

-250 **Mass storage error**

An error occurred while accessing the external mass storage device.

-321 **Out of memory**

The E4981A has insufficent memory to perform the requested operation.

-350 **Queue overflow**

The queue contains a certain code other than the code that caused this error. This indicates that an error has occurred due to insufficient space in the queue but has not been recorded.

-400 **Query error**

A comprehensive Query error has occurred for which the E4981A cannot detect further details. This code simply indicates the occurrence of a Query error that is defined in IEEE488.2,11.5.1.1.7 and 6.3.

-410 **Query INTERRPUTED**

This indicates the status that causes an "INTERRUPTED" Query error. (Refer to IEEE488.1,6.3.2.3.) This error occurs, for example, when data byte (DAB) or GET is received after Query but before the response has been completely sent.

-420 **Query UNTERMINATED**

This indicates the status that causes an "UNTERMINATED" Query error. (Refer to IEEE488.2,6.3.2.) This error occurs, for example, when the E4981A is specified as a talker and an incomplete program message is received.

-430 **Query DEADLOCKED**

This indicates the status that causes a "DEADLOCKED" Query error. (Refer to IEEE488.2,6.3.1.7.) This error occurs, for example, when both input and output buffers become full and the E4981A cannot continue processing.

-440 **Query UNTERMINATED after indefinite response**

In a certain program message, a Query that requests an ambiguous response has not yet been completely executed when a different Query is received. (Refer to IEEE488.2,6.5.7.5.7.)

Warning Messages (WARNING)

Warning messages are displayed to warn users. They are displayed in the lower row of the display of the E4981A. You cannot read them out using the SCPI command.

WARNING: Need corr meas

When the OPEN correction, SHORT correction or LOAD correction is ON, this is displayed when you change the setup of the cable length or measurement frequency shift (1 MHz). In this case, the OPEN correction, SHORT correction and LOAD correction are automatically turned OFF.

WARNING: Need load meas

This is displayed when you turn ON the LOAD correction from the front panel although the setups of the cable length and measurement frequency shift (1 MHz) differ from those when measuring/setting up the LOAD correction data. In this case, the LOAD correction is turned ON, but you need to measure the LOAD correction data again for accurate measurement.

WARNING: Need open meas

This is displayed when you turn ON the OPEN correction from the front panel although the setups of the cable length and measurement frequency shift (1 MHz) differ from those when measuring/setting up the OPEN correction data. In this case, the OPEN correction is turned ON, but you need to measure the OPEN correction data again for accurate measurement.

WARNING: Need short meas

This is displayed when you turn ON the SHORT correction from the front panel although the setups of the cable length and measurement frequency shift (1 MHz) differ from those when measuring/setting up the SHORT correction data. In this case, the SHORT correction is turned ON, but you need to measure the SHORT correction data again for accurate measurement.

WARNING: Out of limit

This is displayed if the correction data is out of the valid range when measuring the correction data. The valid range for each type of correction is as follows.

Type of correction	Valid range
OPEN correction	|Y| < 20 μS
SHORT correction	|Z| < 20 Ω
LOAD correction	|Zref| × 0.9 < |Z| < |Zref| × 1.1

In the above table, Y is the measured admittance value, Z is the measured impedance value, and Zref is the LOAD correction standard definition value.

WARNING: Improper high/low limits

The upper limit value is less than the lower limit value. Set the lower limit value to less than the upper limit value.

WARNING: Incompatible state file

The setting file recalled from external mass storage device has been saved using an E4981A with a different firmware version or different options. There may be some parameters set up incorrectly. Check the setting.

This message may appear due to option mismatch, firmware mismatch, check-sum error or state format mismatch.

KEYSIGHT
TECHNOLOGIES

This information is subject to change without notice.

Edition 3, October 8, 2019

E4981-90001
www.keysight.com