Table Of Contents

Agilent E5061B Network Analyzer Help

The E5061B help provides an easy access to the information related to the use of Agilent E5061B Network Analyzer. Pressing **Help** key on the front panel displays the topic related to the selected softkey.

You can navigate this help system through the navigation tools provided in the left or you can directly access the chapters:

For impedance measurement, see Option 005 Impedance Analysis

E5061B Firmware Revision	A.02.0x
Help Edition No.	2.0
Help Revision Date	October 15,2010

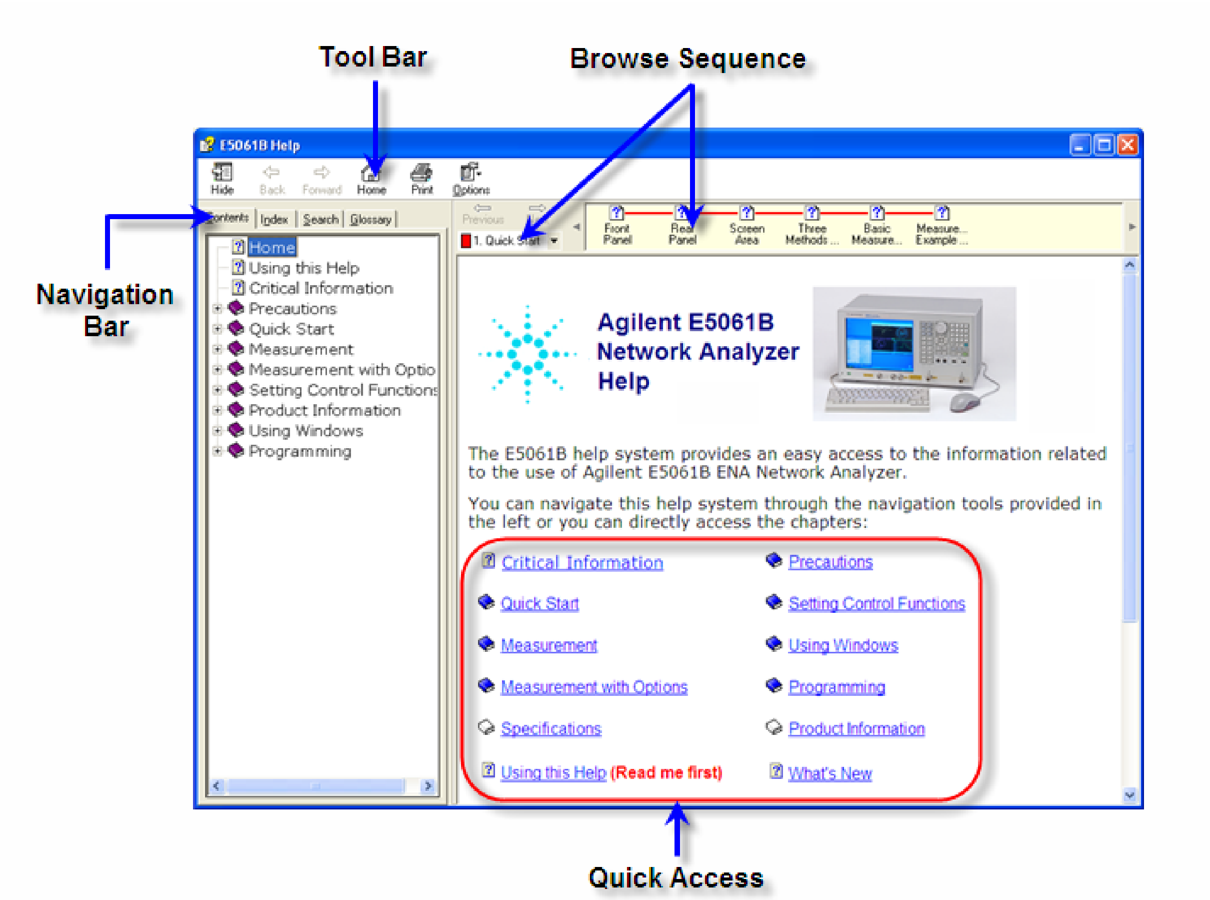
Agilent Technologies

Using this Help

This help provides the user and programming documentation in an searchable electronic format for the E5061B Network Analyzer. This section describes the usage of this help system.

- Opening E5061B Help
- Context Sensitive Help
- Closing E5061B Help
- Viewing E5061B Help on PC
- Switching between E5061B Help and Measurement View
- Navigating E5061B Help
- Conventions used in E5061B Help

E5061B Online Help



Opening E5061B Help

This help system is provided in Microsoft Compressed HTML Help format (.chm). This help can be also be viewed on a normal PC.

To open E5061B help, use either of the following methods:

- By pressing the **Help** key located in the **ENTRY** Block.
- By pressing the **F1** key on a keyboard attached with the E5061B.
- By double-clicking **E5061B_Help.chm** (located in **D:\Agilent\Help**).

Context Sensitive Help

Context sensitive help is a great feature of the E5061B help. It allows you to get information about the selected softkey by pressing the **Help** key in the E5061B or by pressing **F1** in a keyboard attached to the E5061B or by clicking the help button in a dialog box. It provides information relevant to the task that needs to be accomplished and reduces the time to search relevant information required to complete a task.

Closing E5061B Help

To close the E5061B help, click , located on the top right of the E5061B help viewer.

Viewing E5061B Help on your PC

The E5061B help can be opened and viewed on a normal PC. The help file (**E5061B_Help.chm**) is located in the E5061B Network Analyzer hard disk at **D:\Agilent\Help**. Copy it to the local hard disk drive on your PC, then double-click it to view. You can download the latest help file from the Agilent web site.

Switching between E5061B Help and Measurement View

The **Foc** Key, located in the **Entry** Block, can be used to switch between the E5061B Measurement View and Help View.

Navigating E5061B Help

The E5061B help system provides several ways to navigate through the information related to the use of E5061B Network Analyzer. These sections describe the navigation system of the E5061B help which consists of:

- Navigation bar

- Quick Access
- Toolbar
- Browse Sequence

Navigation bar

Navigation bar comprises of tabs related to **Contents**, **Index**, **Search** and **Glossary**. The **Contents** tab contains the main navigational structure of the E5061B help. The **Glossary** tab contains explanation of the terms significant to the E5061B Network Analyzer. The **Index** tab is an additional tool that can be used to navigate different topics according to their alphabetical listing. The **Search** tab can be used to search any term/phrase used in the E5061B online Help.

Toolbar

The toolbar can be used to navigate through the help. The **Home** option can be used to return to the home page of the E5061B help which contains Quick Access to chapters in the help. The **Back** and **Forward** options can be used to toggle between visited topics. The **Print** option can be used to print the selected, or all the topics of the E5061B help.

Browse Sequence

The E5061B help contains pre-defined browse sequence to ease the navigation and to save time required to perform some commonly used procedures. It also enables users to toggle between steps of the browse sequence.

Quick Access

Quick access, located on the **Home Page** of the E5061B help, is a convenient and quick way to access the contents of the E5061B help.

Conventions used in E5061B Help

Naming Conventions Used in this Help

Convention	Description	Example
File Name/Path	File names and path associated with them as displayed as **Bold** in Arial with 12 pt size.	**C:\Documents and Settings\test.txt** **example2.xls**

Hard keys	Hard Keys (Keys located on the Front panel of E5061B) are displayed in Blue color, **Bold** in Verdana with 12 pt size.	**Trace Max** **Foc** **Marker**
Softkeys	Also known as menu keys, are the names of menu that appear in the Firmware (Software) of E5061B and are displayed as **Bold** in Arial with 12 pt size.	**Auto Scale All** **S22**
BLOCK Names	E5061B Front Panel is divided into 7 blocks. These blocks are displayed in ALL CAPS, **Green Color**, **Bold** in Arial with 12 pt size, followed by the word 'Block'.	**RESPONSE** Block **ENTRY** Block **STIMULUS** Block
Note	This Note sign denotes important information. It calls attention to a condition that is essential for the user to understand.	NOTE This is a Note.
Caution	This Caution sign denotes a hazard. It calls attention to a procedure, practice, or condition that, if not correctly performed or adhered to, could result in damage to or destruction of part or all of the instrument.	CAUTION This is a Caution.
Warning	This warning sign denotes a hazard. It calls attention to a procedure, practice, or condition that, if not correctly performed or adhered to, could result in injury or death to personnel.	WARNING This is a Warning.

Critical Information

Do not apply the exceeded DC on the Input Ports

Do not apply the DC voltage or current to the test port except the following conditions.

- Port 1 and 2: AC Coupling
- Port R and T (Option 3L5): 1 MΩ Input impedance

Applying DC voltage or current may lead to device failure. In particular, the capacitor might remain charged. Connect the measurement sample (DUT) to the test port (or the test fixture, cables, etc. connected to the test port) after the analyzer has been completely discharged.

The maximum DC limit of Port1, Port 2, R and T ports is 7V for 50 Ω input impedance, 42V for 1 MΩ input impedance.

When the exceeded DC is detected, the Overload Detection function is activated to protect the analyzer input circuit.

Using Probe with Connector Pin (Option 3L5)

When you use the probe which has a pin on the connector, place the plastic ring (E5061-25006) on the ports T and R in order to avoid connector from damage. E5061-25006 are furnished with the E5061B option 3L5 at shipment.

Precautions

Precautions

- Safety
- Notices
- Installing Software
- Protecting the E5061B
- Before Contacting us

Safety

Safety Summary

The following general safety precautions must be observed during all phases of operation, service, and repair of this instrument. Failure to comply with these precautions or with specific WARNINGS elsewhere in this manual may impair the protection provided by the equipment. Such noncompliance would also violate safety standards of design, manufacture, and intended use of the instrument. Agilent Technologies assumes no liability for the customer's failure to comply with these precautions.

NOTE The E5061B complies with INSTALLATION CATEGORY II as well as POLLUTION DEGREE 2 in IEC61010-1. The E5061B is an INDOOR USE product.

NOTE The LEDs in the E5061B are Class 1 in accordance with IEC60825-1, CLASS 1 LED PRODUCT

- Ground the Instrument

  To avoid electric shock, the instrument chassis and cabinet must be grounded with the supplied power cable's grounding prong.

- DO NOT Operate in an Explosive Atmosphere

  Do not operate the instrument in the presence of inflammable gasses or fumes. Operation of any electrical instrument in such an environment clearly constitutes a safety hazard.

- Keep Away from Live Circuits

  Operators must not remove instrument covers. Component replacement and internal adjustments must be made by qualified maintenance personnel. Do not replace components with the power cable connected. Under certain conditions, dangerous voltage levels may remain even after the power cable has been disconnected. To avoid injuries, always disconnect the power and discharge circuits before touching them.

- DO NOT Service or Adjust the Instrument Alone

  Do not attempt internal service or adjustment unless another person, capable of rendering first aid and resuscitation, is present.

- DO NOT Substitute Parts or Modify the Instrument

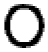
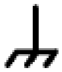
To avoid the danger of introducing additional hazards, do not install substitute parts or perform unauthorized modifications to the instrument. Return the instrument to an Agilent Technologies Sales and Service Office for service and repair to ensure that safety features are maintained in operational condition.

- Dangerous Procedure Warnings

  Warnings, such as the example below, precede potentially dangerous procedures throughout this manual. Instructions contained in the warnings must be followed.

  WARNING Dangerous voltage levels, capable of causing death, are present in this instrument. Use extreme caution when handling, testing, and adjusting this instrument.

Safety Symbols

	Instruction Manual symbol: the product is marked with this symbol when it is necessary for the user to refer to the instrument manual.
	Alternating current.
	Direct current.
	On (Supply).
	Off (Supply).
	A chassis terminal; a connection to the instrument's chassis, which includes all exposed metal structure.
	Stand-by.

Notices

The information contained in this document is subject to change without notice.

This document contains proprietary information that is protected by copyright. All rights are reserved. No part of this document may be photocopied, reproduced, or translated to another language without the prior written consent of Agilent Technologies.

Microsoft®,MS-DOS®,Windows®,Visual C++®,Visual Basic®,VBA® and Excel® are registered trademarks of Microsoft Corporation.

Java® is registered trademark of Sun Microsystems Corporation.

Certification

Agilent Technologies certifies that this product met its published specifications at the time of shipment from the factory. Agilent Technologies further certifies that its calibration measurements are traceable to the United States National Institute of Standards and Technology, to the extent allowed by the Institution's calibration facility or by the calibration facilities of other International Standards Organization members.

Documentation 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, Agilent 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. Agilent 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 Agilent 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.

Exclusive Remedies

The remedies provided herein are Buyer's sole and exclusive remedies. Agilent Technologies shall not be liable for any direct, indirect, special, incidental, or consequential damages, whether based on contract, tort, or any other legal theory.

Assistance

Product maintenance agreements and other customer assistance agreements are available for Agilent Technologies products.

For any assistance, contact your nearest Agilent Technologies Sales and Service Office.

Sample Program

The customer shall have the personal, non-transferable rights to use, copy, or modify SAMPLE PROGRAMS in this manual for the customer's internal operations. The customer shall use the SAMPLE PROGRAMS solely and exclusively for their own purposes and shall not license, lease, market, or distribute the SAMPLE PROGRAMS or modification of any part thereof.

Agilent Technologies shall not be liable for the quality, performance, or behavior of the SAMPLE PROGRAMS. Agilent Technologies especially disclaims any responsibility for the operation of the SAMPLE PROGRAMS to be uninterrupted or error-free. The SAMPLE PROGRAMS are provided AS IS.

AGILENT TECHNOLOGIES DISCLAIMS ANY IMPLIED WARRANTY OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE.

Agilent Technologies shall not be liable for any infringement of any patent, trademark, copyright, or other proprietary right by the SAMPLE PROGRAMS or their use. Agilent Technologies does not warrant that the SAMPLE PROGRAMS are free from infringements of such rights of third parties. However, Agilent Technologies will not knowingly infringe or deliver software that infringes the patent, trademark, copyright, or other proprietary right of a third party.

Useful Sample VBA Library for ENA Series Network Analyzers are available at http://www.agilent.com/find/ena_support.

VBA Macro

The customer shall have the personal, non-transferable rights to use, copy, or modify the VBA macros for the customer's internal operations.

The customer shall use the VBA macros solely and exclusively for their own purposes and shall not license, lease, market, or distribute the VBA macros or modification of any part thereof.

Agilent Technologies shall not be liable for any infringement of any patent, trademark, copyright, or other proprietary right by the VBA macros or their use. Agilent Technologies does not warrant that the VBA macros are free from infringements of such rights of third parties. However, Agilent Technologies will not knowingly infringe or deliver software that infringes the patent, trademark, copyright, or other proprietary right of a third party.

Installing Software

Updating Pre-Installed Software

Do not update pre-installed software in except when Agilent recommends to do so. Before updating or installing software refer to Windows Support Information in http://www.agilent.com/find/ena_support for more information.

Installing User Application Software

Users can install commercial application software for Windows on the E5061B at their own risk. Some application software may affect the measurement performance, especially measurement speed.

Protecting the E5061B

To protect your E5061B, follow the instructions below:

Read the warning labels and specifications

Do not exceed the values provided in the specifications guide or as indicated by the yellow warning labels on the front panel of the E5061B. Refer to the specifications for the conditions required to meet the listed specifications. There will be information regarding E5061B settings, and calibration requirements.

Do NOT Plug off Power Cable during Shutdown Process

Do Not Plug off Power cable until completing the shutdown process.

If you directly interrupt the power supply to the power cable receptacle when the power supply is on, or while turning off the Line Switch (Always ON), the shutdown process will not work. This could damage the HDD of the E5061B.

Do NOT Modify or Reconfigure the Operating System

The Microsoft Windows operating system has been modified and optimized by Agilent to improve the performance of the E5061.

- Do NOT install standard version of the Windows operating system on the E5061B.
- Do NOT change advanced performance settings or group policies.
- Do NOT add or delete any hard disk drive partitions on the E5061B.
- Do NOT delete the Agilent user account.
- Do NOT modify any of the Agilent software registry entries.
- Do NOT change the settings of Standards and Formats in Regional Options and Languages from default setting (English).

Install Antivirus Protection

The E5061B does NOT have antivirus protection when shipped. Use of an antivirus program is strongly recommended if you connect the E5061B to the LAN (Internet).

In addition, the use of a firewall could help to protect the E5061B from viruses. However, some firewalls could limit DCOM connectivity of the E5061B.

Install Windows Critical Updates

The E5061B is always shipped with the latest service packs and critical updates that were available at the time when firmware is updated. Agilent recommends you to maintain the latest available protection for your E5061B by automatically accepting and installing the latest critical security patches from the Microsoft Windows Update website: http://windowsupdate.microsoft.com

Run Error Check and Disk Defragmenter

When the E5061B is shutdown unexpectedly or power is removed without properly shutting down, large amounts of Hard Disk Drive space is rendered unusable. If shutdown is done in this manner a number of times, the E5061B could become unstable and no longer work.

This HDD space can be recovered by first running Windows Error-checking to find and correct errors on the disk, and then the Disk Defragmenter to recover Hard Disk Drive space. See the Windows Help file for more information about Disk Defragmenter.

To run error check and defragmenter follow the procedures below:

1. Click **Windows Start Menu** > **My Computer**.
2. Select the drive you want to perform defragmentation or Error-checking.
3. Click **File**, then **Properties**.
4. Click the **Tools** tab.

Error-checking

1. Click **Check Now**.
2. Check **Automatically fix file system errors**.
3. Click **Start**.
4. Click **Yes** to run disk check on next restart.
5. Manually restart the E5061B. The disk check will run before Windows restarts.

Periodically, check the second box in addition to the first box. The error-checking process takes longer time, but performs a complete check.

Defragmentation

1. Click **Defragment Now...**
2. Click **Defragment** to begin the defragment process.
3. Click **Close** when defragmentation is complete.

Precaution for Use of Hard Disk

Do NOT modify or delete any Files and Folders in the drives other than D drive. Doing so will result in malfunctioning of the device. It is required to execute the System recovery if there is any trouble with the above operations.

Precaution for input connector and cable

- Do not apply excessive DC voltage or current to the test port. Applying excessive DC voltage or current may lead to device failure. In particular, the capacitor might remain charged. Connect the measurement sample (DUT) to the test port (or the test fixture, cables, etc. connected to the test port) after the analyzer has been completely discharged. The damage level is described in the data sheet.
- Do NOT bend, bump or flex any device under test (DUT) connected to the input of the E5061B (such as filters, couplers etc). This will reduce the amount of strain placed on the input connector and the mounting hardware. Make sure externally connected items are properly supported (not freely suspended) from the input.
- Do NOT bend cables repeatedly, as this may damage the cables instantly. Limit the number of connections and disconnections to reduce wear and tear. Inspect connectors prior to use; look for dirt, nicks, and other signs of damage or wear. A bad connector can ruin a good connector instantly. Clean dirty connectors to prevent poor electrical connections and damage to the connector. For more information on cable and connector care, refer to http://www.agilent.com/find/cable_care

Precautions for Electrostatic Discharge (ESD)

ESD can damage or destroy electronic components. Whenever possible, conduct testing at a static-safe workstation. Keep static-generating materials at least one meter away from all components. Before connecting any coaxial cable to an analyzer, momentarily short the center and outer conductors of the cable together.

Maintain working environment condition

Control your environment. Maintain temperature & humidity with a satisfactory range within the instruments specification and prevent large fluctuations.

Precautions for transportation

- Do NOT lift the instrument with your hands at the front panel. If the instrument slips, damage may occur to the keypad, knob, or input connectors. Lift the Instrument by the handles when transporting.
- Do NOT use styrene pellets as packaging materials as these may cause damage to E5061B by generating static electricity.

Check for Proper Ventilation

Periodically check and clean the cooling vents of the E5061B. Inadequate airflow can result in excessive operating temperatures which can lead to instrument failures. When installing the product in a cabinet, the convection in and out of the instrument must not be restricted.

See the installation guide for details.

Precautions for Proper Grounding

- Proper grounding prevents building-up of static charge which may be harmful to the E5061B.
- Do NOT defeat the earth-grounding protection by using an extension cable, power cable, or autotransformer without a protective ground conductor.

Before Contacting us

If you encounter the following problems during startup or operation of the E5061B, in which initial registration of the Windows Operating System has been properly performed, execute system recovery and update the Firmware revision.

The system starts up, but the normal measurement screen does not appear

- The system automatically shuts down immediately after the startup, or the startup process stops.
- The measurement screen appears, but "Power on test fail" or "Calibration data lost" is displayed in the instrument message/warning area against a red background in the lower-left part of the screen. The system enters the service mode. (The instrument status bar in the lower-right displays SVC in red).

Unstable Operation

- The system hangs while the instrument is controlled from VBA or external PCs.
- The blue screen appears and the system hangs.
- The response is much slower than usual.

When execution of system recovery does not result in normal operation, it indicates that failure may have occurred. Contact Agilent customer contacts.

For other problems, refer to Troubleshooting.

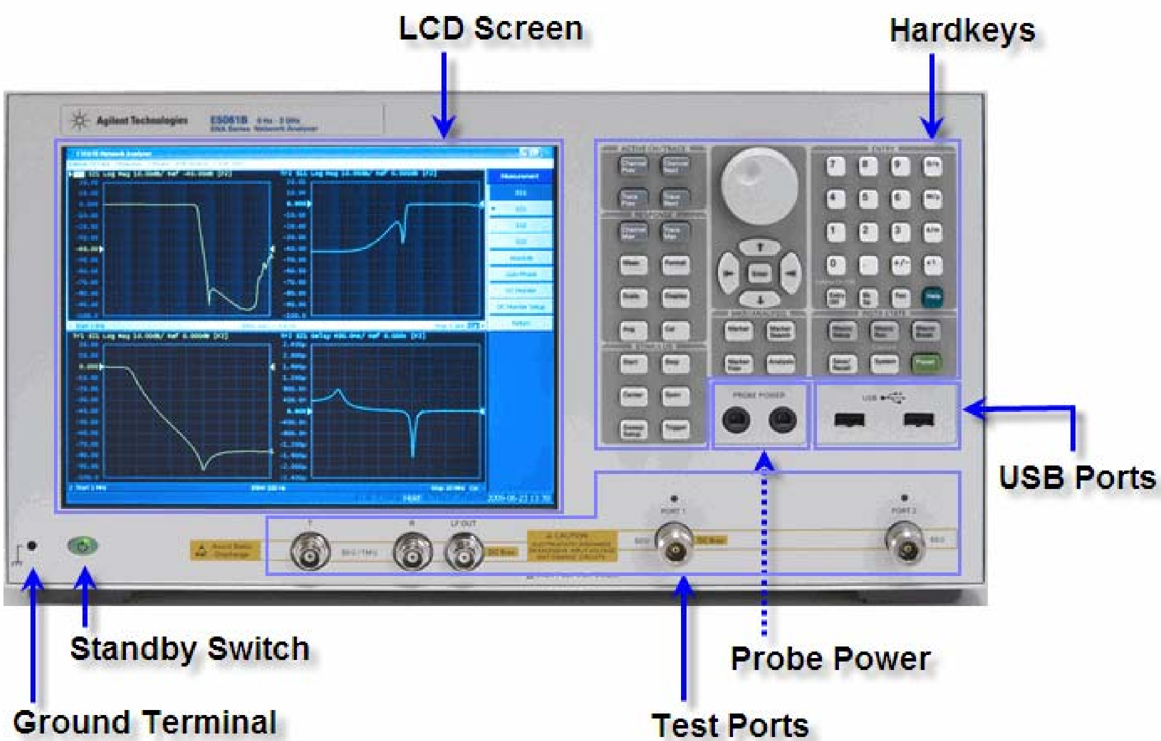
Quick Start

Quick Start helps you to understand the E5061B operation quickly.

- Overview
  - Front Panel
  - Rear Panel
  - Screen Area
- Three Methods of Operation
- S-Parameter Measurement
  - Basic Measurement Procedures
  - Measurement Example of a Bandpass Filter
- Impedance Measurement
  - Basic Measurement Procedure
  - Measurement Method
  - Measurement Example of a Capacitor (Port 1 Reflection)
  - Measurement Example of Crystal (Port 1-2 Series)
  - Measurement Example of a Capacitor (Port 1-2 Shunt)
  - Measurement Example of a Ceramic Resonator (Gain-Phase/Series-Through)
  - Measurement Example of a Capacitor (Gain-Phase/Shunt-Through)

Overview

Front Panel: Names and Functions of Parts



Ground Terminal

Ground terminal is provided with the E5061B and is connected to the chassis of the E5061B. You can connect a banana-type plug to this terminal for grounding.

Hardkeys

ACTIVE CH/TRACE Block

A group of keys for selecting active channels and traces. For more on the concepts of channels and traces, see Setting Channels and Traces.

Hardkey Name	Description
Channel Next	Selects the next channel as the active channel. (Each time the key is pressed, the active channel steps up from the current channel to the channel with one number larger than the current channel number). A channel must be active before you can define such parameters as the sweep range. To change the

	settings of a channel, use this key to first make the channel active.
Channel Prev	Selects the previous channel as the active channel. (Each time the key is pressed,the active channel steps down from the current channel to the channel with one number smaller than the current channel number).
Trace Next	Selects the next trace as the active trace. (Each time the key is pressed, the active trace steps up from the current trace to the trace with one number larger than the current trace number). A trace must be active before you can define measurement parameters and other settings. To change the settings of a trace, use this key to first make the trace active.
Trace Prev	Selects the previous trace as the active trace. (Each time the key is pressed, active trace steps down from the current trace to the trace with one number smaller than the current trace number).

ENTRY Block

A group of keys used for entering numeric data is provided on the front panel of the E5061B.

Hardkey Name	Description
0, 1, 2, 3 ..... 9 , . Keys (numeric keys)	Type numeric characters or a decimal point at the position of the cursor in the data entry area.
+/-	**+/-** alternately changes the sign (+, -) of a numeric value in the data entry area.
G/n, M/u, k/m, x1	Adds a prefix to the numeric data typed by using the numeric key and **+/-** key. One of the two prefixes written on the surface of the key is automatically selected depending on the parameter to be entered. **x1** is entered without a prefix.
Softkey On/Off , Entry Off	Turns off the data entry bar if it is displayed. If the dialog box is displayed, it cancels the entry and closes the dialog box. If the data entry bar and dialog box are not displayed, it turns the softkey menu display on/off.
Bk Sp	Backspace key.
Foc	Changes the selection (focus) among the objects to be

	manipulated by the **NAVIGATION** Block keys and **ENTRY** Block keys. The objects to be manipulated by the **NAVIGATION** Block keys and **ENTRY** Block keys include softkey menus, data entry areas, tables (e.g., segment tables, limit tables, and marker tables), and dialog boxes. When two or more of these objects are displayed on the screen and need selecting, use this key to change the selection (focus) among the objects to be manipulated. When a softkey menu is selected, the menu name area at the top of the menu is displayed in blue. When a data entry area is selected, the data entry bar is displayed in blue. When a table is selected, the frame of the table window is displayed in light gray. While a dialog box is displayed, the focus is fixed on the dialog box and cannot be changed.
Help	Displays help for E5061B.

INSTR STATE Block

A group of keys related to the macro function, store and call function, control/management function, and presetting of the E5061B (returning it to the preset state).

Hardkey Name	Description
Macro Setup	Displays the Macro Setup Menu in Softkey Menu Bar. Manipulating the Macro Setup Menu enables you to start up the VBA editor or to create, call, or store a VBA project.
Macro Run	Executes a VBA procedure called "main" that has a VBA module named Module1.
Macro Break	Stops the VBA procedure being executed.
Save/Recall	Displays the Save/Recall Menu in Softkey Menu Bar. Manipulating the Save/Recall Menu enables you to store the setup conditions to or read from the storage devices, calibration data, and trace data of the analyzer.
Capture/System	First, temporarily saves the data for the image displayed on the LCD screen the moment this key is pressed to the internal memory (clipboard). Immediately after that, displays the System Menu in Softkey Menu Bar. Manipulating the System Menu enables you to define the setup for the limit test and then execute it, or to define the setup for the control and management of the analyzer. Using the **Dump Screen Image** option enables you to store the image data in the clipboard to a file on the storage devices. Also, using the **Print** option in the System menu enables you to print the image data in the clipboard to a

	printer.
Preset	Displays the Preset Menu in Softkey Menu Bar. Clicking **OK** in the Preset Menu enables you to return the analyzer to the initial setup state, called the preset setup.

MKR/ANALYSIS Block

A group of keys used for analyzing the measurement results by using the markers and etc.

Hardkey Name	Description
Marker	Displays the Marker Menu in Softkey Menu Bar. Manipulating the Marker Menu enables you to turn the markers on/off and move them by entering stimulus values. You can place up to 10 markers on each trace.
Marker Search	Displays the Marker Search Menu in Softkey Menu Bar. Manipulating the Marker Search Menu enables you to move a marker to a specific point (maximum, minimum, peak, and a point with a target value) on a trace. You can also find the bandwidth parameters (up to six) and display them.
Marker Fctn	Displays the Marker Function Menu in Softkey Menu Bar. Manipulating the Marker Function Menu enables you to not only specify the marker sweep range and the coupling of markers on a channel but also to display statistics data on traces.
Analysis	Displays the Analysis Menu in Softkey Menu Bar. Manipulating the Analysis Menu enables you to use the analytical function of the fault location, the SRL, and each limit test.

NAVIGATION Block (No Label on Front Panel)

The keys and Rotary knob in the **NAVIGATION** Block are used to navigate between softkey menus, tables (limit table, segment table, etc.), or selected (highlighted) areas in a dialog box as well as to change a numeric value in the data entry area by stepping up or down. When selecting one of two or more objects (softkey menus, data entry areas, etc.) to manipulate with the **NAVIGATION** Block keys displayed on the screen, first press the **Foc** (Focus) key in the ENTRY Block to select the object to be manipulated (placing focus on the object) and then manipulate the **NAVIGATION** Block keys (knob) to move among selected (highlighted) objects or change numeric values.

The following descriptions show how the **NAVIGATION** Block keys work both when the focus is on a softkey menu and when the focus is on the data entry area. For more on tables and dialog boxes manipulation, refer to the manipulation procedure for each of these functions.

- **When the focus is on a softkey menu (softkey menu is selected)**
- **When the focus is on the data entry area (data entry area is selected)**

RESPONSE Block

A group of keys used mainly for setting up response measurements on the E5061B.

Hardkey Name	Description
Channel Max	Changes between normal and maximum display of the active channel window. In normal display, all of the defined channel windows (both active and non-active) are displayed in split views on the screen. In maximum display, only the active channel window is displayed over the entire area, with non-active windows not displayed. To maximize the active channel, double-click the channel window frame. Measurements are also carried out on the non-active channels that are not displayed.
Trace Max	Changes between normal and maximum display of the active trace window. In normal display, all of the defined trace windows (both active and non-active) are displayed in split views on the screen. In maximum display, only the active trace is displayed over the entire channel window, with non-active traces not displayed. To maximize the trace, double-click anywhere in the channel window. Measurements are also carried out on the non-active traces that are not displayed.
Meas	Displays the Measurement Menu in Softkey Menu Bar. Manipulating the Measurement Menu enables you to specify the measurement parameters (types of S-parameters) for each trace.
Format	Displays the Format Menu in Softkey Menu Bar. Manipulating the Format Menu enables you to specify the data format (data transformation and graph formats) for each trace.
Scale	Displays the Scale Menu in Softkey Menu Bar. Manipulating the Scale Menu enables you to specify the scale for displaying a trace (magnitude per division, value of the reference line, etc.) for each trace. You can also specify the electrical delay and phase offset for each trace.
Display	Displays the Display Menu in Softkey Menu Bar. Manipulating the Display Menu enables you to specify the number of channels and the channel window array, the number and arrangement of

	traces, the setup for data math, etc.
Avg	Displays the Average Menu in Softkey Menu Bar. Manipulating the Average Menu enables you to define the averaging, smoothing, and IF bandwidth.
Cal	Displays the Calibration Menu in Softkey Menu Bar. Manipulating the Calibration Menu enables you to turn the calibration and error correction on/off and change definitions for calibration kits.

STIMULUS Block

A group of keys for defining the stimulus values (signal sources and triggers).

Hardkey Name	Description
Start	Displays the data entry bar for specifying the start value of the sweep range in the upper part of the screen. (It also displays the Stimulus Menu for specifying the sweep range in Softkey Menu Bar.)
Stop	Displays the data entry bar for specifying the stop value of the sweep range in the upper part of the screen. (It also displays the Stimulus Menu in the same way as **Start**.)
Center	Displays the data entry bar for specifying the center value of the sweep range in the upper part of the screen. (It also displays the Stimulus Menu in the same way as **Start**.)
Span	Displays the data entry bar for specifying the span value of the sweep range in the upper part of the screen. (It also displays the Stimulus Menu in the same way as **Start**.)
Sweep Setup	Displays the Sweep Setup Menu in Softkey Menu Bar. Manipulating the Sweep Setup Menu enables you to specify the signal source power level, sweep time, number of points, sweep type, etc.
Trigger	Displays the Trigger Menu in Softkey Menu Bar. Manipulating the Trigger Menu enables you to specify the trigger mode and trigger source. You can specify the trigger mode for each channel.

LCD Screen

The E5061B is equipped with a 10.4-inch TFT color, touch-sensitive LCD screen for displaying traces, scales, settings, softkeys and other measurement related information. The touch screen LCD allows you to manipulate softkeys by touching the LCD screen directly with a finger. For more on the LCD screen, see Screen Area: Names and Functions of Parts.

CAUTION Do not press the surface of the LCD screen with a sharp object (e.g., nail, pen, or screwdriver). Pressing the surface with a sharp-pointed object will damage the LCD screen surface or cause the screen to fail.

NOTE Valid pixels are 99.998 % and more. Below 0.002 % of fixed points of black, blue, green or red are not regarded as failure.

Probe Power

The E5061B Option 3L5 comes with two ports that can be used to provide power to external probes. See the Data sheet for the voltage and maximum current.

NOTE Other E5061B options do not have probe power ports.

Standby Switch

This switch can turn on/off the E5061B. The color on the button shows the status as shown below:

Indicator Color	Description
Green	Normal power on status.
Orange	Standby status.
Red	Illegal power on status.

To turn off the power of the E5061B, be sure to follow the steps described below:

1. First, press this standby switch or send a shutdown command from the external controller to activate the shutdown process (the processing of software and hardware necessary to turn off the power supply). This will put the E5061B into the standby state.
2. Next, if necessary, turn off power supply to the Power Cable Receptacle (to LINE) on the rear panel.

   CAUTION Under normal use, never directly interrupt the power supply to the power cable receptacle on the rear panel when the power supply is on. Always keep the Line Switch (Always ON) at (|). Never turn it off (O).

   If you directly interrupt the power supply to the power cable receptacle when the power supply is on, or turn off the Line Switch (Always ON), the shutdown process will not work. This could damage the software and hardware of the E5061B and lead to device failure.

Turning on the power supply after a faulty shutdown may cause the system to start up in a condition called "safe mode." If this occurs, first shut down the system to set it to the standby state and then turn on the power supply again to start up the system in normal mode.

Test Ports

Port 1 and Port 2

Port 1 and Port 2 are for S-parameter measurement. Port 1 can apply DC Bias. While signals are being output from a test port, the yellow LED above the test port is lit.

LF Out, R and T Ports

LF Out, R and T Ports are for Gain-Phase measurement. LF Out is a source port and it can apply DC Bias on measurement signal. The R and T ports are receiver ports. The input impedance for R and T ports can be selected from 50 Ω or 1 MΩ.

LF Out can be used for DC voltage source during S parameter measurement.

NOTE Only E5061B Option 3L5 has LF Out, R and T Ports. Other E5061B options do not have these ports.

CAUTION Do not apply the exceeded DC voltage or current to the test port.

CAUTION When you use the probe which has a pin on the connector, place the plastic ring to avoid connector from damage.

The test ports comply with Installation Category I of IEC 61010-1.

USB Ports

Two USB (Universal Serial Bus) Ports are provided that can be used for connecting to ECal (Electronic Calibration) module, USB keyboard, USB mouse, USB memory or a printer. Connecting a designated ECal module to this port enables ECal measurements to be taken. Connecting a compatible printer to this port enables screen information on the E5061B to be printed. See Using USB for more detail.

Rear Panel: Names and Functions of Parts

24 Bit (Handler) I/O Port

The terminal to which an automatic machine (handler) used on a production line is connected. See 24 Bit (Handler) I/O Port.

Connector type: 36-pin Ribbon (Centronics) connector

Certificate of Authenticity Label

The label shows the licence information of the Windows Operating System

Ethernet Port

A terminal for connecting the E5061B to a LAN (Local Area Network). Connecting this instrument to a LAN enables you to access the hard disk drive of this instrument from an external PC or to control this instrument by using SICL-LAN or telnet.

Specification	Value
Connector type	8-pin RJ-45 connector
Base standard	10Base-T/100Base-TX/1000Base-T

External Monitor Output Port (VIDEO)

A terminal to which an external color monitor (display device) can be connected. By connecting a color monitor to this terminal, the same information shown on the LCD screen of the main body can be displayed on an external color monitor.

Connector type: 15-pin VGA connector, female

External Trigger (EXT TRIG)

Input

A connector to which external trigger signals are input. This connector detects the downward transition from the HIGH state in TTL signals as the trigger signal. To use this connector to generate a trigger, you must set the trigger source to the "external" side. The connector type is BNC, female.

Output

The External Trigger Output Port can output the pulse with the specified polarity and position either before or after the measurement of each point. The connector type is BNC, female.

Parameter	Typical Value	Condition
HIGH Level Output Voltage	5 V	I_{out}=-50μA

LOW Level Output Voltage	0 V	I_{out}=50μA

NOTE The External Trigger Output Port can safely handle a maximum output current of 50 mA.

GPIB Port

The connection of an external controller through General Purpose Interface Bus (GPIB) connector allows you to configure an automatic measurement system.

This GPIB port is used only for controlling the E5061B from an external controller. Use USB/GPIB interface to control other devices from the E5061B. You cannot control other devices from the E5061B through this GPIB port.

Hard Disk Drive (HDD)

Built in hard disk drive.

Reference Signal Ports (10MHz)

The specification for the signals is specified in the data sheet.

Input (REF IN)

The reference signal input connector is used for phase-locking the measurement signal from the E5061B to the external frequency reference signal. Inputting reference signal to this connector improves the accuracy and frequency stability of the measurement signal from the E5061B. When the frequency reference signal is input to this connector, the measurement signal from the E5061B is automatically phase-locked to the reference signal. When an input signal is not present, the frequency reference signal inside the E5061B is automatically used. The ExtRef on the instrument status bar is displayed in blue when the system is phase-locked to the external reference signal and in gray when not phase-locked.

When using Option 1E5 (high stability time base), connect this connector to the REF OVEN by using the BNC(m)-BNC(m) cable included with the option.

Output (REF OUT)

A connector for outputting the internal frequency reference signal from the E5061B. By connecting this output connector to the external reference signal input connector of another device, the device can be phase-locked to the internal reference signal of the E5061B and used under this condition.

High Stability Frequency Reference Output (REF OVEN, OPT 1E5)

When Option 1E5 (high stability time base) is installed, the reference signal is output from this connector. Connect this connector to the REF IN by using the BNC(m)-BNC(m) adapter included with the option.

Line Switch (Always ON)

Always keep this switch on (|).

CAUTION Do not use this switch to turn off (O) the mains. Doing so may cause the analyzer to fail. For more information, see the description of the Standby Switch.

Power Cable Receptacle (to LINE)

The receptacle (outlet) to which the power cable is connected.

NOTE To connect the device to a power source (outlet), use the supplied three-prong power cable with a ground conductor. The plug attached to the power cable (on the power outlet side or device side of the cable) serves as the disconnecting device (device that cuts off power supply) of the E5061B. When the power supply must be cut off to avoid such danger as electric shock, pull out the power cable plug (on the power outlet side or device side of the cable). For the procedure for turning off the main in normal use, see the description in Standby Switch.

For more on the power supply, see the Installation Guide.

Serial Number Plate

The label showing the product number, serial number and the installed option number. The accessary and system rack options are not listed on this label. (CFGxxx or ATOxxx in the first line is Agilent Use Only.)

Test Set I/F Port

This interface is reserved and currently not available.

USB Interface Port (USBTMC)

Through this port, you can control the E5061B from external controllers. For more information on the measurement system using the USB port, see the USB Remote Control System.

Specification	Value
Connector type	Universal serial bus (USB), type B (4 contact positions), Female (jack)
Compliance Standards	USBTMC-USB488 and USB2.0

USB Ports

Four USB (Universal Serial Bus) ports are provided that can be used for connecting to ECal (Electronic Calibration) module, USB keyboard, USB mouse, USB memory or a printer. Connecting a designated ECal module to this port enables ECal measurements to be obtained. Connecting a compatible printer to this port enables screen information on the E5061B to be printed. See Using USB for more detail.

Screen Area

Screen Area: Names and Functions of Parts

Click on the name or area for details of the topic.

- Menu Bar
- Data Entry Bar
- Softkey Menu Bar
- Instrument Status Bar
- Channel Window
- **Windows XP Status Bar** - Windows XP Status Bar may or may not appear, depending on the Windows license. Under the same

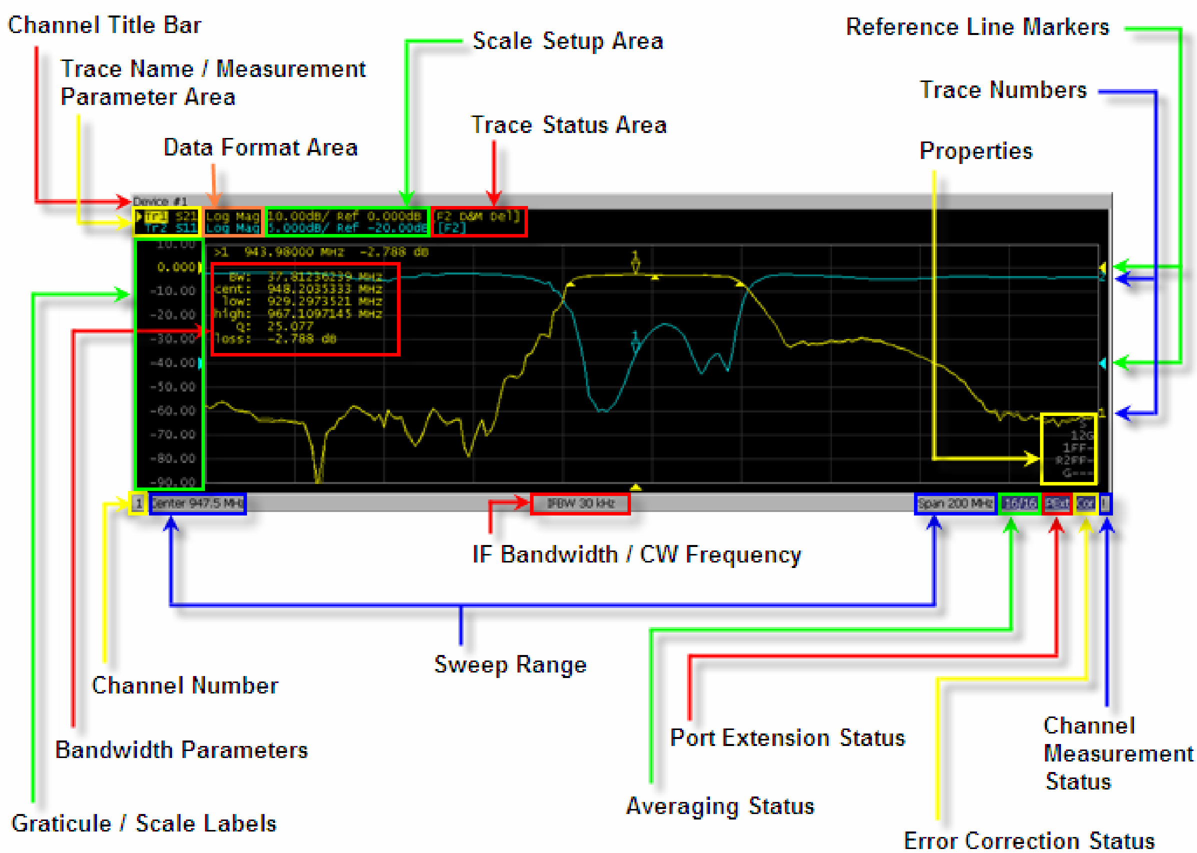
condition, either Windows resize buttons or E5061B resize button will appear. For more information, refer to Windows License.

Channel Window

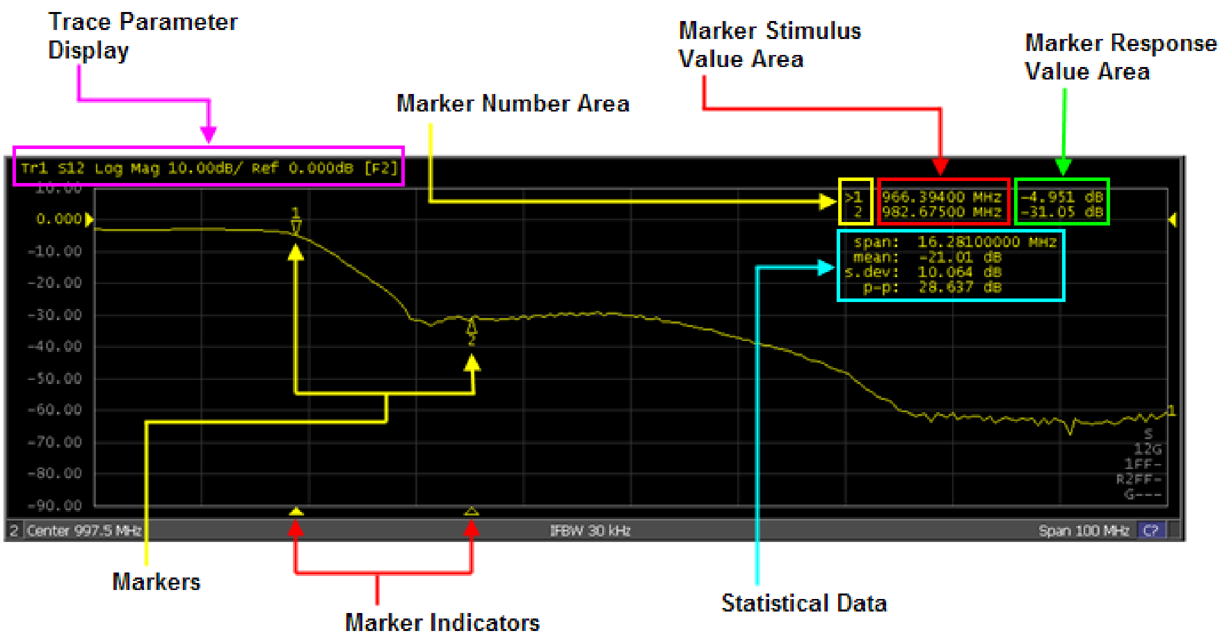
Window for displaying the traces. Because a channel corresponds to a window, it is called a channel window. When the outer frame of a channel window is displayed in light gray, it shows that the channel is an active channel (the channel for which setup is being performed). In the following figure, Channel 1 (the upper window) is the active channel. To make a channel active, use **Channel Next** or **Channel Prev**. Clicking inside a channel window will also make the channel active.

Channel 1 Window and Channel 2 Window describes different measurement parameters available in the channel measurement window. The measurement parameters described in the Channel 1 and 2 Window correspond to the same channel measurement window and are displayed in separate windows for ease of read.

Channel 1 Window



Channel 2 Window



e5061b080

Bandwidth Parameters

Turning on the bandwidth search function displays the bandwidth parameters here.

Channel Number

Indicates the channel number.

Channel Status Bar

The status of each channel is displayed here.

Averaging Status

Displays the averaging factor and averaging count when averaging is turned on.

n/m (displayed in blue)	Averaging: ON (m: averaging factor; n: averaging count)
(not displayed)	Averaging: OFF

Channel Measurement Status

Displays the update status of traces on the channel.

!	Measurement in progress. When the sweep time exceeds 1.5 seconds, ↑ is displayed at the point on the trace.
#	Invalid traces. The measurement conditions have changed,

	but the traces on the channel currently displayed have not been updated to match the new conditions.
(No display)	The measurement has not been executed.

Error Correction Status

Displays the execution status of error correction on the channel.

IF Bandwidth/CW Frequency

Indicates the IF bandwidth when the sweep type is linear/log frequency or the CW frequency when the sweep type is power.

Port Extension Status

Shows whether the port extension and impedance port extension (Option 005 only) or is turned ON or OFF. port extension and impedance port extension can be truned on both at the same time.

PExt (displayed in blue)	Port extension: ON
ZExt (displayed in blue)	Z port extension: ON
(not displayed)	Port extension: OFF

Sweep Range

Indicates the sweep range by using the start/stop or center/span.

Channel Title Bar

You can assign a title to each channel and have the title displayed on the bar.

Data Format

The data format of each trace is displayed here.

See Also: Trace Parameter Display

Graticule Labels

Y-axis divisions in the rectangular display format. When traces in the rectangular display format are overlaid, the Y-axis divisions for the active trace are displayed. The value of the reference line (the division line between ▶ and ◀) is entered numerically by opening the data entry bar using the keys: **Scale** > **Reference Value**. You can change values of the reference line at one-division intervals by placing the mouse pointer in the area of the graticule label (the pointer changes from to), moving the pointer vertically with the left mouse button pressed, and then releasing the button at the desired location.

Marker Indicators

Indicates the positions of markers on the stimulus axis.

▲	Active marker indicator
△	Non-active marker indicator

You can also move a marker to the desired position by placing the mouse pointer on the marker indicator or position of the marker itself (the pointer changes from to), moving the indicator vertically with the left mouse button pressed, and then releasing the button at the desired location.

Marker Numbers

Displayed marker number is listed. For the active marker (the one for which setup and analysis are being performed), > is displayed at the left of the marker number. For the reference marker, △ is displayed instead of the marker number.

Marker Response Values

The marker response value for each marker (the measurement value at the marker point) is displayed here. Two (or three) response values are displayed for data in Smith chart or polar display format.

Markers

The markers used for reading values on a trace. Up to 10 markers can be displayed for each trace.

n ▽	Active marker (the one for which setup and analysis are being performed)
△ n	Non-active marker

Here, "n" denotes a marker number. For the reference marker, however, nothing is displayed at the location of n. Clicking the marker or one of the Marker Indicators makes the marker active.

Marker Stimulus Values

The marker stimulus value for each marker (the frequency/power level at the marker point) is displayed here.

Properties

Displays the status of the obtained calibration coefficients on the channel.

Reference Line Indicators

The indicators that indicate the position of the reference line for the Y-axis scale in the rectangular display format. One indicator is to the right and the other is to the left of the scale (▶and ◀). To enter a numeric value for the position of the reference line, open the data entry bar using the keys: **Scale** > **Reference Position**. You can also move the position of the reference line by placing the mouse pointer on either of the two reference line indicators (the pointer changes from to .), moving the indicator vertically with the left mouse button kept pressed, and then releasing the button at the desired location (i.e., a drag-and-drop operation).

Scale Settings

The scale setting for each trace is displayed here. This example shows that "10.00dB/" corresponds to 10 dB per division. "Ref 0.000dB" shows that the value of the reference line is 0 dB.

See Also: Trace Parameter Display

Statistics Data

Turning on the statistics data function displays statistics data here.

Trace Name/Measurement Parameter

The names of the traces, such as Tr1, on the channel and their measurement parameters are displayed here. ▶ to the right of the trace name indicates the active trace (the trace for which setup is being performed). To make a trace active, use **Trace Next** or **Trace Prev**. Clicking the line where the trace name is placed (the mouse pointer changes from to) also makes a trace active.

See Also: Trace Parameter Display

Trace Number

In the rectangular display format, the trace number is displayed in the same color as the trace at the right end of each trace.

Trace Status Area

The setup for each trace is displayed here.

See Also: Trace Parameter Display

Trace status display

Classification	Contents inside [ ]	Meaning

Error correction	**RO**	Error correction: ON (OPEN (n) response calibration)
	RS	Error correction: ON (SHORT (n) response calibration)
	RT	Error correction: ON (THRU (n) response calibration)
	ER	Error correction: ON (Enhanced response calibration)
	F1	Error correction: ON (1-port calibration)
	F2	Error correction: ON (Full 2-port calibration)
	Zcor	Error correction: ON Impedance Calibration
Fixture Compensation (Opt 005)	**Zcomp**	Fixture Compenation ON
Turning on / off traces	Nothing	Data trace: ON, Memory trace: OFF
	M	Data trace: OFF, Memory trace: ON
	D&M	Data trace: ON, Memory trace: ON
	off	Data trace: OFF, Memory trace: OFF
Performing data math **When a memory trace is ON, see the contents inside ()**	**D+M** (**D+M&M**)	Execution of Data+Mem math
	D- M (**D- M&M**)	Execution of Data- Mem math
	D*M (**D*M&M**)	Execution of Data*Mem math
	D/M (**D/M&M**)	Execution of Data/Mem math
Electrical delay	**Del**	A numeric value other than 0 (zero) is specified as the electrical delay or phase offset.
Smoothing	**Smo**	Smoothing: ON
Gating	**Gat**	Gating: ON
Fault location	**FL(RT)**	Fault location: ON, Reflection type: Round Trip

	FL(OW)	Fault location: ON, Reflection type: One Way
SRL	**SRL(xxxΩ)**	SRL: ON (xxx is average cable impedance value)
Parameter conversion	**Zr**	Conversion: ON (Impedance: Reflection measurement)
	Zt	Conversion: ON (Impedance: Transmission measurement)
	Ztsh	Conversion: ON (Impedance: Transmission-Shunt measurement)
	Yr	Conversion: ON (Admittance: Reflection measurement)
	Yt	Conversion: ON (Admittance: Transmission measurement)
	Ytsh	Conversion: ON (Admittance: Transmission-Shunt measurement)
	1/S	Conversion: ON (Inverse S-parameter)
	Conj	Conversion: ON (Conjugation)
Reference Tracking	**PTrk**	Peak Track
	FTrk	Frequency Track
Equation Editor	**Equ**	Equation Editor: ON

Trace Parameter Display

As mentioned above, measurement parameters are displayed at the upper-left of the trace window. However, there is difference in parameter display for network measurement and impedance measurement.

Network Measurement

When network measurement is selected, for example, S11, the following parameters are displayed at the upper-left of the trace window:

Trace number	Measurement Parameter	Format	Scale/Div	Reference value

Impedance Measurement (Option 005)

When impedance measurement is selected, the measurement parameter and format is replaced with Z-parameter and measurement method. Hence, the following parameters are displayed:

Trace number	Impedance (Z)-Parameter	Method	Scale/Div	Reference value

Y-Axis (Log)

Regardless of the measurement parameter, when Log scale is selected at Y-Axis, the following parameters change.

If Linear scale is selected:

Scale/Div	Reference value

If Log scale is selected:

Top xx.xxx	Bottom xx.xxx

Example

Below are examples of several scenarios.

Network Measurement (with Linear Y-Axis):

Network Measurement (with Log Y-Axis):

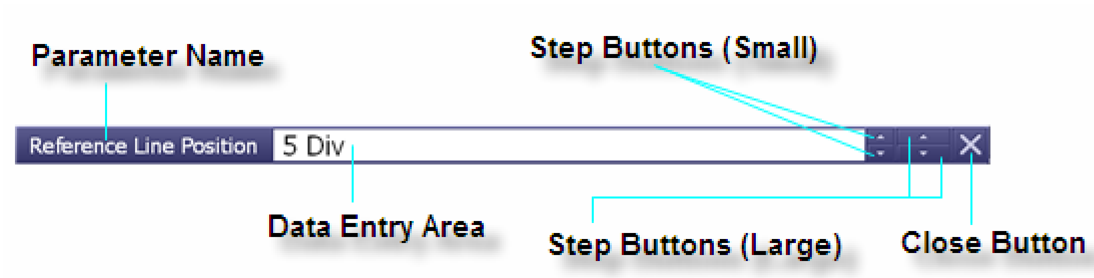
Z Measurement (with Linear Y-Axis):

Z Measurement (with Log Y-Axis):

Data Entry Bar

Used to enter numeric data into the E5061B. Press a hardkey or softkey to enter data, and the data entry bar will appear at the top of the screen. To assign a title to a channel window, an entry bar that allows you to enter letters and symbols by using the front panel keys or mouse is displayed.

Data entry bar



NOTE To manipulate the data entry bar by using the front panel keys, the data entry bar must be selected as the object to manipulate (with the focus placed on it). When the focus is placed on the data entry bar, the entire bar is displayed in blue. Pressing or clicking **Foc** Key in the ENTRY Block enables you to move the focus to the desired object.

Close Button

Closes the data entry area (turns off the display). Use mouse to manipulate this button.

Data Entry Area

When the data entry bar is displayed for the first time, the current settings are displayed on it. You can change numeric values by typing from the keyboard or in the ENTRY block on the front panel.

You can hide the frequency information in order to ensure its confidentiality or for other reasons. For detailed information, see Hiding Softkey's Frequency Information.

Parameter Name

Displays the name of the parameter for which data will be entered.

Step Button (Small)

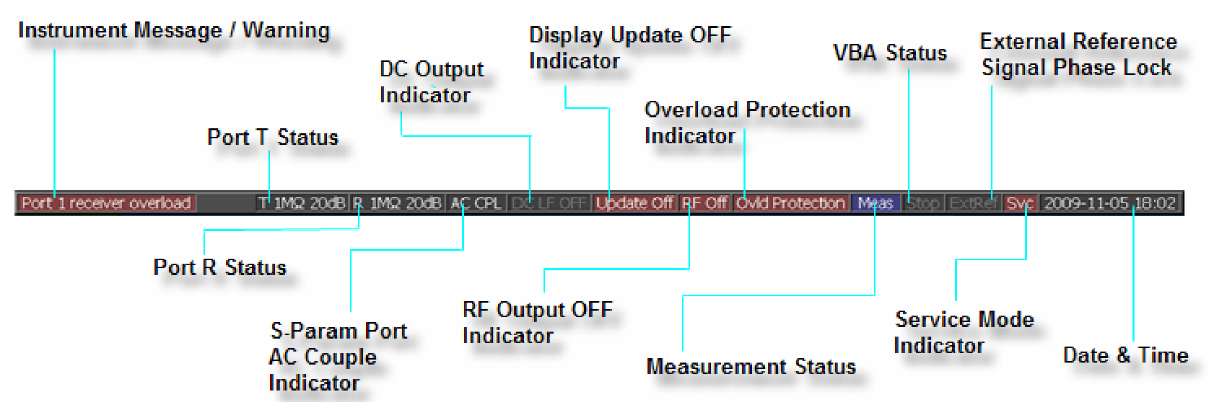
Increases or decreases the numeric value in the data entry area in small steps. Use the mouse to manipulate this button.

Step Button (Large)

Increases or decreases the numeric value in the data entry area in large steps. Use the mouse to manipulate this button.

Instrument Status Bar

The instrument status bar displays the status of the entire instrument.



Date and Time

Displays the date and time generated by the internal clock. The display format is as follows:

YYYY-MM-DD HH:MM

Where:

YYYY: Year (AD)
MM: Month
DD: Day
HH: MM: Time (0:00 to 23:59)

NOTE You can turn the date and time display on/off by: **System** > **Misc Setup** > **Clock Setup** > **Show Clock.**

Display Update OFF Indicator

When information update display on the LCD screen is turned off, this indicator is displayed.

DC Output Indicator

Displays the output port (LF or P1[Port 1]) and status (ON or OFF) for DC output.

External Reference Signal Phase Lock

When the frequency reference signal is input to the Reference Signal Input (REF IN) on the rear panel and the measurement signal of the E5061B is phase-locked to the reference signal, ExtRef is displayed in blue.

Value	Description
ExtRef (displayed in blue)	Measurement signal is phase-locked to the external reference signal.
ExtRef (displayed in	Measurement signal is not phase-locked to the

gray)	external reference signal.

NOTE Even when the High Stability Frequency Reference Output (REF OVEN) and Reference Signal Input (REF IN) are connected, phase-locking may not occur immediately after power-on in a low-temperature environment. (The "ExtRef" display remains gray, not blue.) In such a case, wait a few minutes until the instrument has warmed up and the "ExtRef" display turns blue.

Instrument Message/Warning

Displays instrument messages and warnings. Instrument messages are displayed in gray and warnings in red.

Measurement Status

Displays the measurement status of the E5061B.

Value	Description
Setup	Setup for measurement in progress
Hold	Measurement on hold (idling)
Init	Measurement being initialized
Man	The trigger source is set to "Manual" and waiting for trigger.
Ext	The trigger source is set to "External" and waiting for trigger.
Bus	The trigger source is set to "Bus" and waiting for trigger.
Meas	A measurement is in progress.

Overload Protection Indicator

Displays when the overload is detected.

Port R Status

Displays the current settings of input impedance and attenuator for Port R.

Port T Status

Displays the current settings of input impedance and attenuator for Port T.

RF Output OFF Indicator

When RF signal output is turned off, this indicator is displayed.

Service Mode Indicator

Indicates the service mode status. The service mode indicator is displayed when E5061B enters the following state.

Value	Description
SVC (displayed	The E5061B is in service mode, which is used for self-diagnosis and repair of the E5061B or 8 term calibration

in blue)	mode. Therefore, measurement performance will not be guaranteed according to the specifications. If, under normal use, the system remains in the service mode and does not return to normal operating mode, there is a possibility that the instrument is out of order.
SVC (displayed in red)	An abnormal condition has been detected inside the E5061B. The unit may be damaged. Notify the Customer Contact listed at the end of this manual or the distributor from whom the unit was purchased.

S-Param Port AC Couple Indicator

This indicator is displayed when the S-param port couple status is AC.

VBA Status

Displays the state of the execution of the VBA program in the E5061B.

Value	Description
Run	A VBA program is currently running.
Stop	A VBA program has stopped.

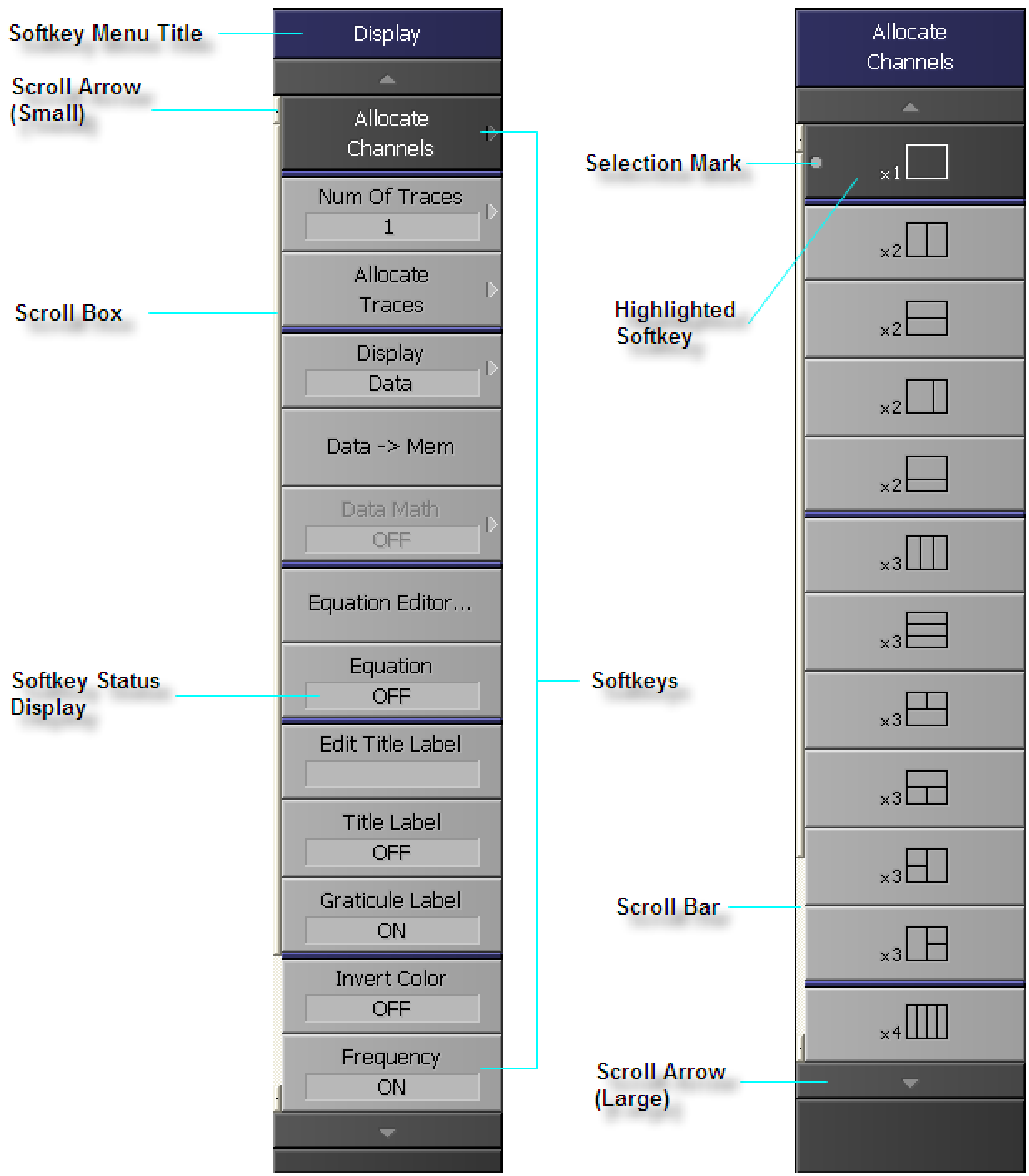
Menu Bar

By using the mouse and keyboard to manipulate the menu bar, you can perform interface operations that are equivalent to those of the keys in the **ACTIVE CH/TRACE** Block, **RESPONSE** Block, **STIMULUS** Block, **MKR/ANALYSIS** Block, and **INSTR STATE** Block on the front panel of the E5061B. The menus on the menu bar correspond to the key blocks, and their submenus to the hardkeys inside the key blocks.

Softkey Menu Bar

A group of keys on the screen called by the softkeys and menu bars. You can manipulate these keys by using the **NAVIGATION** Block keys on the front panel, the mouse, or the keyboard. You can perform manipulations by directly touching the screen with your finger instead of using a mouse.

Softkey Menu Bar



NOTE To manipulate a menu bar, it has to be selected as the object to manipulate (with the focus placed on it). When the focus is placed on a menu bar, the menu title area at the top is displayed in blue. Pressing or clicking on **Foc** Key in the ENTRY Block enables you to move the focus to the desired object.

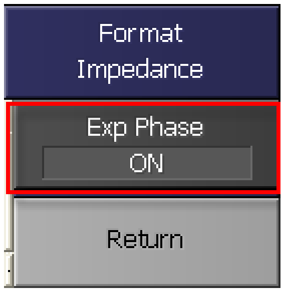
E5061B Option 005

With E5061B Option 005, **Impedance Analysis Menu** under the **Measurement** menu is activated. Else, this button is grayed out or disabled.

e5061b083

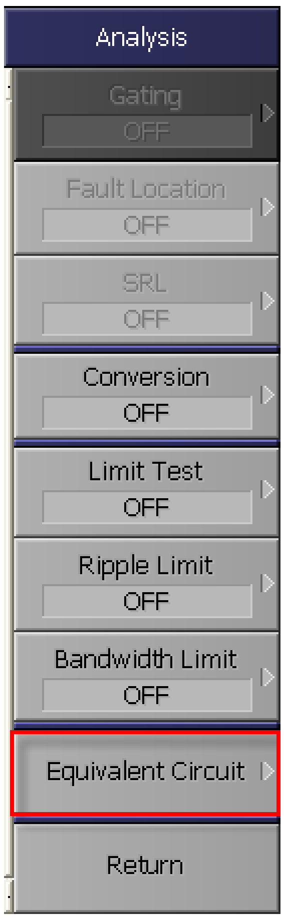
Selecting Impedance measurement, **|Z|** for an active trace changes:

- **Format** menu display. The only available Format option is **Expand Phase**.



e5061b084

- **Analysis** menu display. The **Equivalent Circuit** button is enabled. Else, this button is grayed out or disabled.



e5061b085

All Other E5061B Options Except for Option 3L5

For all other E5061B options except for E5061B Option 3L5, limited buttons as available under the **Measurement** menu, as shown below. Other buttons such as **Gain-Phase**, **Gain-Phase Setup**, **DC Monitor Setup** and **Impedance Analysis Menu** are not available:

e5061b086

Highlighted Softkey

Pressing and **Enter** key on the front panel or pressing **Enter** key on the keyboard causes the highlighted (selected) softkey to be executed. You can change which softkey in the menu is highlighted by turning or pressing on the front panel or by pressing on the keyboard.

Pressing the key on the front panel or the key on the keyboard brings up the upper level softkey menu, and pressing the key on the front panel or the key on the keyboard brings up the lower level softkey menu.

Scroll Arrow (Large)

When the softkeys in a menu overflow the screen, use this key to enable you to scroll the menu page by page. Both upward and downward scroll arrows are available. Use the mouse to manipulate these buttons.

Scroll Arrow (Small)

Using this button, you can scroll the menu one softkey at a time. Both upward and downward scroll arrows are available. Use the mouse to manipulate these buttons.

Scroll Bar

When the softkeys in a menu overflow the screen, clicking on the blank part of the scroll bar enables you to scroll the softkey menu up or down.

Scroll Box

You can scroll the softkey menu up or down by using the mouse to select and drag the scroll box (pressing the button on the object to be moved and then releasing the button at the desired location). The length and position of the scroll box indicate the length and position of the currently displayed part of the softkey menu relative to the entire menu.

Selection Mark

Shows which softkey function is currently selected.

Softkeys

These are the actual keys you would use to perform setup. A ▶ displayed to the right of a softkey indicates that pressing that softkey displays the lower layer of softkeys.

Softkey Menu Title

The title of the softkey menu is displayed here. Double-clicking on this part of the menu bar displays the top layer of softkeys.

Softkey Status Display

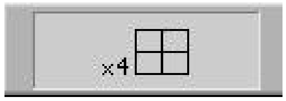
Displays a softkey's setup status.

You can hide the frequency information in order to ensure its confidentiality or for other reasons. See Hiding Softkey's Frequency Information.

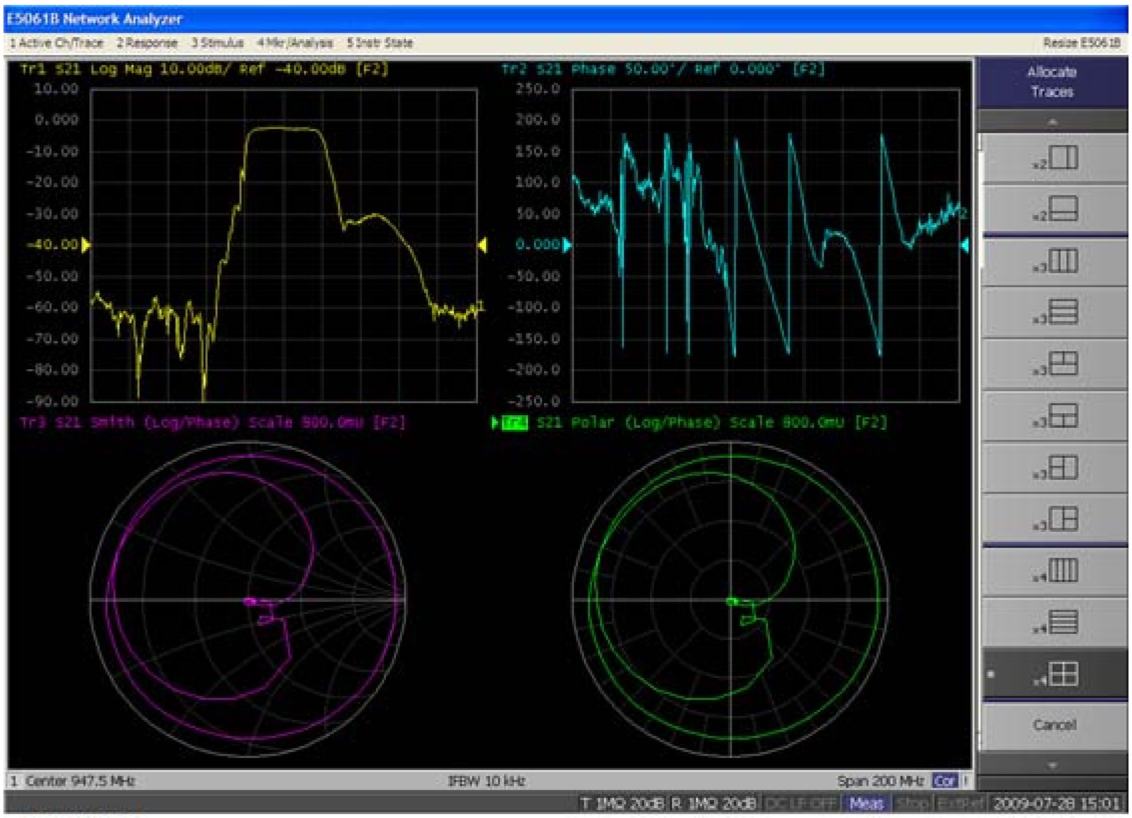
Three Methods of Operation

You can operate the E5061B using one of three operating methods: using keys on the front panel, using a mouse and keyboard, and using the touch screen. This section illustrates these three operating methods through the example in which the channel window layout is set to four-channel display as shown in the following figure.

In the next section and those following it, a series of operations is expressed as follows:

Display > **Allocate Traces** > 

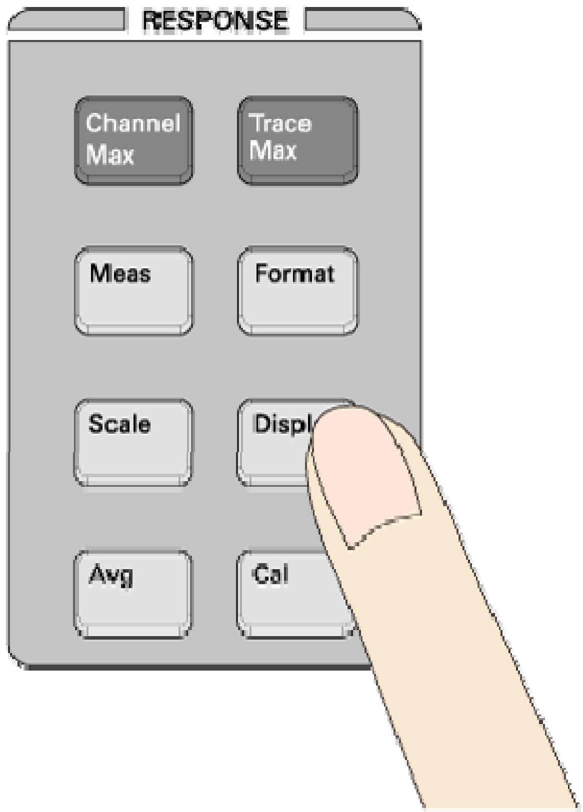
Four-Trace Display



e5061b076

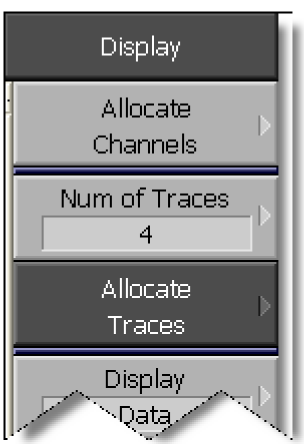
Operating Method Using Keys

1. Press the **DISPLAY** key in the **RESPONSE** Block.

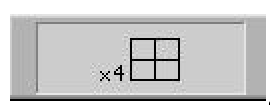


e5071c307

2. Press or key to move the cursor to the **Allocate Traces**, then press **Enter** or ( ) key.

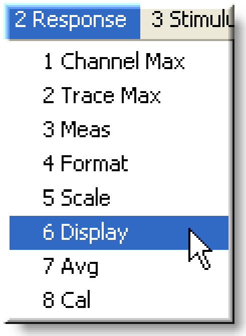


e5071c002

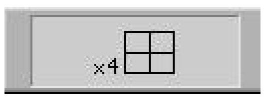
3. Press or key to move the cursor to , then press **Enter** key.

Operation Method Using a Mouse

1. From the **Response** menu, press **Display** key.

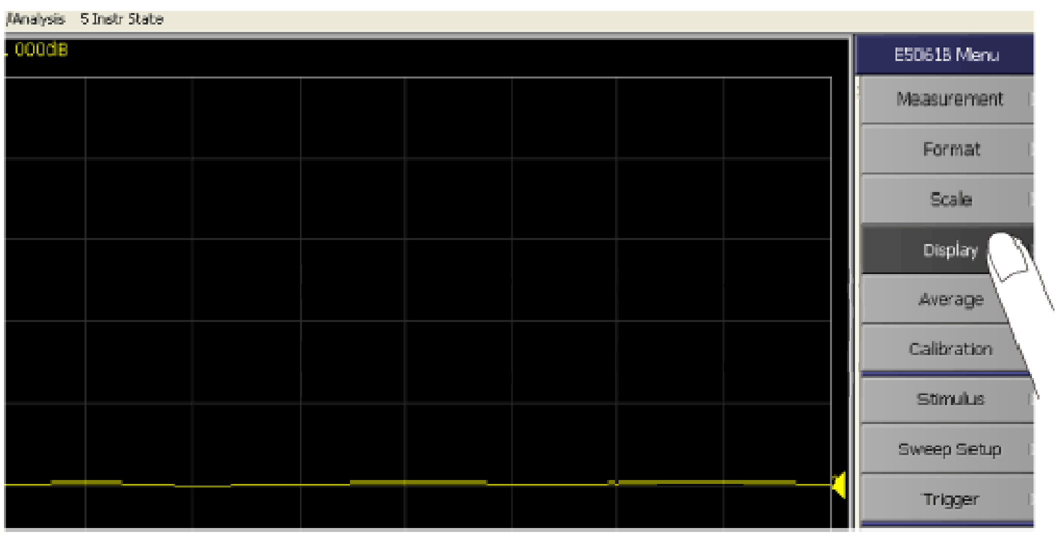


e5071c051

2. Click **Allocate Traces**.
3. Click 

Operation Method Using the Touch Screen

1. Press **Display** key after presetting the E5061B (Executing **Preset** key).



2. Click/Press **Allocate Traces.**
3. Click/Press any desired setting.

S-Parameter Measurement

S-Parameter Measurement

- Basic Measurement Procedure
- Measurement Example of a Bandpass Filter

Basic Measurement Procedures

This section describes the basic measurement procedure using the E5061B and presents an example of the transmission measurement of a bandpass filter, for better understanding of how to use the E5061B.

Basic Measurement Flow

1. **Determining measurement conditions**
   - Initializing Parameters
   - Select:
     - Measurement Parameter
     - Data format
     - Sweep Type and range
     - Power level
     - IF bandwidth
2. **Calibration**
   - Select calibration kit
   - Make a calibration
3. **Connecting the Device Under Test (DUT)**
   - Connect DUT
   - Adjust the scale
4. **Analyzing measurement results**
   - Analysis using markers
5. **Outputting measurement results**
   - Store measurement result into a file

Measurement Example of a Bandpass Filter

This section describes how to measure the transmission characteristics of a 947.5 MHz bandpass filter. The measurement conditions for this measurement example are those suitable for a 947.5 MHz bandpass filter. To measure another device under test (DUT), change the measurement conditions to suit the particular DUT.

STEP 1. Determining Measurement Conditions

1. Preset the E5061B.

   Preset > **OK**

2. Set the S-parameter to S21.

   Meas > **S21**

   NOTE When measuring the reverse transmission characteristics, set the S-parameter to S12.

3. Set the data format to the log magnitude format

   Format > **Log Mag**

4. Set the center frequency to the bandpass filter center frequency. Next, specify the span frequency, which is set to 200 MHz in this measurement example.

   Center > **9** > **4** > **7** > **.** > **5** > **M/m**
   Span > **2** > **0** > **0** > **M/m**

   NOTE When entering the frequency unit using the keyboard, type "G" for GHz, "M" for MHz, and "k" for kHz.

5. Specify the number of measurement points per sweep. The number of measurement points in this measurement example is set to 401.

   Sweep Setup > **Points** > **4** > **0** > **1** > **x1**

6. Specify the power level of the signal source. The power level in this measurement example is set to -10 dBm.

   Sweep Setup > **Power** > **+/-** > **1** > **0** > **x1**

7. Specify the IF bandwidth of the receiver as necessary. In this measurement example, the IF bandwidth is set to 10 kHz because of the need to lower the noise floor.

   Avg > **IF Bandwidth** > **1** > **0** > **k/m**

STEP 2. Calibration

To turn the error correction ON, set the calibration type to the full 2-port calibration and measure the calibration data.

For details about calibration, see Calibration.

1. Select the calibration kit suitable for the measurement cable. In this measurement example, Calibration Kit 85032F is selected.

   Cal > **Cal Kit** > **85032F**

2. Set the calibration type to the full 2-port calibration using the test port 1 and 2.

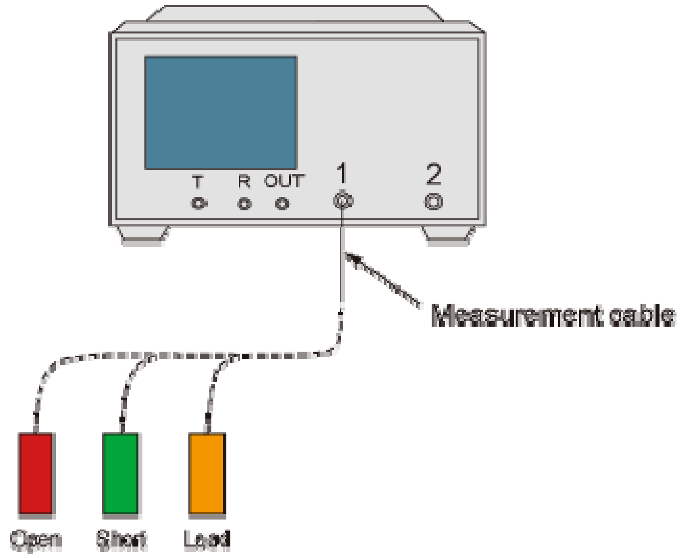
   Cal > **Calibrate** > **2-Port Cal**

3. Connect the OPEN standard (included in the calibration kit) to the other end of the measurement cable that is connected to the test port 1 as shown in the following figure, and measure the open calibration data at the test port 1. After measuring the open calibration data, a checkmark √ is displayed to the left of the **Port 1 Open** menu.

   Cal > **Calibrate** > **2-Port Cal** > **Reflection** > **Port1 Open**

   In the same way, measure the calibration data for the SHORT/LOAD standards at the test port 1.

   Connecting the OPEN/SHORT/LOAD standards

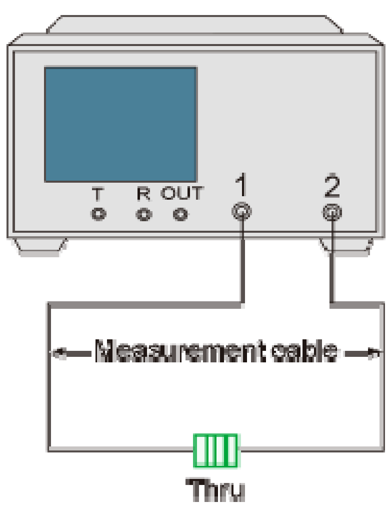


e5061b042

4. In the same way as described above, measure the calibration data for the OPEN/SHORT/LOAD standards at the test port 2.
5. Connect the measurement cables as shown in the figure below, and measure the thru calibration. After measuring the thru calibration data, a checkmark √ is displayed to the left of the **Port 1-2 Thru** button.

Cal > **Calibrate** > **2-Port Cal** > **Transmission** > **Port 1-2 Thru**

Making the through calibration



e5061b044

6. Set the full 2-port calibration measurement to DONE. The calibration factor is calculated based on the calibration data acquired, and the error correction is turned ON.

Cal > **Calibrate** > **2-Port Cal** > **Done**

7. Select the type in which the data is to be saved before saving the calibration factor (calculated based on the calibration data).

Save/Recall > **Save Type** > **State & Cal**

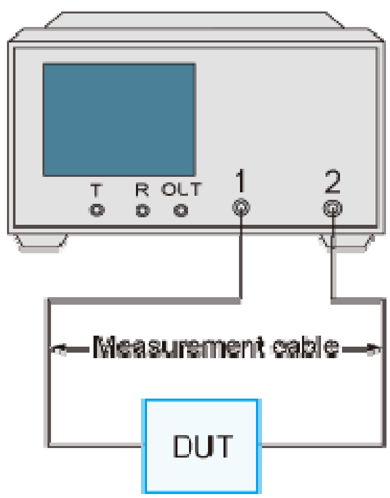
8. Store the calibration file to the disk of the E5061B. The symbol "X" appearing in the operations below represent the assigned numbers to be used when the file is saved.

Save/Recall > **Save State** > **State 0X**

STEP 3. Connecting the Device Under Test (DUT)

1. Connect to the DUT to the E5061B. (See the below figure)

Connecting the DUT



e5061b044

2. Set the appropriate scale by executing the auto scale. (See the below figure)

Scale > **Auto Scale**

You can also adjust the scale by entering arbitrary values in the **Scale/Div** button, **Reference Position** button, and **Reference Value**.

S21 trace after executing the auto scale

STEP 4. Analyzing Measurement Results

This section describes how to use the marker function to read out important parameters for the transmission measurement of the bandpass filter (insertion loss, -3 dB bandwidth).

Measuring the Insertion Loss

1. Display a marker.

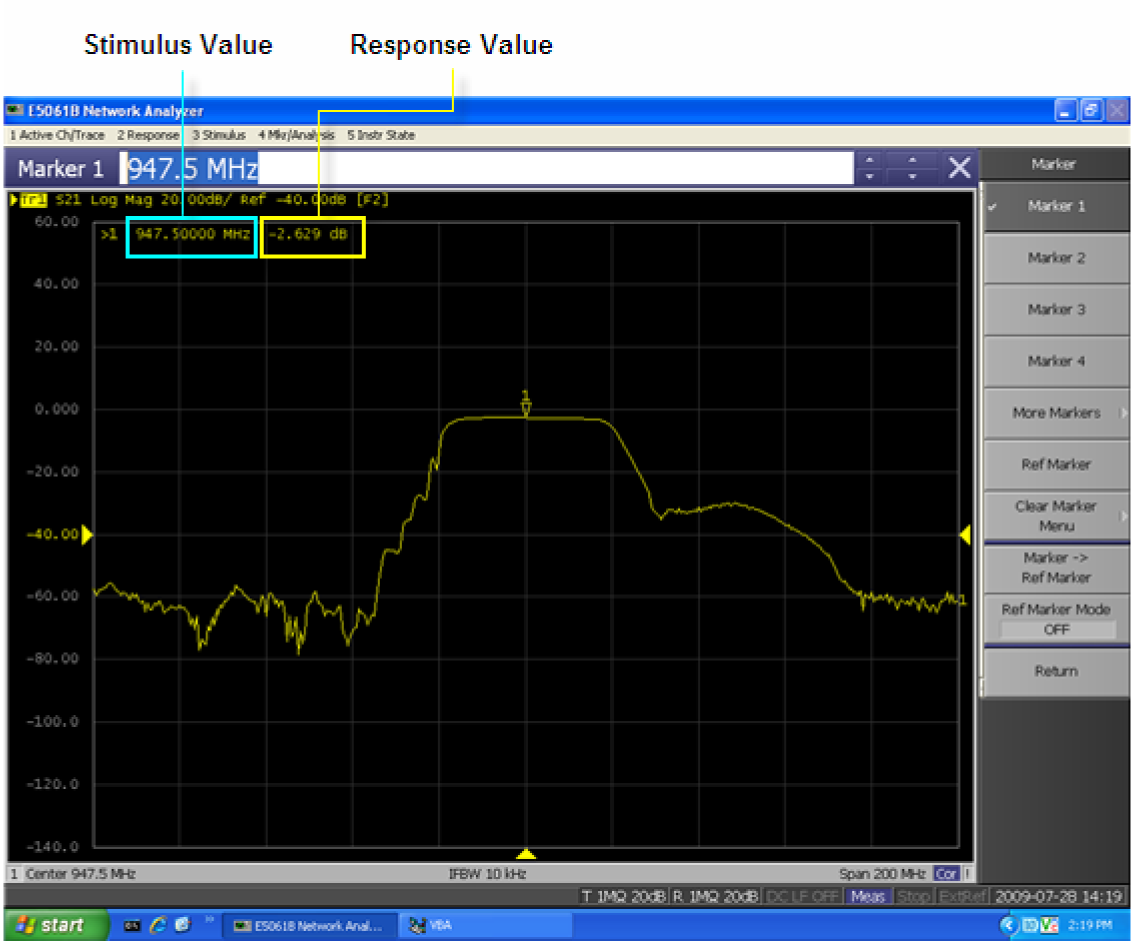
   Marker > **Marker 1**

2. Using one of the following methods to move the marker to the center frequency of the bandpass filter.

- On the entry bar, press **9**> **4** > **7** > **.** > **5** > **M/m**
- Turn the rotary knob  on the front panel to set it to the center frequency (947.5 MHz).

3. Read the marker value displayed as shown in the figure below. In this example, the response value denotes the insertion loss.

Measuring an Insertion Loss



Measuring the -3 dB Bandwidth

Using the marker bandwidth search function, the bandwidth, center frequency between two cutoff frequency points, Q value, and insertion loss are all read out. These parameters are described in the following table.

NOTE If the two cutoff frequency points are not found, all data items except the insertion loss revert to zero.

Parameter	Description
BW (Bandwidth)	Stimulus width between two cutoff frequency points (low and high)
cent (Center Frequency)	Center point between cutoff frequency points (low and high)
low (Left-side Cutoff Frequency)	The lower frequency of the two cutoff frequency points
high (Right-side Cutoff	The higher frequency of the two cutoff

Frequency)	frequency points
Q (Q Value)	Q = cent/BW
loss (Insertion Loss)	The measured value of the active marker.

1. Display a marker.

Marker > **Marker 1**

2. Using one of the following methods to move the marker to the center frequency of the bandpass filter.

- On the entry bar, press **9** > **4** > **7** > **.** > **5** > **M/m**
- Turn the rotary knob on the front panel to set it to the center frequency (947.5 MHz).

3. Specify the bandwidth definition value that defines the pass band of the filter. In this measurement example, it is set to -3 dB.

Marker Search > **Bandwidth Value** > **+/-** > **3** > **x 1**

4. Set the bandwidth search function ON.

Marker Search > **Bandwidth**

5. The bandwidth data items (BW, cent, low, high, Q, loss) is displayed. (See the following figure.)

Measuring the -3 dB Bandwidth

STEP 5. Outputting Measurement Results (Save)

You can save not only the internal data but also the measurement results such as trace data and display screens to the disk.

Saving the Trace Data(in CSV format)

You can save the trace data to the disk of the E5061B in CSV file format (extension: .csv). Since the CSV-formatted data to be saved is a text file, you can analyze the data using Microsoft Excel.

Follow the step below to save the trace data:

Save/Recall > **Save Trace Data**

Saving the Display Screen

You can save the screen displayed on the E5061B to the disk of the E5061B in Windows bitmap file format (extension: .bmp) or Portable Network Graphics format (extension: .png).

Follow the step below to save the display screen:

System > **Dump Screen Image**

NOTE The image on the LCD display memorized in the volatile memory (clipboard) (the image on the LCD display when the **Capture/System** key is pressed) is saved.

Impedance Measurement (Option 005)

Impedance Measurement

- Basic Measurement Procedure
- Measurement Method
- Measurement Example of a Capacitor (Port 1 Reflection)
- Measurement Example of a Crystal (Port 1-2 Series)
- Measurement Example of a Capacitor (Port 1-2 Shunt)
- Measurement Example of a Ceramic Resonator (Gain-Phase/Series-Through)
- Measurement Example of a Capacitor (Gain-Phase/Shunt-Through)

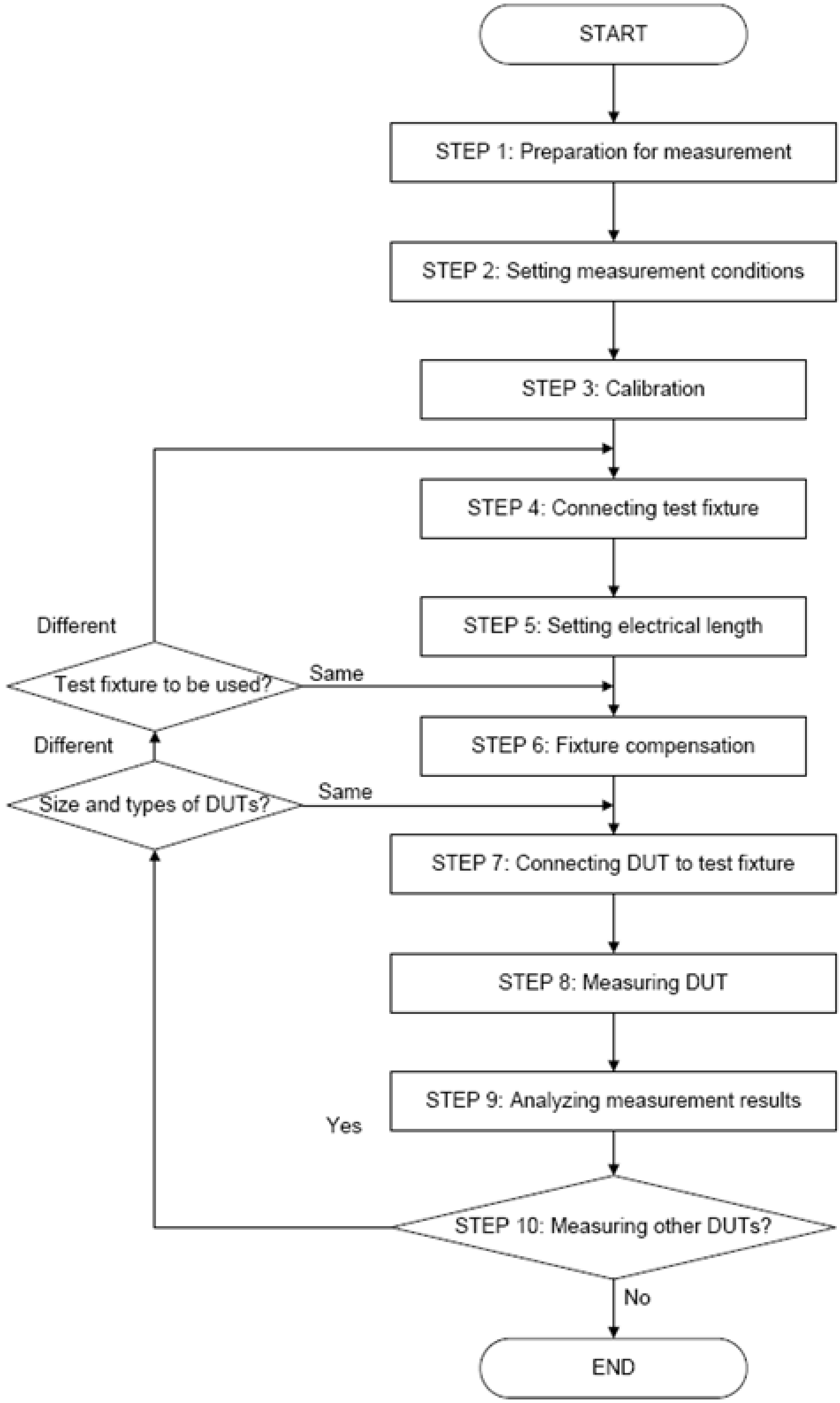
Basic Measurement Procedure

- Procedure for Port 1 Reflection
- Procedure for Port 1-2 Shunt/Series and GP Shunt/Series

Other topics about Impedance Measurement Quick Start

Procedure for Port 1 Reflection

The basic procedure for Port 1 Reflection method is shown in the flow chart below:



STEP 1:Preparation for measurement

STEP 2: Setting measurement conditions

STEP 3: Calibration

STEP 4: Connecting test fixture

STEP 5: Setting electrical length

STEP 6: Fixture compensation

STEP 7: Connecting DUT to test fixture

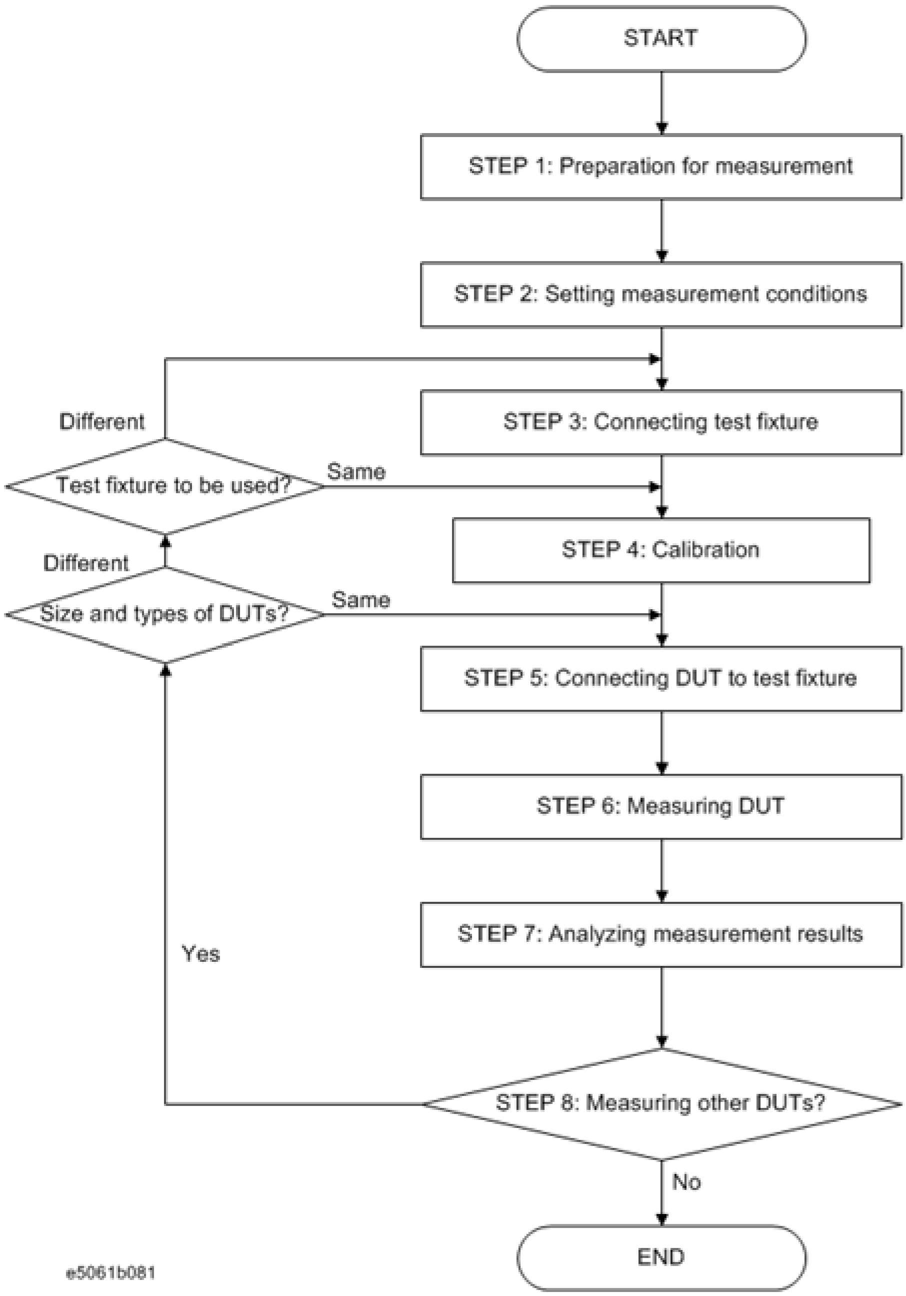
STEP 8: Measuring DUT

STEP 9: Analyzing measurement results

STEP 10: Measuring other DUTs

Procedure for Port 1-2 Shunt / Series and GP Shunt / Series

The basic procedure for Port 1-2 Shunt/Series and GP Shunt/Series method is shown in the flow chart below:



STEP 1:Preparation for measurement

STEP 2: Setting measurement conditions

STEP 3: Connecting test fixture

STEP 4: Calibration

STEP 5: Connecting DUT to test fixture

STEP 6: Measuring DUT

STEP 7: Analyzing measurement results

STEP 8: Measuring other DUTs

Measurement Method

- Available Methods
- Measurement DUT Impedance Range for Each Method
- Configuration for Each Method

Other topics about Impedance Measurement Quick Start

Available Methods

This section describes the impedance measurement method. Five methods shown in the following table can be used to make an impedance measurement. For the connection for each method, see Preparation for Measurement.

When you select Impedance (**Meas** > **Impedance Analysis Menu**) as the Measurement type, the measurement method is made available (**Meas** > **Impedance Analysis Menu** >**Method**).

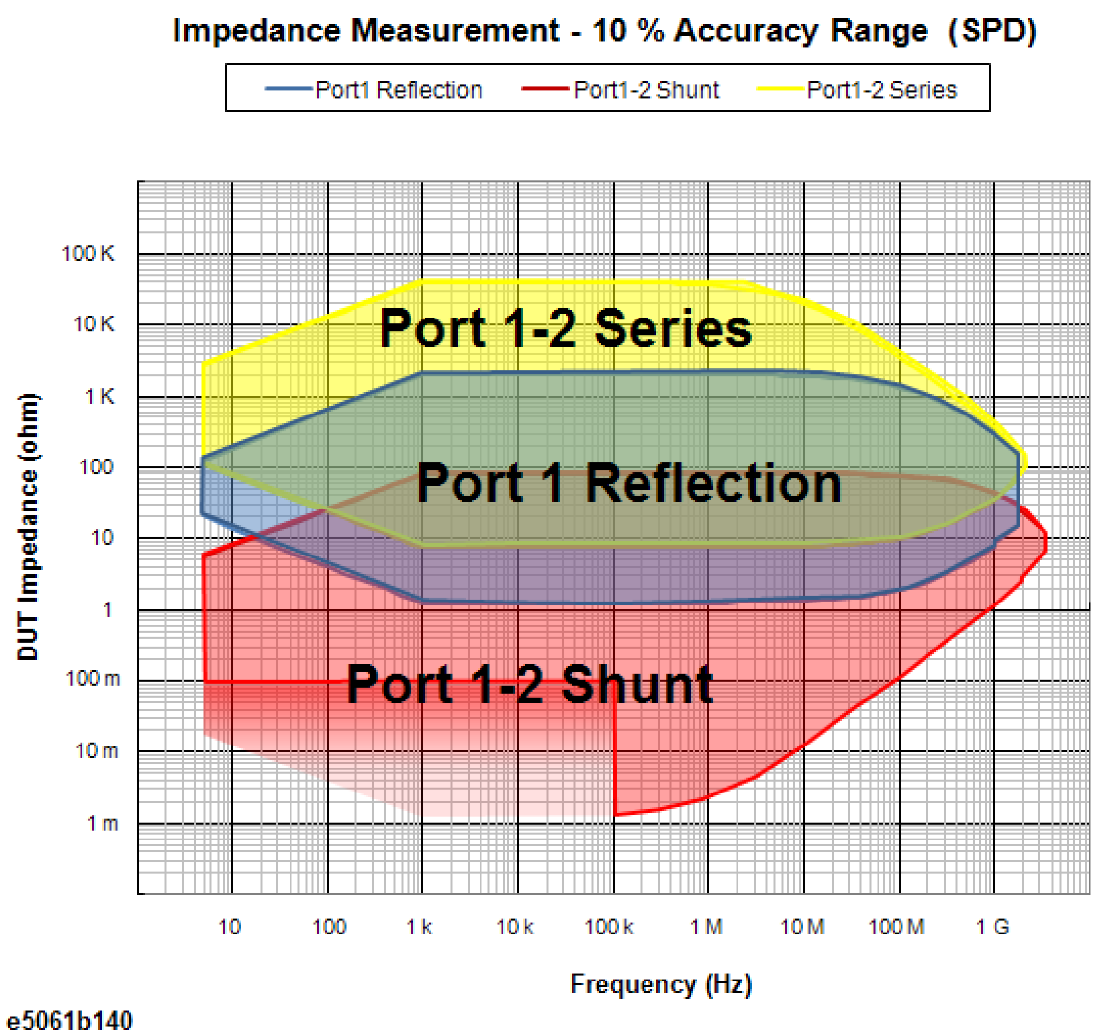
The characteristics of the measurement method is as described in the following table:

Method	Port 1 (or 2) Reflection	Port 1-2 Series	Port 1-2 Shunt	GP Series	GP Shunt
Measurement DUT Impedance Range	Low to middle impedance	Middle to high impedance in the high frequency range Not applicable to grounded DUTs	Very low impedance in the high frequency range	Middle to high impedance in the low frequency range	Very low impedance in the low frequency range
Formula	Zdut = 50 x (1+S11)/(1-S11)	Zdut = 50 x 2 x (1-S21)/S21	Zdut = 50 x S21/(2 x (1-S21))	Zdut = 50 x (1-S21)/S21	Zdut = 50 x S21/(2 x (1-S21))

Measurement DUT Impedance Range for Each Method

The following figures show the 10% accuracy range for each method. You can select the appropriate method according to your DUT impedance.

Ports 1 and 2



Conditions of 10 % measurement accuracy range

The following table shows the condition where the 10% measurement accuracy range shown above is specified.

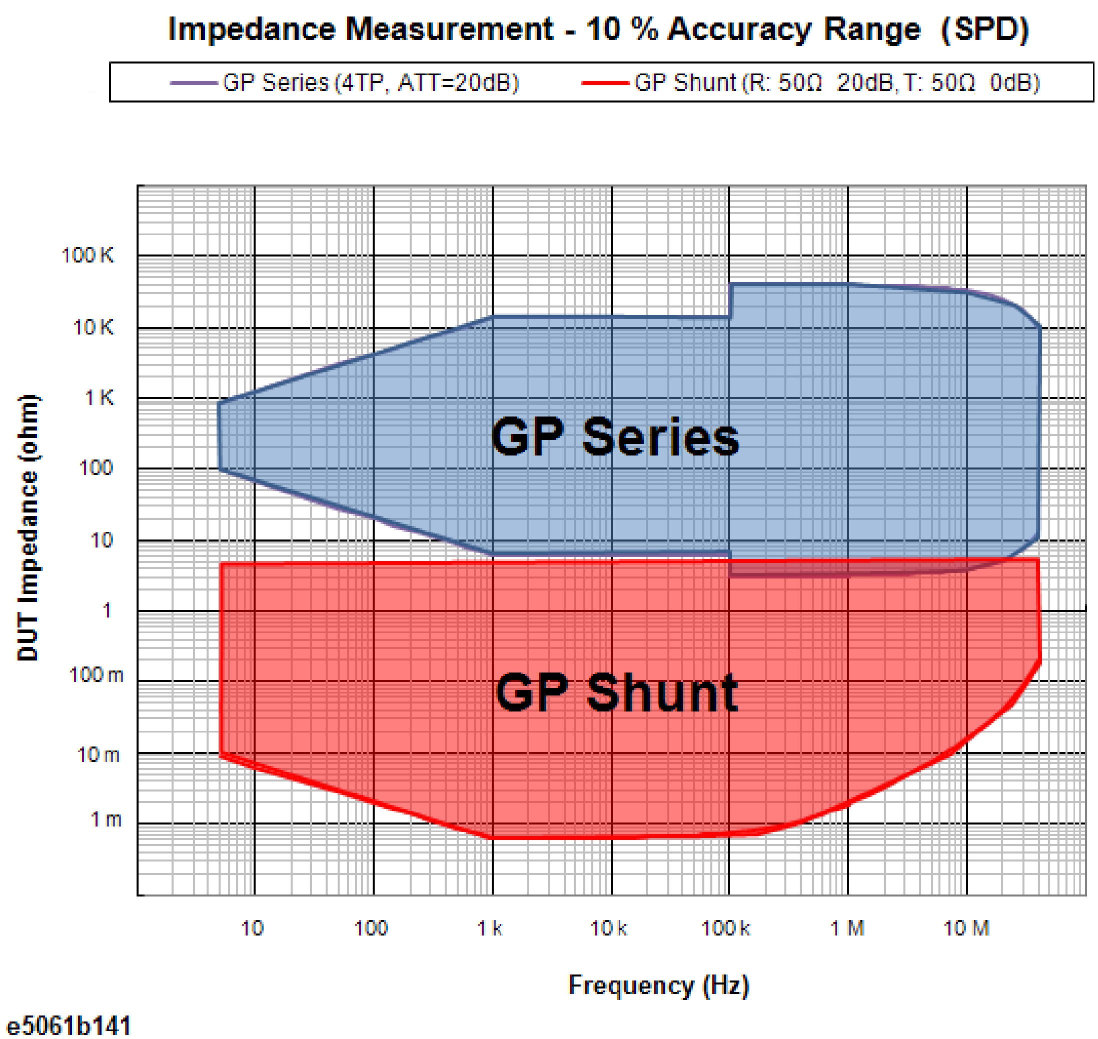
<table>
<tr><th>Method</th><th>Frequency</th><th>Calibration</th><th>IFBW</th><th>Source Power</th><th>Note</th></tr>
<tr><td>Port 1-2 Series</td><td>5 Hz to 3 GHz</td><td>Full 2-port calibration at measurement terminals of fixture
or
Full 2-port calibration + Open/Short/L</td><td>See the following table<table><tr><th>Measurement Frequency</th><th>IF BW</th></tr><tr><td>< 200 Hz</td><td>≤ (1/5 × Measurement</td></tr></table></td><td>-20 to 0 dBm</td><td></td></tr>
</table>

		oad fixture compensation **Note**	Frequency) Hz ≥ 200 Hz ≤ 40 Hz		
Port 1 Reflection		Open/Short/Load calibration at 7 mm terminal of the 16201A. Calibration kit: 16195B or 85031B			
Port 1-2 Shunt	100 kHz to 3 GHz	Full 2-port calibration at measurement terminals of fixture or Full 2-port calibration + Open/Short/Load fixture compensation **Note**	10 Hz	10 dBm	Measurement error in the short calibration is included. (10 pH residual inductance of short standard is included.) A ferrite core is required to measure DUTs with 100 mΩ or below at ≤100 KHz.

Temperature Condition

- 23±5 °C at calibration
- (calibration temperature) ±1 °C at measurement

Gain-Phase Ports



Conditions of 10 % measurement accuracy range

The following table shows the condition where the 10% measurement accuracy range shown above is specified.

Method	Frequency	Calibration	Receiver Setup	IFBW	Source Power	Note
GP Series	5 Hz to 30 MHz	Open/Short/Load calibration at measurement terminals of fixture Fixture: 16047E or	Rch: Zin=50 Ω , Att=20 dB Tch: Zin=50 Ω , Att=2	See the following table <table><tr><th>Meas. Frequency</th><th>IF BW</th></tr><tr><td>< 200 Hz</td><td>≤ (1/5 × Measurement</td></tr></table>	-20 to 0 dBm	Only with the response-thru calibration at the terminals of fixture, the measure

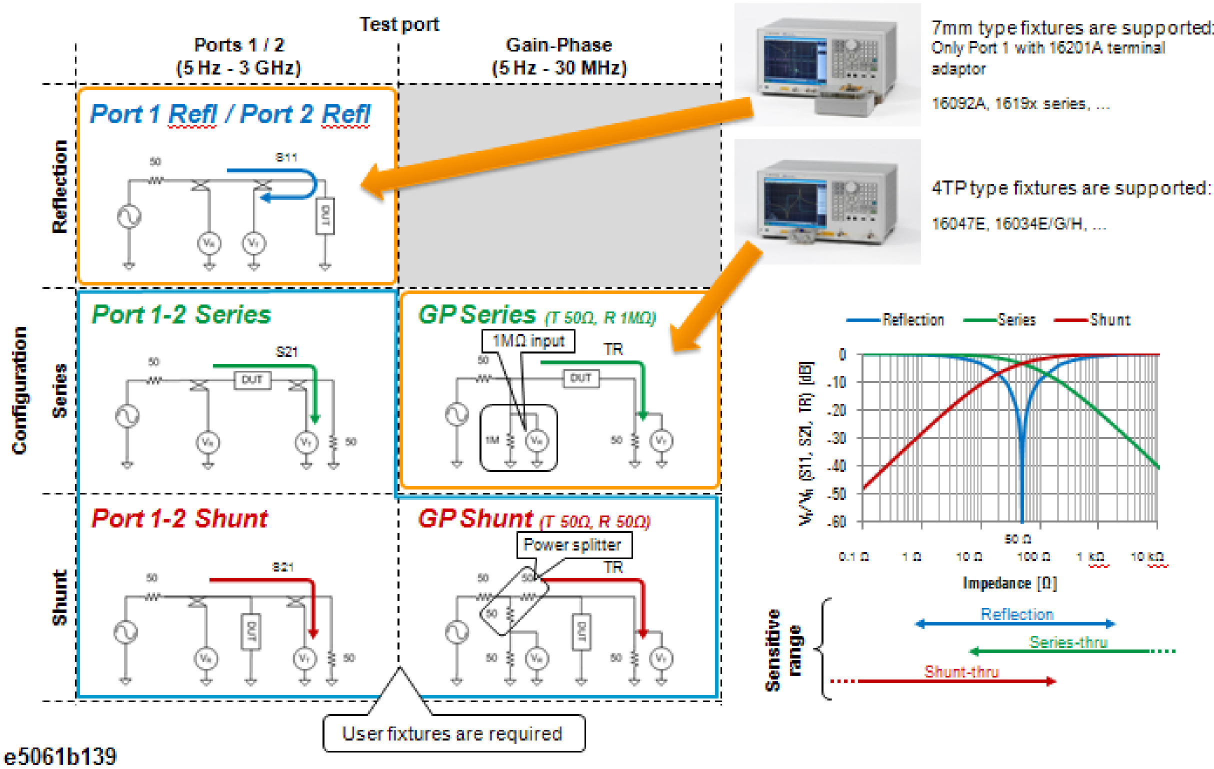
		16034E/G/H Load Standard: Agilent PN 5012-8646 (THT) or 0699-2829 (SMD)	0 dB	| | Frequency) Hz | | ≥ 200 Hz | ≤ 40 Hz |		ment accuracy may be degraded due to a parasitic capacitance of receiver port at RF range (≥1 MHz)
GP Shunt		Open/Short /Load calibration at measurement terminals of fixture (Source=-10 dBm at calibration) **Note**	Rch: Zin=50 Ω , Att=20 dB Tch: Zin=50 Ω , Att=0 dB	See the following table | **Meas. Frequency** | **IF BW** | | < 50 Hz | ≤ (1/5 × Measurement Frequency) Hz | | ≥50 Hz | ≤ 40 Hz |	10 dBm	Measurement error in the short calibration is included (10 pH residual inductance of short standard is included.) Maximum DUT impedance is 5 Ω in this condition in order to avoid a receiver saturation

Temperature Condition

- 23±5 °C at calibration
- (calibration temperature) ±1 °C at measurement

Configuration for Each Method

The following figure shows the configuration for each method.



Test port
Ports 1 / 2
(5 Hz - 3 GHz)
Gain-Phase
(5 Hz - 30 MHz)
Configuration
Reflection
Series
Shunt
Port 1 Refl / Port 2 Refl
S11
DUT
Port 1-2 Series
S21
GP Series (T 50Ω, R 1MΩ)
1MΩ input
TR
Port 1-2 Shunt
GP Shunt (T 50Ω, R 50Ω)
Power splitter
User fixtures are required
7mm type fixtures are supported:
Only Port 1 with 16201A terminal adaptor
16092A, 1619x series, ...
4TP type fixtures are supported:
16047E, 16034E/G/H, ...
Reflection
Series
Shunt
Impedance [Ω]
Sensitive range
Reflection
Series-thru
Shunt-thru

e5061b139

Measurement Example of a Capacitor (Port 1 / Reflection)

This section describes how to measure a Capacitor. In this example, apart from E5061B option 005, 16201A terminal adapter and 16196A test fixture are used. The measurement is performed with 10 pF capacitor, hence, to measure another device under test (DUT), change the measurement conditions to suit accordingly. Prior to the measurement, ensure that the 16201A terminal adapter is connected to the E5061B network analyzer. See Connecting Terminal Adapter.

STEP 1. Setting Measurement Conditions

1. Preset the E5061B.

   Preset > **OK**

2. Set the trace display settings.

   Display > **Num of Traces** > **2**

   Display > **Allocate Traces** > **x2**

3. Set the measurement port to S-Parameter.

   Meas > **Measurement Port** > **S-Parameter**

4. Set the measurement method to Port 1 Reflection.

   Meas > **Impedance Analysis Menu** > **Method** > **Port 1 Refl**

5. Set the measurement type of each trace.

   Select Trace 1 as the active trace. **Meas** > **Impedance Analysis Menu** > **|Z|**

   Select Trace 2 as the active trace. **Meas** > **Impedance Analysis Menu** > **θz**

6. Set the format of the measurement of each trace.

   Select Trace 1 as the active trace. **Format** > **Exp Phase** > **OFF**

   Select Trace 2 as the active trace. **Format** > **Exp Phase** > **OFF**

7. Set the sweep setup power.

   Sweep Setup > **Power** > **-10dBm**

8. Set the sweep type.

   Sweep Setup > **Sweep Type** > **Log Freq**

9. Set the frequency bandwidth.

Avg > **IF Bandwidth** > **100 Hz**

STEP 2. Calibration

Once the measurement condition is set, impedance calibration should be performed. The 16195B calibration kit is required to perform the calibration.

1. Connect the E4991-60022 OPEN standard to the 16201A terminal adapter (which is connected to Port 1 of E5061B).

   Cal > **Cal Kit** > **16195B**

   Cal > **Calibrate** > **Impedance Calibration** > **Open**

   Once the open calibration is completed, a checkmark √ is displayed to the left of the **Open** menu.

2. Remove the OPEN standard and connect the E4991-60021 SHORT standard to the terminal adapter.

   Cal > **Calibrate** > **Impedance Calibration** > **Short**

   Once the short calibration is completed, a checkmark √ is displayed to the left of the **Short** menu.

3. In the same way, measure the calibration data for LOAD standard and LOW LOSS C standard. Use 04287-60021 50 Ω termination LOAD standard and 04287-60022 LOW LOSS Capacitor standard.

   Cal > **Calibrate** > **Impedance Calibration** > **Load**

   Cal > **Calibrate** > **Impedance Calibration** > **Low-Loss C**

   Once the calibrations are completed, a checkmark √ is displayed to the left of the **Load** and **Low-Loss C** menu.

4. Set the calibration to DONE to save the performed calibration.

STEP 3. Fixture Compensation

As 16196A test fixture is used in this measurement example, fixture compensation should be performed to reduce possible errors induced by the test fixture. Ensure that the insulator assembly used is appropriate with the DUT. Refer to 16196A Test Fixture Operation and Service Manual to learn more about the fixture.

1. Connect the 16196A test fixture to the terminal adapter and set the electrical length:
   a. Turn the adapter's 7-mm connector in the counterclockwise direction when viewed from above and screw the connection sleeve in fully.

   b. Align the text fixture with the adapter's mount post and 7-mm connector and set it gently in place.
   c. Turn the adapter's 7-mm connector counterclockwise, connecting the bottom of the test fixture with the connector.
   d. **Cal** > **Fixture Compen** > **Fixture** > **16196A**
2. Set the open state by using the open state supplied.
   a. Using the Tweezers, place the open plate on top of the insulator assembly.
   b. Set the open plate with the protruding surface down.
   c. Fit the cap in place with the mark toward the front, and turn it to the right until it is locked.
   d. **Cal** > **Fixture Compen** > **Compensate** > **Open**
   e. Once the open compensation is completed, a checkmark is displayed to the left of the **Open** menu.
3. Set the short state by using the open state supplied.
   a. Remove the cap. Remove the open plate used to measure the open compensation data.
   b. Place the short plate on the insulator assembly with tweezers. Place the rod-shaped protrusion of the short plate downward, and insert it into the DUT insertion hole.
   c. Fit the cap in place with the mark toward the front, and turn it to the right until it is locked.
   d. **Cal** > **Fixture Compen** > **Compensate** > **Short**
   e. Once the short calibration is completed, a checkmark is displayed to the left of the **Short** menu.
4. Set the compensation to DONE to save the performed fixture compensation. Now, the fixture compensation should be automatically turned ON (**Cal** > **Fixture Compen** > **ON**).

STEP 4. Connecting Device Under Test (DUT)

1. Remove the cap.
2. Insert the DUT into the insulator hole with tweezers. Use a magnifying glass to check that the DUT is inserted deeply enough into the insulator hole for it to contact the bottom electrode.
3. Fit the cap in place with the mark toward the front, and turn it to the right until it is locked.
4. Set the log scale for Trace 1

   Select Trace 1 as the active trace. **Scale** > **Y-Axis** > **Log**

5. Set the appropriate scale for both traces by executing the auto scale.

Scale > **Auto Scale All**

STEP 5. Analyzing Measurement Results

This section describes how to use Equivalent Circuit function to analyze the measurement.

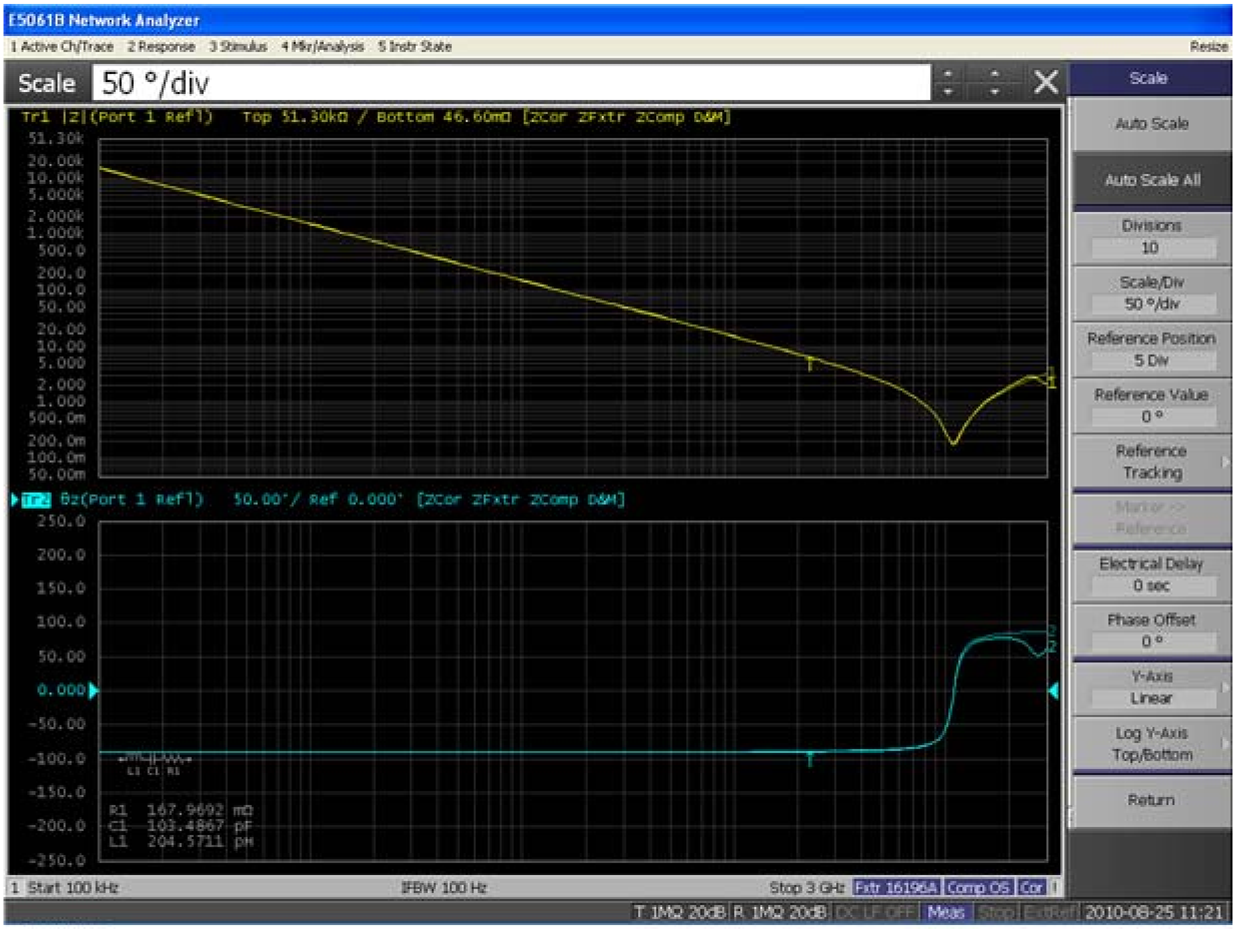
1. **Analysis** > **Equivalent Circuit** > **Select Circuit** > **D**.
2. **Analysis** > **Equivalent Circuit** > **Calculate.** The calculated equivalent circuit parameters are displayed in each box of R1, C1 and L1.
3. **Analysis** > **Equivalent Circuit** > **Simulate** > **ON.**
4. **Analysis** > **Equivalent Circuit** > **Display** > **ON**.

Sample results as shown below:

DUT: 10 pF capacitor

e5061b095

DUT: 100 pF capacitor


E5061B Network Analyzer
1 Active Ch/Trace
2 Response
3 Stimulus
4 Mkr/Analysis
5 Instr State
Resize
Scale
50 °/div
Tr1 |Z|(Port 1 Refl) Top 51.30kΩ / Bottom 46.60mΩ [ZCor ZFxtr ZComp D&M]
Tr2 θz(Port 1 Refl) 50.00°/ Ref 0.000° [ZCor ZFxtr ZComp D&M]
R1 167.9692 mΩ
C1 103.4867 pF
L1 204.5711 pH
1 Start 100 kHz
IFBW 100 Hz
Stop 3 GHz
Auto Scale
Auto Scale All
Divisions
10
Scale/Div
50 °/div
Reference Position
5 Div
Reference Value
0 °
Reference Tracking
Marker -> Reference
Electrical Delay
0 sec
Phase Offset
0 °
Y-Axis
Linear
Log Y-Axis Top/Bottom
Return
2010-08-25 11:21
e5061b096

Measurement Example of a Crystal (Port 1-2 / Series-Through)

This section describes how to measure the frequency characteristics of a crystal by using the Series-Through method on Ports 1 and 2.

In this example, the following items are used.

Description	Product / Agilent Part Number	Note
Test Fixture	User Fixture	-
Shorting device	User Shorting device	-
Leaded Load	5012-8646	50 Ω leaded resister. Furnished with E5061B option 720.
DUT	Crystal	-
Cable and Adapter	Cables and adapters for Type-N or 3.5 mm	Connect your fixture to ports 1 and 2 of the E5061B

To measure another device under test (DUT), change the measurement conditions to suit the particular DUT.

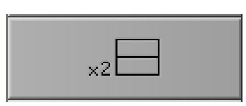
STEP 1. Determining Measurement Conditions

1. Preset the E5061B.

   Preset > **OK**

2. Set the number of traces at two and display each trace in one frame.

   Display > **Num of Traces** > **2**

   Display > **Allocate Traces** > **x2** 

3. Set the measurement port to S Parameter.

   Meas > **Measurement Port** > **S-Parameter**

4. Set the method to Port 1-2/Series-Through configuration.

   Meas > Impedance Analysis Menu > Method > Port 1-2 Series

5. Set the measurement parameter at |Z| for the trace 1 and θ type of each trace.

   Select Trace 1 as the active trace. **Meas** > **Impedance Analysis Menu** > **|Z|**

   Select Trace 2 as the active trace. **Meas** > **Impedance Analysis Menu** > **θz**

6. Specify the center and span frequencies to observe the frequency characteristic. In this example, the center is set at 32 MHz and span is set at 30 kHz

   Center > **3** > **2**> **M/μ**
   Span > **3** > **0** >**k/m**

   NOTE When entering the frequency unit using the keyboard, type "G" for GHz, "M" for MHz, and "k" for kHz.

5. Set the power level at 0 dBm (224 mV @ 50 Ω).

   Sweep Setup > **Power** > **0** > **x1**

6. Set the sweep type at Log.

   Sweep Setup > **Sweep Type** > **Lin Freq**

7. Set the IF Bandwidth at 100 Hz.

   Avg > **IF Bandwidth** > **1** > **0** > **0** > **x1**

STEP 2. Calibration

1. Select the calibration kit for leaded 50 Ω.

   Cal > **Cal Kit** > **Leaded 50ohm**

2. Connect the test fixture on the ports 1 and 2.
3. Set the open state of the test fixture.
4. Measure the calibration data for open.

   Cal > **Calibrate** > **Impedance Calibration** > **Open**

5. Set the short state of the test fixture.
6. Measure the calibration data for short.

   Short

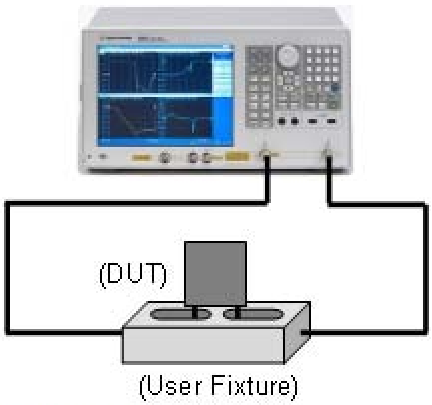
7. Set the load on the test fixture.
8. Measure the calibration data for load.

   Load

9. Finalize the calibration measurement. The calibration factor is calculated based on the calibration data acquired, and the error correction is turned ON. **Cor** is displayed at the bottom of the channel window.
10. **Done**

STEP 3. Connecting the Device Under Test (DUT)

1. Set the DUT on the fixture.



2. Set the log scale for Trace 1.

   Select Trace 1 as the active trace. **Scale** > **Y-Axis** > **Log**
3. Set the appropriate scale for both traces by executing the auto scale.

   Scale > **Auto Scale All**

STEP 4. Analyzing Measurement Results

This section describes how to use the Equivalent circuit analysis.

Reading the values of resonant points

1. Display a marker.

   Marker > **Marker 1**
2. Search the minimum point.

   Marker Search > **Min**

Reading the values of anti-resonant points

1. Display the second marker.

   Marker > **Marker 2**
2. Search the maximum point.

   Marker Search > **Max**

Using Equivalent Circuit Analysis

1. Select the Equivalent circuit model.

   Analysis > **Equivalent Circuit** > **Select Circuit** > **E**.

2. Turn the Equivalent Circuit Display ON.

   Analysis > **Equivalent Circuit** > **Display**

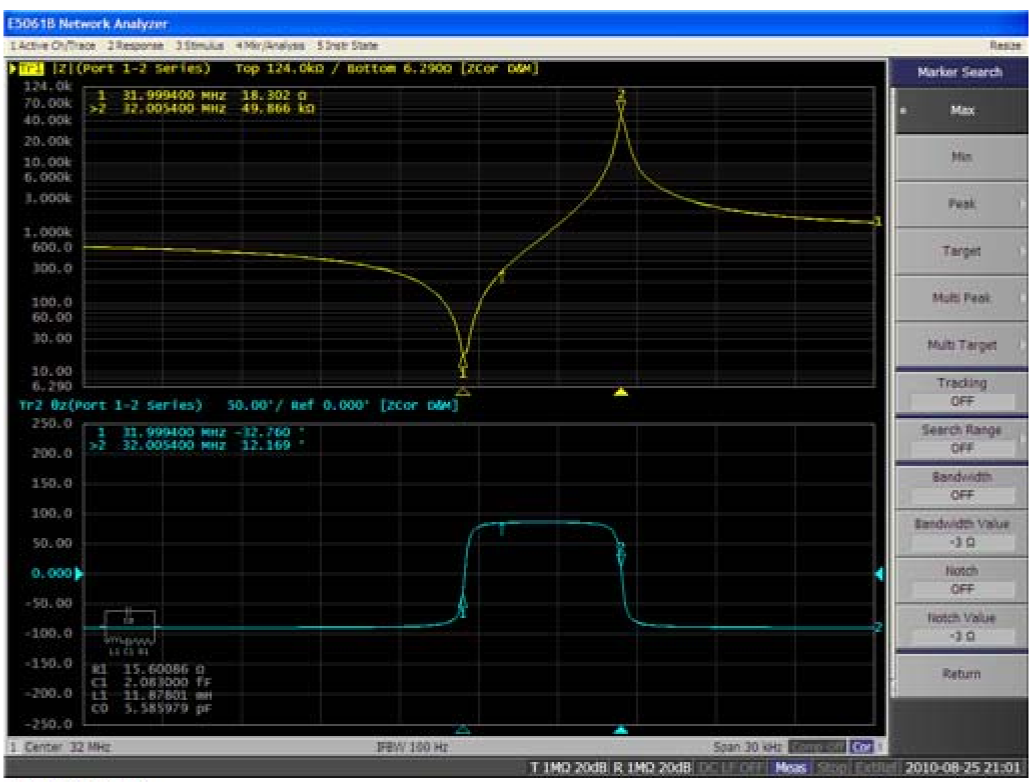
3. Calculate each parameter of the circuit model.

   Calculate. The calculated parameters are displayed in each box of R1, C1 and L1.

4. You can simulate the frequency characteristics by using the approximate value obtained from the above calculation.

   Analysis > **Equivalent Circuit** > **Simulate**

Measurement Result



e5061b108

Measurement Example of a Capacitor (Port 1-2 / Shunt-Through)

This section describes how to measure the frequency characteristics of a capacitor using the Series-Through method on ports 1 and 2.

In this example, the following items are used.

Description	Product / Agilent Part Number	Note
Test Fixture with DUT (Capacitor)	PC board user fixture	-
Test Fixture with Open or Short	PC board user fixture	-
Calibration Kit	85033E	Mechanical Calibration Kit, DC to 9 GHz, 3.5 mm
Cable and Adapter	Cables and adapters for Type-N or 3.5 mm	Connect your PC board to ports 1 and 2 of the E5061B

To measure another device under test (DUT), change the measurement conditions to suit the particular DUT.

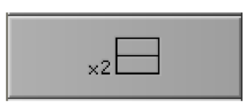
STEP 1. Determining Measurement Conditions

1. Preset the E5061B.

   Preset > **OK**

2. Set the number of traces at two and display each trace in one frame.

   Display > **Num of Traces** > **2**

   Display > **Allocate Traces** > **x2** 

3. Set the measurement port to Gain-Phase.

   Meas > **Measurement Port** > **S-Parameter**

4. Set the method to Port 1-2 Shunt-Through configuration.

   Meas > Impedance Analysis Menu > Method > Port 1-2 Shunt

5. Set the measurement parameter at |Z| for the trace 1 and θ type of each trace.

   Select Trace 1 as the active trace. **Meas** > **Impedance Analysis Menu** > **|Z|**

   Select Trace 2 as the active trace. **Meas** > **Impedance Analysis Menu** > **θz**

6. Specify the center and span frequencies to observe the frequency characteristic. In this example, the start is set at 100 kHz and stop is set at 1 GHz

   Start > **1** > **0** > **0** > **k/m**
   Stop > **1** > **G/n**

   NOTE When entering the frequency unit using the keyboard, type "G" for GHz, "M" for MHz, and "k" for kHz.

5. Set the power level at 0 dBm (224 mV @ 50 Ω).

   Sweep Setup > **Power** > **0** > **x1**

6. Set the sweep type at Log.

   Sweep Setup > **Sweep Type** > **Log Freq**

7. Set the IF Bandwidth at 100 Hz.

   Avg > **IF Bandwidth** > **1** > **0** > **0** > **x1**

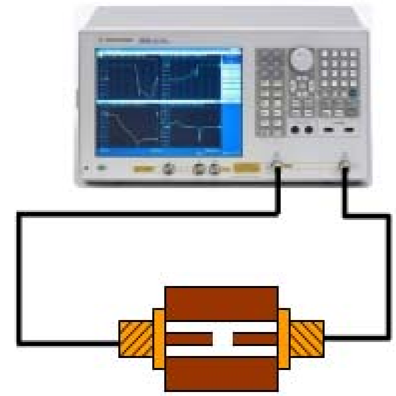
STEP 2. Calibration

1. Select the calibration kit for 85033E.

   Cal > **Cal Kit** > **85033E**

2. Connect the two 3.5 mm cables on both ports 1 and 2.
3. Perform Full 2-Port Calibration at the end of each cables.

4. Connect the fixture with open (or short) between the cables.

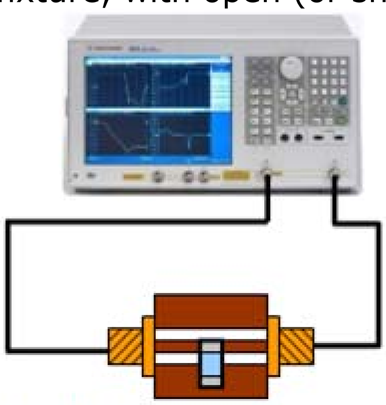


e5061b078

5. Perform Auto Port Extension.

STEP 3. Connecting the Device Under Test (DUT)

1. Connect the fixture with DUT between the cables instead of the fixture, with open (or short).



e5061b079

2. Set the log scale for Trace 1.

   Select Trace 1 as the active trace. **Scale** > **Y-Axis** > **Log**

3. Set the appropriate scale for both traces by executing the auto scale.

   Scale > **Auto Scale All**

STEP 4. Analyzing Measurement Results

This section describes how to use the Equivalent circuit analysis.

Using Equivalent Circuit Analysis

1. Select the Equivalent circuit model.

   Analysis > **Equivalent Circuit** > **Select Circuit** > **E**.

2. Turn the Equivalent Circuit Display ON.

   Analysis > **Equivalent Circuit** > **Display**

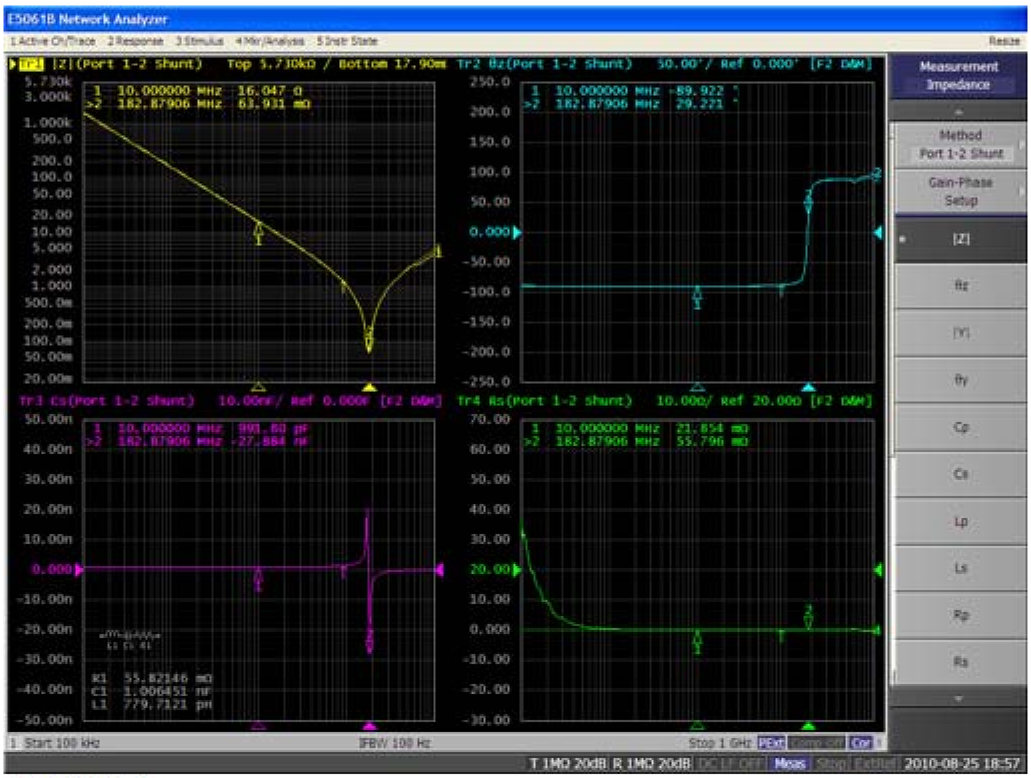
3. Calculate each parameter of the circuit model.

   Calculate. The calculated parameters are displayed in each box of R1, C1 and L1.

4. You can simulate the frequency characteristics by using the approximate value obtained from the above calculation.

   Analysis > **Equivalent Circuit** > **Simulate**

Measurement Result



e5061b109

Measurement Example of a Ceramic Resonator (Gain-Phase / Series-Through)

This section describes how to measure the frequency characteristics of a ceramic resonator using the Series-Through method on Gain-Phase ports using Agilent test fixture. You can connect Agilent 4 terminal pair type fixture on the Gain-Phase ports.

In this example, the following items are used.

Description	Product / Agilent Part Number	Note
Test Fixture	Agilent 16047E	-
Shorting Bar	16047-00621	Furnished with the 16047E
Leaded Load	5012-8646	50 Ω leaded resister. Furnished with E5061B option 720.
DUT	Ceramic resonator	-

To measure another device under test (DUT), change the measurement conditions to suit the particular DUT.

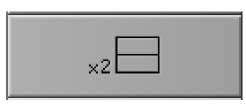
STEP 1. Determining Measurement Conditions

1. Preset the E5061B.

   Preset > **OK**

2. Set the number of traces at two and display each trace in one frame.

   Display > **Num of Traces** > **2**

   Display > **Allocate Traces** > **x2** 

3. Set the measurement port to Gain-Phase.

   Meas > **Measurement Port** > **Gain-Phase**

4. Set the method to Series-Through configuration.

   Meas > Impedance Analysis Menu > Method > GP Series T 50Ω, R 1MΩ

5. Set the measurement parameter at |Z| for the trace 1 and θ type of each trace.

Select Trace 1 as the active trace. **Meas** > **Impedance Analysis Menu** > **|Z|**

Select Trace 2 as the active trace. **Meas** > **Impedance Analysis Menu** > **θz**

6. Specify the center and span frequencies to observe the frequency characteristic. In this example, the center is set at 395 kHz and span is set at 40 kHz.

   Center > **3** > **9** > **5** > **k/m**
   Span > **4** > **0** > **k/m**

   NOTE When entering the frequency unit using the keyboard, type "G" for GHz, "M" for MHz, and "k" for kHz.

5. Set the power level at 0 dBm (224 mV @ 50 Ω).

   Sweep Setup > **Power** > **0** > **x1**

6. Set the sweep type at Linear.

   Sweep Setup > **Sweep Type** > **Lin Freq**

7. Set the IF bandwidth at AUTO.

   Avg > **IF BW Auto** .

8. Set the IFBW auto limit at 100 Hz.

   Avg > **IFBW Auto Limit** > **1** > **0** > **0** > **x1**

STEP 2. Calibration

1. Select the calibration kit for leaded 50 Ω.

   Cal > **Cal Kit** > **Leaded 50ohm**

2. Connect Agilent 16047E test fixture on the ports R, T and LF Out.
3. Set the open state of the 16047E.
   - Fasten the electrode securing screws at the HIGH and LOW sides without the DUT attached
4. Measure the calibration data for open.

   Cal > **Calibrate** > **Impedance Calibration** > **Open**

5. Set the short bar on the 16047E.

- Loosen the electrode securing screws so that the shorting bar is caught in the electrodes.
- Fasten the electrode securing screws.

6. Measure the calibration data for short.

   Short

7. Set the load on the 16047E.
   - Loosen the electrode securing screws so that the load (leaded register) device is caught in the electrodes.
   - Fasten the electrode securing screws.
8. Measure the calibration data for load.

   Load

9. Finalize the calibration measurement. The calibration factor is calculated based on the calibration data acquired, and the error correction is turned ON. **Cor** is displayed at the bottom of the channel window.

Done

STEP 3. Connecting the Device Under Test (DUT)

1. Set the ceramic resonator on the 16047E.
2. Set the log scale for Trace 1.

   Select Trace 1 as the active trace. **Scale** > **Y-Axis** > **Log**

3. Set the appropriate scale for both traces by executing the auto scale.

   Scale > **Auto Scale All**

STEP 4. Analyzing Measurement Results

This section describes how to use the marker function to read out the resonant point and the Equivalent circuit analysis.

Reading the values of resonant points

1. Display a marker.

   Marker > **Marker 1**

2. Search the minimum point.

   Marker Search > **Min**

Reading the values of anti-resonant points

1. Display a second marker.

   Marker > **Marker 2**

2. Search the maximum point.

   Marker Search > **Max**

Using Equivalent Circuit Analysis

1. Select the Equivalent circuit model.

   Analysis > **Equivalent Circuit** > **Select Circuit** > **E**.

2. Turn the Equivalent Circuit Display ON.

   Analysis > **Equivalent Circuit** > **Display**

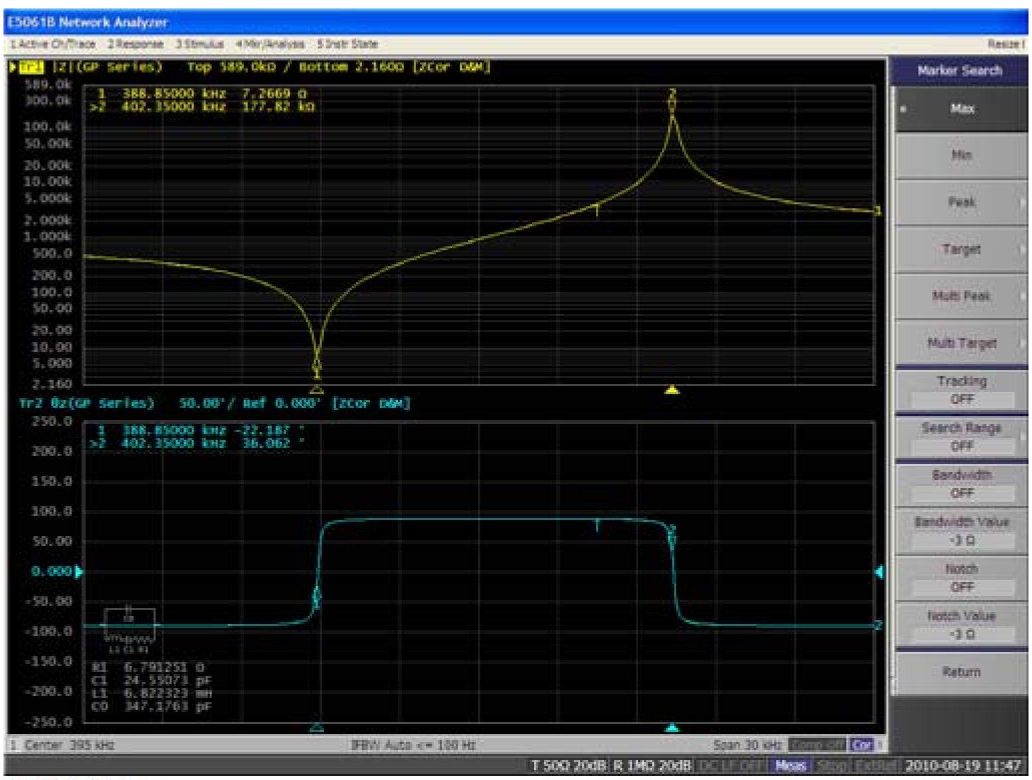
3. Calculate each parameter of the circuit model.

   Calculate. The calculated parameters are displayed in each box of R1, C1 and L1.

4. You can simulate the frequency characteristics by using the approximate value obtained from the above calculation.

   Analysis > **Equivalent Circuit** > **Simulate**

Measurement Result



e5061b110

Measurement Example of a Capacitor (Gain-Phase/Shunt-Through)

This section describes how to measure the frequency characteristics of a capacitor by using the Shunt-Through method on Gain-Phase ports.

In this example, the following items are used.

Description	Product/Agilent Part Number	Note
PC Board with DUT	PC board user fixture	-
PC Board	PC board user fixture	-
Power Splitter	11617L	DC to 2 GHz Power Splitter with BNC connector.
DUT (Capacitor)	Capacitor	-
Cable and Adapter	Cables and adapters for BNC	Connect your PC board to LF Out/T and R connector of the E5061B.

To measure another device under test (DUT), change the measurement conditions to suit the particular DUT.

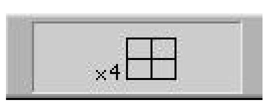
STEP 1. Determining Measurement Conditions

1. Preset the E5061B.

   Preset > **OK**

2. Set the number of traces at two and display each trace in one frame.

   Display > **Num of Traces** > **4**

   Display > **Allocate Traces** > **x4** 

3. Set the measurement port to Gain-Phase.

   Meas > **Measurement Port** > **Gain-Phase**

4. Set the method to Shunt-Through configuration.

   Meas > **Impedance Analysis Menu** > **Method** > **GP Shunt T 50Ω, R 50Ω**

5. Set the measurement parameter at |Z| for the trace 1 and θ type for each trace.

   Select Trace 1 as the active trace. **Meas** > **Impedance Analysis Menu** > **|Z|**.

   Select Trace 2 as the active trace, then click **θz.**

   Select Trace 3 as the active trace, then click **Cs**.

   Select Trace 4 as the active trace, then click **Rs**.

6. Specify the center and span frequencies to observe the frequency characteristic. In this example, the start is set at 5 Hz and stop is set at 30 MHz.

   Start > **5** > **x1**
   Stop > **3** > **0** > **M/μ**

   NOTE When entering the frequency unit using the keyboard, type "G" for GHz, "M" for MHz, and "k" for kHz.

5. Set the power level at -10 dBm (70.7 mV @ 50 Ω).

   Sweep Setup > **Power** > **-** > **1** > **0** > **x1**

6. Set the sweep type at Linear.

   Sweep Setup > **Sweep Type** > **Log Freq**

7. Set the IF bandwidth at AUTO.

   Avg > **IF BW Auto** .

8. Set the IFBW auto limit at 100 Hz.

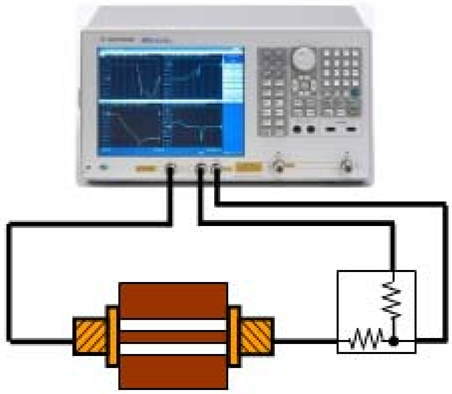
   Avg > **IFBW Auto Limit** > **1** > **0** > **0** > **x1**

STEP 2. Calibration

1. Select the calibration kit for leaded 50 Ω.

   Cal > **Cal Kit** > **Leaded 50ohm**

2. Connect the through fixture on the ports R, T and LF Out as shown below.



e5061b106

3. Measure the response calibration data.

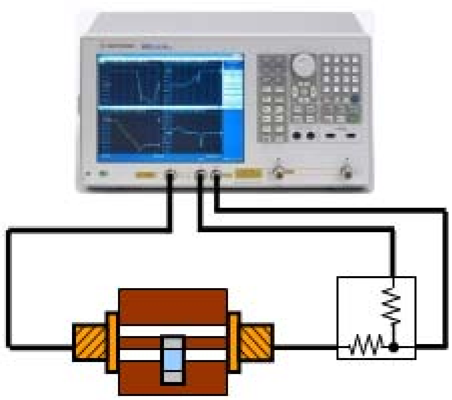
   Cal > **Calibrate** > **Response (Thru)** > **Thru**

4. Finalize the calibration measurement. The calibration factor is calculated based on the calibration data acquired, and the error correction is turned ON. **Cor** is displayed at the bottom of the channel window.

   Done

STEP 3. Connecting the Device Under Test (DUT)

1. Set the PC board with DUT.



e5061b105

2. Set the log scale for Trace 1.

   Select Trace 1 as the active trace. **Scale** > **Y-Axis** > **Log**

3. Set the appropriate scale for both traces by executing the auto scale.

   Scale > **Auto Scale All**

STEP 4. Analyzing Measurement Results

This section describes how to use the marker function to read out the resonant point and the Equivalent circuit analysis.

Using Equivalent Circuit Analysis

1. Select the Equivalent circuit model.

   Analysis > **Equivalent Circuit** > **Select Circuit** > **D**.

2. Turn ON the Equivalent Circuit Display option.

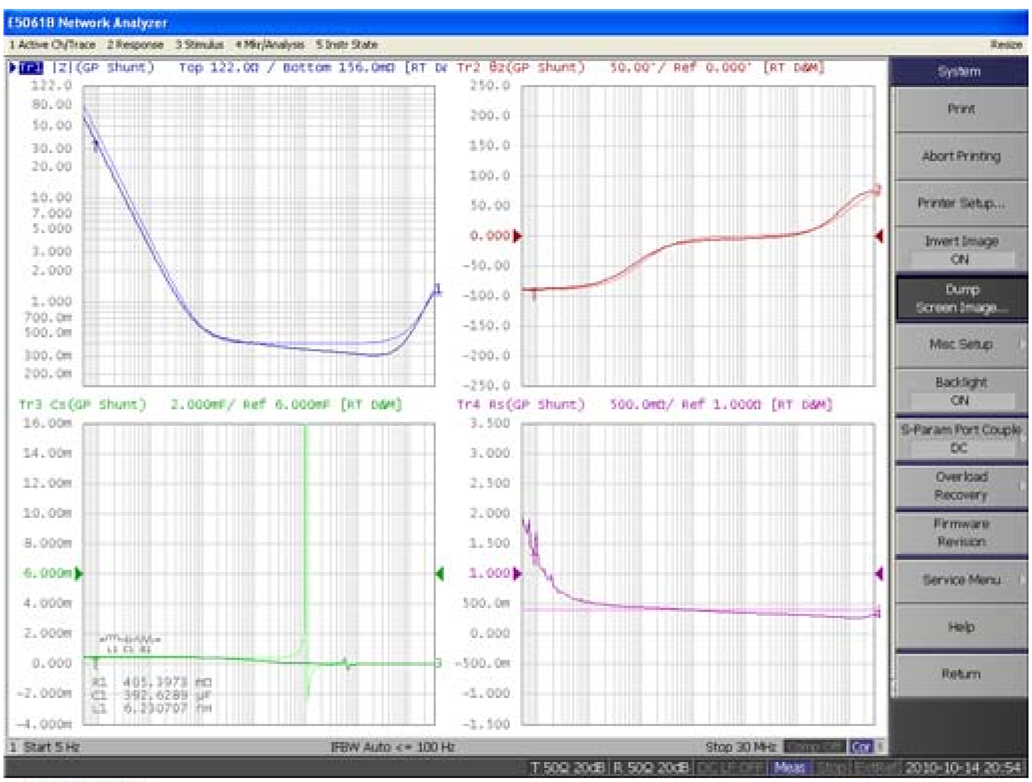
   Analysis > **Equivalent Circuit** > **Display**

3. Calculate each parameter of the circuit model.

   Calculate. The calculated parameters are displayed in each box of R1, C1 and L1.

4. You can simulate the frequency characteristics by using the approximate value obtained from the above calculation.

   Analysis > **Equivalent Circuit** > **Simulate**



e5061b093

Measurement

Measurement

This chapter provides information related to the measurements done through E5061B.

- Setting Measurement Conditions
- Calibration
- Making Measurements
- Data Analysis
- Data Output
- Optimizing Measurements
- Measurement Examples

Setting Measurement Conditions

Setting Measurement Conditions

- Initializing Parameters
- Setting the System Z0
- Setting Channels and Traces
- Selecting Measurement Parameters
- Setting Stimulus Conditions
- Applying DC bias
- Selecting a Data Format
- Setting the Scales
- Setting Window Displays
- Setting Port Coupling

Initializing Parameters

The E5061B has three different initial settings as shown below.

Initial setting	Restore method
Preset state	Press **Preset** > **OK** the front panel or Execute the :SYST:PRES command
*RST state	Execute the *RST command
Factory default setting	E5061B factory (default) settings

The user can set items to be preset freely. For more information, see Setting the user preset function.

Other topics about Setting Measurement Conditions

Setting the System Z0

The procedure for setting the system characteristic impedance (Z_0) is as follows:

1. Press **Cal** key.
2. Click **Set Z0**, then input the system Z_0.

Other topics about Setting Measurement Conditions

Setting Channels and Traces

- Overview of Channel and Trace
- Number of Channels Traces
- Setting Channel Display (Layout of Channels)
- Setting Trace Display
- Active Channel

Other topics about Setting Measurement Conditions

Overview of Channel and Trace

The E5061B allows you to setup multiple channels to perform measurement under different stimulus conditions.

As multiple traces (measurement parameters) can be displayed for each channel, no feature is provided to link the stimulus conditions between channels, and each channel is always independent of the others. In other words, you need to set the measurement conditions and execute calibration for each channel you use for measurement.

When you set items whose setting target is channels/traces (refer to Parameter setting for each setup item), the target is the selected (active) channel/trace. You can specify only the displayed channels/traces as active channels/traces. Therefore, set the display of channels/traces before setting the measurement conditions.

Setting Parameter for each Setup Item (Analyzer, Channel, Trace)

The following table lists the setting parameters and indicates the setup item (analyzer, channel, or trace) that each parameter controls along with the applicable setup key(s).

Parameter	Controlled Setup Items			Setup Key(s)
	Analyzer	Channel	Trace	
Stimulus Settings				
Sweep range		x		**Start**, **Stop**, **Center**, **Span**
Power, CW frequency		x		**Sweep Setup** > **Power**
Sweep time/Sweep delay time		x		**Sweep Setup** > **Sweep Time, Sweep Delay**

Number of points		x		**Sweep Setup** > **Points**
Segment sweep		x		**Sweep Setup** > **Sweep Type**, **Edit Segment Table**, **Segment Display**
DC Bias		x		**Sweep Setup** > **Sweep Type**
Trigger Settings				
Trigger mode		x		**Trigger** > **Hold/Single/ Continuous**
	x			**Hold All Channels/Continuous Disp Channels**
Trigger source, Trigger Event, Trigger Scope	x			**Trigger** > **Trigger Source**, **Trigger Event**, **Trigger Scope**
Trigger	x	-		**Trigger** > **Restart/Trigger**
Ext Trigger Input, Trigger Delay	x	-		**Trigger** > **Ext Trig Input**, **Trigger Delay**
Ext Trigger Output, Polarity, Position, Pulse Width	x	-		**Trigger** > **Ext Trig Output**, **Polarity**, **Position**, **Pulse Width**
Response Settings				
Measurement parameter			x	**Meas**
Data format			x	**Format**
Scale, Electrical delay, Phase offset			x	**Scale**
Memory trace and data math			x	**Display** > **Display/Data-> Mem/ Data Math**

Equation Editor		x		Display > **Equation Editor/Equation (ON/OFF)**
Window title		x		Display > **Edit Title Label/ Title Label (ON/OFF)**
Graticule label in rectangular form		x		Display > **Graticule Label (ON/OFF)**
Color inversion	x			Display > **Invert Color**
Frequency display	x			Display > **Frequency (ON/OFF)**
Display update	x			Display > **Update (ON/OFF)**
Averaging		x		Avg > **Averaging Restart/ Avg Factor/Averaging (ON/OFF)**
Averaging Trigger	x			Avg > **Avg Trigger (ON/OFF)**
Smoothing			x	Avg > **Smo Aperture/ Smoothing (ON/OFF)**
IF bandwidth, IF BW Auto, IF BW Auto Limit		x		Avg > **IF Bandwidth, IFBW Auto, IFBW Auto Limit**
Calibration		x		Cal
System Impedance	x			Cal > **Set Z0**
Marker			x	Marker, Marker Search, Maker Fctn
Market Table	x			Maker Fctn > **Maker Table**
Analysis				
Time domain, Fault Location,			x	Analysis > **Gating, Fault Location**

SRL		x		**Analysis** > **SRL**
Parameter conversion			x	**Analysis** > **Conversion**
Limit test, Ripple Test, Bandwidth Test			x	**Analysis** > **Limit Test, Ripple Test, Bandwidth Limit**
Saving and recalling data	x			**Save/Recall**
Macro	x			**Macro Setup**, **Macro Run**, **Macro Break**
System				
Printing/Saving display Screen/Beeper/GPIB settings/Network Settings/Date & Time/Key Lock/Backlight/Firmware Revision/Service menu	x			**System**
Preset	x			**Preset**

Number of Channels/Traces

The number of channels and the number of traces are 4. The maximum number of points is 1601.

Setting Channel Display (Layout of Channels)

The measurement result for each channel is displayed in its dedicated window (channel window). You cannot have a single window to display the measurement results from more than one channel. This means that the setting of the window layout determines the number of channels displayed on screen.

NOTE The execution of measurement for each channel does not depend on how the channel is displayed (channels that are not displayed can be measured). For information on executing measurement for each channel (trigger mode and trigger source), refer to Making Measurements.

The procedure for setting the window layout is as follows:

1. Press **Display** > **Allocate Channels**.
2. Press the desired softkey to select the window layout.

Setting Trace Display

Setting the number of traces

Depending on the measurement parameters of the traces displayed for each channel, the sweep necessary for each channel is executed. For more information, refer to Sweep Order in Each Channel.

You specify the trace display by setting the number of traces (upper limit of displayed trace numbers). For example, if you set the number of traces to 3, traces 1 through 3 are displayed.

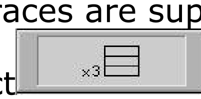
The procedure for setting the number of traces is as follows:

1. Press **Channel Next** (or **Channel Prev**) to select the channel for which you want to set the number of traces.
2. Press **Display** > **Number of Traces**.
3. Press the desired softkey to set the number of traces.

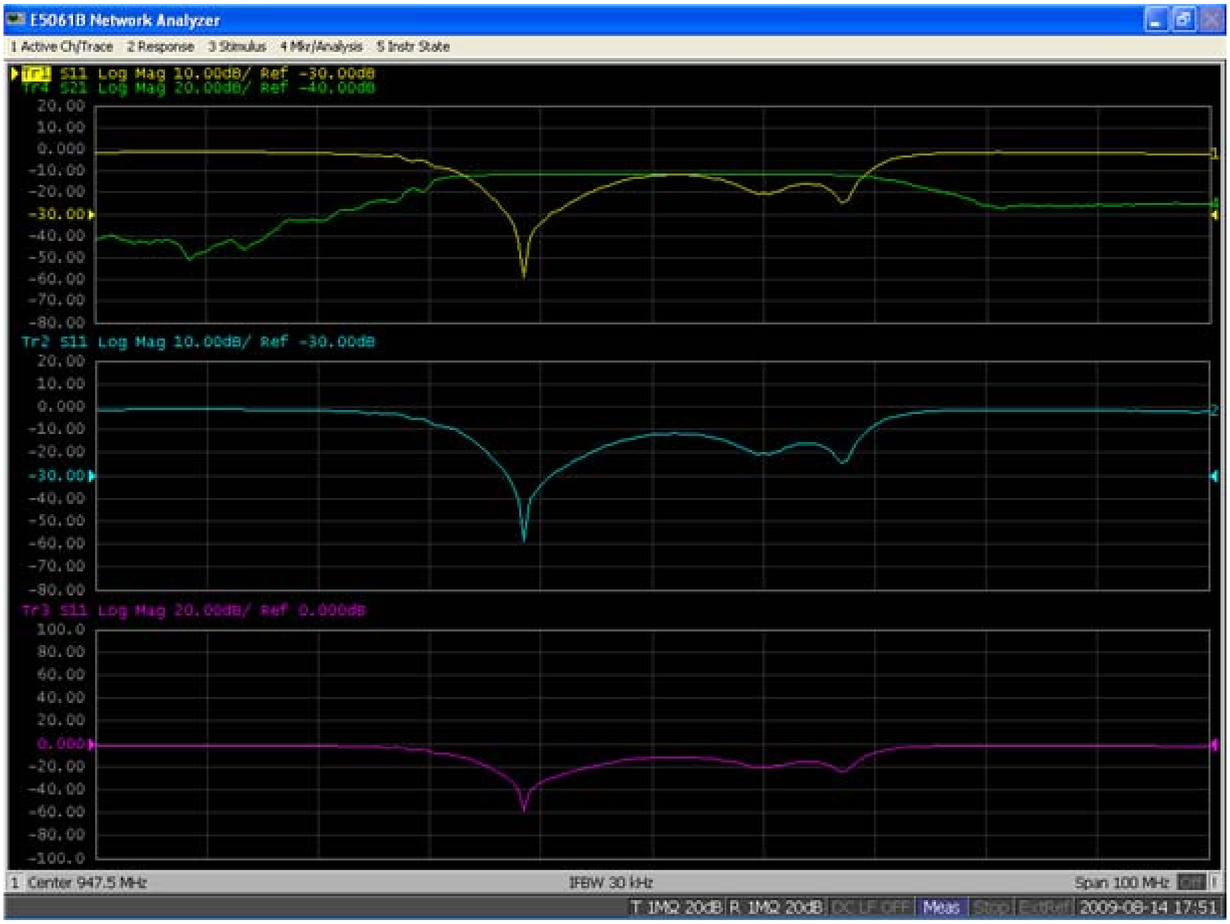
Setting trace layout (graph layout)

Traces are laid out and displayed in the order of the trace number from graph 1 according to the graph layout in the channel window.

You can select the graph layout from the windows layout.

If the number of traces is less than the number of graphs, nothing is displayed in the remaining area. If the number of traces you set exceeds the number of graphs, excess traces are superimposed from the first graph. For example, if you select 

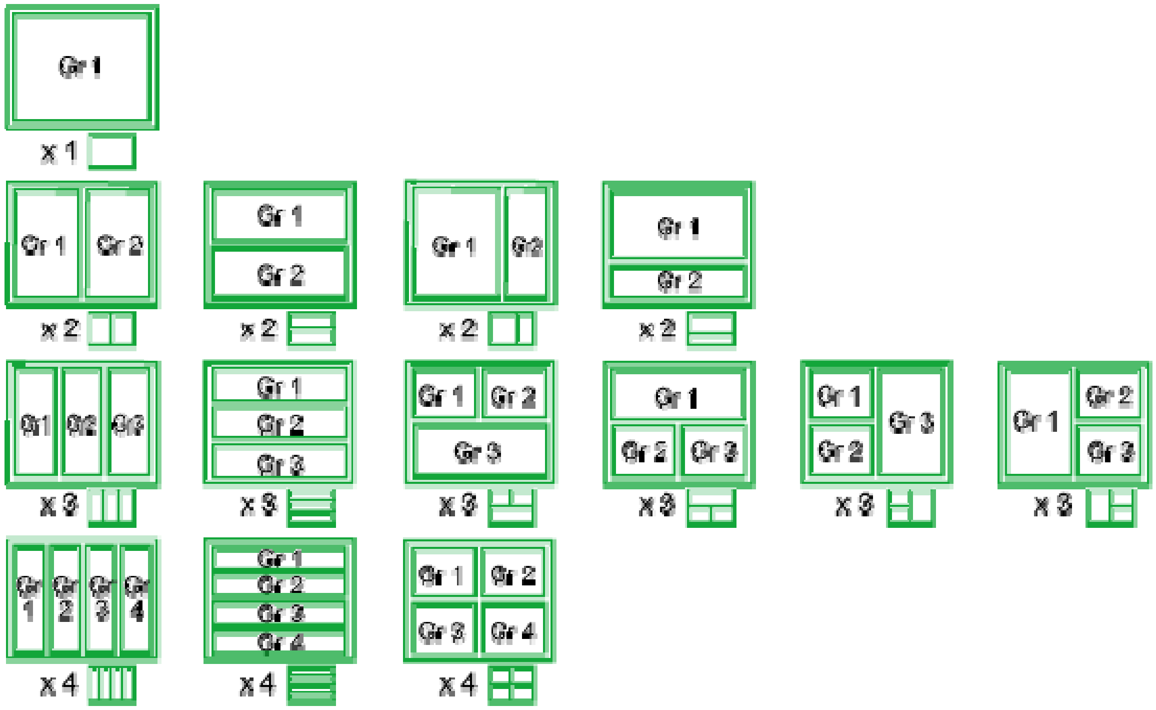
as the graph layout and set the number of traces to 4, graph 1 (Gr1 in Graph layout) display traces 1 and 4, respectively, by superimposing, and graph 2 (Gr2 in Graph layout) and graph 3 (Gr3 in Graph layout) displays trace 2 and trace 3 as shown in the figure below.



The procedure for setting the graph layout is as follows:

1. Press **Channel Next** (or **Channel Prev**) to select the channel for which you want to set the graph layout.
2. Press **Display** > **Allocate Traces**.
3. Press the desired softkey to select the graph layout shown below.

Graph Layout



Active Channel

The active channel is the one whose settings can currently be changed. The window frame of the active channel is displayed brighter than the window frames of the other channels. To change the settings specific to a certain channel, you must first activate the channel.

To change the active channel, use the following hardkeys:

Hardkey	Function
Channel Next	Change the active channel to the next channel with the larger channel number.
Channel Prev	Change the active channel to the previous channel with the smaller channel number.

Active trace

The active trace is the one whose settings can currently be changed. The trace name on the screen (for example, Tr2) of the current active trace is highlighted and indicated with ► to the left. To change the settings specific to a certain trace, you must first activate the trace.

To select the active trace, use the following hardkeys:

Hardkey	Function
Trace Next	Change the active trace to the next trace with the larger trace number.
Trace Prev	Change the active trace to the previous trace with the smaller trace number.

Selecting Measurement Parameters

The E5061B allows users to evaluate the DUT (device under test) characteristics by using the following measurement parameters.

- Select Measurement Port
- S-parameters
- Absolute
- Gain-Phase

Other topics about Setting Measurement Conditions

Select Measurement Port (Option 3L5 Only)

For each channel, the measurement should select either S-parameter or Gain-Phase. This function is available only in option 3L5 (Gain-Phase).

1. Press **Meas** > **Measurement Port.**
2. Select **S-Parameter** or **Gain-Phase**.
3. All traces in the selected channel are set to either S11 or T/R, respectively.

NOTE By using the commands, the parameters of S-parameters (Sxx) and Gain-Phase (T/R, T, R) can exist in one channel. The **Measurement Port** softkey has no equivalent SCPI command.

S-parameters

S-parameters (scattering parameters) are used to evaluate how signals are reflected by and transferred through the DUT. An S-parameter is defined by the ratio of two complex numbers and contains information on the magnitude and phase of the signal. S-parameters are typically expressed as follows:

$S_{out\ in}$

out: port number of the DUT from which the signal is output

in: port number of the DUT to which the signal is input

For example, S-parameter S_{21} is the ratio of the output signal of port 2 on the DUT with the input signal of port 1 on the DUT, both expressed in complex numbers.

1. Press **Channel Next** (or **Channel Prev**) and **Trace Next** (or **Trace Prev**) to select the trace.

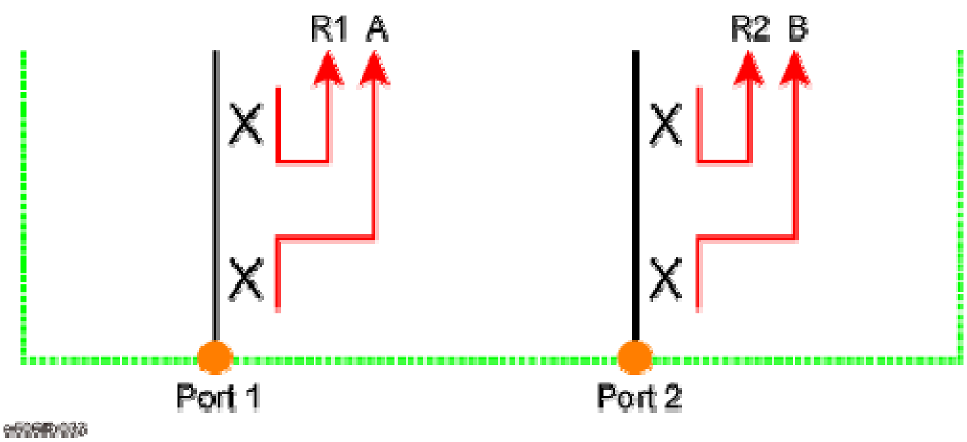
2. Select a softkey that corresponds to the desired measurement parameter. If the desired softkey is gray out, press **Meas** > **Measurement Port** > **S-Parameter**.

Absolute

Absolute shows the absolute power for reference and received signals on the port.

Softkey	Description
A (n)	Absolute measurement in Port 1, test receiver
B (n)	Absolute measurement in Port 2, test receiver
R1 (n)	Absolute measurement in Port 1, reference receiver
R2 (n)	Absolute measurement in Port 2, reference receiver

where n in the parentheses is the stimulus port number. For example, R1(1) means the reference level while the signal is output from the port 1, and A(2) means the received signal level into port 1 while the signal is output from the port 2.



1. Press **Channel Next** (or **Channel Prev**) and **Trace Next** (or **Trace Prev**) to select the trace.
2. Press **Meas** > **Absolute.** If **Absolute** is gray out, press **Meas** > **Measurement Port** > **S-Parameter**.
3. Select a softkey that corresponds to the desired measurement parameter.

Gain-Phase (Option 3L5 Only)

1. Press **Channel Next** (or **Channel Prev**) to select the channel.
2. Press **Meas** > **Measurement Port** > **Gain-Phase.**
3. Press **Trace Next** (or **Trace Prev**) to select the trace.

4. Press **Meas** > **Gain-Phase**. Then select a softkey that corresponds to the desired measurement parameter.

Input Impedance

The input impedance can be selected from 50 Ω or 1 MΩ.

1. Press **Channel Next** (or **Channel Prev**) and **Trace Next** (or **Trace Prev**) to select the trace.
2. Press **Meas** > **Gain-Phase Setup** > **Input Impedance** > **R Input Z** (or **T Input Z**).
3. Select **50 Ω** or **1 MΩ**.

Input Attenuator

The input attenuator can be selected from 0 dB or 20 dB. When your input signal exceeds the signal over -5 dBm (50 Ω input) or 0.18 Vpeak (1 MΩ input), the attenuator should be set at 20 dB.

1. Press **Channel Next** (or **Channel Prev**) and **Trace Next** (or **Trace Prev**) to select the trace.
2. Press **Meas** > **Gain-Phase Setup** > **Input Attenuator** > **R Attenuator** (or **T Attenuator**).
3. Select **0 dB** or **20 dB**.

Setting Stimulus Conditions

You can set the stimulus condition for each channel independently.

- Setting Sweep Type
- Setting Sweep Range
- Enable Stimulus Signal Output
- Setting Power Level
- Setting Fixed Frequency at Power/DC Bias Sweep
- Setting Number of Measurement Points
- Setting Sweep Delay and Sweep Time
  - Measuring in Time Series (Time Sweep)

Other topics about Setting Measurement Conditions

Setting Sweep Type

You can select the sweep type from the following four types.

SoftKey	Description
Linear Freq	Sweeps frequencies in linear scale.
Log Freq	Sweeps frequencies in logarithmic scale.
Segment	Performs a sweep with linear sweep conditions (segments) combined. For more information, refer to Performing a Segment-by-Segment Sweep (segment sweep).
Power Sweep	Sweeps power levels in linear scale.
DC Bias Sweep	Sweeps DC bias levels. This is available only when option is 3L5.

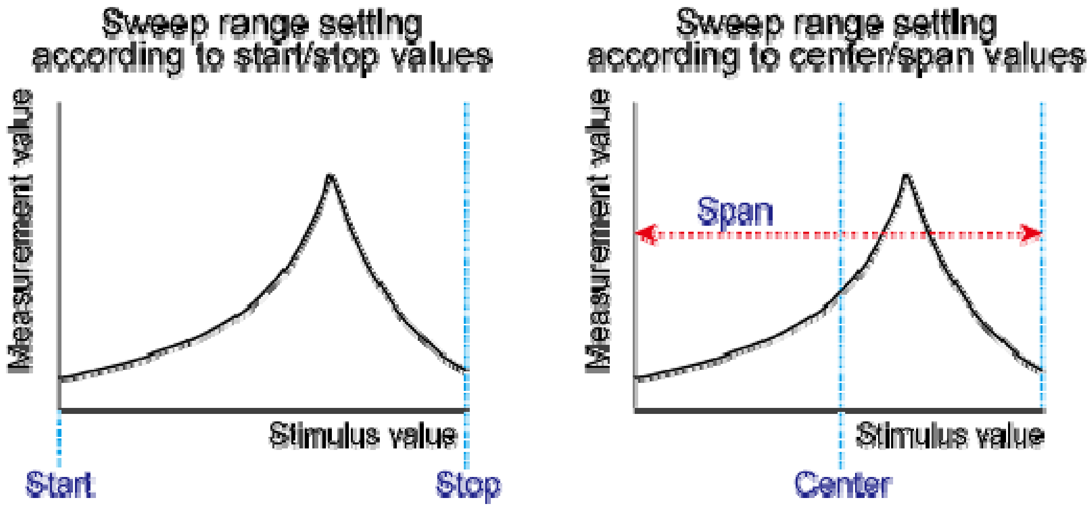
The procedure for selecting the sweep type is as follows:

1. Press **Channel Next** (or **Channel Prev**) to select the channel for which you want to set the sweep type.
2. Press **Sweep Setup** > **Sweep Type**.
3. Press the desired softkey to select the sweep type.

NOTE The time sweep is shown in Measuring in Time Series. This allows you to display the measurement parameter versus time.

Setting Sweep Range

There are two ways to set the sweep range: by specifying the lowest and the highest values and by specifying the center value and a span. Once the sweep range is set, it is possible to change the range by substituting the lowest value, the highest value, or the center value with a value (stimulus value) represented by a marker on the trace.



e50F1e443

Setting the Sweep Range with the Lowest and Highest Values

1. Press **Channel Next** (or **Channel Prev**) to select the channel of which sweep range will be set.
2. Click **Start**, then input the lowest value.
3. Click **Stop**, then input the highest value.

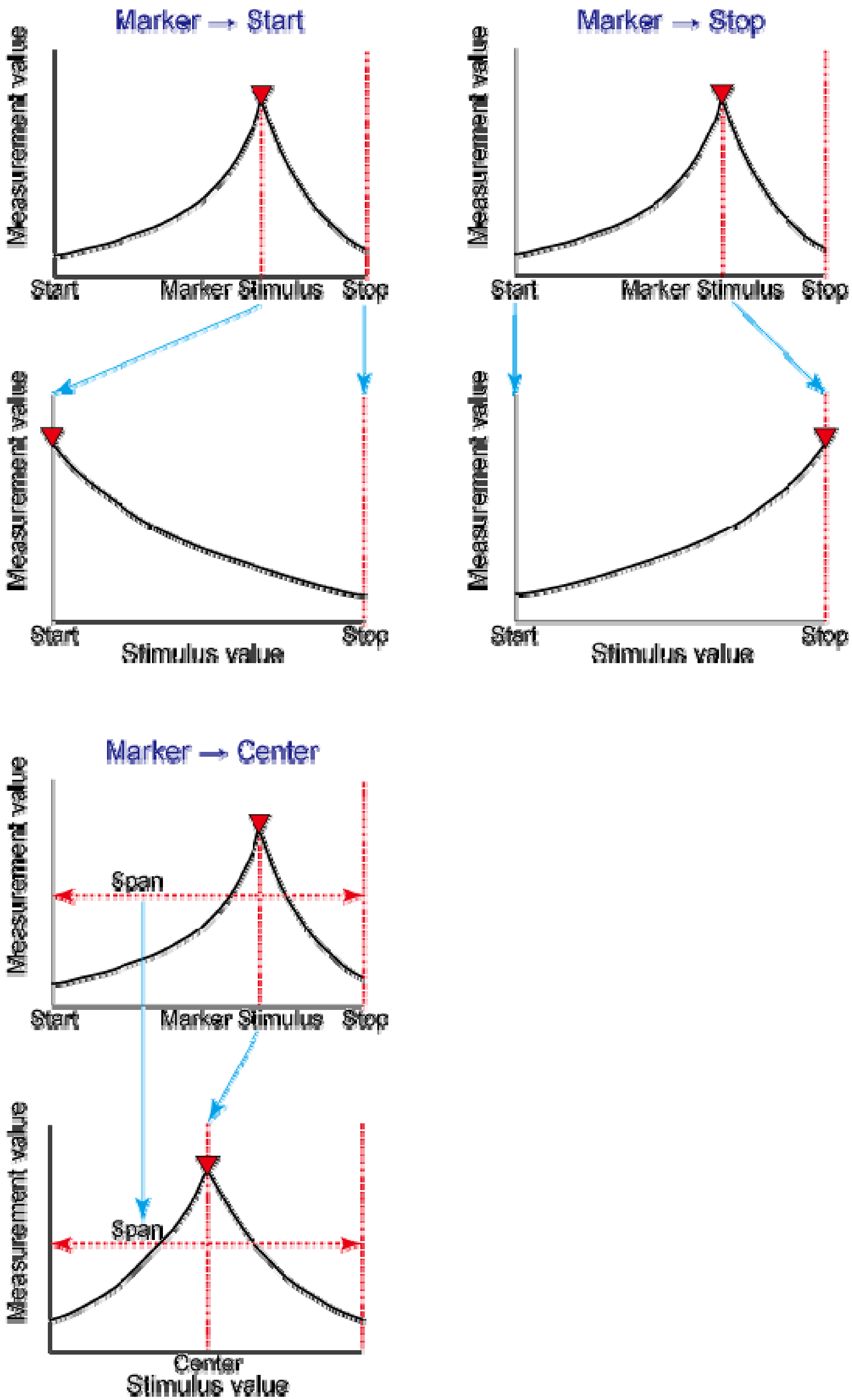
Setting the Sweep Range with the Center Value and a Span

1. Press **Channel Next** (or **Channel Prev**) to select the channel of which sweep range will be set.
2. Click **Center**, then input the center value.
3. Click **Span**, then input the span value.

Setting Sweep Range Using the Marker

1. In the channel window of which range must be set, place the active marker on the active trace to a position that corresponds to the new range (to the lowest, highest, or center value).
2. Press **Marker Fctn**.
3. Click the **softkey** that corresponds to each value.

   NOTE If the reference marker is on and the stimulus value of the active marker is expressed by a value relative to the reference marker, the absolute stimulus value will be used to set the new sweep rang.


Marker → Start
Measurement value
Start
Marker Stimulus
Stop
Measurement value
Start
Stimulus value
Stop
Marker → Stop
Measurement value
Start
Marker Stimulus
Stop
Measurement value
Start
Stimulus value
Stop
Marker → Center
Measurement value
Span
Start
Marker Stimulus
Stop
Measurement value
Span
Center
Stimulus value

e5071c385

Enable Stimulus Signal Output

You can turn on/off the stimulus signal output, but this prevents you from performing the measurement. Therefore, normally this feature will not be used. This feature is mainly used to turn the output ON back after it has been turned OFF by the power trip feature.

Follow these steps to turn the stimulus signal output on/off:

1. Press **Sweep Setup**.
2. Click **Power** > **RF Out** (Each press toggles between on/off).

When set to OFF, "**RF OFF**" is displayed in Instrument Status Bar.

Power trip

The power trip is a feature that the instrument uses to automatically turn OFF the output of the stimulus signal to protect the instrument when a signal of which level exceeds the upper limit is inputted to the test port.

If the power output is automatically turned off by the power trip feature, remove the cause of the over-input and turn ON the power output according to the above steps to restart the measurement.

Setting Power Level

1. Press **Channel Next** (or **Channel Prev**) to select the desired channel.
2. Press **Sweep Setup** key.
3. Click **Power** > **Port Couple**, then select the **on / off setting** of the level coupling for all the ports.

   NOTE The power level of port 1 is coupled with the power level for all ports.

   NOTE If you change the on/off setting of the level coupling, all ports are automatically changed to the same level value as that of port 1.

4. Follow the procedure below according to the Port Couple.

- When setting level for all ports (Port Couple ON)
  a. Click **Power**, then enter the power level.
- When setting level for each port (Port Couple OFF)
  a. Press **Port Power**, then click the softkey corresponding to each port (**Port 1 Power**, **Port 2 Power**, **LF OUT Power**)
  b. Enter the power level.

Correcting attenuation of power level (using power slope feature)

You can use the power slope feature to correct the attenuation of a power level so that it is simply proportional to the frequency (attenuation due to cables and so on), which improves the accuracy of the level actually applied to the DUT.

Turning power slope feature on/off

1. Press **Channel Next** (or **Channel Prev**) to select the desired channel.
2. Press **Sweep Setup** key.
3. Click **Power** > **Slope [OFF]** (**Slope [ON]**). Each press toggles between on/off.

Setting correction coefficient (correction amount per 1 GHz)

1. Press **Channel Next** (or **Channel Prev**) to select the desired channel.
2. Press **Sweep Setup** key.
3. Click **Power** > **Slope [xxx dB/GHz]** ("xxx" represents the current set value.).
4. Enter the correction coefficient using the **ENTRY** block keys on the front panel.

Setting Fixed Frequency at Power / DC Bias Sweep

The procedure for setting the fixed frequency (CW frequency) at the power sweep and DC Bias sweep (Option 3L5 only) is as follows:

1. Press **Channel Next** (or **Channel Prev**) to select the desired channel.
2. Press **Sweep Setup** key.
3. Click **CW Freq**, then enter the fixed frequency.

Setting Number of Measurement Points

The number of points is the number of data items collected in one sweep. It can be set to any number from 2 to 1601 for each channel independently.

- To obtain a higher trace resolution against the stimulus value, choose a larger value for the number of points.
- To obtain higher throughput, keep the number of points to a smaller value within an allowable trace resolution.
- To obtain higher measurement accuracy after calibration, perform calibration using the same number of points as in actual measurements.

1. Press **Channel Next** (or **Channel Prev**) to select the desired channel.
2. Press **Sweep Setup** key.
3. Click **Points**, then input the desired number of points.

Setting Sweep Delay and Sweep Time

Sweep time is the time taken to complete a sweep for each stimulus (source) port. Two modes are available for setting the sweep time: manual sweep time mode and automatic sweep time mode.

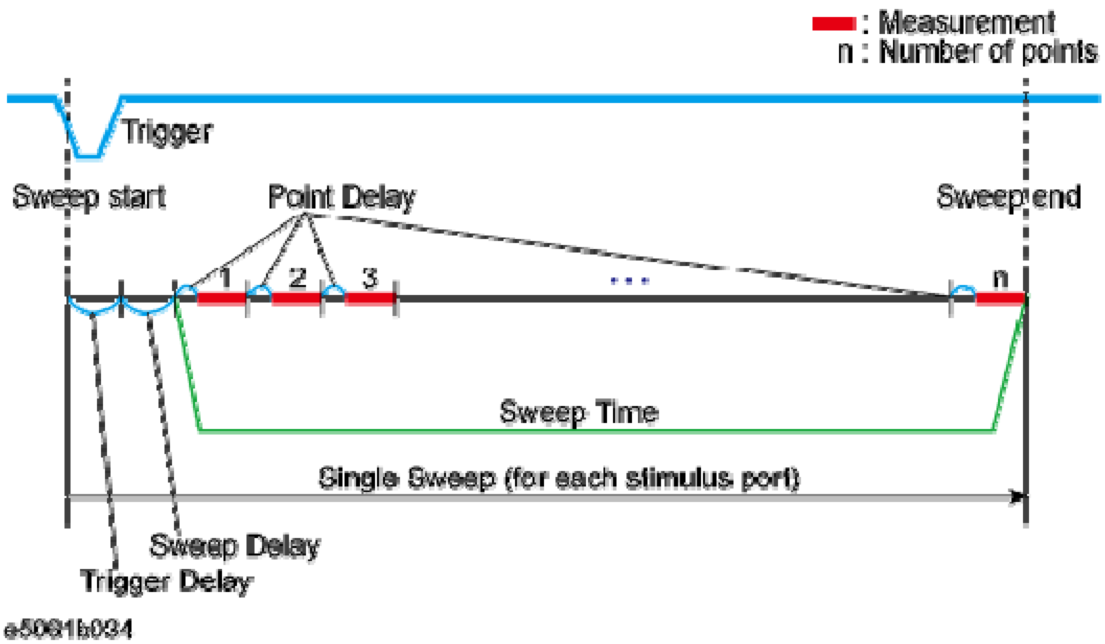
Sweep Time Mode	Description
Manual	In this mode, the sweep time is set manually. Once the sweep time is set, changes in the measurement conditions do not affect the sweep time as long as it is within the analyzer's capability. If the sweep time becomes lower than the analyzer's lower sweep time limit, the sweep time is reset to the shortest time within the conditions. If the sweep time exceeds the analyzer's upper sweep time limit, the sweep time is reset to the longest time within the conditions.
Auto (Default)	The sweep time is always kept to the shortest time possible with the current measurement conditions.

The following figure shows the definitions of the sweep time and the sweep delay time.

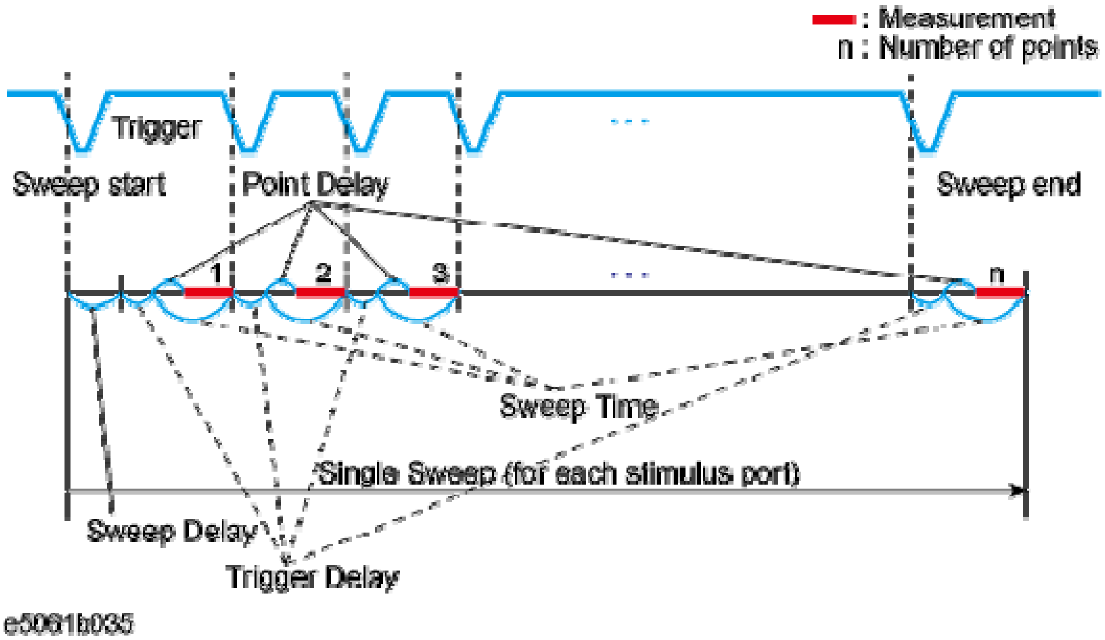
Sweep delay is time before starting a sweep for each stimulus (source) port. Sweep time does not include the sweep delay.

Timing Chart for Sweep

When the trigger mode is set at "On Sweep".



When the trigger mode is set at "On Point".



NOTE Trigger delay is available when the trigger source is set at external.
Sweep time is the total of point delay and measurement time in all measurement points when the trigger mode set at "On Point".

Setting Sweep Delay Time

1. Press **Channel Next** (or **Channel Prev**) to select the desired channel.
2. Press **Sweep Setup** > **Sweep Delay**.
3. Using the **ENTRY** block keys on the front panel, input the desired sweep delay time (in seconds).

Setting Up Sweep Time (Manual Sweep Time Mode)

1. Press **Channel Next** (or **Channel Prev**) to select the desired channel.
2. Press **Sweep Setup** > **Sweep Time Auto** to turn OFF.
3. The softkey named **Sweep Time** is activated. Press **Sweep Time**.
4. Using the **ENTRY** block keys on the front panel, input the desired sweep time (in seconds).

NOTE The sweep time is not correctly displayed in DC Bias sweep at 90 kHz and above when there are the traces for both Gain-Phase and S-Parameter measurements in one channel.

Measuring in Time-Series (Time Sweep)

The following procedure allows you to time sweep and the measurement parameter is displayed versus time.

1. Press **Channel Next** (or **Channel Prev**) to select the channel of which sweep range will be set.
2. Press **Span** > **0** > **x1** to set the span value to 0 (zero span).
3. Press **Center**, then input the desired value (frequency, power, or DC bias).
4. Press **Sweep Setup** > **Sweep Time Auto** to turn OFF.
5. Press **Sweep Time**, then input the duration of the sweep which is displayed on X-axis.
6. Press **Marker** to display the marker 1. The time at the marker shows as the marker position value at the upper left corner on the screen.

Applying DC bias and DC Source (Option 3L5 only)

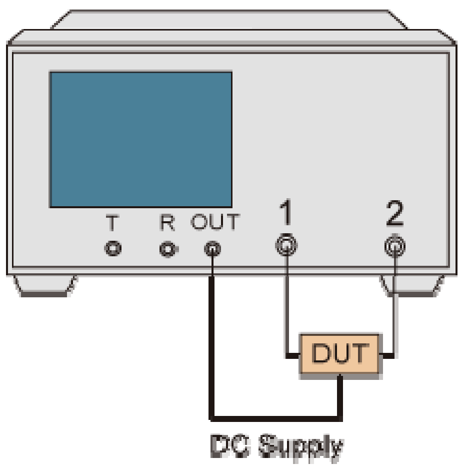
Other topics about Setting Measurement Conditions

Overview

E5061B option 3L5 allows you to apply the DC bias on the source signal of Port 1 or LF output port. The range of DC bias is -40 V to +40 V.

DC Source

When you make S-parameter measurements, the LF output port can be used as the DC supply.



e5061b052

Setting DC Bias

1. Press **Sweep Setup** > **DC Bias Port**, then select **LF Out** or **Port 1** which you want to output the DC bias to.
2. Press **DC Bias Level**, then enter the DC bias level.

Applying DC Bias

1. Press **Sweep Setup** > **DC Bias** to turn it ON.

When an ECal is connected to the E5061B USB port or **Cal** > **Calibrate** is pressed, DC Bias is turned OFF in order to avoiding the calibration kit having damage.

Using as DC Source

1. Setup the S-Parameter measurements.
2. Press **Sweep Setup** > **DC Bias Port** > **LF Out**.

3. Press **DC Bias Level**, then enter the DC level.
4. Press **DC Bias** to turn ON.
5. Make the S-Parameter measurements.

Selecting a Data Format

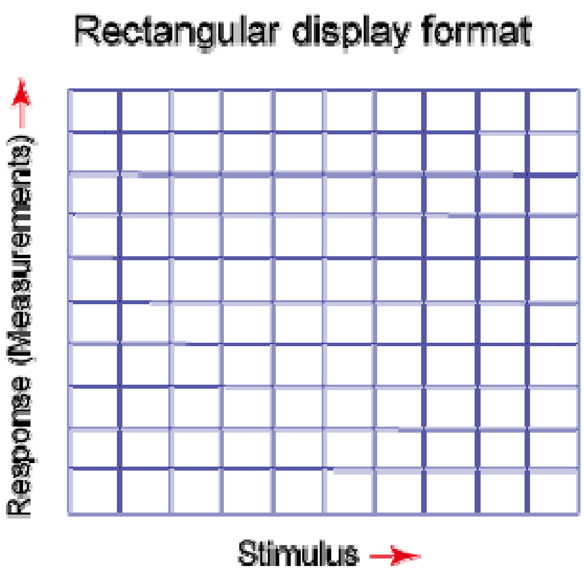
The E5061B allows you to display the measured S-parameters by using the following data formats. The data format can be preset to factory settings using the Preset option.

- Rectangular display formats
- Polar format
- Smith chart format

Other topics about Setting Measurement Conditions

Rectangular display formats

Rectangular display formats draw traces by assigning stimulus values (linear scale) to the X-axis and response values to the Y-axis. Eight different formats are available depending on the selection of data for the Y-axis.



e5071c451

Type	Y-axis Data Type	Y-axis Unit	Application Examples
Log magnitude format	Magnitude	dB	• Return loss measurement • Insertion loss measurement (or gain measurement)
Phase	Phase (displayed in range from -180 °	Degrees (	Measurement of deviation

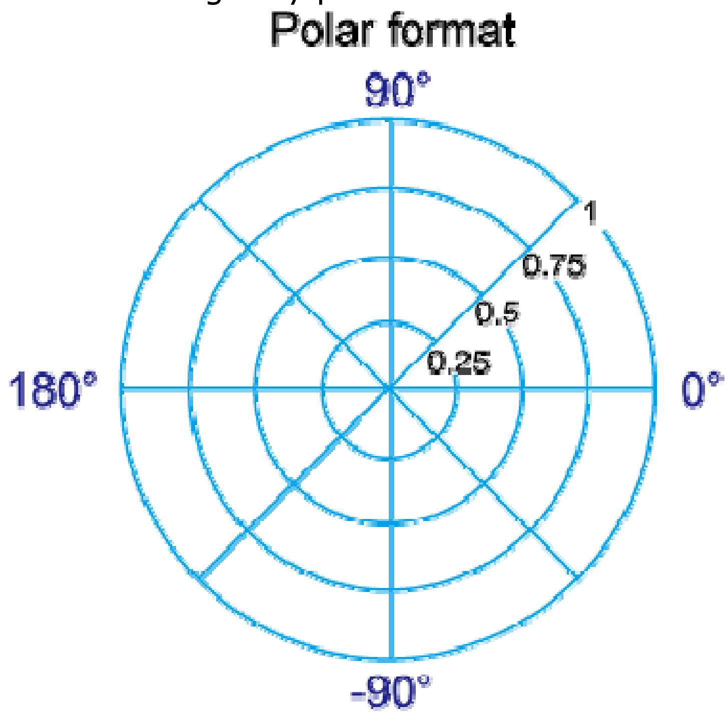
format	to +180 °)	°)	from linear phase
Expanded phase format	Phase (can be displayed above +180 ° and below -180 °)	Degrees (°)	Measurement of deviation from linear phase
Positive phase format	Phase (displayed in range from 0 ° to +360 °)	Degrees (°)	Measurement of deviation from linear phase
Group delay format	Signal transfer delays within the DUT	Seconds (s)	Group delay measurement
Linear magnitude format	Magnitude	(Abstract number)	Reflection coefficient measurement
SWR format	$\frac{1+\rho}{1-\rho}$ (ρ: reflection coefficient)	(Abstract number)	Measurement of standing wave ratio
Real format	Real part of measured complex parameter	(Abstract number)	
Imaginary format	Imaginary part of the measured complex parameter	(Abstract number)	

Polar format

In the polar format, traces are drawn by expressing the magnitude as a displacement from the origin (linear) and phase in an angle counterclockwise from the positive X-axis. This data format does not have a stimulus axis, so frequencies must be read by using the marker. The polar format allows users to select one of the following three data groups to display the marker response values.

- Linear magnitude and phase (°)
- Log magnitude and phase (°)

- Real and imaginary parts

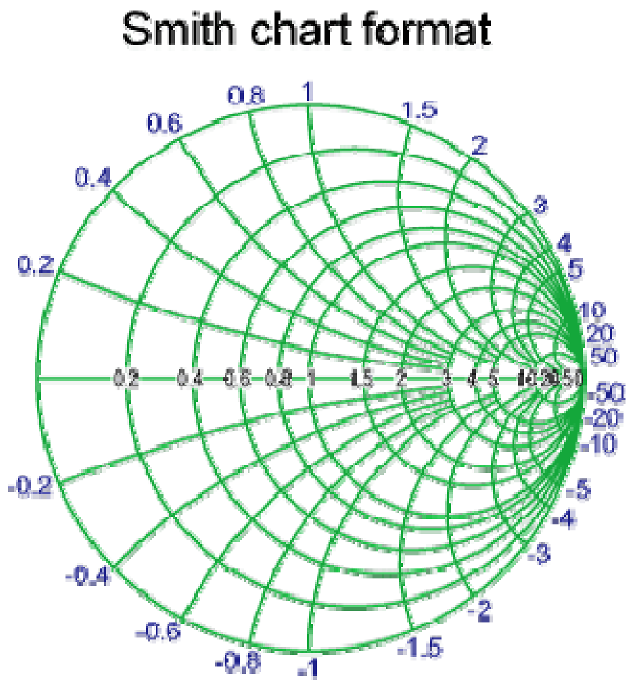


e5071c382

Smith chart format

The Smith chart format is used to display impedances based on reflection measurement data of the DUT. In this format, traces are plotted at the same spots as in the polar format. The Smith chart format allows users to select one of the following five data groups to display the marker response values.

- Linear magnitude and phase (°)
- Log magnitude and phase (°)
- Real and imaginary parts
- Resistance (ohm), Reactance (ohm), and inductance (H) or capacitance (F)
- Conductance (S), susceptance (S), and capacitance (F) or inductance (H)



e5071c383

Use the following procedure to select a data format:

1. Press **Channel Next** (or **Channel Prev**) and **Trace Next** (or **Trace Prev**) to select the trace for which the data format will be set.
2. Press **Format.**
3. Press the softkey that corresponds to the desired data format.

Softkey	Function
Log Mag	Selects the log magnitude format
Phase	Selects the phase format
Group Delay	Selects the group delay format
Smith > Lin / Phase	Selects the Smith chart format (with linear magnitude and phase as the marker response values)
Smith > Log / Phase	Selects the Smith chart format (with log magnitude and phase as the marker response

	values)
Smith > **Real / Imag**	Selects the Smith chart format (with the real and imaginary parts as the marker response values)
Smith > **R + jX**	Selects the Smith chart format (with resistance and reactance as the marker response values)
Smith > **G + jB**	Selects the Smith chart format (with conductance and susceptance as the marker response values)
Polar > **Lin / Phase**	Selects the polar format (with linear magnitude and phase as the marker response values)
Polar > **Log / Phase**	Selects the polar format (with log magnitude and phase as the marker response values)
Polar > **Real / Imag**	Selects the polar format (with the real and imaginary parts as the marker response values)
Lin Mag	Selects the linear magnitude format
SWR	Selects the SWR (standing wave ratio) format
Real	Selects the real format
Imaginary	Selects the imaginary format
Expand Phase	Selects the expanded phase format
Positive Phase	Selects the positive phase format

Setting the Scales

- Auto Scale
- Manual Scale Adjustment
- Setting Reference Line Value Using Marker

Other topics about Setting Measurement Conditions

Auto Scale

The auto scale function is used to tailor each scale (scale/division and the reference line value) automatically in such a way that traces will appear at the proper size on the screen for easy observation.

NOTE The scale data can be preset to factory settings using the Preset option.

Single Trace Auto Scale

Follow the procedure below to perform the auto scale function on a specific trace.

1. Press **Channel Next** (or **Channel Prev**) and **Trace Next** (or **Trace Prev**) to select the trace of which the auto scale function will be performed.
2. Press **Scale** > **Auto Scale**.

Auto Scale on All Traces Within a Channel

1. Press **Channel Next** (or **Channel Prev**) to select the channel of which the auto scale function will be performed.
2. Press **Scale** > **Auto Scale All**.

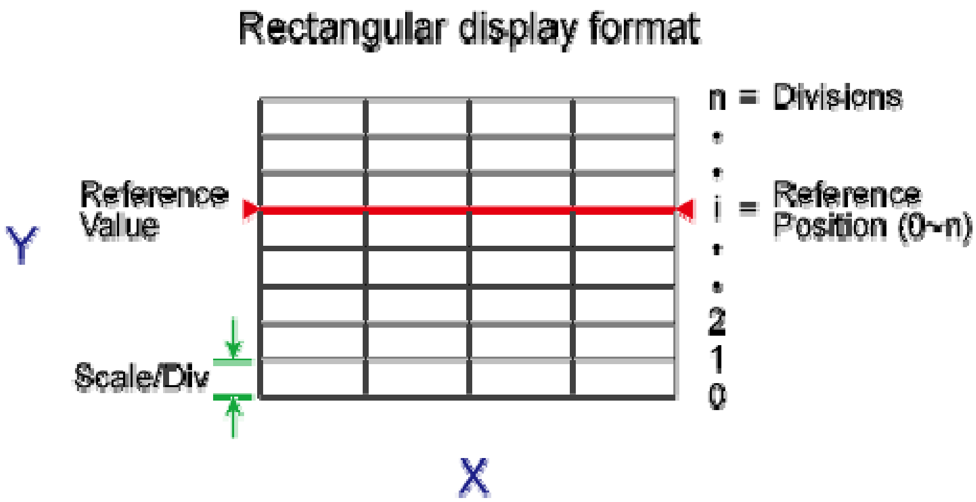
Manual Scale Adjustment

Manual scale adjustment on the rectangular display format

For a rectangular display format, four parameters are used to manually adjust the scales.

Adjustable feature	Description
Divisions	Defines the number of divisions on the Y-axis. An even number from 4 to 30 must be used. Once set, it is commonly applied to all traces displayed in any rectangular format within that channel.
Scale/Division (Scale/Div)	Defines the number of increments per division on the Y-axis. The value applies only to the active trace.
Reference	Defines the position of the reference line. The position must

position	be specified using the number assigned to each division on the Y-axis starting at 0 (the least significant) running up to the number of divisions being used (the most significant). The position applies only to the active trace.
Reference line value (Reference Value)	Defines the value corresponding to the reference line. It must be set using the unit on the Y-axis. The reference line value applies only to the active trace.



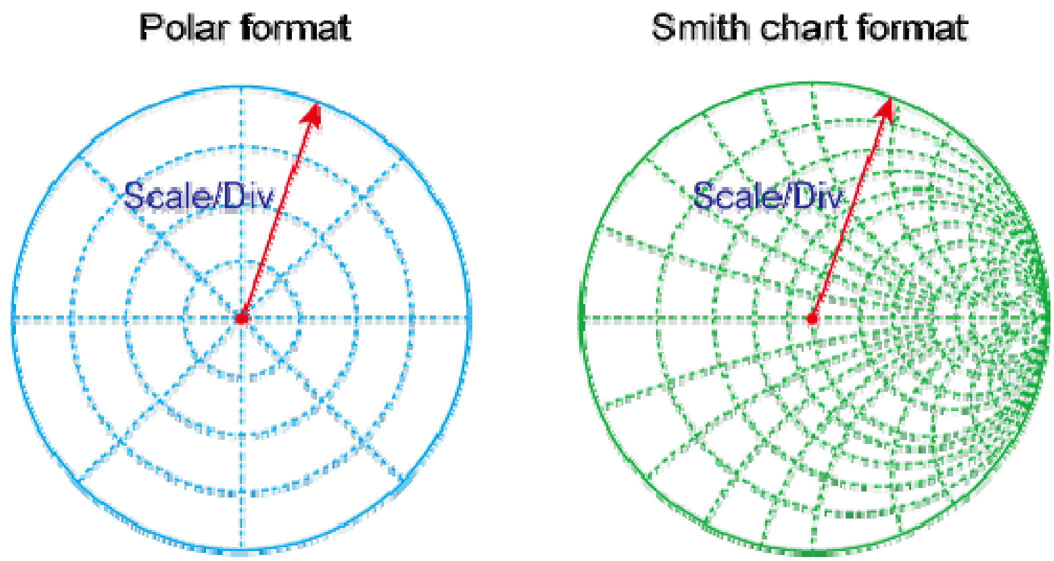
e5071c385

1. Press **Channel Next** (or **Channel Prev**) and **Trace Next** (or **Trace Prev**) to select the trace of which scale features will be adjusted.
2. Press **Scale**.
3. Select the **softkey** that corresponds to the particular feature that needs to be adjusted.

NOTE It is also possible to turn off the display of graticule labels. For details, refer to Turning off the display of graticule labels.

Manual scale adjustment on the Smith chart/polar format

Manual scale adjustment on the Smith chart format or the polar format is done by using the displacement (**Scale/Div** of the outermost circle).



e5071c384

1. Press **Channel Next** (or **Channel Prev**) and **Trace Next** (or **Trace Prev**) to select the trace for which the scale will be adjusted.
2. Press **Scale**.
3. Click **Scale/Div**, then input the displacement of the outermost circle.

Setting Reference Line Value using Marker

When using a rectangular display format, it is possible to change the reference line value to be equal to the response value of the active marker on the active trace.

1. Place the active marker on the active trace on the position that corresponds to the new reference line value.
2. Press **Scale** or **Marker Fctn**.
3. Click **Marker -> Reference** to change the reference line value to the marker response value.

NOTE If the reference marker is ON and the stimulus value of the active marker is expressed using a value relative to the reference marker, the absolute stimulus value is used to set the new reference line value.

Setting Window Displays

- Maximizing the specified window trace display
- Turning off the display of graticule labels
- Hiding Frequency Information
- Labeling a Window
- Setting display colors
- Setting display magnification
- Resizing the screen

Other topics about Setting Measurement Conditions

Maximizing the specified window / trace display

When using multiple channels, it is possible to maximize a specific channel window on the screen. When multiple traces are displayed in a channel window, it is also possible to maximize a specific trace displayed within that channel window.

NOTE The Window/Trace Display data can be preset to factory settings using the Preset option

Maximizing a window

1. Press **Channel Next** (or **Channel Prev**) to select the channel of which window will be maximized.
2. Press **Channel Max** to maximize the channel window.
3. Press **Channel Max** one more time to reduce the window to its previous size.

Maximizing a trace display

1. Press **Channel Next** (or **Channel Prev**) to select the channel to which the trace belongs.
2. Press **Trace Next** (or **Trace Prev**) to select the trace of which display will be maximized.
3. Press **Trace Max** to maximize the trace display.
4. Press **Trace Max** one more time to reduce the display to its previous size.

Turning off the display of graticule labels

When using a rectangular display format, the graph area can be expanded to the left by turning OFF the display of graticule labels.

Turning OFF graticule label display

1. Press **Channel Next** (or **Channel Prev**) to select the channel of which graticule label display will be turned ON or OFF.
2. Press **Display**.
3. Click **Graticule Label** to turn graticule label display ON or OFF.

Hiding frequency information

You can hide the frequency information from the screen in order to ensure its confidentiality or for other reasons.

Hiding Frequency Information on the Screen

Follow the steps below to hide frequency information on the measurement screen.

1. Press **Display** key.
2. Click **Frequency** to turn OFF the frequency display.

NOTE Turning OFF the frequency display using **Display** > **Frequency** key does not erase the frequency display within the **Stimulus** softkey, which is turned on by pressing **Start**, **Stop**, **Center**, and **Span**. The display of the softkey bar itself can be switched ON or OFF by pressing **Softkey On/Off**.

Hiding Softkey's Frequency Information

You can delete the frequency information from the measurement screen, which changes the frequency information displayed in the Stimulus softkey and the data entry area for Hz unit to asterisks (***).

1. Press **System** key.
2. Click **Service Menu**, then click **Security Level** and select any of the following options for the frequency display.

Softkey	Function
None	Displays the frequency information.
Low	Hides the frequency information with a series of asterisks. **Save/Recall** > **Save Trace Data** and **Save SnP** are inactive. This can be turned OFF by the Security Level menu.
High	Hides the frequency information with a series of asterisks. **Save/Recall** > **Save Trace Data** and **Save SnP** are inactive. This cannot be turned OFF by the Security Level menu. Resetting to OFF is only possible by executing Preset or Recall.

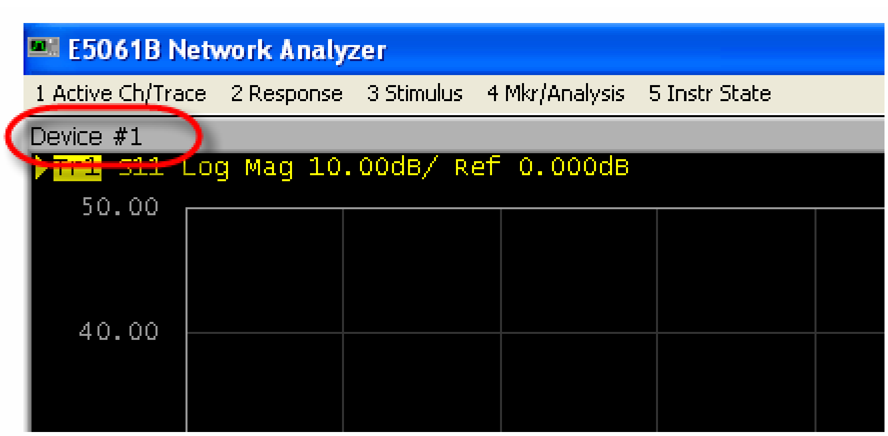
Labeling a window

It is possible to assign a unique name to a channel and display it on the screen. This feature is useful in saving and/or printing measurement results for future reference.

Labeling a window

1. Press **Channel Next** (or **Channel Prev**) to select the channel to be labeled.
2. Press **Display** > **Edit Title Label**, and the title label input dialog box appears.

3. Using the keys in the dialog box, type a label and click **Enter**.
4. Click **Title Label** to turn ON the title display. The title appears within a frame at the top of the channel window.



Setting display colors

Selecting display mode

You can select the display mode of the LCD display from two modes: normal display (background: black) or inverted display (background: white). In normal display, the colors of items are preset so that you can recognize them easily on the display of the instrument. On the other hand, in inverted display, they are preset to colors obtained by inverting the default settings to the normal display so that you can use data easily when storing it into a graphic file.

The selection procedure is as follows:

1. Press **Display**.

2. Click **Invert Color** to select the display color. OFF indicates the normal display; ON the inverted display.

Setting display color for each item

You can set the display color to the normal display or inverted display separately for each of the following items:

- Data/memory trace
- Labels and lines of graphs
- File display of the limit test and limit lines
- Background

Set the color of each item by specifying the amounts of red (R), green (G), and blue (B) contained in the color. You can specify each level of R, G, and B in 6 steps (0 to 5). Therefore, total of 216 colors are available by combining them. The table below shows the R, G, and B values for the main colors as a reference.

	R	G	B		R	G	B		R	G	B
White	5	5	5	**Gray**	2	2	2	**Black**	0	0	0
Light red	5	3	3	**Red**	5	0	0	**Dark Red**	2	0	0
Light yellow	5	5	3	**Yellow**	5	5	0	**Dark Yellow**	2	2	0
Light green	3	5	3	**Green**	0	5	0	**Dark Green**	0	2	0
Light cyan	3	5	5	**Cyan**	0	5	5	**Dark cyan**	0	2	2
Light blue	3	3	5	**Blue**	0	0	5	**Dark Blue**	0	0	2
Light magenta	5	3	5	**Magenta**	5	0	5	**Dark Magenta**	2	0	2

The setting procedure is as follows:

1. Press **System** > **Misc Setup** > **Display Setup** > **Color Setup.**
2. Click **Normal** (for normal display) or **Invert** (for inverted display).
3. Click the **softkey** corresponding to the item of which you want to set the display color.

4. Click **Red** (or, **Green**, or **Blue**).
5. Select the amount of the selected color from 0 to 5.

Resetting the display colors to the factory state

You can reset the display colors in normal display and inverted display to the preset factory state.

The selection procedure is as follows:

1. Press **System** > **Misc Setup** > **Display Setup** > **Color Setup**.
2. Click **Normal** (for normal display) or **Invert** (for inverted display).
3. Click **Reset Color** > **OK.**

Setting display magnification

You can reset the display magnification to Small, Normal or Large.

The selection procedure is as follows:

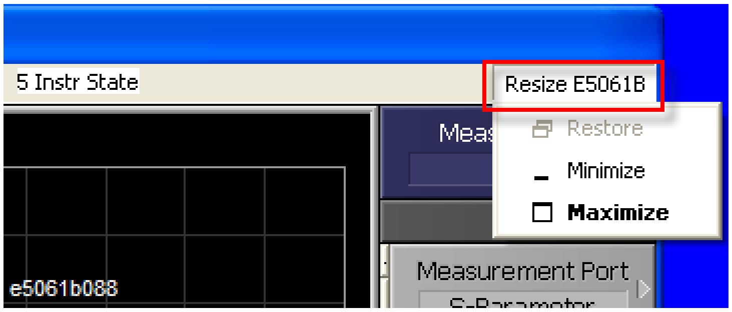
1. Press **System** > **Misc Setup** > **Display Setup** > **Magnification.**
2. Click **Normal,Small** or **Large**.

Resizing the screen

You can resize the E5061B screen by minimizing, maximizing or restoring it to its original size.

The resizing procedure is as follows:

1. Click **Resize E5061B** at the top right corner of the screen.



2. A drop-down menu prompts and the available options are:

Softkey	Function
Restore	Restores the screen to its default size.
Minimize	Minimizes the screen.
Maximize	Displays the screen in full page size.

3. Click **Restore,Minimize** or **Maximize**.

4. When the screen is resized according to an option, its the related softkey is disabled. For example, when the screen is displayed in full page size, **Maximize** is disabled.

Another option to minimize the E5061B screen is by using the Menu Bar and the procedure is as follows:

1. Press **Display**.
2. Click **Minimize E5061B.**

You can also hide and restore the title bar of the E5061B screen and the procedure is as follows:

1. Press **Display**.
2. Click **E5061B Title bar.**
3. Click **ON** to restore the title bar.
4. Click **OFF** to hide the title bar.

Setting Port Coupling (Option 3L5 only)

In Option 3L5, Port 1 and 2 can be selectively AC or DC coupled. DC coupling (Default setting) allows both DC and AC signals through, while AC coupling accepts only AC signal. In DC coupling, the port 1 and 2 has 50 Ω input impedance. In AC coupling, a blocking capacitor is inserted at the ports 1 and 2. It may cause the signal level difference between AC and DC coupling at lower frequency.

1. Press **System** > **S-Param Port Couple**, then select **AC** or **DC**.

When the port coupling is set at AC, there are limitations as follows:

- DC bias can not be turned ON for the port 1.
- The minimum frequency is 100 kHz.

NOTE When the port coupling is set at AC and the frequency is set below 100 kHz, the trigger becomes hold state.

Calibration

Calibration

Overview

- Measurement Errors and their Characteristics
- Calibration Types and Characteristics
- Checking Calibration Status
- Clear Calibration

Basic Calibration

- Selecting Calibration Kit
- OPEN/SHORT Response Calibration (reflection test)
- THRU Response Calibration (transmission test)
- Enhanced Response Calibration
- 1-Port Calibration (reflection test)
- Full 2-Port Calibration

Calibration with ECal (Electronic Calibration)

- ECal (Electronic Calibration)
- ECal Driver Installation
- Calibration Using ECal Module

Advanced Calibration with ECal

- Improving Calibration Accuracy along with ECal
- Confidence Check on Calibration Coefficients Using ECal
- Turning off ECal Auto-detect Function
- User-characterized ECal

Advanced Calibration

- Modifying Calibration Kit Definition
- Partial Overwrite
- Adapter Removal-Insertion

Measurement Errors and their Characteristics

- Overview
- Drift Errors
- Random Errors
- Systematic Errors

Other topics about Calibration

Overview

It is important to understand the factors contributing to measurement errors in order to determine the appropriate measures that should be taken to improve accuracy. Measurement errors are classified into three categories:

Drift Errors

Drift errors are caused by deviations in the performance of the measuring instrument (measurement system) that occur after calibration. Major causes are the thermal expansion of connecting cables and thermal drift of the frequency converter within the measuring instrument. These errors may be reduced by carrying out frequent calibrations as the ambient temperature changes or by maintaining a stable ambient temperature during the course of a measurement.

Random Errors

Random errors occur irregularly in the course of using the instrument. Since random errors are unpredictable, they cannot be eliminated by calibration. These errors are further classified into the following sub-categories depending on their causes:

- Instrument noise errors
- Switch repeatability errors
- Connector repeatability errors

Instrument noise errors

Instrument noise errors are caused by electric fluctuations within components used in the measuring instrument. These errors may be reduced by increasing the power of the signal supplied to the DUT, narrowing the IF bandwidth, or enabling sweep averaging.

Switch repeatability errors

Switch repeatability errors occur due to the fact that the electrical characteristics of the mechanical RF switch used in the measuring instrument change every time it is switched on. These errors may be reduced by carrying out measurements under conditions in which no switching operation takes place.

Connector repeatability errors

Connector repeatability errors are caused by fluctuations in the electrical characteristics of connectors due to wear. These errors may be reduced by handling connectors with care.

Systematic Errors

Systematic errors are caused by imperfections in the measuring instrument and the test setup (cables, connectors, fixtures, etc.). Assuming that these errors are repeatable (i.e., predictable) and their characteristics do not change over time, it is possible to eliminate them mathematically at the time of measurement by determining the characteristics of these errors through calibration. There are six types of systematic errors, as follows.

Errors caused by signal leaks in the measuring system:

- Directivity
- Isolation (cross-talk)

Errors caused by reflections in the measuring system:

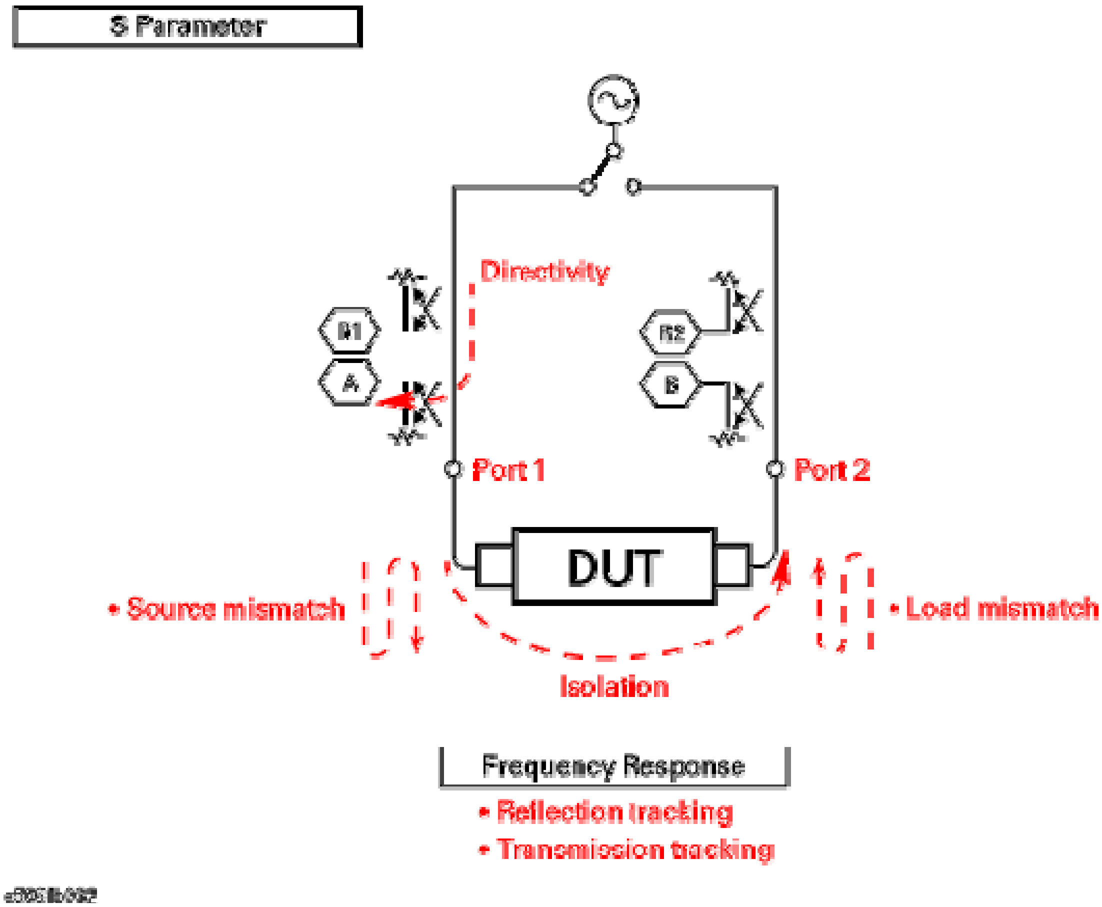
- Source match
- Load match

Errors caused by the frequency response of the receiver within the measuring instrument:

- Reflection tracking
- Transmission tracking

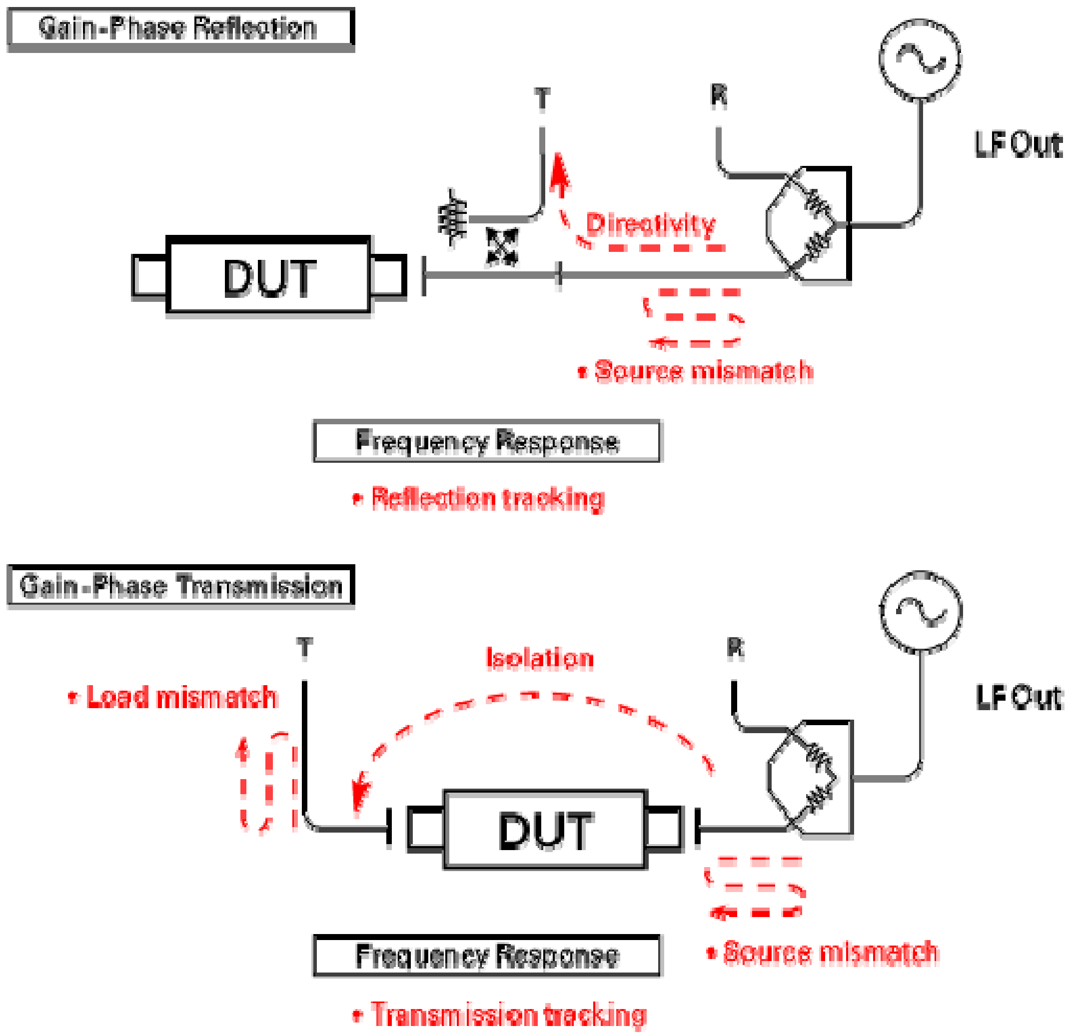
The E5061B has two receivers for each S-parameter test port: the reference receiver and the test receiver (transmission measurement or reflection measurement). You can perform measurements with both of these receivers at the same time.

E5061B port architecture and systematic errors in S-parameter Measurement



The ports T and R are receivers and LF out supplies the source signal.

E5061B port architecture and systematic errors in Gain-Phase Measurement



Directivity error (Ed)

Directivity errors are caused by the fact that, in a reflection measurement, signals other than the reflection signal from the DUT are received by the receiver through the directivity coupler. When a certain port is a stimulus port, this error can be defined as a constant value for each stimulus port because the state of the termination at the other ports does not change. The number of directivity errors of the E5061B is the number of stimulus ports you use.

Isolation error (Ex)

An isolation error (crosstalk error) is caused by signals other than the transmission signal of the DUT leaking to the test receiver of the transmission measurement port in transmission measurements. When a certain port is a stimulus port, an isolation error is defined for each of the ports. Therefore, the number of isolation errors for the E5061B is the total number of combinations of stimulus ports and response ports.

Source match error (Es)

A source match error is caused when the reflection signal of the DUT reflects at the signal source and enters the DUT again. When a certain port is a stimulus port, this error can be defined as a constant value for each stimulus port because the state of the signal source switch does not change. The number of source match errors in the E5061B is equivalent to the number of stimulus ports you use.

Load match error (El)

A load match error is caused when part, but not all, of the signal transmitted in the DUT reflects at a response port is measured by the receiver of the response port. When a certain port is a stimulus port, a load match error is defined for each of the ports. Therefore, the number of load match errors for the E5061B is the total number of combinations of stimulus ports and response ports.

Reflection tracking error (Er)

A reflection tracking error is caused by the differences in frequency response between the test receiver and the reference receiver of a stimulus port in reflection measurements. This error can be defined as a constant value for each stimulus port because the combination of the test receiver and the reference receiver of a stimulus port is always the same. The number of reflection tracking errors for the E5061B is simply the number of stimulus ports you use.

Transmission tracking error (Et)

A transmission tracking error is caused by the differences in frequency response between the test receiver of a response port and the reference receiver of a stimulus port in transmission measurements. When a certain port is a stimulus port, a transmission tracking error is defined for each of the ports. Therefore, the number of transmission tracking errors for the E5061B is the total number of combinations of stimulus ports and response ports.

Calibration Types and Characteristics

The table shows different types of calibrations and features of each method.

Calibration Method	Standard (s) Used	Corrected Error Factor	Measurement Parameters	Characteristics
No calibration	None	None	All parameters	• Low accuracy • Calibration not required
Response Calibration	• OPEN or SHORT • LOAD (Optional)	Following 2 error terms: • Reflection Tracking (Er) • Directivity (Ed)	S11(Reflection characteristics at 1 port) T/R (reflection setting)	• Medium-level accuracy • Quick calibration • Isolation calibration improves the accuracy in a reflection measurement of a DUT with high return loss
	• THRU • LOAD (Optional)	Following 2 error terms: • Transmission Tracking (Et) • Isolation (Ex)	S21 (1 direction transmission characteristics at 2 ports) T/R (Transmission setting)	• Medium-level accuracy • Quick calibration • Isolation calibration improves the accuracy in a transmission measurement of a device with high insertion loss
1-Port Calibration	ECal module (2-port/4-port)	Following 3 error terms: • Directivity (Ed) • Source Match (Es) • Reflection Tracking (Er)	S11 (Reflection characteristics at 1 port) T/R	• 1-port measurement with the highest degree of accuracy • Quick calibration with low chance of operator error
	• OPEN • SHORT • LOAD			• Highly accurate 1-port measurement
Enhanced Response	Ecal module (2-port)	Following 5 error terms: • Directivity (Ed1)	S11, S21 (1 direction transmission/Reflection characteristics at 2 ports)	• Highly accurate 2-port measurement (higher than response calibration)

Calibration		• Isolation (Ex21) • Source Match (Es1) • Transmission Tracking (Et21) • Reflection Tracking (Er1)		• Quick calibration with low chance of operator error
	• OPEN • SHORT • LOAD • THRU			• Highly accurate 2-port measurement (higher than response calibration)
Full 2-Port Calibration	ECal module (2-port/4-port)	Following 12 error terms: • Directivity (Ed1,Ed2) • Isolation (Ex21,Ex12) • Source Match (Es1,Es2) • Load Match (El12,El21) • Transmission Tracking (Et21,Et12) • Reflection Tracking (Er1,Er2)	S11,S21,S12,S22 (All S-parameters at 2 ports)	• Highly accurate 2-port measurement • Quick calibration with low chance of operator error
	• OPEN • SHORT • LOAD • THRU			• Highly accurate 2-port measurement

Checking Calibration Status

Other topics about Calibration

Execution Status of Error Correction for Each Channel

You can check the execution status of error correction for each channel with the error correction status.

The error correction status is indicated in the channel status bar at the lower part of the window by the symbols shown in the below table.

Symbol	Execution status of error correction
Cor (displayed in blue)	Error correction: On (enabled for all traces)
Cor (displayed in gray)	Error correction: On (enabled for some traces)
Off (displayed in gray)	Error correction: Off
--- (displayed in gray)	Error correction: On (no calibration data)
C? (displayed in blue)	Error correction: On (Interpolation is being executed or the IF bandwidth, power level, power range, sweep time, sweep delay time, or sweep type is different from that when the calibration was executed.) NOTE When the attenuator value of T/R port is changed after executing calibration, this symbol is displayed.
C! (displayed in blue)	Error correction: On (Extrapolation is being executed.) NOTE When the T/R input Z value is changed after executing calibration, this symbol is displayed.

NOTE When one of the trace is set as DC monitor, COR status does not change even if the sweep time is changed.

Execution Status of Error Correction for Each Trace

You can check the status of the error correction actually executed for each trace with the trace status area.

For a trace of which error correction is executed, the applied calibration type is indicated in the trace status area by the symbols in the table below.

If none of the symbols described above is displayed, error correction is not executed for the trace.

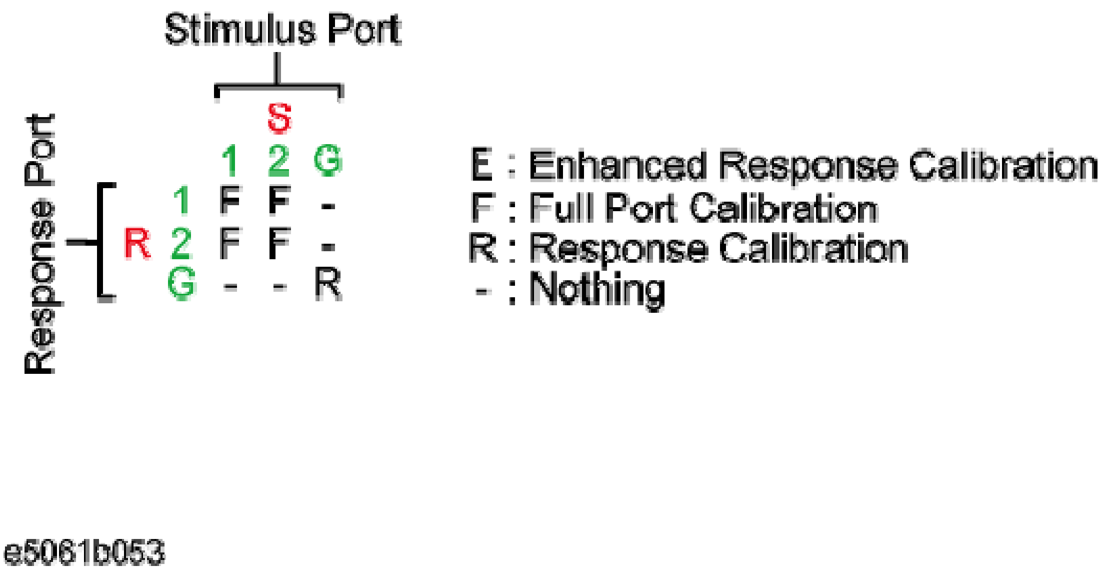
Acquisition Status of Calibration Coefficient for Each Channel

You can check the acquisition status of the calibration coefficient for each channel with the calibration property.

The calibration property displays the acquisition status of the calibration coefficient between test ports for each channel in matrix format. The following example shows 2 port full calibration which is done on ports 1 and 2, and one of the response calibration is done on the ports for gain-phase measurement.

For impedance calibration in option 005, see Acquisition Status of Calibration Coefficient in the impedance measurement section.

Example of calibration property display



Procedure to turn on/off calibration property display

Follow these steps to turn on/off the calibration property display.

1. Press **Channel Next** (or **Channel Prev**) to select the channel of which you want to turn on/off the calibration property display.
2. Press **Cal** > **Property**. Each press toggles the on/off setting.

Conditions for clearing the calibration coefficients which are already acquired

In the following cases, the calibration coefficients which are already acquired are cleared.

- Executing preset clears all the calibration coefficients.
- If S parameters are required to calculate the calibration coefficient for the specified calibration type and test ports and those required for the existing calibration coefficient overlap, executing the acquisition of the calibration coefficient (measuring necessary data and then clicking the **Done** softkey) clears the calibration coefficient of which necessary S parameters overlap. Taking the Example of calibration property display as an example, if you acquire the calibration coefficient of the response calibration for test port 1, calibration coefficients for s-parameters are cleared.

Clear Calibration

This softkey clears the user calibration data. When Calibration is done for a particular DUT, the data get stored in the E5061B. To clear this data, **Clear Calibration** can be used which removes the User calibration data from the E5061B.

Other topics about Calibration

Basic Calibrations

Selecting Calibration Kit

- Overview
- Procedure

Other topics about Basic Calibration

Overview

Before performing calibration, you need to select a calibration kit.

If you use a calibration kit other than a predefined one, you need to define it. If the connector type of the standard calibration kit you use has polarity (the distinction between male and female), you need to change the standard class definition of the calibration kit depending on the standard you actually use. For more information, see Modifying Calibration Kit Definition.

> NOTE If you select a predefined calibration kit, (m) and (f) in the name (label) of the standard displayed in the softkey indicate male (m) and female (f) for the analyzer's connector, respectively.

Procedure

1. Press **Channel Next** (or **Channel Prev**) to select the channel of which you want to select the calibration kit.
2. Press **Cal** > **Cal Kit**, then select the calibration kit.

> NOTE If the name (label) of the calibration kit has been changed, the label is displayed as the softkey.

> NOTE An asterisk (*) on the upper right of the softkey corresponding to a predefined calibration kit indicates that its definition value has been changed from the factory setting by the user.

OPEN/SHORT Response Calibration (reflection test)

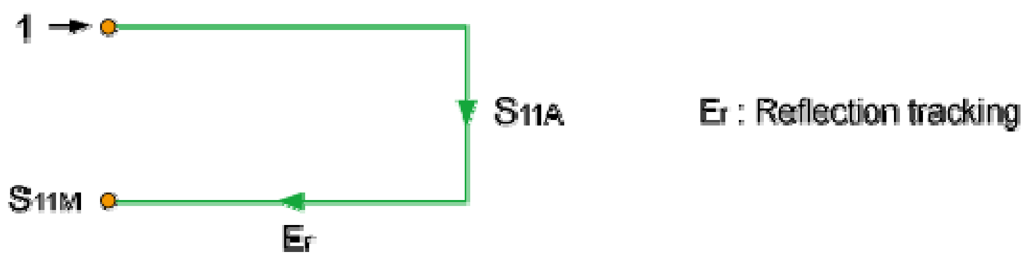
- Overview
- S-Parameter Measurement
- Gain-Phase Measurement

Other topics about Basic Calibration

Overview

In OPEN or SHORT response calibration, calibration data are measured by connecting an OPEN or SHORT standard, respectively, to the desired test port. For frequency response, these calibrations effectively eliminate the reflection tracking error from the test setup reflection test using that port.

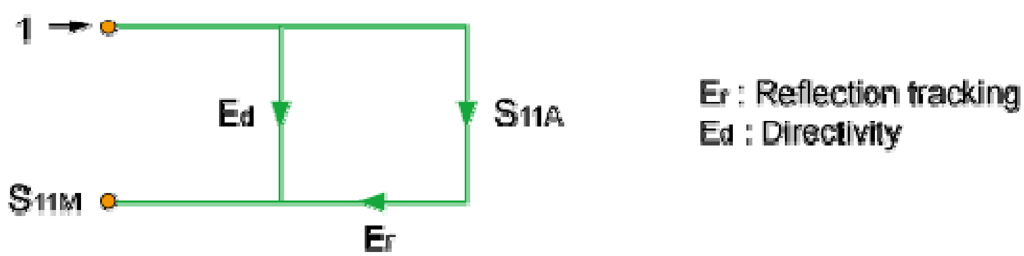
1-Port error model (OPEN/SHORT response)



e5071c398

It is also possible to carry out isolation calibration with a LOAD standard during OPEN/SHORT response calibration. An isolation calibration eliminates the directivity error from the test setup in a reflection test using that port.

1-Port error model (OPEN/SHORT response + isolation)



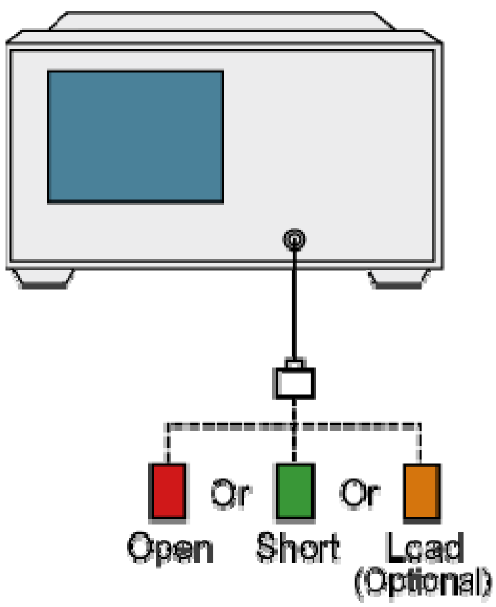
e5071c399

S-Parameter Measurement

1. Press **Channel Next** (or **Channel Prev**) to select the channel for which you want to perform the calibration. Confirm if your required setup is set to the channel.
2. Press **Cal** > **Calibrate.**

3. Select **Response (Open) or Response (Short) calibration**.
4. Click **Select Port.**
5. Select the test port upon which you will perform OPEN/SHORT response calibration.
6. Connect a calibration standard (OPEN or SHORT) to the selected test port (connector to which the DUT is to be connected).
7. Click **Open** or **Short** to start the calibration measurement.
8. If an isolation calibration must be performed using a LOAD standard, follow the procedure below.
9. Connect a LOAD standard to the selected test port (connector to which the DUT is to be connected).
10. Click **Load (Optional)** to start the measurement on the LOAD standard.
11. Click **Done** to terminate the response calibration (and the LOAD isolation calibration) process. Upon pressing this key, calibration coefficients are calculated and saved. The error correction function is also enabled automatically.

Connecting standards in OPEN/SHORT Response calibration



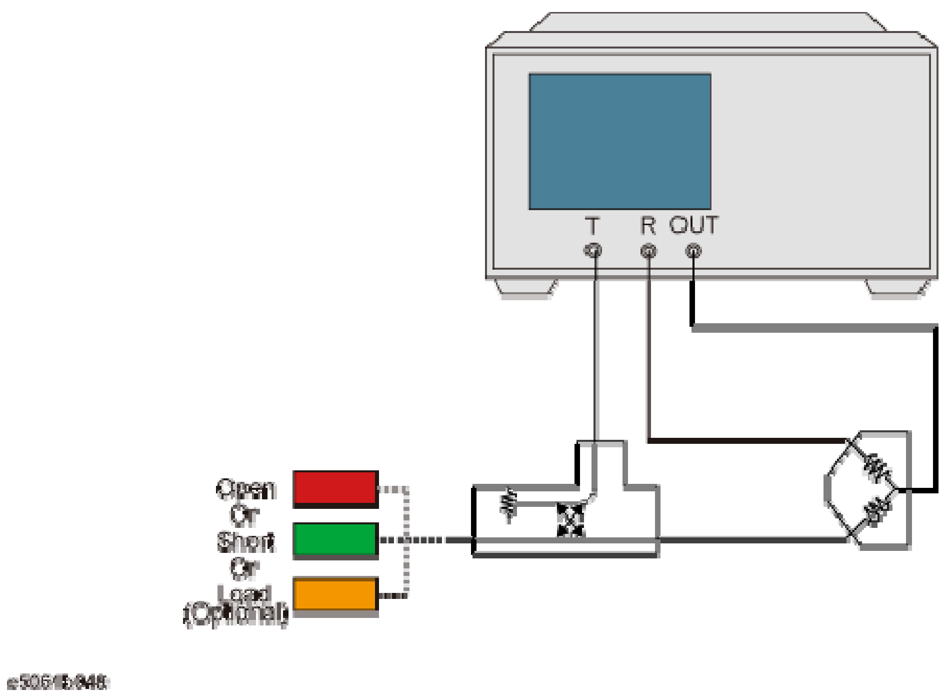
e5061b047

Gain-Phase Measurement

1. Press **Channel Next** (or **Channel Prev**) to select the channel of which you want to perform the calibration. Confirm if your required setup is set to the channel.
2. Press **Cal** > **Calibrate.**

3. Select **Response (Open) or Response (Short) calibration.**
4. Click **Select Port** > **GP Port**.
5. Connect a calibration standard (OPEN or SHORT) with the position specified in the figure below.
6. Click **Open** or **Short** to start the calibration measurement.
7. If an isolation calibration is required, follow the procedure below:
    a. Connect a LOAD standard with the position specified in the figure below.
    b. Click **Load (Optional)** to start the measurement on the LOAD standard.
8. Click **Done** to terminate the response calibration process. Upon pressing this key, calibration coefficients are calculated and saved. The error correction function is also enabled automatically.

Connection standards in Open/Short response calibration (Gain-Phase measurement)



THRU Response Calibration (transmission test)

- Overview
- S-Parameter Measurement
- Gain-Phase Measurement

Other topics about Basic Calibration

Overview

In THRU response calibration, calibration data are measured by connecting a THRU standard to the desired test port. This calibration effectively eliminates the frequency response transmission tracking error from the test setup in a transmission test using that port.

2-Port error model (THRU response)

e5071c400

It is also possible to carry out an isolation calibration using a LOAD standard in the process of THRU response calibration. An isolation calibration eliminate sisolation error (crosstalk error) from the test setup in a transmission test using that port.

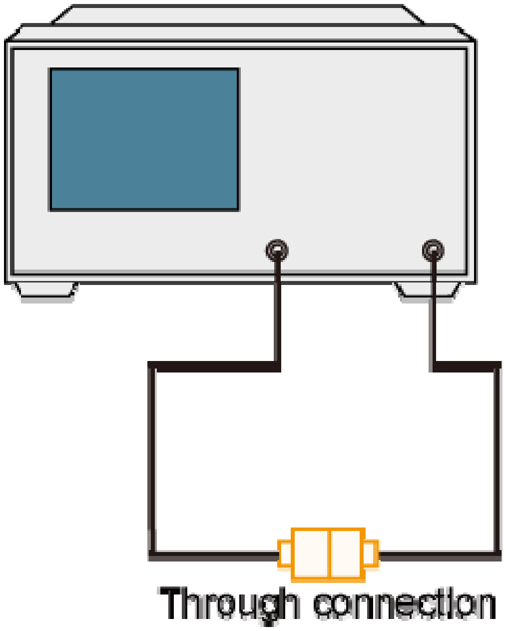
2-Port Error model (THRU response + isolation)

e5071c401

S-Parameter Measurement

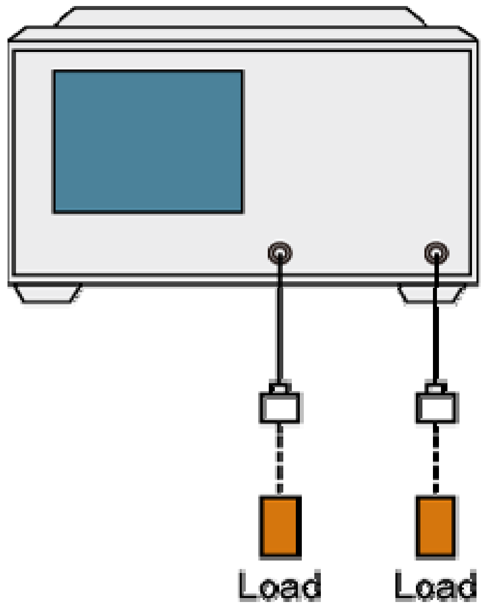
1. Press **Channel Next** (or **Channel Prev**) to select the channel for which you want to perform the calibration. Confirm if your required setup is set at the channel.
2. Press **Cal** > **Calibrate** > **Response (Thru)** > **Select Ports**.
3. Select the test ports (and corresponding S parameters) upon which a THRU response calibration is performed.

4. Make a connection between the selected test ports (between the connectors to which the DUT will be connected).



e5071c317

5. Click **Thru** to start the calibration measurement.
6. If an isolation calibration must be performed using a LOAD standard, follow the procedure below:
    a. Connect a LOAD standard to each of the two selected test ports (connectors to which the DUT is to be connected).



e5071c319

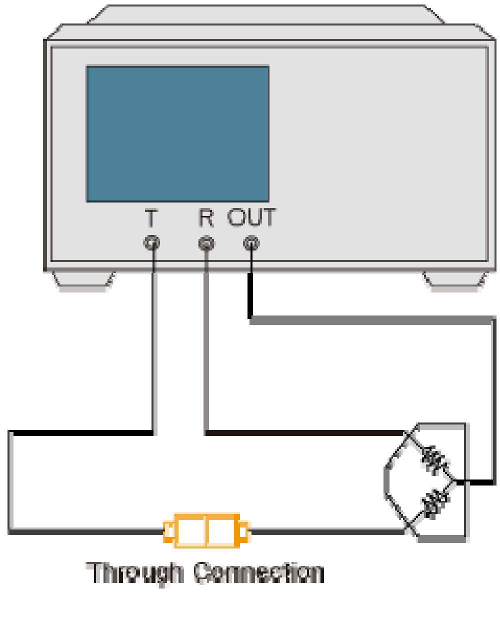
b. Click **Isolation (Optional)** to start the calibration measurement.

7. Click **Done** to terminate the response calibration (and the LOAD isolation calibration) process. Upon pressing this key, calibration coefficients are calculated and saved. The error correction function is also enabled automatically.

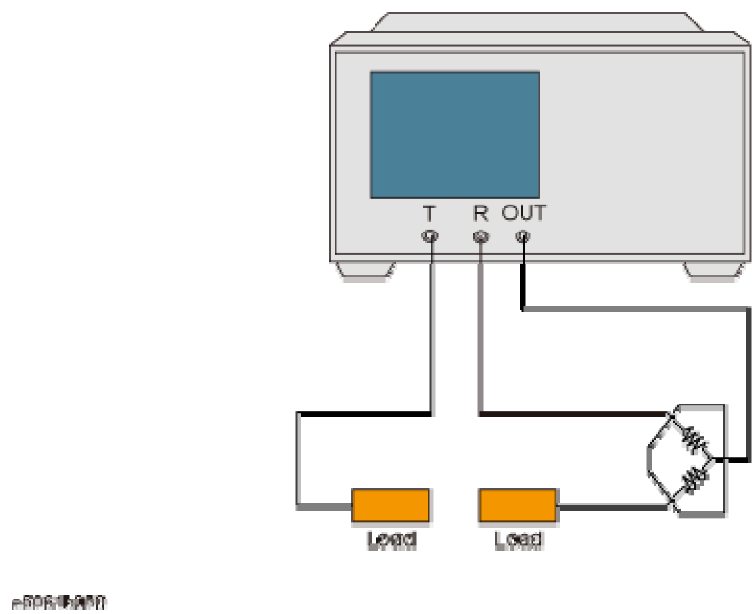
Gain-Phase Measurement

1. Press **Channel Next** (or **Channel Prev**) to select the channel for which you want to perform the calibration. Confirm if your required setup is set at the channel.
2. Press **Cal** > **Calibrate** > **Response (Thru)** > **Select Ports** > **GP Ports**.

3. Connect the cable according to following figure.



4. Click **Thru** to start the calibration measurement.
5. If an isolation calibration is required, follow the procedure below:
    a. Connect the LOAD standards in the position specified in the figure below.



    b. Click **Isolation (Optional)** to start the calibration measurement.
6. Click **Done** to terminate the response calibration (and the LOAD isolation calibration) process. Upon pressing this key, calibration

coefficients are calculated and saved. The error correction function are also enabled automatically.

Enhanced Response Calibration

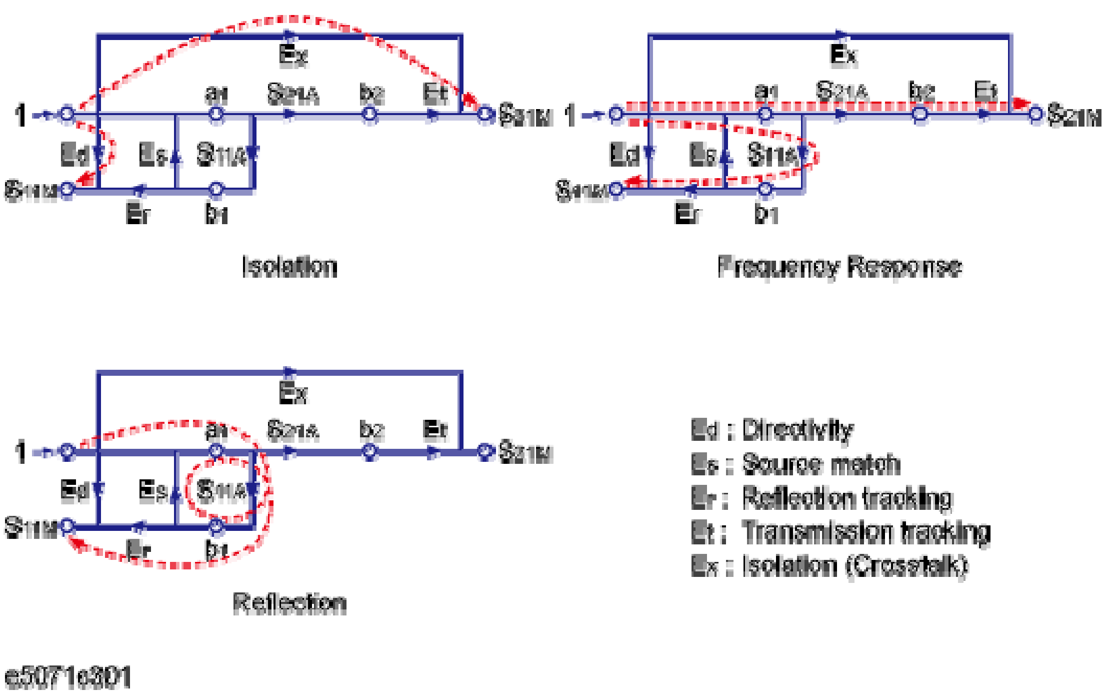
- Overview
- Procedure

Other topics about Basic Calibration

Overview

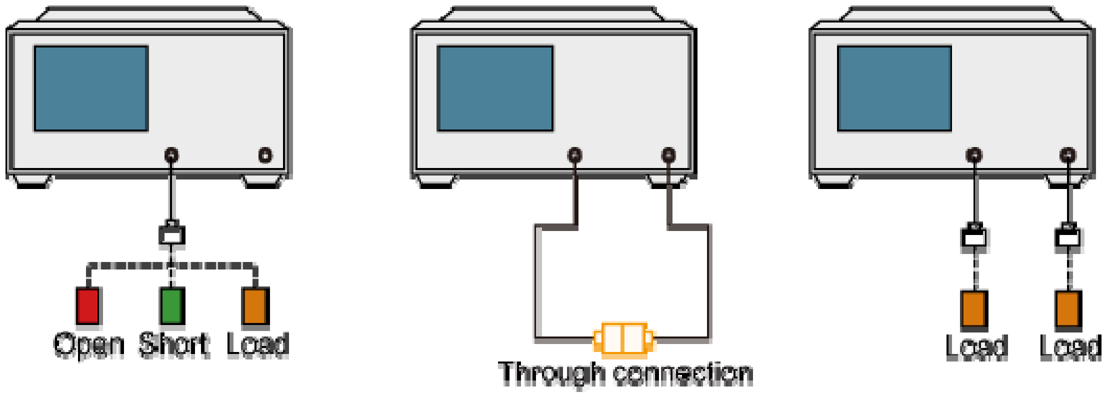
In enhanced response calibration, calibration data are measured by connecting an OPEN standard, a SHORT standard, or a LOAD standard to the output port (or a THRU standard between two ports). This calibration effectively eliminates the directivity error, crosstalk, source match error, frequency response reflection tracking error, and frequency response transmission tracking error from the test setup in a transmission or reflection test that uses those ports .

2-Port Error Model (Enhanced Response)



Procedure

Connecting the Standard at Enhanced Response Calibration



e5071c332

1. Press **Channel Next** (or **Channel Prev**) to select the channel of which you want to perform the calibration. Confirm if your required setup is set to the channel.
2. Press **Cal** > **Calibrate** > **Enhanced Response** > **Ports** to select the test ports on which an enhanced response calibration is performed.
3. Connect an OPEN calibration standard to the output port.
4. Click **Open** to start the calibration measurement.
5. Disconnect the OPEN calibration standard and replace it with a SHORT calibration standard.
6. Click **Short** to start the calibration measurement.
7. Disconnect the SHORT calibration standard and replace it with a LOAD standard.
8. Click **Load** to start the calibration measurement.
9. Make a THRU connection between the two ports.
10. Click **Thru** to start the calibration measurement.
11. If an isolation calibration must be performed using a LOAD standard, follow the procedure below:
    a. Connect a LOAD standard to the two test ports.
    b. Click **Isolation (Optional)** to start the calibration measurement.
12. Click **Done** to terminate the enhanced response calibration process. Upon pressing the key, calibration coefficients are calculated and saved. The error correction function is also enabled automatically.

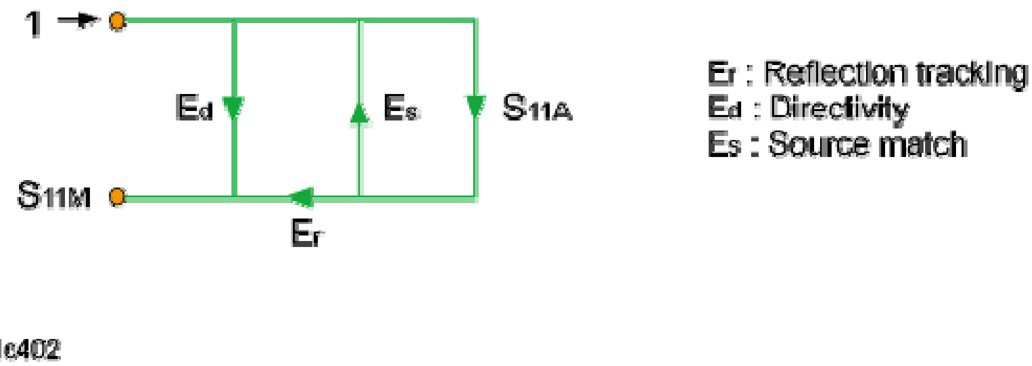
1-Port Calibration (reflection test)

- Overview
- S Parameter Measurement
- Gain-Phase Measurement

Other topics about Basic Calibration

Overview

In 1-port calibration, calibration data are measured by connecting an OPEN standard, a SHORT standard, and a LOAD standard to the desired test port. This calibration effectively eliminates the frequency response reflection tracking error, directivity error, and source match error from the test setup in a reflection test using that port.



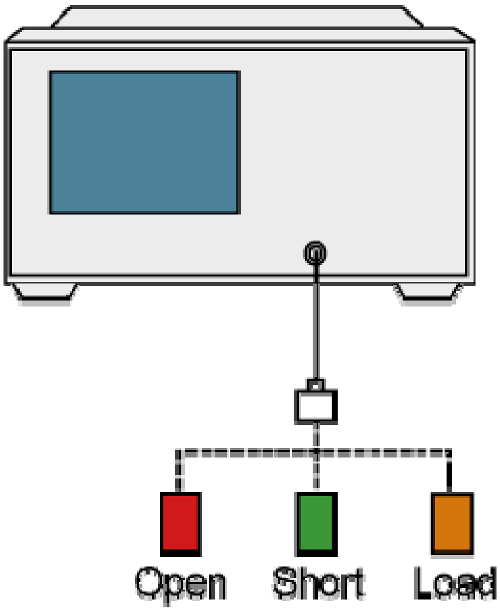
1-Port error model (1-port calibration)

S-Parameter Measurement

1. Press **Channel Next** (or **Channel Prev**) to select the channel of which you want to perform the calibration. Confirm if your required setup is set to the channel.
2. Press **Cal** > **Calibrate** > **1-Port Cal** > **Select Port**.
3. Select a test port (and corresponding S parameter) on which 1-port calibration will be performed.
4. Connect an OPEN calibration standard to the selected test port (connector to which the DUT is to be connected).
5. Click **Open** to start the calibration measurement.
6. Connect a SHORT calibration standard to the selected test port (connector to which the DUT is to be connected).
7. Click **Short** to start the calibration measurement.
8. Connect a LOAD calibration standard to the selected test port (connector to which the DUT is to be connected).
9. Click **Load** to start the calibration measurement.

10. Click **Done** to terminate the 1-port calibration process. Upon pressing this key, calibration coefficients are calculated and saved. The error correction function is also enabled automatically.

Connecting the standard for 1-port calibration

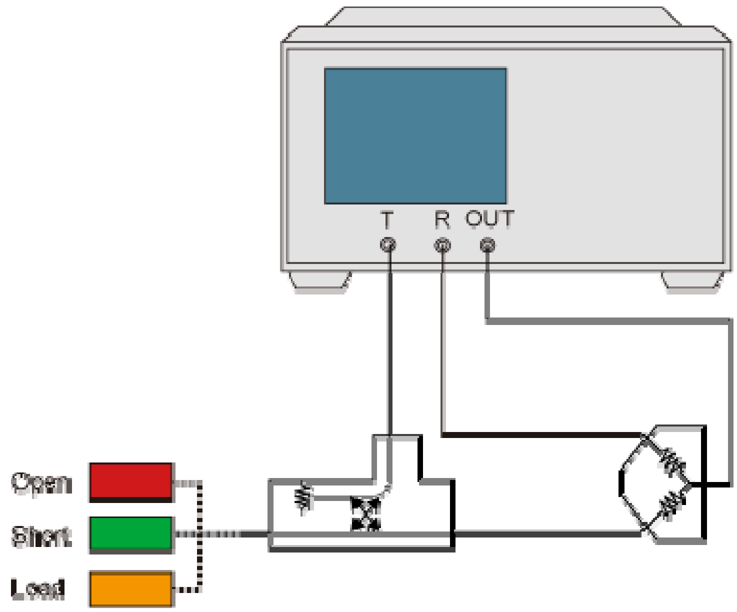


e5071c800

Gain-Phase Measurement

1. Press **Channel Next** (or **Channel Prev**) to select the channel of which you want to perform the calibration. Confirm if your required setup is set to the channel.
2. Press **Cal** > **Calibrate** > **1-Port Cal** > **Select Port** > **GP Port**.
3. Connect an OPEN calibration standard in the position specified in the figure below.
4. Click **Open** to start the calibration measurement.
5. Connect a SHORT calibration standard in the position specified in the figure below.
6. Click **Short** to start the calibration measurement.
7. Connect a LOAD calibration standard in the position specified in the figure below.
8. Click **Load** to start the calibration measurement.
9. Click **Done** to terminate the 1-port calibration process. Upon pressing this key, calibration coefficients are calculated and saved. The error correction function is also enabled automatically.

Connecting the standard for 1-port calibration (Gain-Phase Measurement)



e5061b048

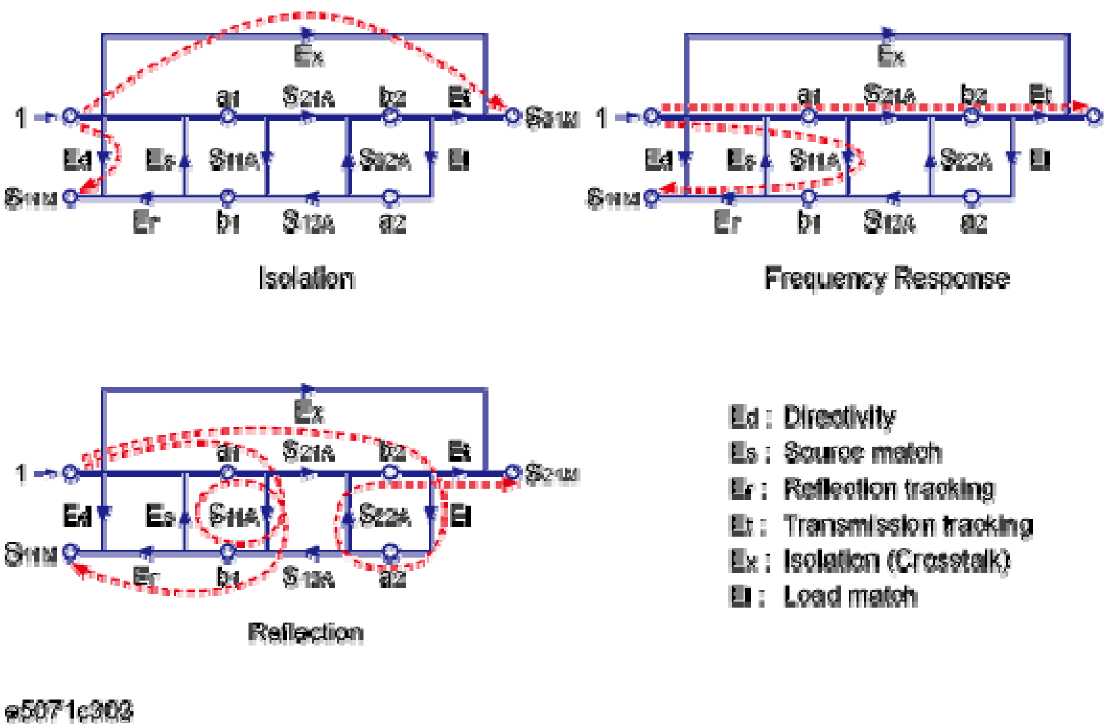
Full 2-Port Calibration

- Overview
- Procedure

Other topics about Basic Calibration

Overview

In full 2-port calibration, calibration data are measured by connecting an OPEN standard, a SHORT standard, or a LOAD standard on ports 1 and 2 for S-parameter (or a THRU standard between two ports).This calibration effectively eliminates the directivity error, crosstalk, source match error, frequency response reflection tracking error, and frequency response transmission tracking error from the test setup in a transmission or reflection test tat uses those ports. This calibration makes it possible to perform measurements with the highest possible accuracy. A total of twelve error terms, six each in the forward direction and the reverse direction, are used in the calibration.

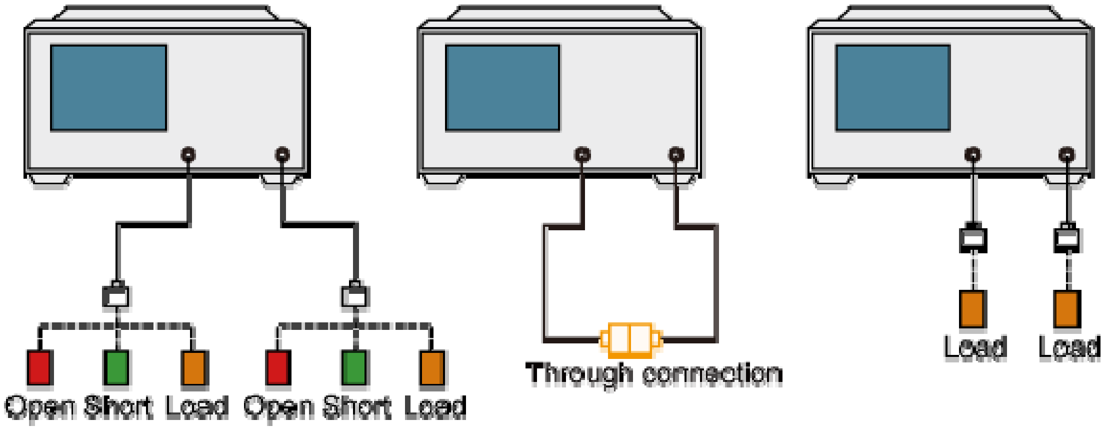


Procedure

1. Press **Channel Next** (or **Channel Prev**) to select the channel fof which you want to perform the calibration.
2. Press **Cal** > **Calibrate** > **2-Port Cal**.
3. Click **Reflection**.

4. Connect an OPEN calibration standard to test port x (the connector to which the DUT is to be connected).
5. Click **Port x Open** to start the calibration measurement (**x** denotes the test port to which the standard is connected).
6. Disconnect the OPEN calibration standard and replace it with a SHORT calibration standard.
7. Click **Port x Short** to start the calibration measurement (**x** denotes the test port to which the standard is connected).
8. Disconnect the SHORT calibration standard and replace it with a LOAD standard.
9. Click **Port x Load** to start the calibration measurement (**x** denotes the test port to which the standard is connected).
10. Repeat the above procedure for port y.
11. Click **Return**.
12. Click **Transmission**.
13. Make a THRU connection between ports x and y (between the connectors to which the DUT is to be connected).
14. Click **Port 1-2 Thru** to start the calibration measurement.
15. Click **Return**.
16. If an isolation calibration must be performed using a LOAD standard, follow the procedure below.
17. Click **Isolation (Optional)**.
18. Connect a LOAD standard to each of the two test ports (connectors to which the DUT is to be connected).
19. Click **Port 1-2 Isol** to start the calibration measurement.
20. Click **Return**.
21. Click **Done** to terminate the full 2-port calibration process. Upon pressing this key, calibration coefficients are calculated and saved. The error correction function is also enabled automatically.

Connecting standards in full 2-port calibration


Open Short Load Open Short Load
Through connection
Load Load

e5071c331

Calibration with ECal (Electronic Calibration)

ECal (electronic calibration)

ECal is a calibration method that uses solid-state circuit technology. ECal offers the following advantages:

- Simplified calibration process.
- Shorter time required for calibration.
- Reduced chance of erroneous operation.
- Little degradation of performance due to wear ad tear because the ECal module employs PIN diodes and FET switches.

NOTE If the frequency sweep range exceeds the frequency range of the ECal, the calibration data for the minimum frequency or maximum frequency is used for the exceeding frequency range and extrapolation is executed.

Refer the following section for ECal calibration.

- ECal Driver Installation
- Calibration Using Ecal Module
- Improving Calibration Accuracy along with ECal
- Confidence Check on Calibration Coefficients Using ECal
- Turning off ECal auto-detect function
- User-characterized ECal

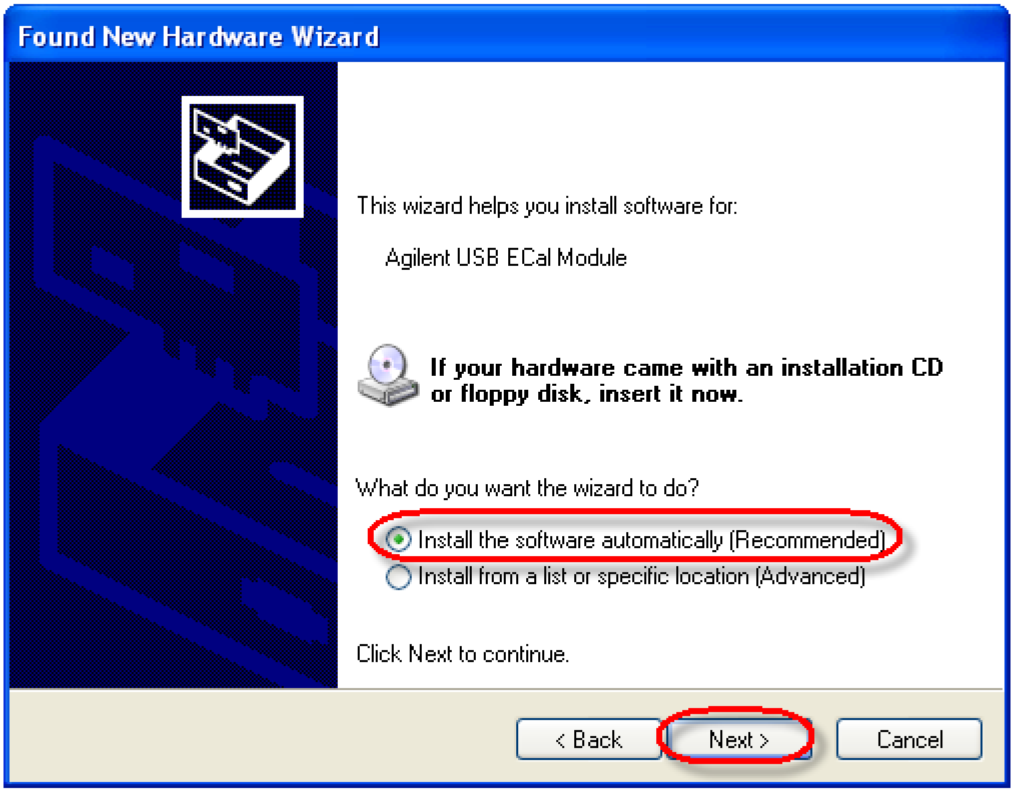
Other topics about Calibration with ECal

ECal Driver Installation

When the ECal is connected t USB ports at the first time, ECal driver installation is required.

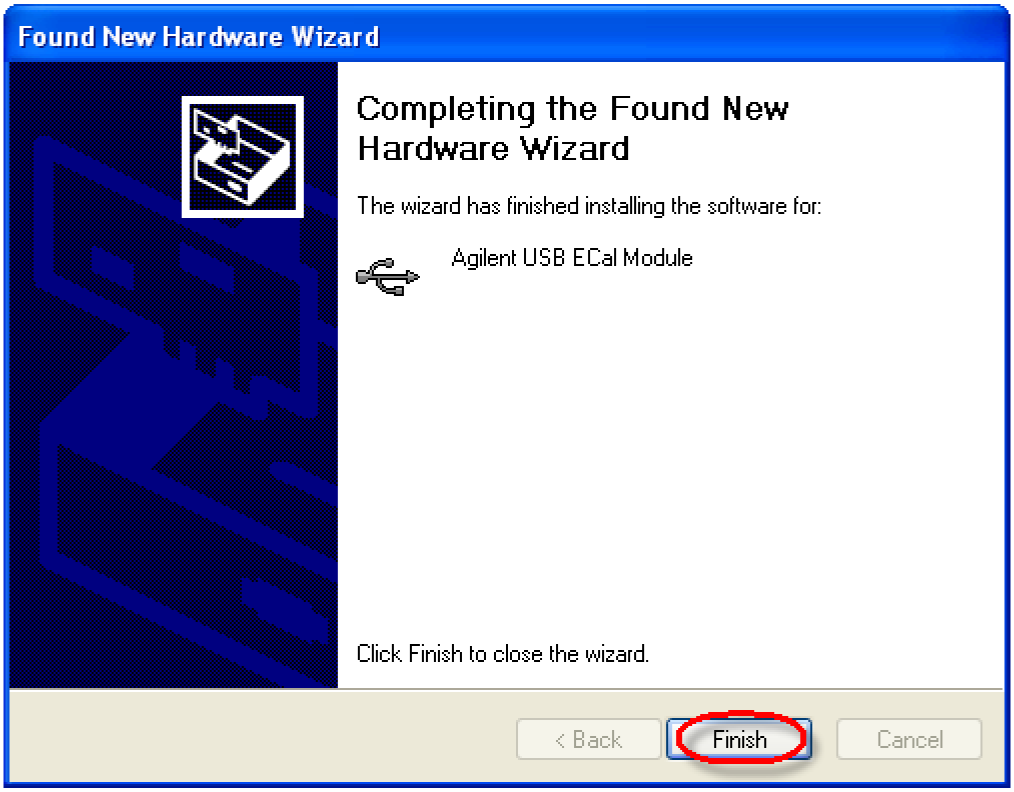
1. Connect an Ecal to the USB port of the E5061B.
2. Select **No, not this time**, then click **Next.**

3. Select **Install the software automatically (Recommended)**, then click **Next.**



e5071c230

4. Click **Finish.**



e5071c231

NOTE Even if you install the driver on a USB port, you will be asked to install the driver again if you connect the ECal with a different USB port.

Other topics about Calibration with ECal

Calibration Using Ecal Module

- Overview
- Procedure

Other topics about Calibration with ECal

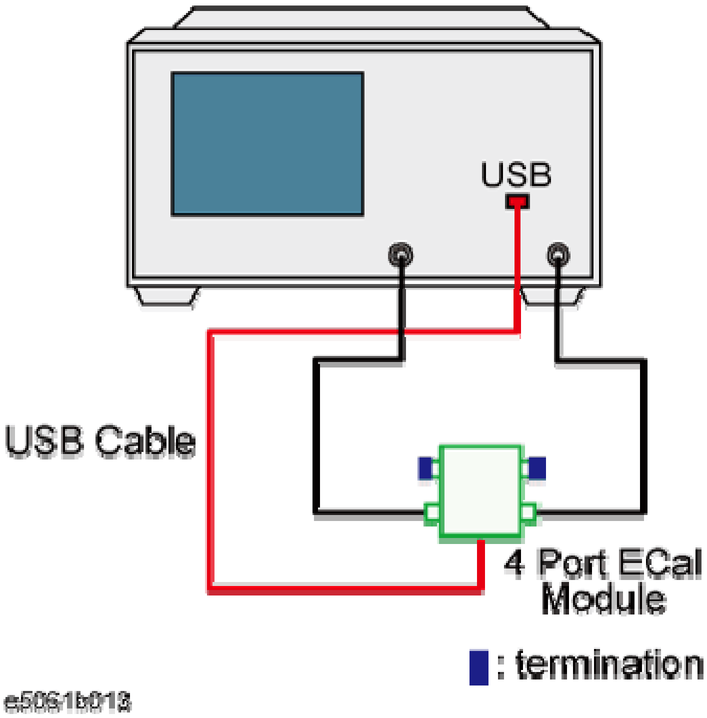
Overview

The E5061B allows you to perform calibration using the 2 or 4-port ECal module in S-parameter measurement. ECal cannot be used for Gain-Phase measurement.

Procedure

1. Connect the USB cable between the USB port of the 2 or 4-port ECal module and the USB port of the E5061B. You can make this connection while the E5061B's power is on.
2. Allows the ECal module to warm up for 20 minutes until the module indicator changes from WAIT to READY.
3. Connect the ports of the ECal module to the test ports you want to calibrate.
4. If you don't use all of the ECal module's ports, connect terminations to the unused ports.
5. Press **Channel Next** (or **Channel Prev**) to select the channel of which you want to perform the calibration.
6. Press **Cal** > **Ecal**.
7. Select the **calibration type**.
8. If you must select a port, the softkey for making this selection is displayed. Select a port and start calibration. If you do not have to select a port, skip this step.
9. The E5061B detects the test ports connected to the ECal and then measurement starts. If the test ports to be calibrated are not connected to the ECal module, error occurs.

Connecting 4-port ECal module (for full 2-port calibration)



NOTE You can connect the ports of the ECal and the test ports of the E5061B arbitrarily. Connected ports can be manually specified although they are automatically detected before the data measurement. For more information, see Turning off ECal auto-detect function.

Improving Calibration Accuracy along with ECal

Inaccuracy caused by thru calibration in the ECal can be reduced by the using the following method:

- Partial Overwrite

Other topics about Calibration with ECal

Partial Overwrite

Using the partial overwrite function allows you to improve the calibration accuracy. For example, follow these steps for full 2-port calibration:

1. Execute full 2-port calibration with ECal and save the calibration coefficients.
2. Execute the procedure of Partial overwrite with the thru standard of the calibration kit.

Confidence Check on Calibration Coefficients Using ECal

- Overview
- Procedure

Other topics about Calibration with ECal

Overview

By using the ECal module, the E5061B allows you to verify the obtained calibration coefficients to determine whether correct measurement is possible with them.

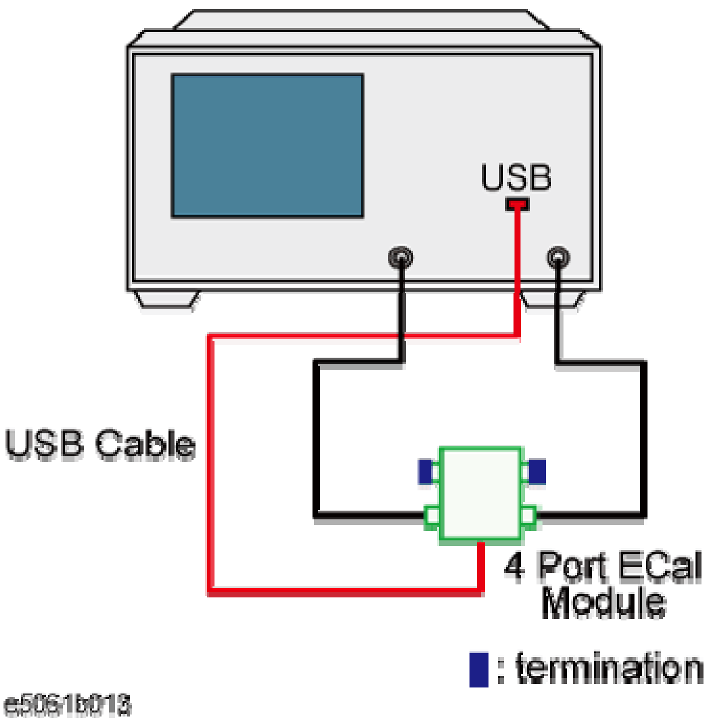
The E5061B is able to set ECal to the state used to verify the measurement parameters and then copy the appropriate characteristics of that verification state to the memory trace from the ECal's built-in memory. This is done according to the measurement parameters of the active trace of the active channel. While measuring ECal in this specified state, you can compare the measurement results with those of the E5061B and with the appropriate measurement results stored in ECal in several different ways. These include simultaneously displaying the data and memory traces or displaying the math operation results between the data and memory traces. This enables you to verify the correctness of measurement for each measurement parameter when the obtained calibration coefficients are used.

Procedure

1. Connect the USB cable between the USB port of the ECal module and that of the E5061B. You can make this connection while the E5061B's power is ON.
2. Allows the ECal module to warm up for 15 to 20 minutes until the module indicator changes from WAIT to READY.
3. Press **Channel Next** (or **Channel Prev**) to select the channel of which you want to perform the verification.
4. Press **Meas** key.
5. Select the S-parameter you want to verify. You cannot verify the mixed mode S-parameter.
6. Connect the test ports of the E5061B corresponding to the selected S-parameter (for example, ports 1 and 2 when the S-parameter is S21) and the ports of the ECal module.
7. If you do not use all of the ECal module's ports, terminate connections to the unused ports.
8. Press **Cal** > **ECal**.

9. When using an adapter to the ECal, click **Characterization** and then press the softkey corresponding to the characterization of the adapter you are using.
10. Click **Confidence Check**.
11. Compare the data trace and memory trace and verify whether measurement is correct.
12. The following is the procedure for comparison when simultaneously displaying the data trace and the memory trace.
    a. Press **Display** > **Display** > **Data & Mem.**
    b. Press **Scale** > **Auto Scale**.
    c. Determine whether the differences between the traces are acceptable. The differences should be read in terms of linear values instead of dB error. If you compare the magnitude of the linear error with the dB delta, the value is very small. So to evaluate the difference between the traces, a linear error scale should be used for comparison instead of the dB error scale.
13. For all of the parameters you want to verify, repeat the procedure.

Connecting ECal module (for verification of S21)



Turning off ECal auto-detect function

The ECal module automatically detects the connection between E5061B's test ports and ECal module's ports. You can turn off this function to set ports manually.

1. Press **Channel Next** (or **Channel Prev**) to select the channel of which you want to turn off the auto-detect function.
2. Press **Cal** > **ECal** > **Orientation** > **Manual**.
3. Specify a test port of the E5061B.
4. Specify an Ecal port for the port of the E5061B you specified.

NOTE If the auto-detect function is turned off, no error is displayed even if the connection is wrong.

Other topics about Calibration with ECal

User-characterized ECal

Other topics about Calibration with ECal

Overview

The E5061B allows you to execute ECal calibration with user-defined characteristics instead of the ECal characteristics defined as the factory default. This feature is called User-characterized ECal, and it is used to execute ECal calibration when an adapter is connected to the ECal module.

Before executing the User-characterized ECal, you have to measure data, such as characteristics when the adapter is connected to the ECal module, and store them to the built-in flash memory of the ECal module as the user characteristics.

Use the following VBA macro to acquire user characteristics and store them to the ECal module's built-in memory.

NOTE The User-characterized VBA macro does not support 4-port ECal modules. Use 2 port ECal modules.

Storage folder	VBA macro name (project name)
D:\Agilent	EcalCharacterization.vba

Precautions to take in using VBA macros

- Never connect/disconnect the USB cable while executing the VBA macro.

  CAUTION In particular, the above precaution must always be observed while the VBA macro is storing data to the ECal module's built-in flash memory; disconnecting the USB cable at this time may damage the ECal module.

- Back up the flash memory contents.
  The VBA macro provides a feature to back up the contents of the ECal module's built-in flash memory. Before storing user characteristics to the ECal module, be sure to use this feature to back up the flash memory's current contents.

Storing user characteristics to the ECal module

Follow these steps to measure the characteristics while an adapter is connected to the ECal module and then to store them to the ECal module's built-in flash memory as user characteristics.

1. Connecting ECal Module

Connect the USB cable between the USB port of the ECal module and that of the E5061B. You can make this connection while the E5061B's power is ON.

2. Setting Stimulus Condition

Set the stimulus condition of the channel for which you want to measure the user characteristics. For optimal accuracy, set the IF bandwidth to 1 kHz or less.

3. Executing Calibration

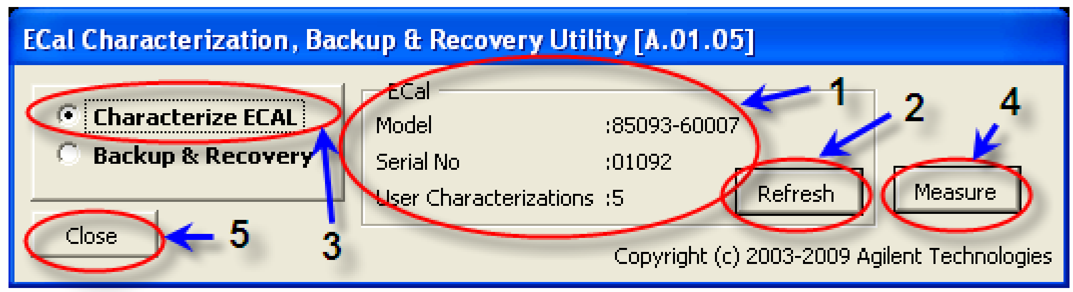
For the channel of which you have set the stimulus condition, execute full 2-port calibration with a mechanical calibration kit when characterizing 2-port ECal. Define the calibration surface as the connector surface connected to each port of the ECal module in the state used to measure the characteristics.

4. Starting the VBA MACRO

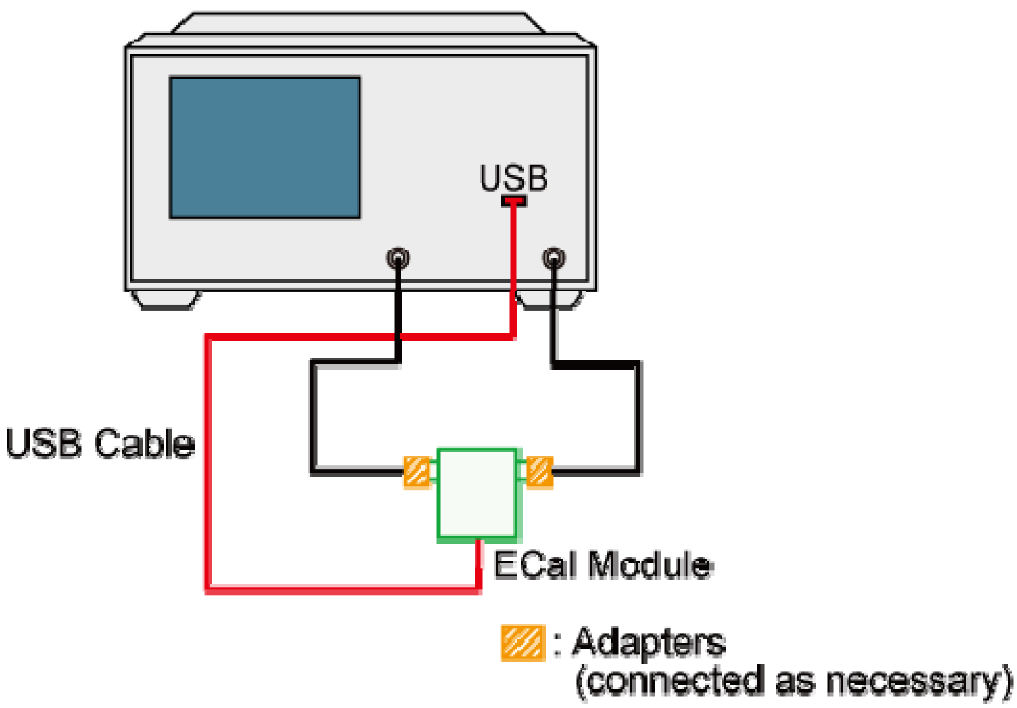
a. Press **Macro Setup** > **Load Project**.
b. The Open dialog box appears. Specify the file name "**D:\Agilent\EcalCharacterization.vba**" and click **Open**.
c. Press **Macro Run** key to start the macro.
d. The ECal (1 in the figure below) displays the information of the ECal module connected to the E5061B.
e. Click **Refresh** (2 in the figure below) to update the information if you have connected another ECal module after the macro has been started.

5. Measuring User Characteristics

a. Select Characterize ECAL (3 in the figure below) to display the User Characteristic Measurement screen.



b. After connecting the adapter to the ECal module as necessary, connect each port of the ECal module and the test port of the E5061B.



e5061b014

c. Click **Measure** (4 in the figure) to start measurement.

NOTE You can select any port of the ECal module and any test port of the E5061B for connection; the E5061B automatically recognizes the connected ports before measurement.

a. When the measurement is complete, the User Characterization Info screen appears.

b. Enter the following information.

Designation	Category	Description
1	Number	Specify a user number (a location number in the memory where you want to store the user characteristics) . If the specified location number is not used for storage, the parts Characterization, Connectors, and Adapter Description are left blank; if already used, the stored contents are displayed.
2	Characterization	Enter the information (operator, used analyzer, and so on) when measuring user characteristics as necessary.
3	Connectors	Select the connector types of the adapters for the ECal module's test ports. Male and female in the list of connected types indicate male and female adapter, respectively. Select "No adapter" if no adapter is used on a port.
4	Adapter Description	Enter the detailed information on the adapters connected to each port as necessary.

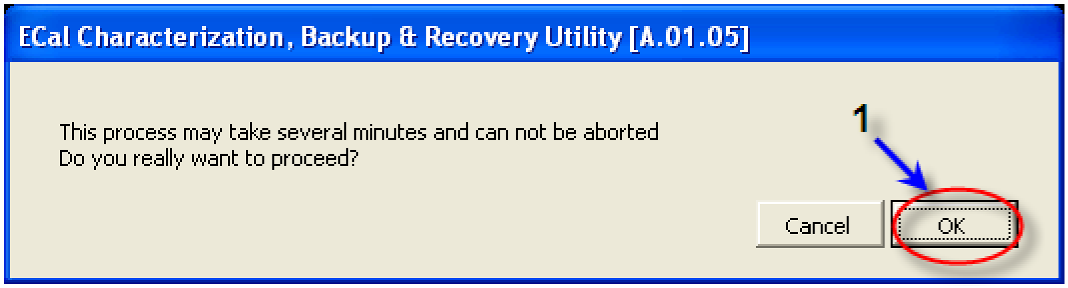
The information you have entered is displayed when checking the user characteristics information by using the key strokes: **Cal** > **Ecal** > **Characterization Info**.

d. Click **Write**.

e. At this time, if user characteristics are already stored for the specified user number, a dialog appears to confirm the overwrite. Click **OK**.

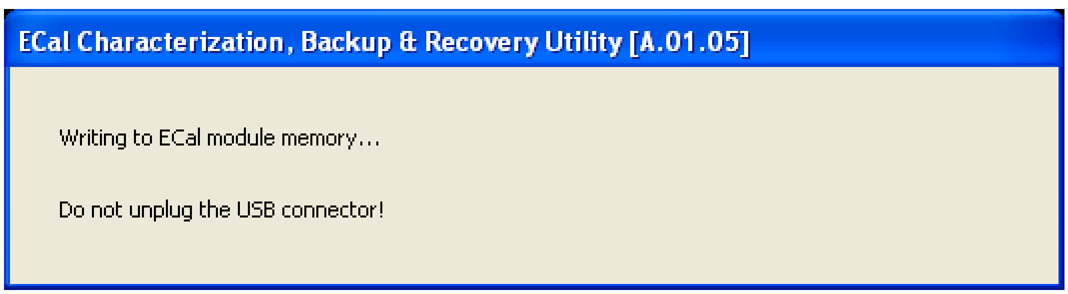
NOTE Although the maximum number of user characteristics stored in the ECal's memory is usually five, this number may be limited by the memory size because the size of user-characteristics data is not fixed and increases in proportion to the number of measurement points. An error occurs when the **Write** button is pressed if the total size added to the new user characteristics exceeds this limitation due to memory size.

f. The following dialog box is displayed to confirm execution. Click **OK** to start storing the user characteristics.

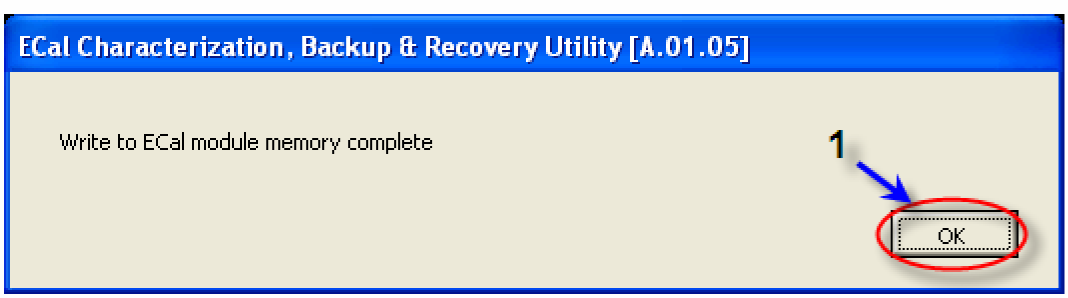


CAUTION Do not disconnect the USB cable or terminate the VBA macro by force while the VBA macro is storing data to the ECal's built-in flash memory. Doing so may damage the ECal module.

g. The following dialog box appears while the VBA macro is storing data to memory. Storing the user characteristics takes a few minutes depending on the amount of data.



h. Another dialog box is displayed to notify completion of data storage. Click **OK**.



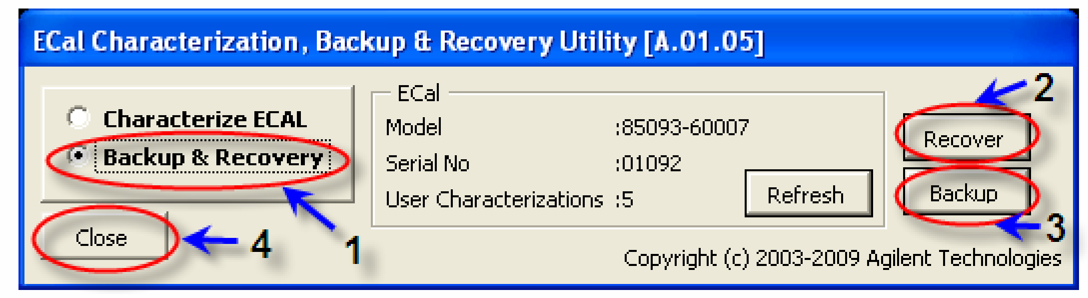
8. Closing the VBA macro

a. Click **Close** (No. 5 in Figure) to close the macro.

Backup and recovery of ECal module's built-in flash memory

Follow these steps to back up the contents of the ECal module's built-in flash memory.

1. Connect the USB cable between the USB port of the ECal module and that of the E5061B. You can make this connection while the E5061B's power is ON.
2. Start the VBA macro according to Starting the VBA MACRO
3. Select **Backup Flash ROM** (No. 1 in the following figure) to display the Backup screen.



Recovery

1. Click **Recover** (No. 2 in the figure above.).
2. The Open dialog box appears. Enter the file name of the contents you want to recover and Click **Open**. If the serial number information stored in the file does not match that of the ECal module connected to the E5061B, a confirmation dialog box appears. Click **OK** to continue the recovery only if a mismatch between these serial numbers is allowed.
3. The dialog box is displayed to confirm execution. Click **OK** to start the recovery of the flash memory. The dialog box appears while the VBA macro is storing data to the memory. The recovery of the flash memory takes a few minutes depending on the amount of data.
4. The Completion screen appears. Click **OK**.

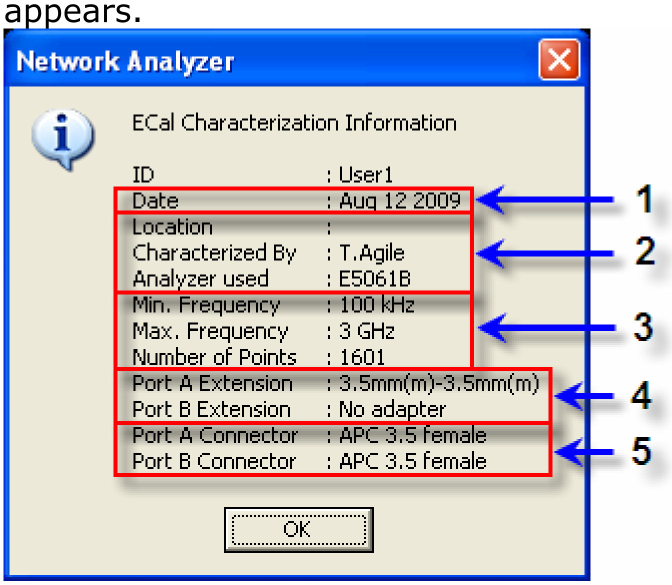
Backup

1. Click **backup**.
2. The Save As dialog box appears. Enter the name of the file you want to save and press **Save**.
3. Click **Close** to close the macro.

Executing User-characterized ECal

The execution procedure for the User-characterized ECal is the same as for normal ECal except that it requires the user characteristics to be selected in advance.

1. Press **Channel Next**/**Channel Prev** to select the channel of which you want to execute calibration.
2. Press **Cal** > **ECal** > **Characterization.**
3. Select a user characteristic which is specified by User Characterization Info screen.
4. To check the information on the user characteristics you have selected, click **Characterization Info**. The following dialog box appears.



1:	The date when the user characteristics are measured
2:	The information you entered in Characterization (1 of User Characterization Info screen)
3:	The stimulus conditions when the user characteristics are measured
4:	The information you entered in Adapter Description (4 of User Characterization Info screen)
5:	The information you entered in Connectors (3 of User Characterization Info screen)

Advanced Calibrations

Modifying Calibration Kit Definition

- Definition of Terms
- Defining Parameters for Standards
- Redefining a Calibration Kit

Other topics about Advanced Calibration

In most measurements, the user can use pre-defined calibration kits as they are. However, it may be necessary to change the definition of a calibration kit (or create a new one) when changing the pre-defined connector between male and female (e.g. from OPEN (f) to OPEN (m)) or when a special standard is used or a high degree of accuracy is demanded. When it is necessary to change the definition of a calibration kit that contains a calibration device but no calibration kit model, the user must fully understand error correction and the system error model.

A user-defined calibration kit may be used in the following circumstances:

- When the user wants to use connectors other than those pre-defined in the calibration kits for the E5061B (e.g., a SMA connector).
- When the user wants to use different standards in place with one or more standards pre-defined in the E5061B. For example, when three offset SHORT standards are used instead of OPEN, SHORT, and LOAD standards.
- When the user wants to modify the standard model of a pre-defined calibration kit and turn it into a more accurate model. It is possible to perform better calibration if the performance of the actual standard is better reflected in the standard model. For example, you may need to define the 7-mm LOAD standard as 50.4 Ω instead of 50.0 Ω.

Definition of Terms

The terms used in this section are defined as follows:

Standard

An accurate physical device, for which the model is clearly defined, used to determine system errors. With the E5061B, the user may define up to 21 standards per calibration kit. Each standard is numbered from 1 to 21. For example, standard 1 for the 85033E 3.5-mm calibration kit is a SHORT standard.

Standard type

The type of standard used to classify a standard model based on its form and construction. Five standard types are available: SHORT, OPEN, LOAD, delay/THRU, and arbitrary impedance.

Standard coefficient

The numeric characteristics of the standard used in the selected model. For example, the offset delay (32 ps) of the SHORT standard in the 3.5-mm calibration kit is a standard coefficient.

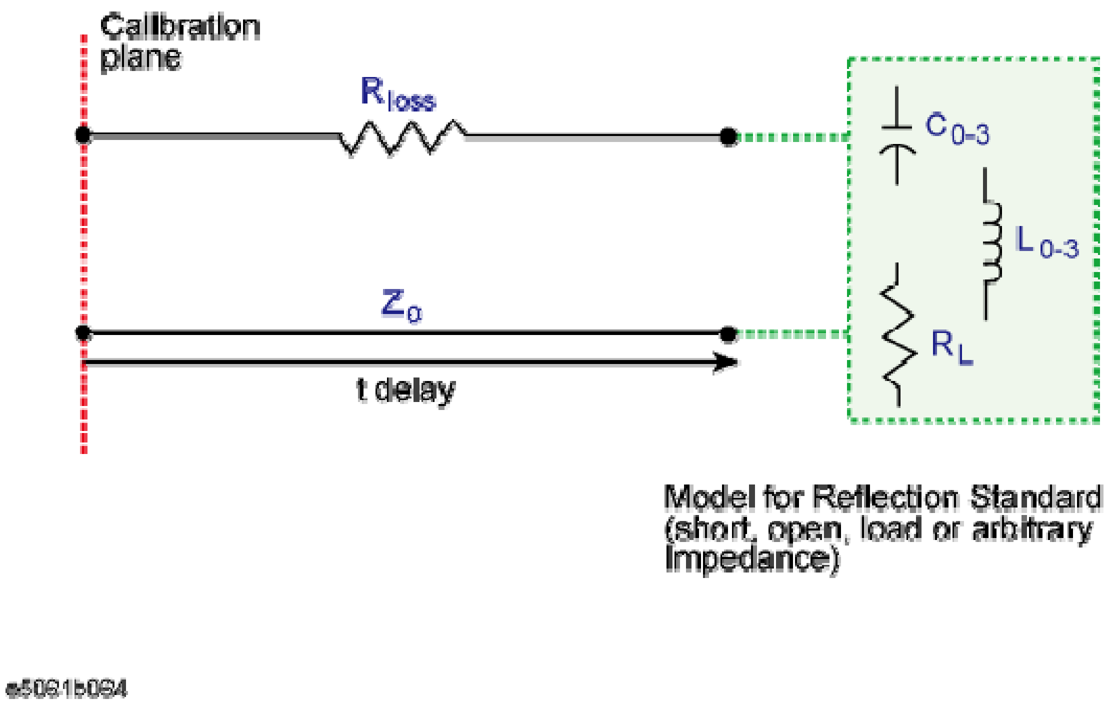
Standard class

A group of standards are used in a calibration process. Class allows you to use the different standard for each port. For each class, the user must select the standards to use from the available standards.

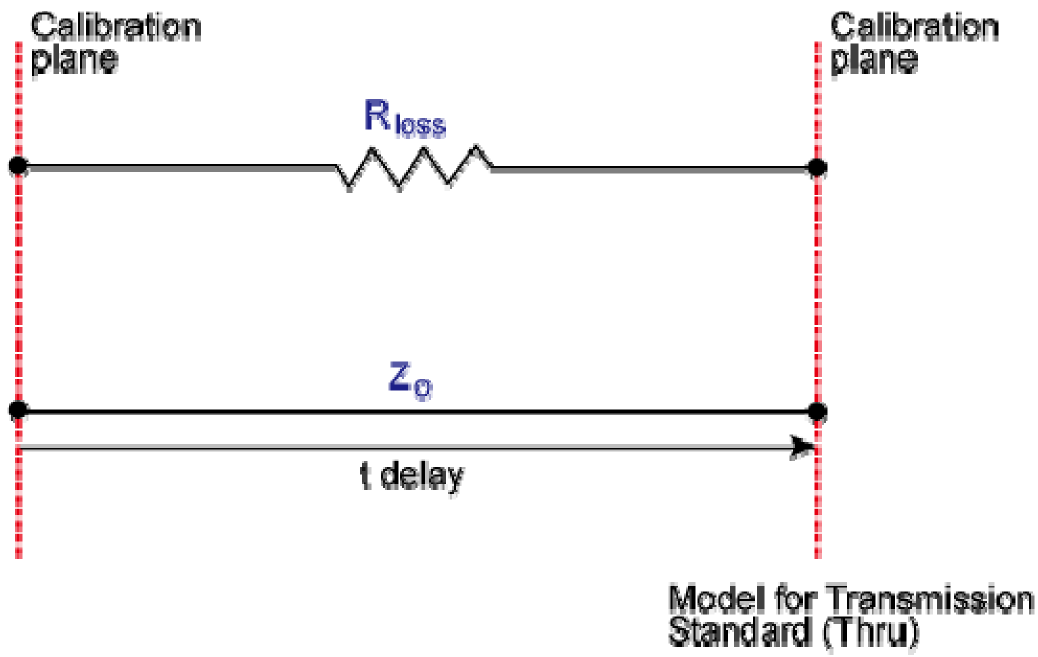
Defining Parameters for Standards

The following figures show the parameters used in defining the standards.

Reflection Standard Model (SHORT, OPEN, or LOAD)



Transmission Standard Model (THRU)



e5061b065

Z0

The offset impedance between the standard to be defined and the actual measurement plane. Normally, this is set to the system's characteristic impedance.

Delay

The delay that occurs depending on the length of the transmission line between the standard to be defined and the actual measurement plane. In an OPEN, SHORT, or LOAD standard, the delay is defined as one-way propagation time (sec.) from the measurement plane to the standard. In a THRU standard, it is defined as one-way propagation time (sec.) from one measurement plane to the other. The delay can be determined through measurement or by dividing the exact physical length of the standard by the velocity coefficient.

Loss

This is used to determine the energy loss caused by the skin effect along the length (one-way) of the coaxial cable. Loss is defined using the unit of ohm/s at 1 GHz. In many applications, using the value 0 for the loss should not result in significant error. The loss of a standard is determined by measuring the delay (sec.) and the loss at 1 GHz and then substituting them in the formula below.

$$Loss\left(\frac{\Omega}{s}\right) = \frac{loss(dB) \times Z_0(\Omega)}{4.3429(dB) \times delay(s)}$$

C0, C1, C2, C3

It is extremely rare for an OPEN standard to have perfect reflection characteristics at high frequencies. This is because the fringe capacitance of the standard causes a phase shift that varies along with the frequency. For internal calculation of the analyzer, an OPEN capacitance model is used. This model is described as a function of frequency, which is a polynomial of the third degree. Coefficients in the polynomial may be defined by the user. The formula for the capacitance model is shown below:

$$C = (C0) + (C1 \times F) + (C2 \times F^2) + (C3 \times F^3)$$

F: measurement frequency
C0 unit: (Farads) (constant in the polynomial)
C1 unit: (Farads/Hz)
C2 unit: (Farads/Hz^2)
C3 unit: (Farads/Hz^3)

L0, L1, L2, L3

It is extremely rare for a SHORT standard to have perfect reflection characteristics at high frequencies. This is because the residual inductance of the standard causes a phase shift that varies along with the frequency. It is not possible to eliminate this effect. For internal calculation of the analyzer, a short-circuit inductance model is used. This model is described as a function of frequency, which is a polynomial of the third degree. Coefficients in the polynomial may be defined by the user. The formula for the inductance model is shown below:

$$L = (L0) + (L1 \times F) + (L2 \times F^2) + (L3 \times F^3)$$

F: Measurement frequency
L0 unit: [Henry] (the constant in the polynomial)
L1 unit: [Henry/Hz]
L2 unit: [Henry/Hz^2]
L3 unit: [Henry/Hz^3]

In most existing calibration kits, THRU standards are defined as "zero-length THRU," i.e., the delay and loss are both "0". Such THRU standard does not exist, however, calibration must be done with two test ports interconnected directly.

NOTE The measurement accuracy depends on the conformity of the calibration standard to its definition. If the calibration standard has been damaged or worn out, the accuracy will decrease.

Redefining a Calibration Kit

This section provides the procedure to change the definition of a calibration kit.

a. Select and define a calibration kit

1. Press **Cal** > **Cal Kit**, then select the calibration kit to be redefined.
2. Click **Modify Kit**.
3. If necessary ,click **Label Kit** and type a new label for the calibration kit.

b. Select the standard type and define standard coefficient

4. Click **Define STDs** and select the standard number to be redefined.
5. If necessary, click **Label**, then type your desired name for the selected standard.
6. Click **STD Type**, then select the **type of standard**.
7. Set the **standard coefficient**.
8. Repeat steps 4 to 7 to redefine all standards for which changes are necessary, then click **Return**.

c. Define standard class

9. Click **Specify CLSs**, then select the **class**.
10. Select the test port. Select **Set All** to use the same standard for all test ports.
11. Select the standards to be registered in the class from the standard number to be redefined. Define classes for all test ports that need to be redefined, then click **Return**.
12. Repeat the steps 9 to 12 to redefine all classes that need to be modified, then click **Return**.

Preset the definition for calibration kits

1. Press **Cal** key.
2. Click **Cal Kit**, then select a calibration kit.
3. Click **Modify Cal Kit** > **Restore Cal Kit** to preset the selected kit definition at the factory setting.

Partial Overwrite Calibration

- Overview
- Procedure

Other topics about Advanced Calibration

Overview

The partial overwrite function is used to perform partial measurement after the execution of calibration, and it overwrites the calibration coefficients.

There are three types of calibration coefficients: Er, Es, Ed for reflection, Et for transmission, and Ex for isolation. If some of them do not provide satisfactory calibration, you can use this function to re-calculate the calibration coefficients by measuring an applicable standard only instead of measuring all the standards again.

> NOTE When the calibration coefficients become inappropriate over time or the status on the E5061B side from the calibration surface changes due to replacement of a cable or connector, you also need to perform thru measurement when partial overwrite is required for reflection or isolation measurement.

> NOTE The adapter removal and partial overwrite function is only available when calibration status is [**Cor**] and not for [**C?**] or [**C!**].
>
> Partial overwrite is not available if no calibration has been done. You cannot append calibration coefficients to previous calibrations. For example, you cannot realize 2-port calibration by performing additional calibration for 1 port after the execution of full 1-port calibration. The partial overwrite function is used to make measurements for previous calibration coefficients and overwrite them.

Procedure

Follow these steps to execute the partial overwrite function. The example demonstrates re-calibration thru calibration only for exiting 2 port full calibration:

1. Press **Channel Next** (or **Channel Prev**) to select the channel of which you already have performed full 2 port calibration.
2. Press **Cal** >**Calibrate** > **2-Port Cal**.
3. Click **Transmission**.
4. Make a thru connection between the ports 1 and 2.

5. Click **Port 1-2 Thru**.
6. Click **Return**.
7. Click **Overwrite** to finish the re-calibration for the full 2-port calibration. At this point, the calibration coefficients are re-calculated with the new thru calibration and saved.

.

Adapter Removal-Insertion

- About Adapter Removal
- About Adapter Insertion
- Procedure for Adapter Removal/Insertion
- Difference between Traditional Network Analyzer & E5061B Adapter Calibration

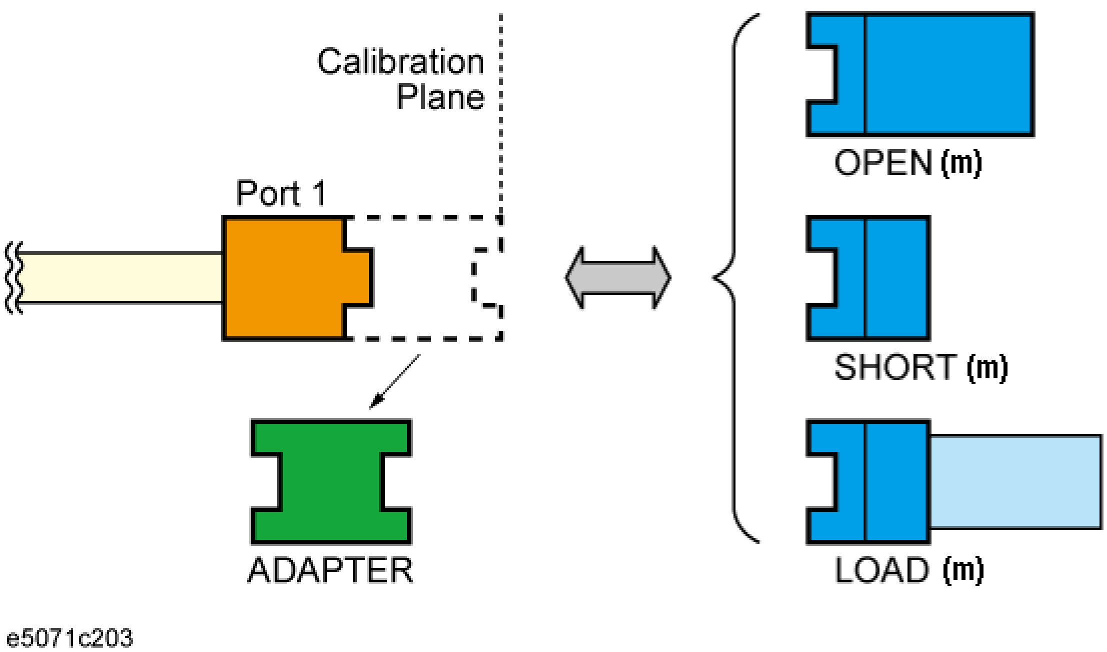
Other topics about Advanced Calibration

About Adapter Removal

Adapter Removal is a technique used to remove any adapter characteristics from the calibration plane. The E5061B uses the following adapter removal process to remove the adapter characteristics:

1. Perform calibration with the adapter in use.
2. Remove the adapter from the port and measure Open, Short, and Load values to determine the adapter's characteristics.
3. Remove the obtained adapter characteristics from the error coefficients in a de-embedding fashion.

Open, Short, and Load values measured with the adapter removed



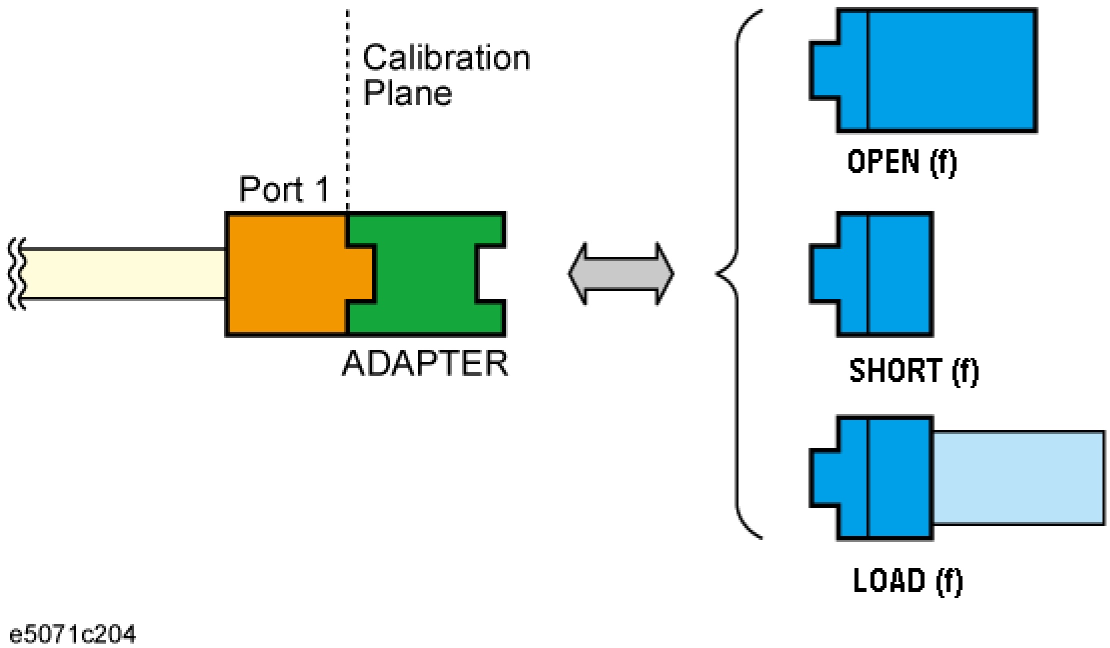
NOTE The adapter removal and partial overwrite function is only available when the calibration status is [**Cor**] and not for [**C?**] or [**C!**].

About Adapter Insertion

The above described method also makes it possible to add adapter characteristics to a port with n-port full calibration. This allows you to make a measurement with the adapter. E5061B uses the following adapter insertion process to insert the adapter characteristics:

1. Perform calibration without adapter in use.
2. Insert the adapter to the port and measure Open, Short, and Load values to determine the adapter's characteristics.
3. Insert the obtained adapter characteristics to the error coefficients in an embedding fashion.

Open, Short, and Load values measured with adapter attached



In order to determine the adapter characteristics (with four unknown parameters) by making three measurements (Open, Short, and Load), the adapter must satisfy the following requirements:

1. Adapter must be Reciprocal (with S21 and S12 equal) in nature. It should have a consistent behavior, and independent of the direction from which it is used.
2. The electrical length of Adapter should be known with the accuracy of ±1/4 of wavelength.

Adapter removal and insertion is not available in Gain-Phase measurement.

Procedure for Adapter Removal/Insertion

The S parameter of a reciprocal adapter can be determined when the following data is available:

- Open, Short, and Load measurements.
- Actual values derived from the **CalKit** definitions.
- An approximate length of the adapter.
- Nature of the intended operation: removal or insertion.

To use Adapter Removal/Insertion, follow the below procedure:

1. Perform a full n-Port calibration using your calibration kit so that the port used to conduct adapter insertion/removal is calibrated.

   NOTE When you need to remove an adapter from the calibration plane (adapter removal) to connect your DUT, perform the calibration with an adapter so that you can make a calibration with your calibration kit. When you need to add an adapter into the calibration plane (adapter insertion) to connect your DUT, perform the calibration without an adapter.

2. Press **Cal** > **Calibrate** > **Adapter Removal**.
3. Select a proper standard Calkit you need to use from **CalKit** (e.g. 85033E) to characterize the adapter. The calibration kit is used at the plane from which the adapter is removed (adapter removal), or the plane in which adapter is inserted (adapter insertion).
4. Select **Port 1** or **Port 2** which you want to insert/remove Adapter characteristics from/to. A * sign appears in front of the port is the valid port to conduct adapter removal/insertion as the full n-port calibration has been performed on the port.
5. Connect Open, Short, and Load of the selected **Calkit** (e.g. 85033E) with the selected port respectively and click/press **Open**, **Short**, and **Load** respectively. ENA measures the cal kit standard, calculates the adapter characteristics, and then conducts adapter removal/insertion.

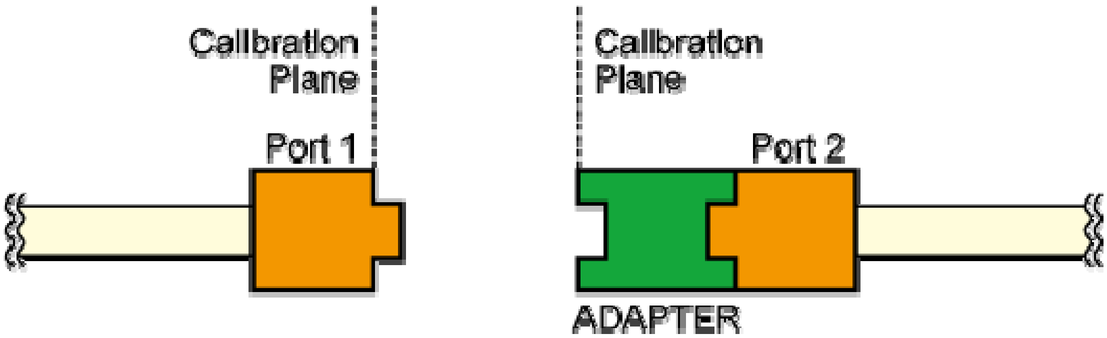
   NOTE A checkmark √ appears in (**Open**, **Short**, and **Load**) menu after each type of calibration is completed.

7. Click **Done** to complete the process.
8. In cases where the auto phase fails, select **Rotate Adapter** to move the adapter's phase (which is removed or inserted) to 180 degrees.
9. When you want to set the adapter length manually, press **Length** then input the value. Input 0s set at the Auto mode.

Difference between Traditional Network Analyzer & E5061B Adapter Calibration

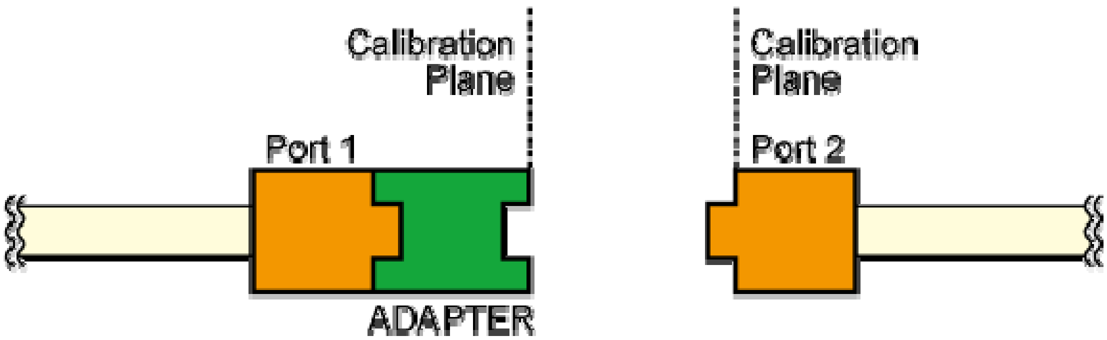
Usually, two-port network analyzers remove the adapter characteristics by performing two sets of Full 2 Port Calibration as shown below:

Calibration performed with the adapter connected to Port2



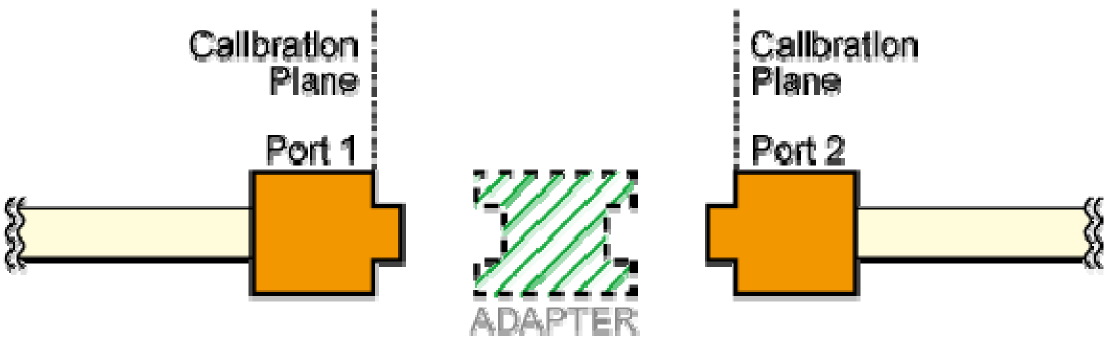
e5071c200

Calibration performed with the adapter connected to Port1



e5071c201

Removing adapter characteristics using two (above) sets of calibration



e5071c202

However, this method is not suitable for a multi-port network analyzer because it requires Full 2 Port Calibration as many as twice the number of port combinations. Therefore, the E5061B uses an advanced method to remove the Adapter characteristics described in About Adapter Insertion.

Making Measurements

Making Measurements

- Setting Up Trigger
- Making Averaging Measurement with Single Trigger
- Distributing Trigger to External Device
- Making Trigger
- Making One Time DC Measurement

Setting Up Trigger

- Overview
- Selecting Trigger Source
- Setting Trigger Mode (Single/Continuous)
- Making Trigger to Active Channel Only (Trigger Scope)
- Making One Point Measurement with Single Trigger (Point Trigger)
- Setting Trigger Delay

Other topics about Making Measurement

Overview

The E5061B has one trigger source. When this trigger source detects a trigger signal that has occurred, a sweep or point measurement is performed for channels.

The execution of measurement for each channel does not depend on whether the channel is displayed. Channels that have been activated can be measured even if they are not displayed.

For each channel, a sweep is performed only for the stimulus ports required to update the parameters of the displayed trace.

Selecting Trigger Source

The trigger source generates a cue signal that initiates a measurement process. Four types of trigger sources are available:

1. Press Trigger > **Trigger Source**, then select the desired **trigger source**.
2. When **External** is selected as a trigger source, click **Ext Trig Input** to select **trigger polarity**.

NOTE The setting for trigger polarity is NOT valid for the external trigger from the 24 Bit I/O (Hander).

Setting Trigger Mode (Single/Continuous)

You can set the trigger mode for each channel independently. This allows you to control the operation of each channel after a trigger signal is detected by setting the channel's status with the trigger mode.

1. Press **Channel Next** (or **Channel Prev**) to select the channel for which the trigger mode will be set.
2. Press **Trigger**, then select the desired **trigger mode**.
3. Repeat the procedure until each channel is set to its trigger mode.

Making Trigger to Active Channel Only (Trigger Scope)

The trigger scope specifies the scope of the triggering, whether it is for all channels or for the active channel.

For example, when **Trigger** > **Continuous** is selected for all the channels and the trigger scope is set to active channel, a measurement channel is automatically changed by changing an active channel.

1. Press **Trigger** > **Trigger Scope**, then select the desired **trigger scope**.

Making One Point Measurement with Single Trigger (Point Trigger)

The point trigger provides a point measurement at every trigger, and it can be used to change the trigger event to point trigger mode.

1. Press **Trigger** > **Trigger Event**, then select the desired **trigger event**.

NOTE When the trigger source is the internal trigger, the point trigger does not work.

Setting Trigger Delay

Set the external trigger delay time at each point. The trigger delay works when the trigger source is set to external.

1. Press **Trigger** > **Trig Delay**.
2. Enter an external trigger delay time.

See the timing chart for sweep.

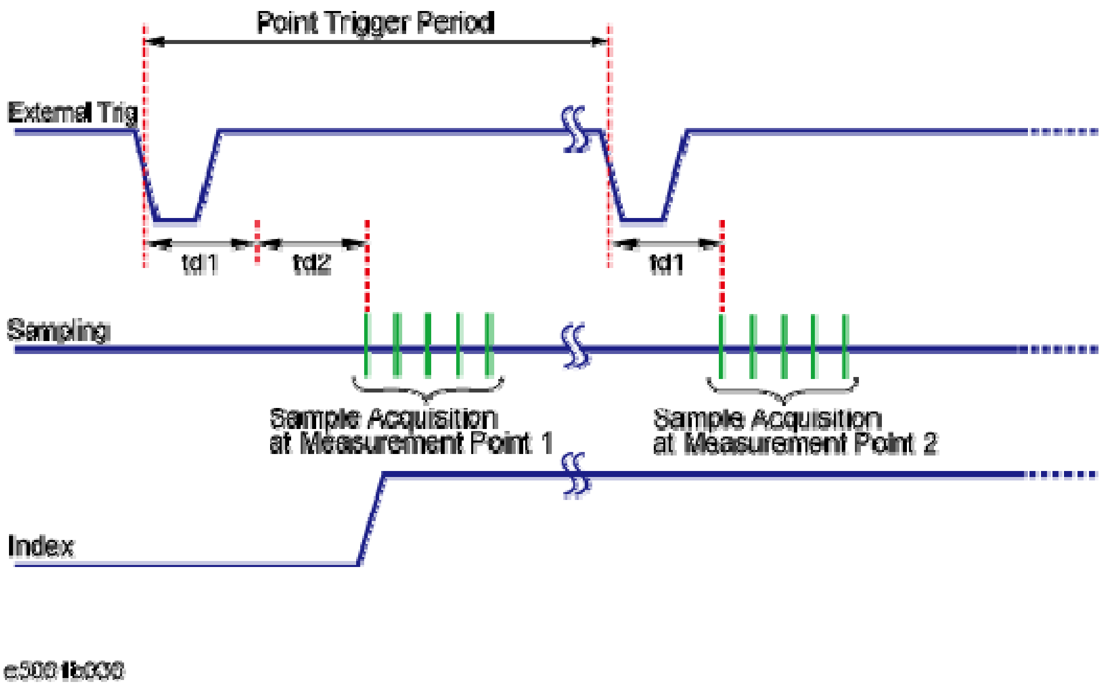
Trigger Delay Time and Point Trigger Interval

External trigger pulses which are supplied until the next measurement becomes ready after the start of a one-point measurement, are ignored, and the next trigger is generated by a pulse supplied after the completion of the one-point measurement.

The time until the next trigger can be accepted after the start of a one-point measurement depends on the IFBW and other settings of the analyzer. For example, in the case of a frequency's zero-span measurement, the time until the next measurement is ready after the start of a one-point measurement is obtained by dividing the time required for a single sweep in On Sweep mode, instead of On Point mode, by the number of measurement points. If you use the point trigger function with external trigger pulses that are wider than this time, point trigger measurement is performed at each pulse input.

The figure below shows the timing chart of an external trigger when the point trigger function is on.

Timing chart of external trigger (trigger source = external)



The table below describes signals and time as shown in the above figure.

Signal, time	Description
External Trig	External trigger signal to be supplied.
Sampling	Time while the E5061B is actually performing measurement.
Index	Index signal of the handler I/O port. When the point trigger function is ON, it goes to the High level only before starting the measurement of the first sweep point and returns to the Low level after completing the measurement of all the measurement points.

Point Trigger Period	Time until the E5061B is ready to accept a trigger for the next measurement point. The value depends on the measurement conditions and the settings of the E5061B.
td1	Time set as the external trigger delay time.
td2	Time for sweep delay

Making Averaging Measurement with Single Trigger

- Overview
- Averaging Trigger Function

Other topics about Making Measurement

Overview

The averaging trigger function is used to execute the sweep the number of times specified by the averaging factor with a single trigger when the sweep averaging function is **ON**.

Averaging Trigger	Function
ON	Performs the sweep the number of times specified by the averaging factor with a single trigger.
OFF	Performs the sweep once with a single trigger.

The averaging factor is cleared before the start of measurement.

NOTE When the point trigger function is ON, its setting has priority, and you need to generate triggers based on "(number of measurement points) × (averaging factor)".

NOTE When the sweep averaging function is OFF, sweep is performed only once even if the averaging trigger function is set to ON.

NOTE The averaging trigger function is valid for all the channels. Note that you can set the sweep averaging function for each channel.

Averaging Trigger Function

Setting Averaging Trigger Function

When the sweep averaging function is **ON**, follow these steps to set the averaging trigger function.

1. Press **Avg** > **Avg Trigger**.
2. Click **ON** to activate the averaging trigger.

Executing Averaging Measurement

1. Press **Trigger**.
2. Click **Single**. The averaging factor is cleared before the start of measurement, the sweep is executed the number of times specified by the averaging factor, and then the instrument waits for the next trigger.

Distributing Trigger to External Device

- Overview
- Setting External Trigger Output

Other topics about Making Measurement

Overview

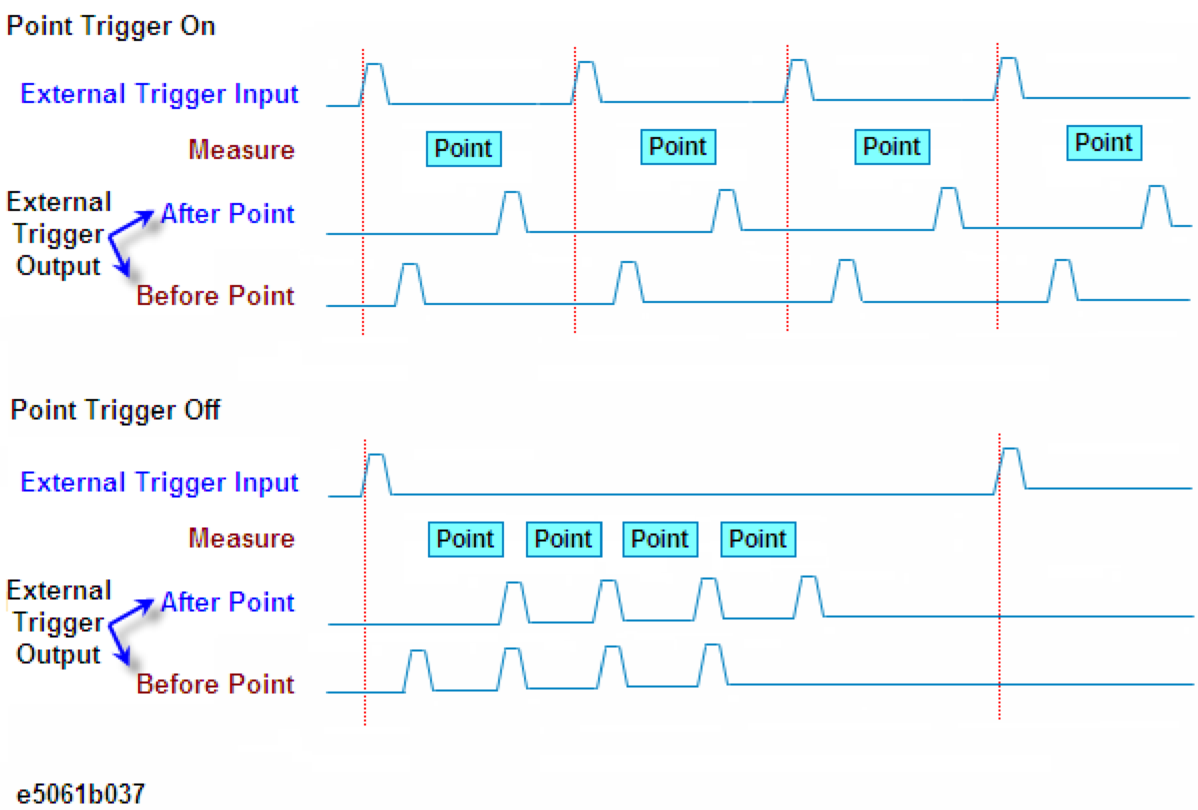
The External Trigger Output port (located at the rear panel) can be used to provide trigger to an external device. This is useful in cases where an external device needs to be triggered through the E5061B.

Setting External Trigger Output

1. Press **Trigger** key, and then set the value of **Ext Trig Output** as **ON**.
2. Click **Polarity**, and then select **polarity**.
3. Click **Position**, and then select **position**.
4. Click **Pulse Width** to define the pulse width of the signal.

Difference between After and Before Point Settings

External trigger output function displays the difference in results by setting the Point trigger as follows:



Making Trigger

- Generating the Trigger
- Sweep Order in Each Channel

Other topics about Making Measurement

Generating the Trigger

To make a measurement, it is necessary to generate a trigger by using the selected trigger source. Once the internal trigger is selected, a series of triggers is continuously generated as soon as the setting becomes effective.

Pressing **Trigger** > **Restart** during a sweep forces the analyzer to abort the sweep.

Sweep Order in Each Channel

In a channel, each test port is set to a stimulus port in the order of port number, and updates each trace accordingly.

Sweep Order	Stimulus Port	Updated Trace
1	Port 1	S11, S21
2	Port 2	S12, S22
3	LF Out	T/R, R, T

NOTE If full 2-port error correction is in effect, no trace between calibrated ports is updated until the last calibrated port is swept as the stimulus port.

NOTE Sweep is not executed for stimulus ports that are not required for updating traces.

Making DC Measurement at Sweep End (Option 3L5 only)

- DC Measurement at end of sweep
- Clear the DC measurement Data

Other topics about Making Measurement

DC Measurement at end of sweep

E5061B option 3L5 allows you to measure DC level at the end of measurement sweep. When this function is turned on, after the sweep is finished, the DC level is measured in the condition of the last point of sweep, then result is displayed at the upper right corner of the screen.

1. Press **Meas** > **DC Monitor Setup.**
2. Click **Function** to select **the port** for DC measurement.
3. Click **DC Monitor Sweep End** > **Monitor** to turn ON the DC monitor function.
4. The result is displayed at the upper right corner of the screen.

Clear the DC measurement Data

You can clear the DC measurement data. This can be used when you want to clear the data at the hold status.

1. Click **Meas** > **DC Monitor Setup** > **DC Monitor Sweep End** > **Clear** to reset the DC measurement at sweep end.

Analyzing Data

Analyzing Data

- Analyzing Data on the Trace Using the Marker
- Searching for Positions that Match Specified Criteria
- Determining the Bandwidth of the Trace (Bandwidth Search)
- Determining the Bandwidth of the Trace (Notch Search)
- Determining the Mean, Standard Deviation, and p-p of the Trace
- Obtaining Span, Gain, Slope, and Flatness between Markers
- Obtaining Loss, Ripple, and Attenuation of the RF Filter
- Comparing Traces/Performing Data Math
- Performing Parameter Conversion of Measurement Results
- Using Limit Test
- Using Bandwidth Test
- Using Ripple Test
- Using Equation Editor

Analyzing Data on the Trace Using the Marker

- About Marker Functions
- Reading Marker Values on Trace
- Reading Relative Value from Reference Point on Trace
- Reading Actual Measurement Point/Value Interpolated between Measurement Points
- Setting up Markers for Each Trace/Setting up Markers for Coupled Operation between Traces
- Listing Marker Values in All Displayed Channels
- Specifying Display Position of Marker Values
- Aligning Marker Value Display
- Displaying All Marker Values for Displayed Traces

Other topics about Data Analysis

About Marker Functions

The marker can be used in the following ways:

- Reading a measured value as numerical data (as an absolute value or a relative value from the reference point)
- Moving the marker to a specific point on the trace (marker search)
- Analyzing trace data to determine a specific parameter
- Using the value of the marker to change the stimulus (sweep range) and scale (value of the reference line)

For the procedure used to change the sweep range and scale by using the marker, refer to Setting the Sweep Range Using the Marker and Setting the value of a reference line using the marker.

The E5061B is capable of displaying up to 10 markers including the reference marker on each trace. Each marker has a stimulus value (the value on the X-axis in rectangular display format) and a response value (the value on the Y-axis in rectangular display format). The Smith chart and polar formats have two marker response values each (log amplitude and phase).

Reading Marker Values on Trace

You can read the value of a marker displayed on the trace.

In rectangular display format, the marker response value is always in the same data format as that of the Y-axis. On the contrary, one format of the marker response values (two values: main and auxiliary) can be selected from several types. The selection is performed in the data format.

Softkey for selecting data format	Marker response value	
	Main	Auxiliary
Smith > **Lin / Phase**	Linear amplitude	Phase
Smith > **Log / Phase**	Log amplitude	Phase
Smith > **Real / Imag**	Real component	Imaginary component
Smith > **R + jX**	Resistance	Reactance
Smith > **G + jB**	Conductance	Susceptance
Polar > **Lin / Phase**	Linear amplitude	Phase
Polar > **Log / Phase**	Log amplitude	Phase
Polar > **Real / Imag**	Real component	Imaginary component

To set up data formats, refer to Selecting a Data Format.

Activating the marker on the Trace

1. Press **Channel Next** (or **Channel Prev**) and **Trace Next** (or **Trace Prev**) to activate the channel on which a marker is used.
2. Press **Marker** key. At this point, marker 1 is turned on and becomes active (you can operate the marker). When using marker 1, you can omit the next step.
3. Select a marker and turn it ON. The softkey used to turn on a marker is also used to activate that marker.

Moving the marker

1. Change the marker stimulus value. This operation enables you to move the marker to a point on the selected trace.

2. Read the marker stimulus value and marker response value displayed in the upper-left part of the trace screen.

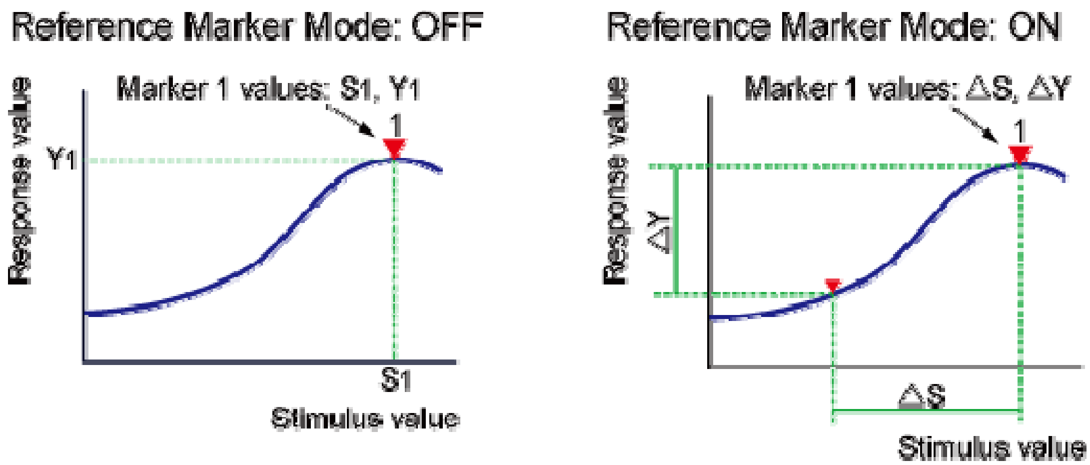
Turning off the marker

1. Press **Marker** key.
2. Click **Clear Marker Menu** and then click one of the options.

NOTE In the preset configuration, the marker settings on traces in a channel are coupled (Marker Couple is turned on). For marker coupling, refer to Setting up markers for each trace/Setting up markers for coupled operations between traces.

Reading Relative Value from Reference Point on Trace

You can convert the marker reading into a relative value from the reference point.



e5071c310

Converting From a Reference Point to a Relative Value

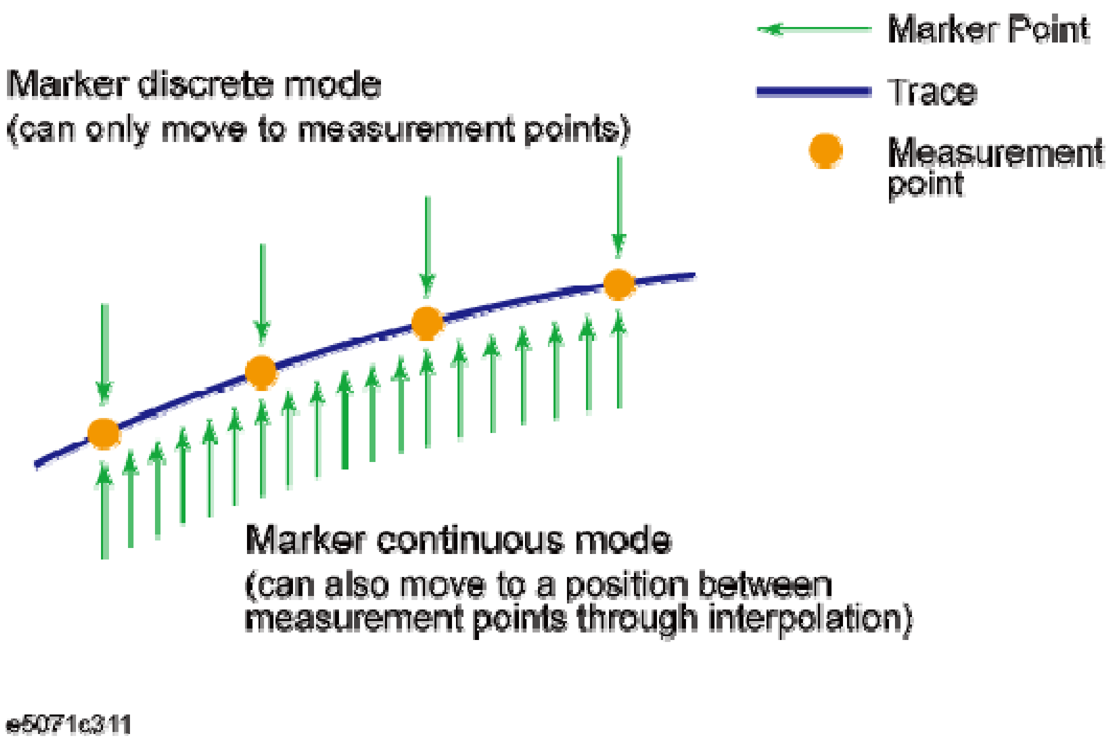
1. Activate the reference marker.
2. Move the reference marker at the point to be used as the reference.
3. Click **Ref Marker Mode** to turn on the reference mode.
4. With the reference mode turned ON, the stimulus values and response values are indicated in relative values referred to by the position of the reference marker.
5. Activate your desired marker, then move it to your desired position.

NOTE Pressing **Marker** > **Marker -> Ref Marker** enables you to place the reference marker at the position of the currently active marker. The reference mode then turns ON automatically.

Reading Actual Measurement Point/Value Interpolated between Measurement Points

The point on the trace on which a marker can be placed differs depending on how the discrete marker mode is set up.

Value	Description
Turning on discrete mode (**Discrete ON**)	A marker moves only between actual measurement points. When a specific marker's stimulus value is specified as a numerical value, the marker is placed at the measurement point closest to the specified value. A marker placed between interpolated points with the discrete mode OFF automatically moves to the nearest measurement point when the discrete mode is turned ON.
Turning off discrete mode (**Discrete OFF**)	The marker can move from one actual measurement point to another. Because it is interpolated, it can also move in the space between measurement points.



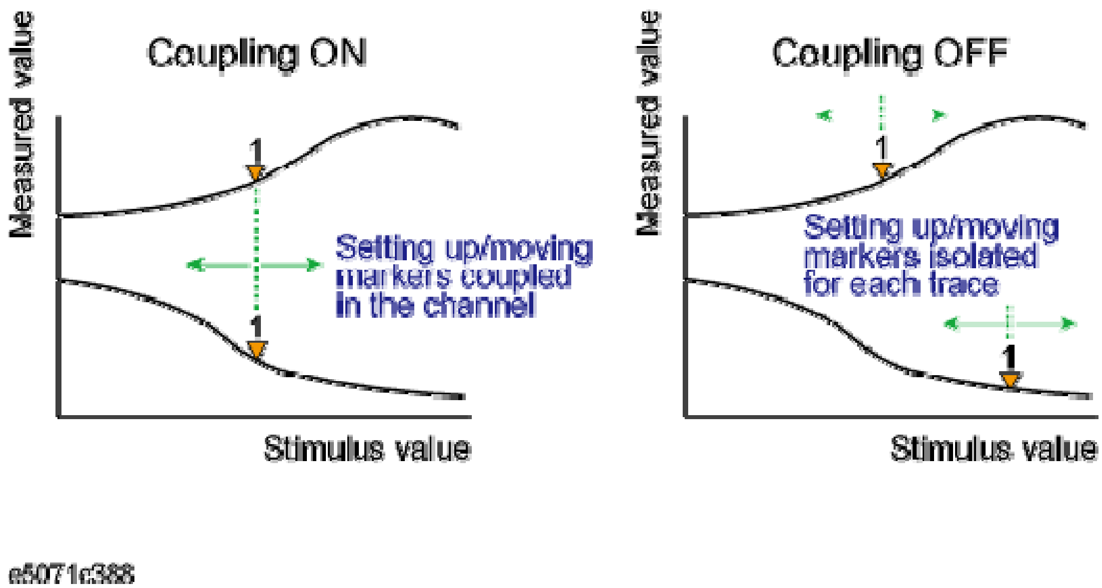
Turning Discrete Mode On or Off

1. Press **Channel Next** (or **Channel Prev**) and **Trace Next** (or **Trace Prev**) to activate the trace on which the discrete mode is set up.

2. Press **Marker Fctn**.
3. Click **Discrete** to turn the discrete mode ON or OFF.

Setting up Markers for Each Trace / Setting up Markers for Coupled Operation between Traces

Makers can be set up and moved either in coupled operation for all traces in a channel or independently for each trace.



Value	Description
Marker Couple is ON (**Coupling ON**)	Markers are set up and moved in coupled operation on all the traces in a channel.
Marker Couple is OFF (**Coupling OFF**)	Markers are set up and moved independently for each trace.

Turning Marker Coupling On or Off

1. Press **Channel Next** (or **Channel Prev**) to activate the channel on which the marker couple is set.
2. Press **Marker Fctn**.
3. Click **Couple** to turn the marker coupling ON or OFF.

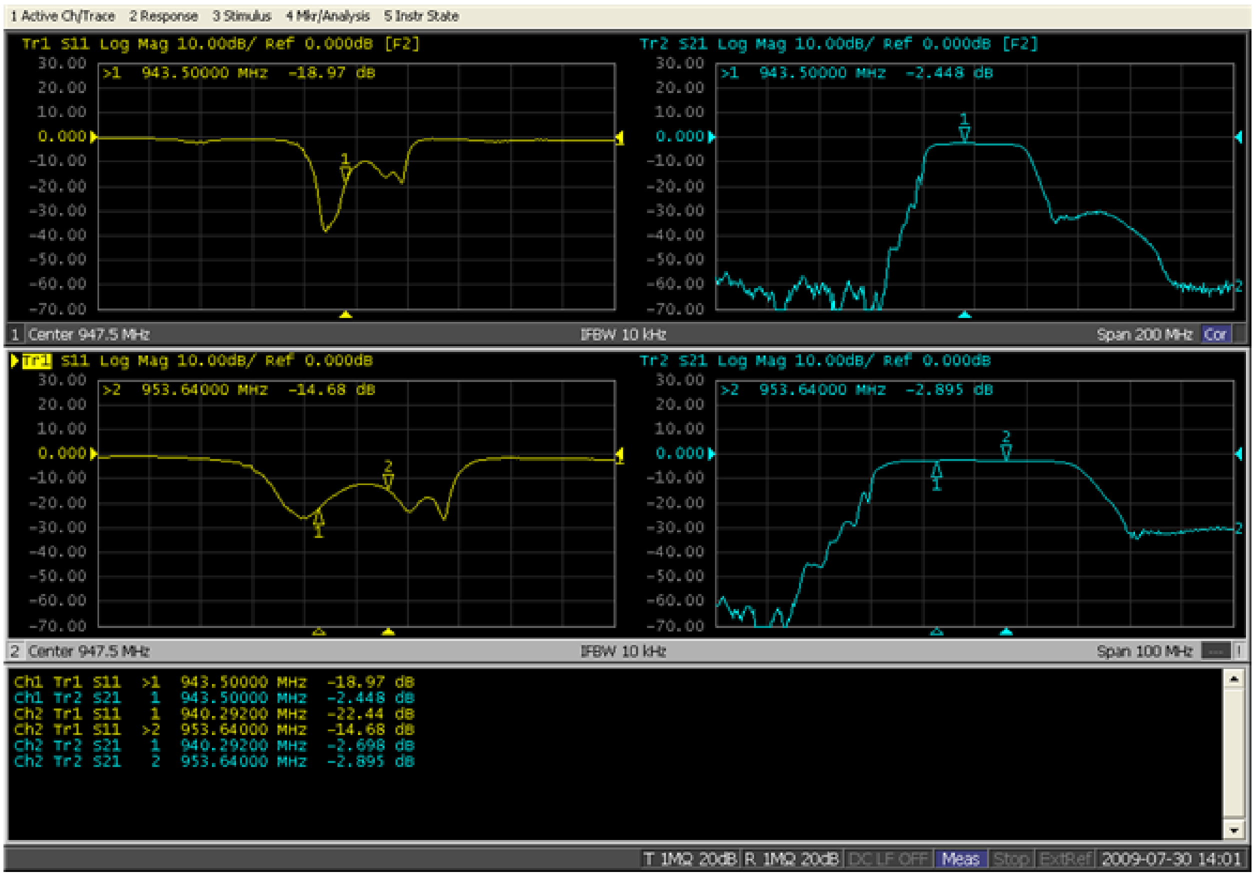
Listing all Marker Values in all Displayed Channels

You can list all of the marker values in all of the displayed channels on the screen.

Turning ON the Marker Table Display

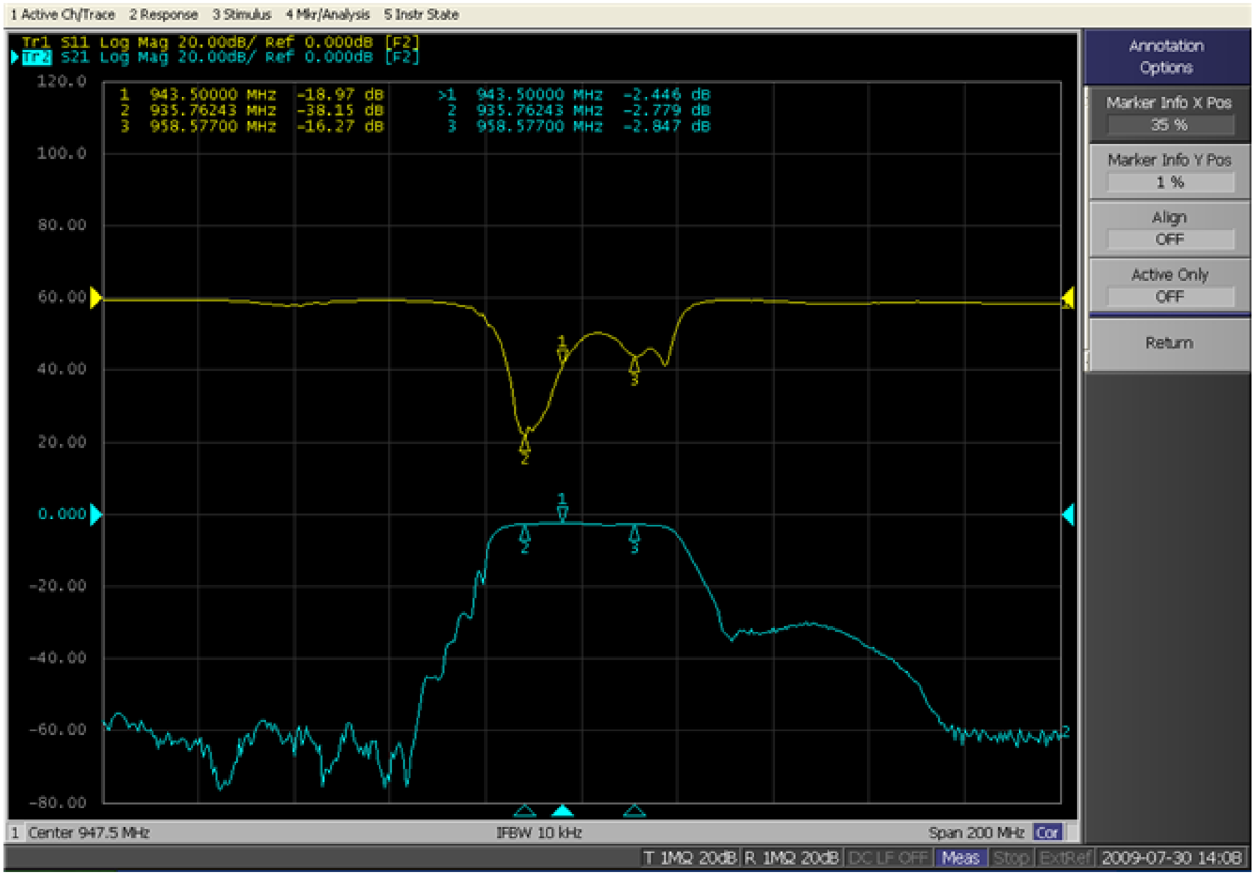
1. Press **Marker Fctn**.
2. Click **Marker Table** to turn ON the marker table display.

The marker table appears in the lower part of the screen.



Specifying Display Position of Marker Values

This section describes how to specify the marker value display position for each active trace.



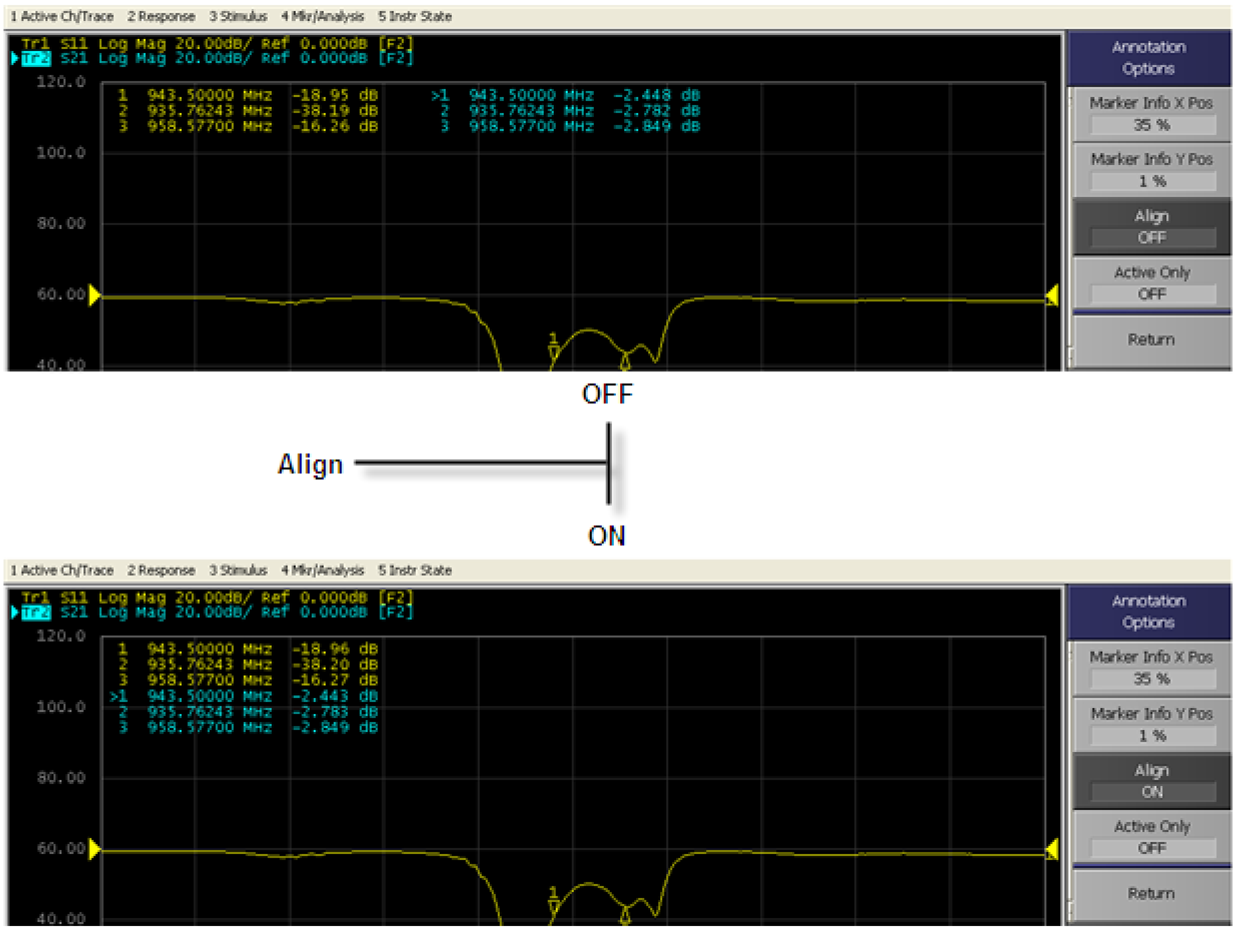
Value	Description
Marker Info X Pos	Specifies the horizontal display position by the width of the display area as a percentage.
Marker Info Y Pos	Specifies the vertical display position by the height of the display area as a percentage.

Operational procedure

1. Press **Channel Next** (or **Channel Prev**) to activate the channel for which you want to set the marker coupling.
2. Press **Marker Fctn** > **Annotation Options**.
3. Click **Marker Info X Pos** to set the horizontal display position.
4. Click **Marker Info Y Pos** to set the vertical display position.

Aligning Marker Value Display

This section describes how to align maker value displays.

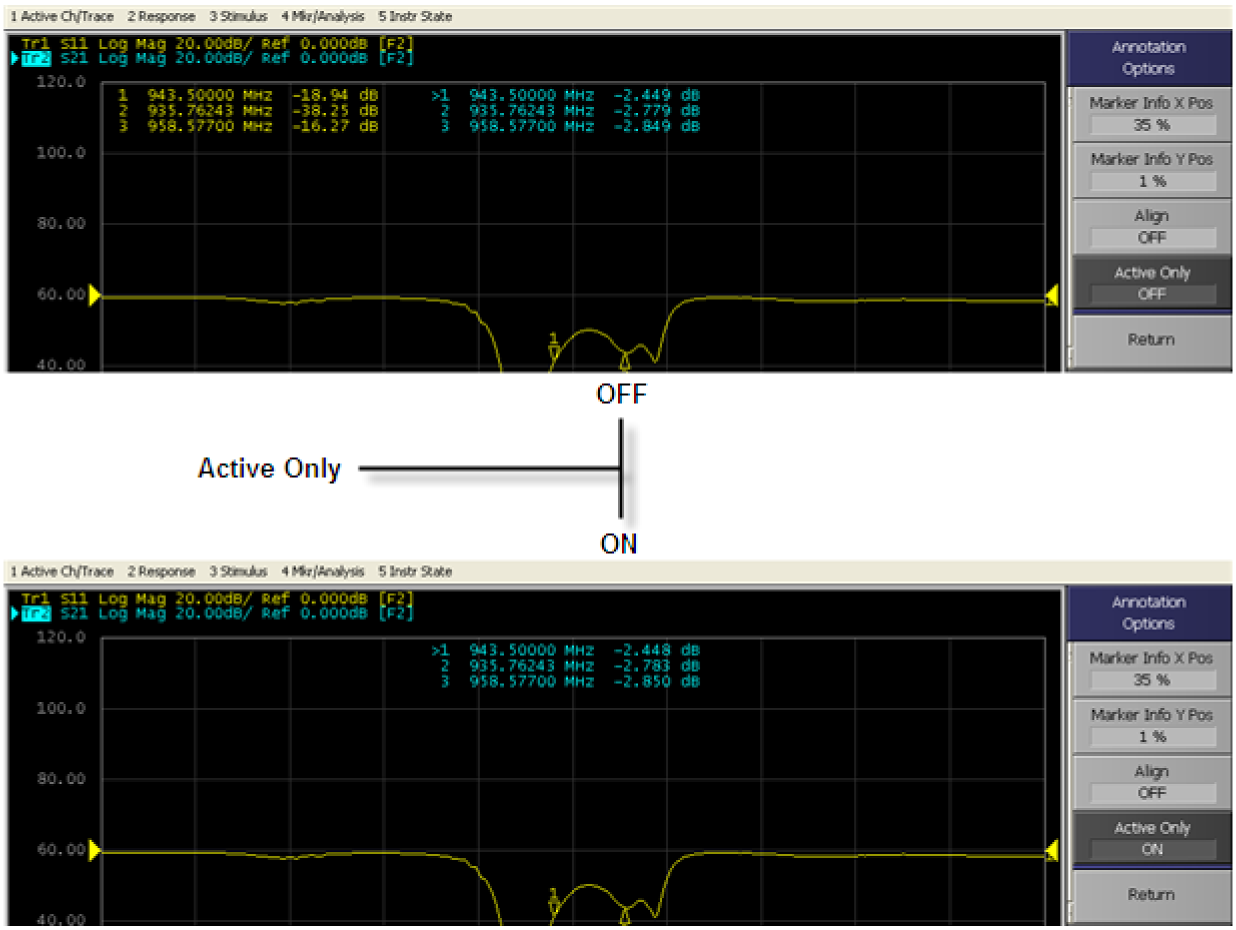


Value	Description
On (**Align ON**)	Displays marker values to align to the display position of trace 1.
Off (**Align OFF**)	Displays marker values in the display position defined for each trace.

1. Press **Marker Fctn** > **Annotation Options**.
2. Click **Align** to toggle ON/OFF.

Displaying All Marker Values for Displayed Traces

This section describes how to display all marker values for the displayed traces.



Value	**Description**
Displays all *(***Active Only OFF***)*	Displays all marker values for displayed traces.
Displays active markers *(***Active Only ON***)*	Displays markers for the active trace only.

1. Press **Marker Fctn** > **Annotation Options**.
2. Click **Active Only** to toggle ON/OFF.

Searching for Positions that Match Specified Criteria

- Overview
- Setting Search Range
- Automatically Executing a Search (Search Tracking)
- Searching for Maximum and Minimum Values
- Searching for the Peak
- Searching for Multiple Peaks
- Searching for the Target Value (Target search)
- Searching for the Multiple Target Values (Multi-target Search)

Other topics about Data Analysis

Overview

You can search for a position that matches your specified criteria by using the Marker Search feature. Marker Search allows you to search for a position that matches any of the following criteria.

- Maximum value
- Minimum value
- Peak
  - Maximum peak (for a positive peak), minimum peak (for a negative peak)
  - Peak on the left-hand side nearest to marker position
  - Peak on the right-hand side nearest to marker position
- Multi Peak
- Target (a point that has a target measurement value)
  - Target nearest to the marker position
  - Target on the left-hand side nearest to marker position
  - Target on the right-hand side nearest to marker position
- Multi Target

Setting Search Range

The Marker Search feature allows you to set part of the sweep range as the search target (Partial Search feature) as well as the entire search range. For the Partial Search feature, you can select whether to couple traces in the channel.

Procedure to Turn ON/OFF Trace Coupling within Search Range

1. Press **Channel Next** (or **Channel Prev**) and **Trace Next** (or **Trace Prev**) to activate the trace for which you want to set the search range.
2. Press **Marker Search** > **Search Range.**
3. Click **Couple** to toggle ON/OFF trace coupling within the search range.

Procedure to Set Search Range

1. Press **Channel Next** (or **Channel Prev**) and **Trace Next** (or **Trace Prev**) to activate the trace for which you want to set the search range.
2. Press **Marker Search** > **Search Range.**
3. Click **Search Range** to turn ON the Partial Search feature.
4. Click **Start**, then enter the start value (lower limit) of the search range.
5. Click **Stop**, then enter the stop value (upper limit) of the search range.

Automatically Executing a Search (Search Tracking)

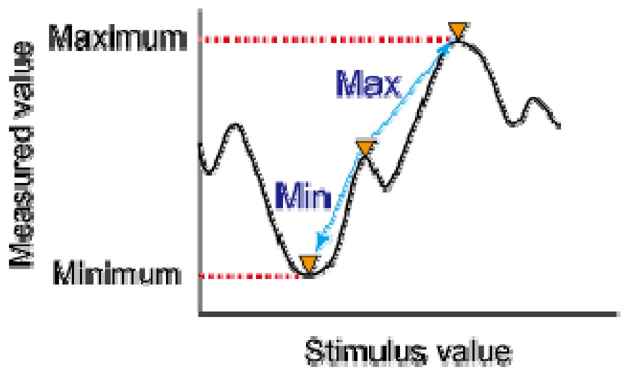
Search tracking is a function that sets a search to be repeated every time a sweep is done even if the execution key for the search (maximum, minimum, peak, and target) is not pressed. This function facilitates observation of measurement results such as the maximum value of traces (e.g., the insertion loss of a band pass filter).

Performing Search Tracking

1. Press **Channel Next** (or **Channel Prev**) and **Trace Next** (or **Trace Prev**) to activate the trace on which you want to set up the search tracking.
2. Press **Marker Search** key.
3. Click **Tracking** and turn the search tracking function ON/OFF.

Searching for Maximum and Minimum Values

You can search for the maximum or minimum measured value on the trace and move a marker to that point.



e5071c386

Search for maximum (**Max**)	Move active marker to point on the trace where measured value is the greatest
Search for minimum (**Min**)	Move active marker to point on the trace where measured value is the lowest

Procedure

1. Activate the marker you are using to search for the maximum and minimum values.
2. Press **Marker Search** key.
3. Click the corresponding softkey to move the marker to the maximum or minimum measured value.

NOTE When the data format is in Smith chart or polar format, execute the search only for the main response value.

Searching for the peak

The peak search function enables you to move the marker to the peak on the trace.

Definition of the peak

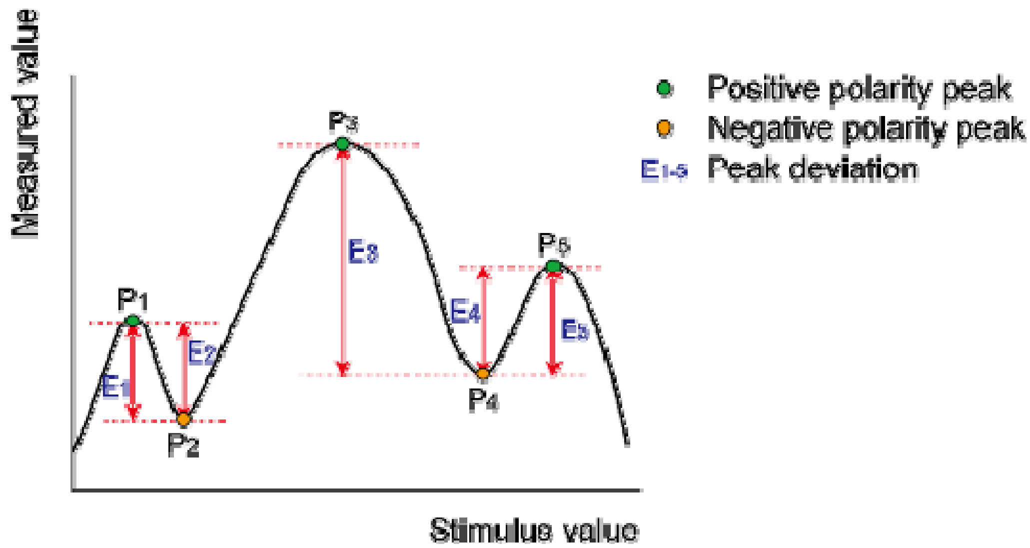
A peak is a measurement point of which the value is greater or smaller than the adjoining measurement points on its right and left sides. Peaks are classified into the following two types depending on the differences in magnitude from the measurement points on either side of it.

Positive peak (**Positive**)	A peak of which the measured value is greater than the measurement points on either side of it (peak polarity: positive)

Negative peak (**Negative**)	A peak of which the measured value is smaller than the measurement points on either side of it (peak polarity: negative)

About Peak Excursion Value

The peak excursion value is smaller among the differences in the measured values from the adjoining peaks of the opposite polarity.

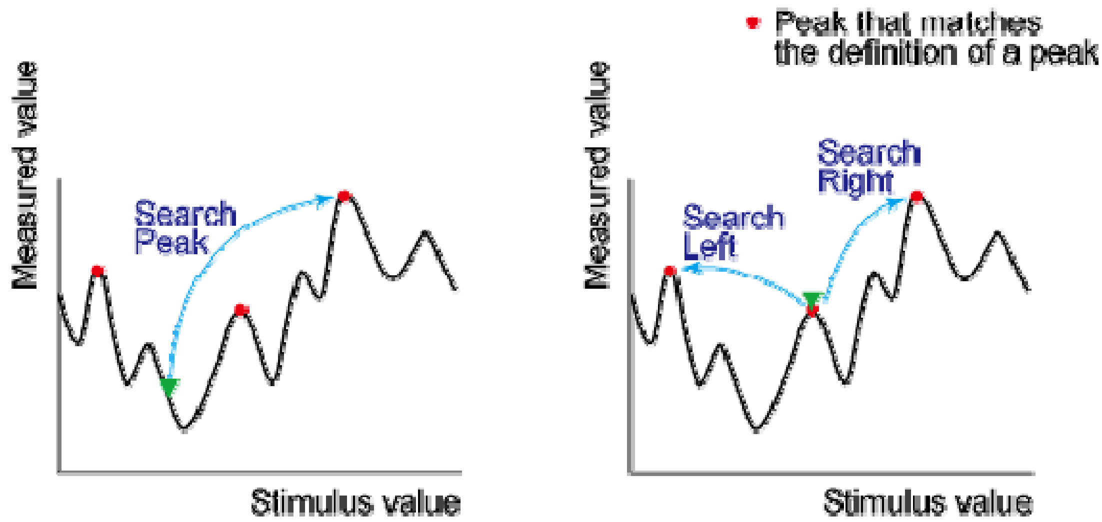


e5071c380

Executing a Peak Search

The following three methods are available for executing the peak search:

Peak search (**Search Peak**)	Moves the marker to the maximum peak when peak polarity is **Positive** or **Both**. Moves the marker to the minimum peak when peak polarity is **Negative**.
Left search (**Search Left**)	Executes the search from current marker position to the smaller stimulus values and moves the marker to first peak encountered.
Right search (**Search Right**)	Execute the search from current marker position to the larger stimulus values and moves the marker to first peak encountered.



e5071c452

Procedure

1. Activate the marker you are using for the peak search.
2. Press **Marker Search** key.
3. Click **Peak** > **Peak Excursion**.
4. Enter the lower limit for the peak excursion value. This sets the peak search to be executed based on the definitions of the newly set lower limit for the peak excursion value and the currently set peak polarity.
5. Click **Peak Polarity**.
6. Select a peak polarity. This sets the peak search to be executed based on the definitions of the currently set lower limit for the peak excursion value and the newly set peak polarity.
7. Click the corresponding softkey to move the marker to the peak.

NOTE When the data format is in Smith chart or polar format, execute the search for the main response value of the two marker response values.

Searching for Multiple Peaks

The multi-peak search function enables you to display markers on multiple peaks on traces.

Definition of the Peaks

A peak is a measurement point whose value is greater or smaller than the adjoining measurement points on its right and left sides. Peaks are classified into the following types depending on the difference in magnitude from the measurement points on either side of it.

Positive peak (**Positive**)	A peak of which the measurement value is greater than the measurement points on either side of it (peak polarity: positive)
Negative peak (**Negative**)	A peak of which measurement value is smaller than the measurement points on either side of it (peak polarity: negative)

About the Multi-peak Search Function (Search Multi Peak)

The multi-peak search is a function that searches for peaks that match with pre-defined lower limit for the peak excursion value and peak polarity (positive or negative) and then displays the markers on the peaks being searched. Depending on number of detected peaks, markers 1 through 9 are displayed from the start frequency.

The peak excursion is the smaller of the differences in measurement values from the adjoining peaks of the opposite polarity.

NOTE When the multi-peak search is executed, search and tracking settings for markers 1 through 9 are ignored and the settings for the multi-peak search are used. Note that the reference marker is not affected.

Positive Peak / Negative Peak and Peak Excursion

Multi-peak Search (when peak polarity is positive)

Executing a Multi-peak Search

1. Activate the marker you are using for the multi-peak search.
2. Press **Marker Search** > **Multi Peak** > **Peak Excursion**
3. Enter the lower limit for the peak excursion value.

   This causes the multi-peak search to be executed based on the definitions of the newly set lower limit for the peak excursion value and currently set peak polarity.
4. Press **Peak Polarity**.
5. Select a peak polarity from **positive**, **negative** or **both**.

   This causes the multi-peak search to be executed based on the definitions of the currently set lower limit for the peak excursion value and newly set peak polarity.
6. Press **Search Multipeak** to move the marker to the peak.

NOTE When the data format is Smith chart or polar format, execute the search for the main response value of the two marker response values.

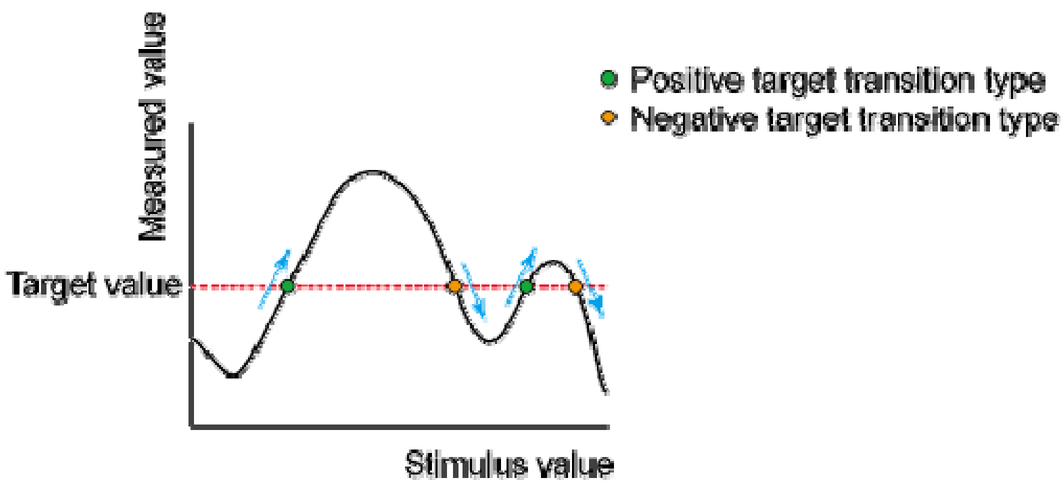
Searching for the target value (target search)

The target search is a function that searches for a target that matches the pre-defined target value and transition type(s) (positive, negative, or both positive and negative) and then moves the marker to that target.

Target and Transition Types

A target is a point that has a specific measured value on the trace. Targets can be divided into the two groups shown below depending on their transition type.

Transition type: Positive (**Positive**)	When the value of the target is larger than the measured value that immediately precedes it (on the left side)
Transition type: Negative (**Negative**)	When the value of the target is smaller than the measured value that immediately precedes it (on the left side)



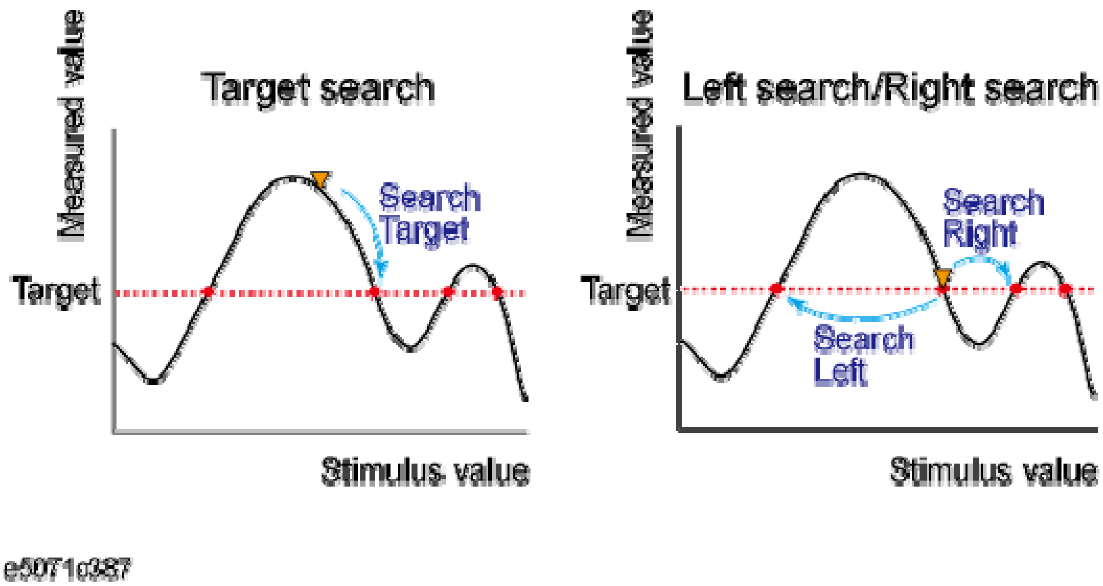
e5071c389

Executing a Target Search

The following three methods are available for executing the target search:

Target search (**Search Peak**)	The marker moves to the peak with maximum response value if the peak polarity is **Positive** or **Both** or to the peak with minimum response value if the peak polarity is **Negative**.
Search left (**Search Left**)	Executes the search from the current marker position to the smaller stimulus values and moves the marker

	to first encountered target.
Search right (**Search Right**)	Executes the search from the current marker position to the larger stimulus values and moves the marker to first encountered target.



Procedure

1. Activate the marker you are using for the target search.
2. Press **Marker Search** key.
3. Click **Target** > **Target Value**
4. Enter the target value in the entry area that appears. This causes the target search to be executed based on the definitions of the newly set target value and the currently set transition type.
5. Click **Target Transition**.
6. Select a transition type. This sets the target search to be executed based on the definitions of the currently set target value and the newly set transition type.
7. Press the corresponding softkey to move the marker to the target.

NOTE When the data format is in Smith chart or polar format, execute the search for the main response value of the two marker response values.

Searching for the Multiple Target Values (Multi-target Search)

The multi-target search function enables you to display a marker on each point having the target measurement value.

Target and Transition Types

A target is a point that has a specific measurement value on the trace. Targets can be divided into two groups shown below depending on their transition type.

Positive	When the value of the target is larger than the measurement value that immediately proceeds it (on the left side).
Negative	When the value of the target is smaller than the measurement value that immediately proceeds it (on the left side).

Target and Transition Types

About the multi-target search function (Search Multi Target)

The multi-target search is a function that searches for targets that match to pre-defined target value and transition type(s) (positive, negative, or both of positive and negative) and displays markers on the targets being searched.

Depending on the number of detected targets, markers 1 through 9 are displayed from the start frequency.

NOTE When the multi-target search is executed, search and tracking settings for markers 1 through 9 are ignored and the settings for the multi-target search are used. Note that the reference marker is not affected.

Multi-target Search (when transition type is set to "both positive and negative")

Procedure

1. Activate the marker you are using for target search.
2. Press **Marker Search** > **Multi Target** > **Target Value.**
3. Enter a target value in the entry box that appears.

   This causes the target search to be executed based on the target value newly set and the transition type defined at this point.
4. Press **Target Transition.**
5. Selects a transition type from **positive**, **negative** or **both**.

This causes the target search to be executed based on the target value set at this point and the transition type newly set.

6. Press **Search Multi Target** to move the marker to the target.

   NOTE When the data format is in Smith chart or polar format, execute the search for the main response value of the two marker response values.

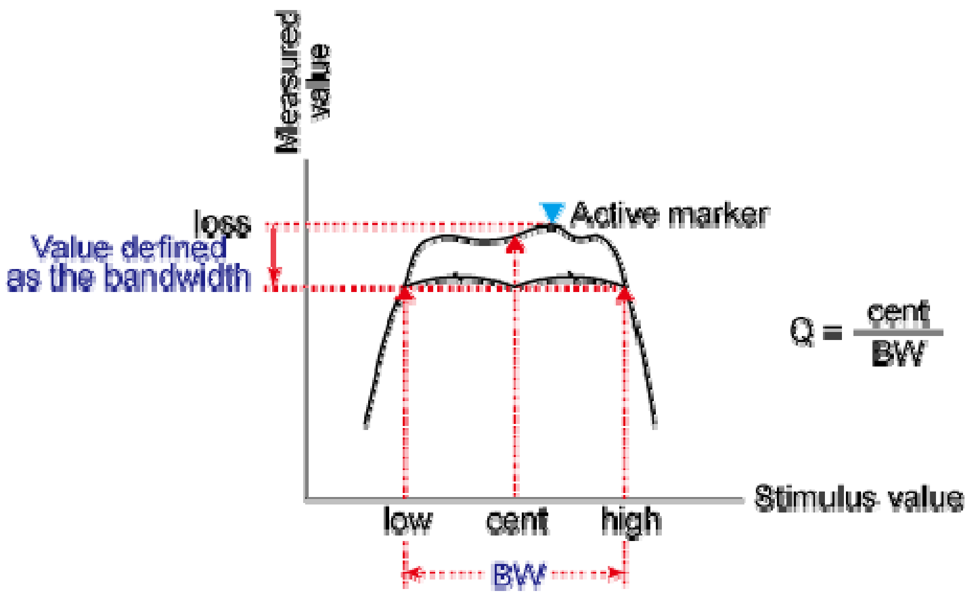
Determining the Bandwidth of the Trace (Bandwidth Search)

- Overview
- Executing a Bandwidth Search

Other topics about Data Analysis

Overview

The bandwidth search is a function for determining the bandwidth of the trace, center frequency, cut-off points (on the higher frequency and the lower frequency sides), Q, and insertion loss based on the position of the active marker. The definitions of the parameters determined through the bandwidth search are shown below.



e5071c392

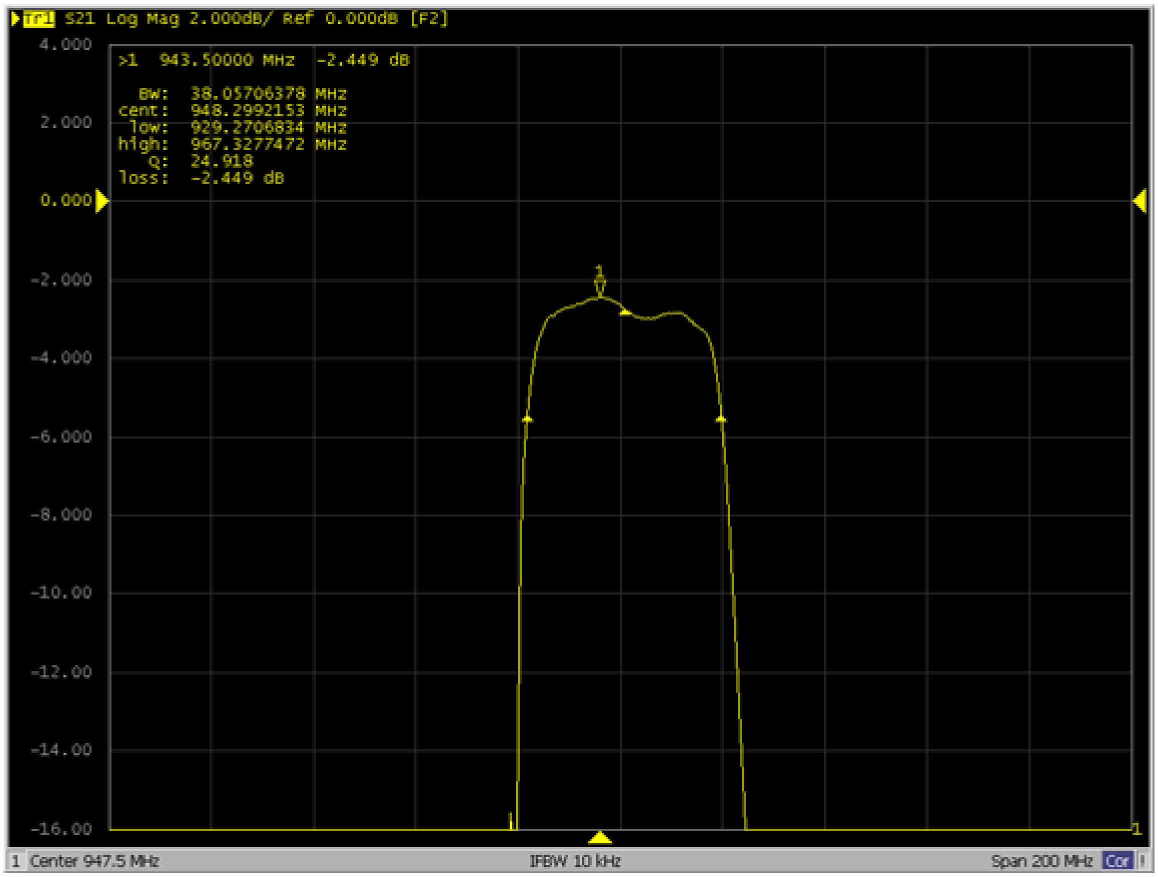
Bandwidth Parameter	Definition
Insertion loss (*loss*)	The measured value of the position of the active marker at the time the bandwidth search is executed.
Lower frequency cut-off point (*low*)	Lowest frequency within two measurement points, both separated by the defined bandwidth value from the active marker position.
Higher frequency cut-off point	Highest frequency within two measurement points, both separated by the defined bandwidth

(*high*)	value from the active marker position.
Center frequency (*cent*)	Frequency at the midpoint between the lower frequency cut-off and higher frequency cut-off points. (*high+low)/2*
Bandwidth (*BW*)	The difference in frequency between the higher frequency cut-off and lower frequency cut-off points. (*high-low)*
Q	Value obtained by dividing the center frequency by the bandwidth.(*cent*/*BW*)

Executing a Bandwidth Search

1. Place the active marker at the desired point on the trace on which the bandwidth search is executed. The response value of this active marker itself is the insertion loss in the bandwidth search (**loss**).
2. Press **Marker Search** > **Bandwidth Value**.
3. Enter the defined bandwidth value in the entry area that appears.
4. Click **Bandwidth** to turn ON the bandwidth search. In the upper left of the trace display, six bandwidth parameters are displayed.

Bandwidth search results


Tr1 S21 Log Mag 2.000dB/ Ref 0.000dB [F2]
>1 943.50000 MHz -2.449 dB
BW: 38.05706378 MHz
cent: 948.2992153 MHz
low: 929.2706834 MHz
high: 967.3277472 MHz
Q: 24.918
loss: -2.449 dB
4.000
2.000
0.000
-2.000
-4.000
-6.000
-8.000
-10.00
-12.00
-14.00
-16.00
1 Center 947.5 MHz
IFBW 10 kHz
Span 200 MHz
Cor

Determining the Bandwidth of the Trace (Notch Search)

- Overview
- Executing a Notch Search

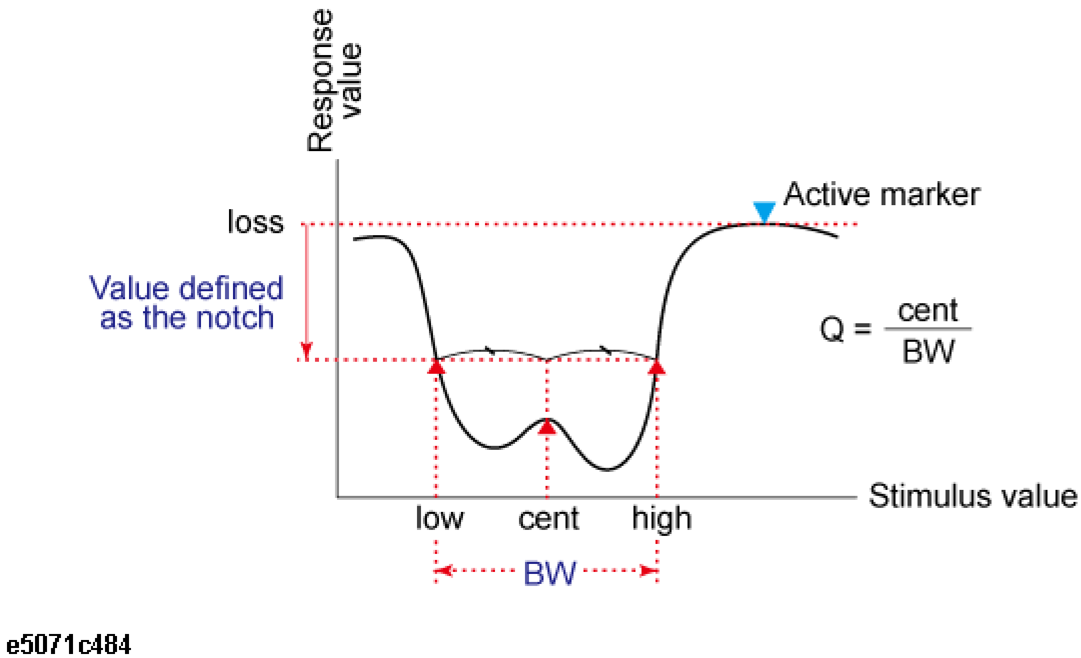
Other topics about Data Analysis

Overview

The notch search function is used to obtain the bandwidth, center frequency, cutoff points (high-frequency side and low-frequency side), Q, and insertion loss of a trace based on the active marker position. The notch search function starts from the left side of the active marker position, and ends when the points meet the conditions.

The figure and table below shows the definition of parameters obtained by notch search function. The notch value in figure below must be specified by the user.

Bandwidth Parameters



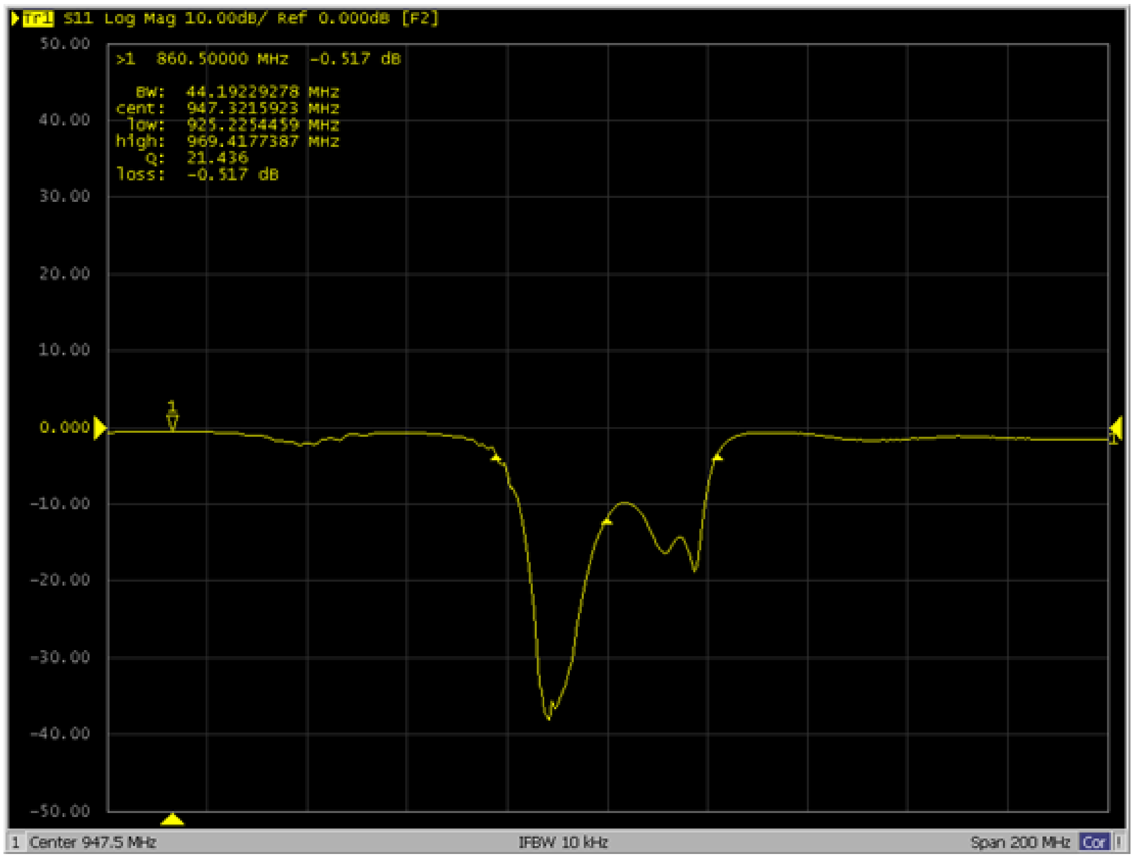
Definition of bandwidth parameters

Bandwidth parameter name	**Definition**
Insertion loss (loss)	Measurement value at the active marker position when the notch search is executed.

Lower cutoff point (low)	Lower frequency of the 2 points on both sides that have the measurement value apart from the active marker position by the notch value.
Higher cutoff point (high)	Higher frequency of the 2 points on both sides that have the measurement value apart from the active marker position by the notch value.
Center frequency (cent)	Frequency of the middle point between the lower cutoff point and the higher cutoff point (high + low)/2.
Bandwidth (BW)	Frequency difference between the higher cutoff point and the lower cutoff point (high − low).
Q	Value obtained by dividing the center frequency by the bandwidth (cent/BW).

Executing a Notch Search

1. Place the active marker on the desired point on the trace on which the notch search is executed. The response value of this active marker itself is the insertion loss in the notch search (loss).
2. Press **Marker Search** > **Notch Value** and enter the notch value in the entry area that appears.
3. Press **Notch** to turn ON the notch search. In the upper left of the trace display, six bandwidth parameters are displayed (see the figure below).



NOTE For more information on displaying the notch search result, see Notch Search

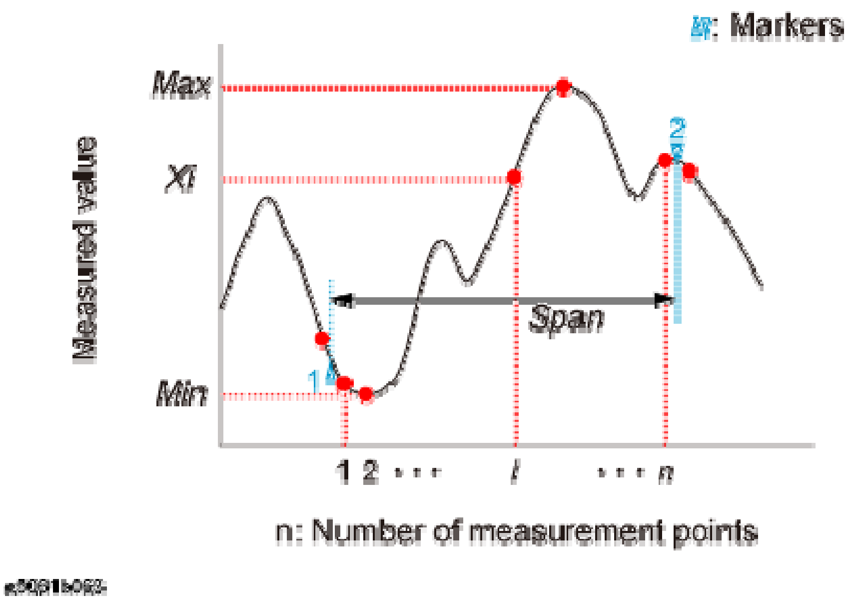
Determining the Mean, Standard Deviation, and p-p of the Trace

- Overview
  - Displaying Statistical Data

Other topics about Data Analysis

Overview

You can easily determine the statistics data for a trace (span, mean, standard deviation, and peak-to-peak). The definitions for the statistics data elements are shown below. It is calculated within the range of markers 1 and 2. The markers 1 and 2 are activated automatically.



Statistics data element	Definition
Span	Span between markers 1 and 2.
Mean (*mean*)	$$\frac{\sum_{i=1}^{n} x_i}{n}$$ (n: number of points between markers 1 and 2; x_i: measured value at the i-th measurement point between markers 1 and 2.)

Standard deviation (*s. dev*)	$$\sqrt{\frac{\sum_{i=1}^{n}(x_i - mean)^2}{n-1}}$$ (n: number of points between markers 1 and 2; x_i: measured value at the i-th measurement point between markers 1 and 2.; mean: Mean)
Peak-to-peak (*p - p*)	Max - Min (Max: greatest measured value between markers 1 and 2.; Min: smallest measured value between markers 1 and 2.)

NOTE The search range does not affect this result.

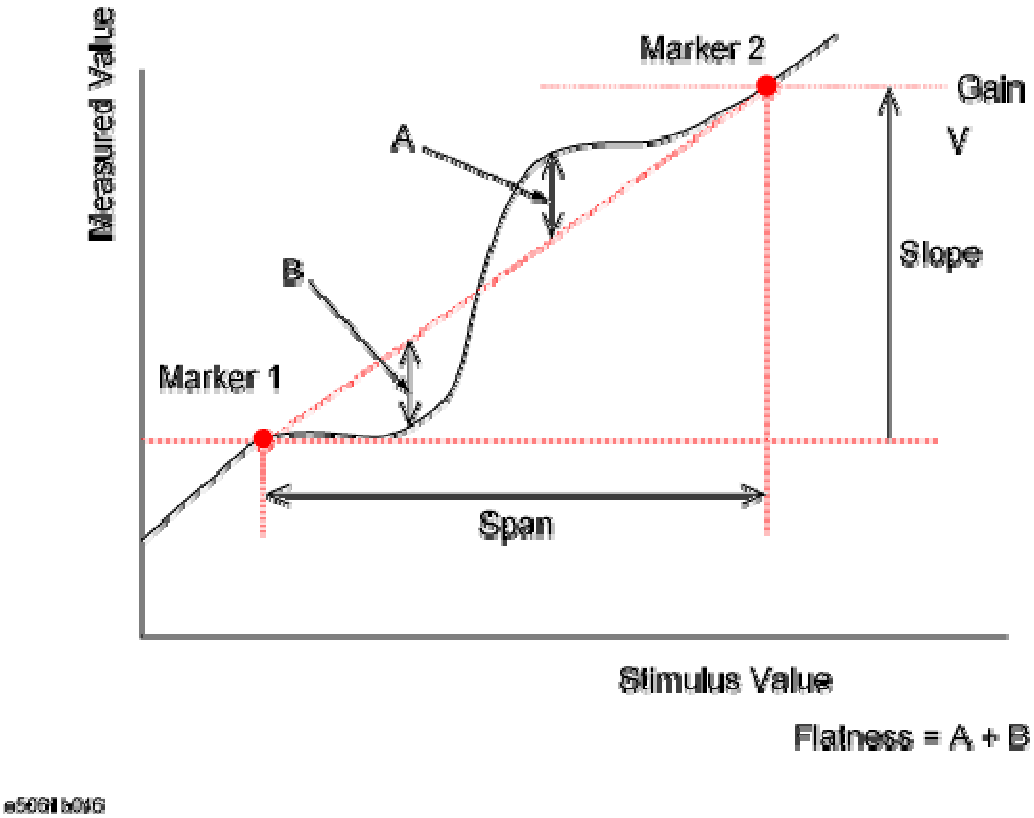
Displaying Statistical Data

1. Press **Channel Next** (or **Channel Prev**) and **Trace Next** (or **Trace Prev**) to activate the trace of which the statistical data is required.
2. Press **Marker Fctn** > **Statistics** to turn ON the display of statistics data.
3. The markers 1 and 2 are activated automatically. Move markers 1 and 2 to the position of the measurement.

Obtaining Span, Gain, Slope, and Flatness between Markers

The span, gain, slope, and flatness between marker 1 and marker 2 on a trace can be obtained. The following figure shows data definitions.

Span, Gain Slope, and Flatness Parameters



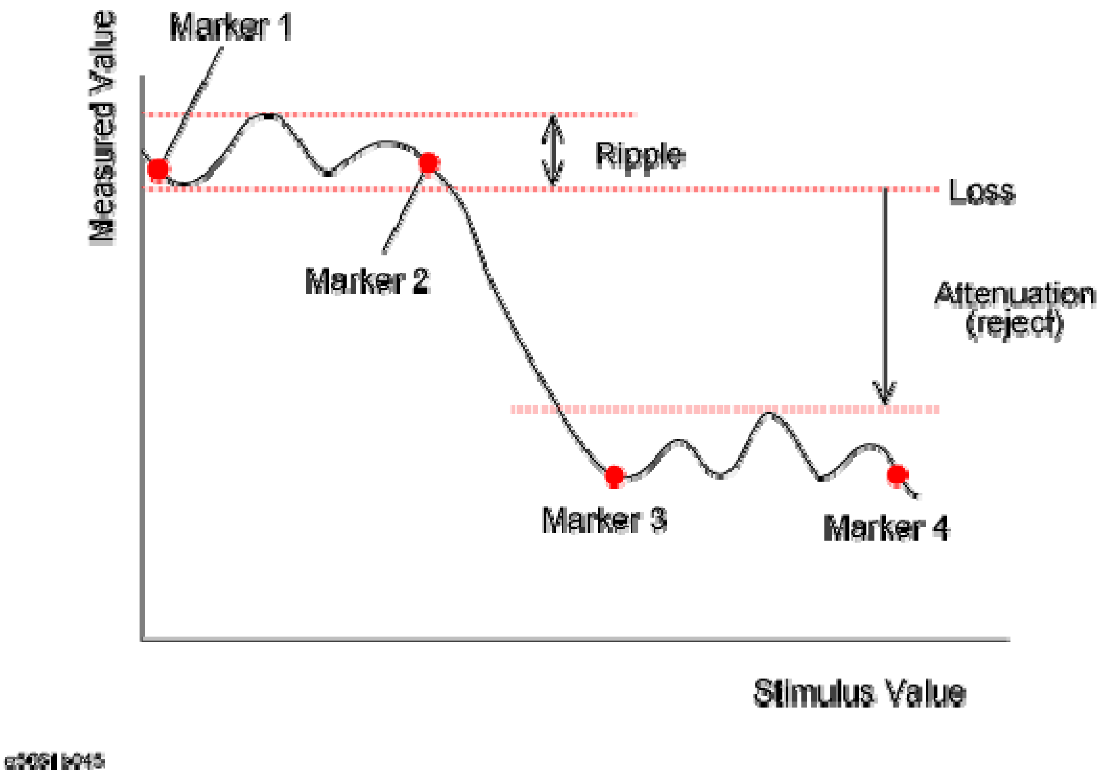
Data Name	Definition
Span	Calculates the distance between marker 1 and marker 2.
Gain	Calculates the larger measurement value between marker 1 and marker 2.
Slope	Calculates the differences in measurement value between marker 1 and marker 2. (marker 2 - marker 1)
Flatness	Calculates the sum of 2 maximum measurement value differences from the line connecting marker 1 and marker 2 (above and below the line).

1. Press **Channel Next** (or **Channel Prev**) and **Trace Next** (or **Trace Prev**) to activate the trace of which you want to obtain the data.
2. Press **Marker Fctn** > **Flatness** to show data.
3. Markers 1 through 2 are turned ON and displayed.
4. Set the positions of markers 1 through 2 on the trace.

Obtaining Loss, Ripple, and Attenuation of the RF Filter

Specify the pass band with markers 1 and 2 on the trace and specify the stop band with markers 3 and 4 to obtain loss, ripple, and attenuation. The following figure shows data definitions.

Loss, Ripple, and Attenuation Parameters



Data Name	Definition
Loss (loss)	Calculates the minimum value between markers 1 and 2.
Ripple (p-p)	Calculates the differences between the maximum value and the minimum value of markers 1 and 2.
Attenuation (reject)	Calculates the differences between the maximum value between markers 3 and 4 and the minimum value (loss) between markers 1 and 2.

1. Press **Channel Next** (or **Channel Prev**) and **Trace Next** (or **Trace Prev**) to activate the trace of which you want to obtain the data.
2. Press **Marker Fctn** > **RF Filter Stats** to show the data.

3. Markers 1 through 4 are turned ON and displayed.
4. Set the positions of markers 1 through 4 on the trace.

Comparing Traces / Performing Data Math

- Overview
- Performing Data Math Operations

Other topics about Data Analysis

Overview

Each of the traces for which measured data is displayed is provided with an additional trace, called a memory trace, that temporarily stores measured data. You can use the memory trace to compare traces on the screen or to perform complex data math between the memory trace and measured data.

The following data math operations are available:

Value	Description
Data / Memory	Divides the measured data by the data in the memory trace. This function can be used to evaluate the ratio of two traces (e.g., evaluating gain or attenuation).
Data * Memory	Multiplies the measured data by a memory trace.
Data - Memory	Subtracts a memory trace from the measured data. This function can be used, for example, to subtract a vector error that has been measured and stored (e.g., directivity) from data subsequently measured on a device.
Data + Memory	Adds the measured data and the data in the memory trace.

Performing Data Math Operations

1. Press **Channel Next** (or **Channel Prev**) and **Trace Next** (or **Trace Prev**) to activate the trace to be stored in memory.
2. Press **Display** > **Data -> Mem** to store the measured data in memory.
3. Press **Data Math.**
4. Select the **data math** operation to perform.
5. Press **Display**.
6. Select the **type of data** to display ON the screen.
7. Send the trigger to make measurements.

Performing Parameter Conversion of Measurement Results

- Overview
- Turning Conversion ON
- Selecting Conversion Target Parameter

Other topics about Data Analysis

Overview

You can use the parameter conversion function to convert the measurement results of the S-parameter (S_{ab}) to the following parameters.

- Equivalent impedance (Z_r) and equivalent admittance (Y_r) in reflection measurement

$$Z_r = Z_{0a} \times \frac{1 + S_{aa}}{1 - S_{aa}}, Y_r = \frac{1}{Z_r}$$

- Equivalent impedance (Z_t) and equivalent admittance (Y_t) in transmission measurement

$$Z_t = \frac{2 \times \sqrt{Z_{0a} \times Z_{0b}}}{S_{ab}} - (Z_{0a} + Z_{0b}), Y_t = \frac{1}{Z_t}$$

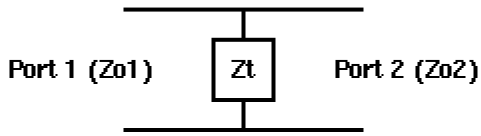
- Inverse S-parameter ($1/S_{ab}$)

where:

Z_{0a}: Characteristic impedance of port a

Z_{0b}: Characteristic impedance of port b

- Z/Y Transmission Shunt

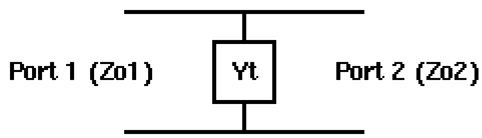


Z Transmission-Shunt

$$Zt = \frac{1}{Yt}$$

$$Yt = \frac{2\sqrt{Yo1 \cdot Yo2}}{S} - (Yo1 + Yo2)$$

$$Yo1 = \frac{1}{Zo1} \quad Yo2 = \frac{1}{Zo2}$$



Y Transmission-Shunt

- Conjugation

  Conjugation converts the measurement value to complex conjugate number.

When the fixture simulator function is ON and the port impedance function is ON, the value set in the port impedance conversion is used. In other cases, the system Z_0 (preset value: 50 Ω) is used.

Turning Conversion ON

1. Press **Channel Next** (or **Channel Prev**) and **Trace Next** (or **Trace Prev**) to activate a trace of which you want to use the marker.
2. Press **Analysis** > **Conversion**.
3. Click **Conversion** to turn ON the conversion function.

Selecting Conversion Target Parameter

1. Press **Channel Next** (or **Channel Prev**) and **Trace Next** (or **Trace Prev**) to activate a trace of which you want to use the marker.
2. Press **Analysis** > **Conversion** > **Function**.
3. Select the **parameter** to which you want to convert the result.

When the conversion function is ON, the selected parameter is displayed in Trace Status Area.

Using Limit Test

- Overview
- Concept of Limit Test
- Displaying Judgement Result of Limit Test
- Defining Limit Line
- Changing the Limit Line Display Mode
- Using Relative Limit Line
- Adding Offset to Limit Line
- Initializing the Limit Table

Other topics about Data Analysis

Overview

The limit test feature allows you to set the limit line for each trace and then perform the pass/fail judgment for the measurement result.

Concept of Limit Test

The limit test is a function to perform pass/fail judgment based on the limit line you set with the limit table.

In the limit test, if the upper limit or lower limit indicated by the limit line is not exceeded, the judgment result is pass; if it is exceeds, the judgment result is fail for all measurement points on the trace. Measurement points in a stimulus range with no limit line are judged as pass.

NOTE The targets of the pass/fail judgment are measurement points only. Parts interpolated between the measurement points are not judged.

You define the limit line by specifying the stimulus value (Begin Stimulus) and response value (Begin Response) of the begin point, the stimulus value (End Stimulus) and response value (End Response) of the end point, and the type (lower limit/upper limit). For more information, refer to Defining the limit line.

When the limit test is ON, measurement points that fail are displayed in red on the screen and the trace's pass/fail judgment result based on the results of individual measurement points (fail if one or more measurement points on the trace fail) is also displayed. You can check the pass/fail judgment result for the channel (fail if one or more traces fail in any of the limit test, the ripple test or the bandwidth test within the channel) on the screen as well. For more information, refer to Displaying judgment result of limit test.

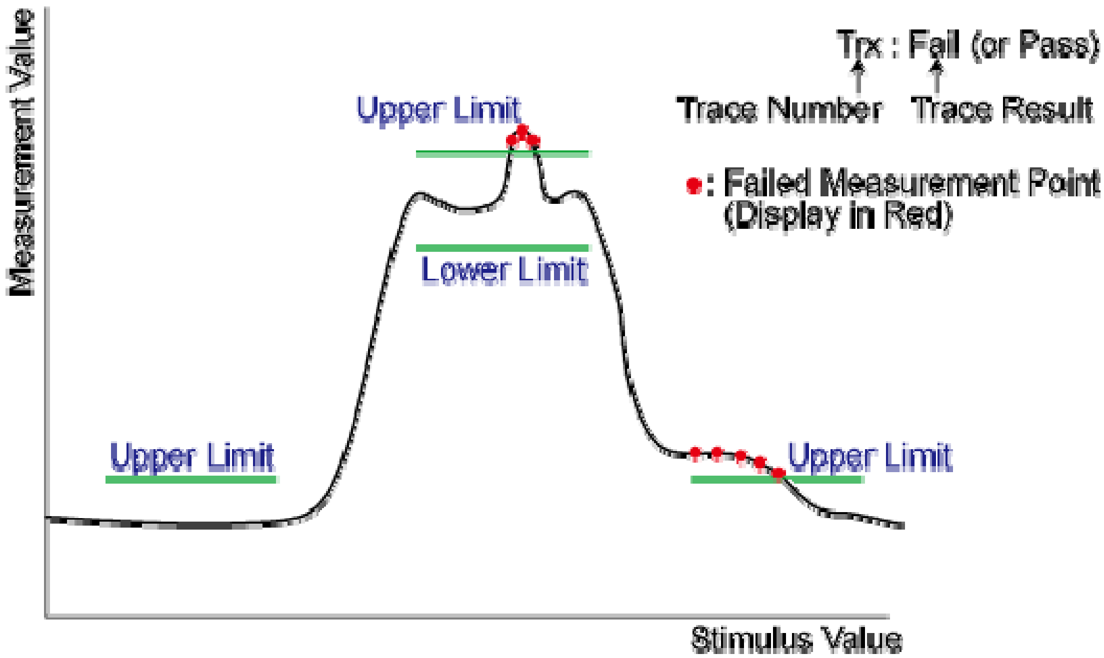
In addition to viewing the screen, you can check the judgment result of the limit test with the following methods.

- Beep that occurs when the judgment result fails.
- Using the status register.

Displaying Judgment Result of Limit Test

Judgment result of measurement points and trace

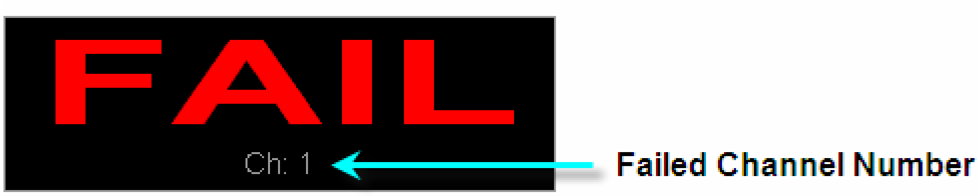
Measurement points that fail are displayed in red on the screen. The judgment result of the trace is indicated by Pass or Fail displayed in the upper right section of the graph.



e50F1c446

Judgment result of channels

If a channel has a judgment result of fail, the message below appears on the screen (it will be judged as fail if one or more unsatisfactory trace exist in any of the limit test, the ripple test or the bandwidth test within the channel.)



Follow these steps to turn ON/OFF the display of the channel fail message.

1. Press **Analysis** > **Limit Test.**
2. Click **Fail Sign**. Each press toggles between ON/OFF.

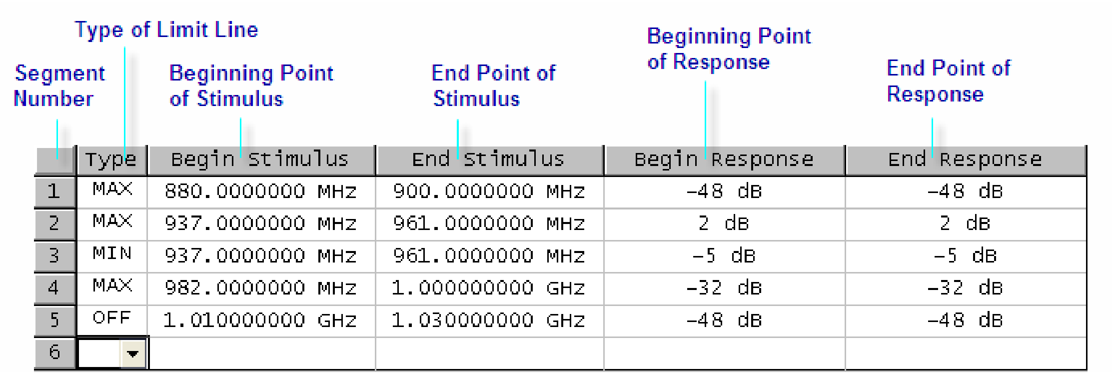
Defining Limit Line

To use the limit test, you must first define the limit line. You can define a limit table for each trace, and you can define up to 100 limit lines (segments) in a limit table.

Defining a segment

The following steps describe how to define a segment.

1. Press **Channel Next** (or **Channel Prev**) and **Trace Next** (or **Trace Prev**) to select the trace on which the limit test function is used.
2. Press **Analysis** > **Limit Test.**
3. Click **Edit Limit Line** to display the limit table.



	Type	Begin Stimulus	End Stimulus	Begin Response	End Response
1	MAX	880.0000000 MHz	900.0000000 MHz	-48 dB	-48 dB
2	MAX	937.0000000 MHz	961.0000000 MHz	2 dB	2 dB
3	MIN	937.0000000 MHz	961.0000000 MHz	-5 dB	-5 dB
4	MAX	982.0000000 MHz	1.000000000 GHz	-32 dB	-32 dB
5	OFF	1.010000000 GHz	1.030000000 GHz	-48 dB	-48 dB
6					

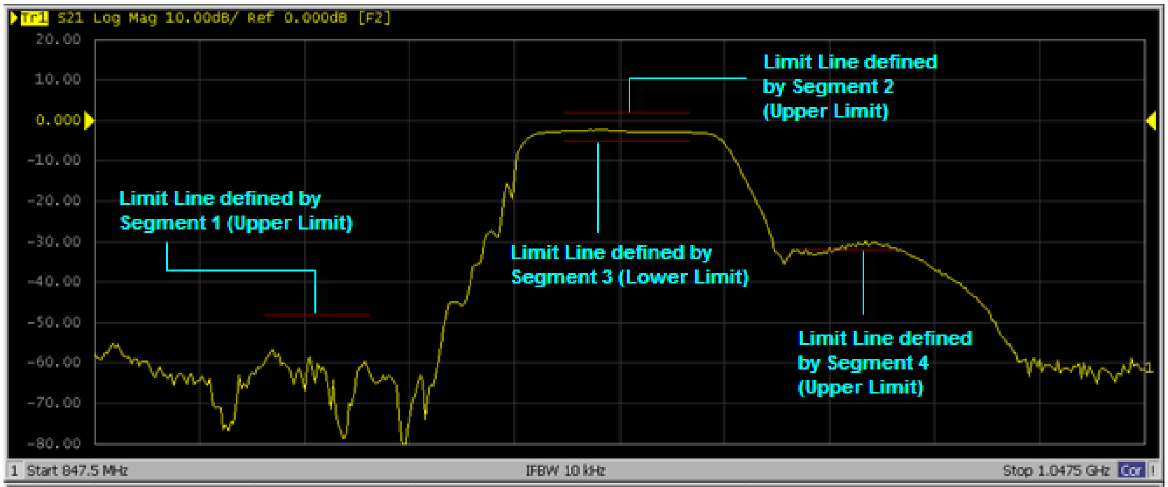
4. Using the limit table, create/edit a segment. Initially, no segments are entered in the limit table. At the same time, the Edit Limit Line menu used to create/edit the limit table is displayed.
5. Click **Add** to add a segment to the limit table and then specify the segment parameter values shown below.

Segment Parameter	Description
Type	Select the type of segment from the following: **OFF** Segment not used for the limit test **MIN** Segment at which the minimum is specified

	MAX Segment at which the maximum is specified
Begin Stimulus	Specify starting point for stimulus value on the limit line
End Stimulus	Specify ending point for stimulus value on the limit line
Begin Response	Specify starting point for response value on the limit line
End Response	Specify ending point for response value on the limit line

NOTE The range in which stimulus values can be specified is from -500 G to +500 G. When a value outside the range is entered, a suitable value within the range is specified. Once the stimulus value is specified, changing the sweep range of the E5061B does not affect the stimulus value.

NOTE The range in which response values can be specified is from -500 M to +500 M. When a value outside this range is entered, a suitable value within the range is specified. After the response value is specified, changing formats results in changing the units but not the value.



	Type	Begin Stimulus	End Stimulus	Begin Response	End Response
1	MAX	880.0000000 MHz	900.0000000 MHz	-48 dB	-48 dB
2	MAX	937.0000000 MHz	961.0000000 MHz	2 dB	2 dB
3	MIN	937.0000000 MHz	961.0000000 MHz	-5 dB	-5 dB
4	MAX	982.0000000 MHz	1.000000000 GHz	-32 dB	-32 dB
5					

NOTE You can define a limit line that is able to freely overlap the stimulus range of another limit line.

Defining one limit line that has the same type as a second limit line whose stimulus range overlaps with the first one results in two or more limit values at the same measurement point. In this case, the limit value to be used in the limit test is defined as follows:

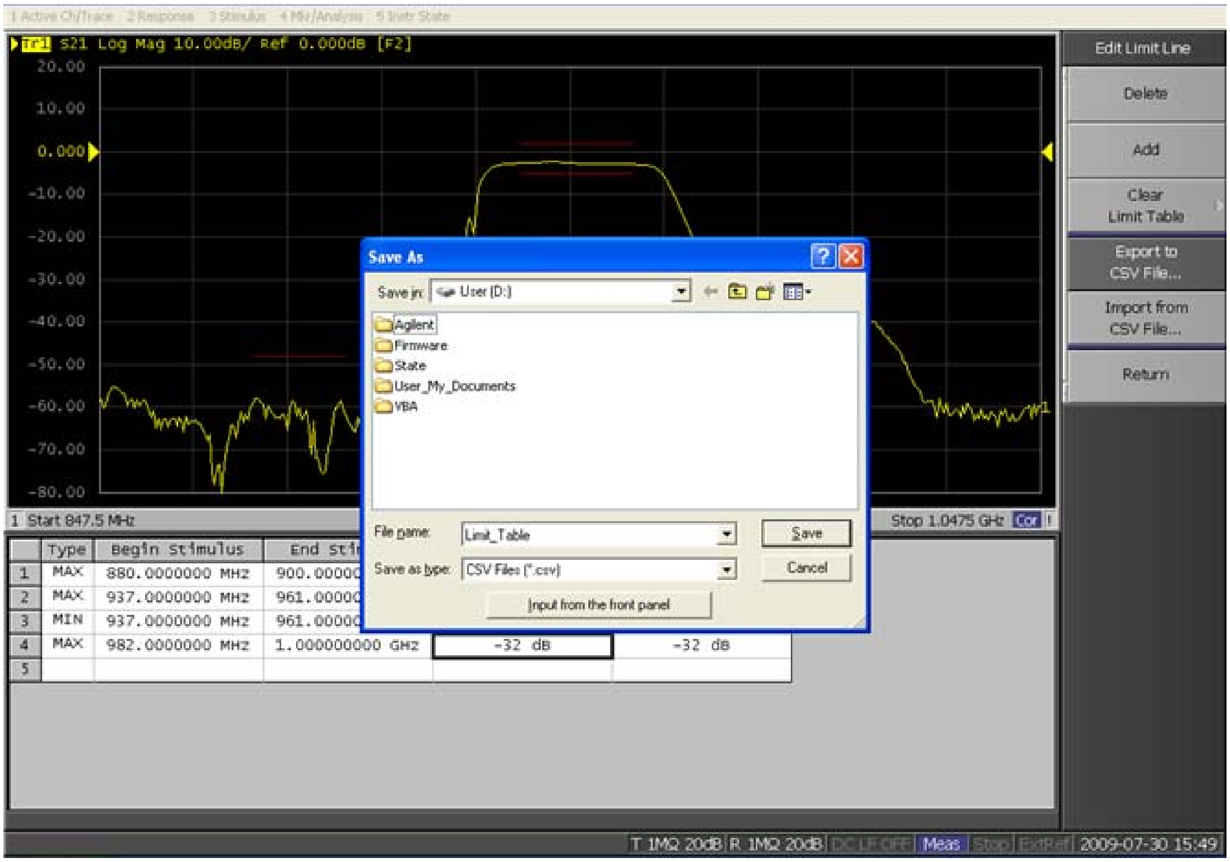
- When two or more limit values of which the type is set to maximum (MAX) exist, the smallest one is used as the maximum.
- When two or more limit values of which the type is set to minimum (MIN) exist, the largest one is used as the minimum.

NOTE Even if the span of the sweep range on the E5061B is set to 0, enter the two parameters of Begin Stimulus and End Stimulus.

NOTE When two or more response values are returned as a result of using the Smith or polar chart format, the first response value of the marker provides the object of the limit test.

Saving/calling the limit table

You can save the limit table to a file that you can then freely bring up on the screen later and use. You can import a file saved in CSV format (extension: *.csv) into spreadsheet software on a PC for later use (a numerical value is saved as strings that includes its unit).



1. Display the limit table.
2. In the Edit Limit Line menu, press **Export to CSV File** to open the Save As dialog box. In this step, CSV (extension: *.csv) is selected as the file type.
3. Specify the folder in which to save the file and enter the file name. Press **Save** to save the limit table displayed on the screen to the file.
4. Conversely, to recall a saved limit table, press **Import from CSV File** in the Edit Limit Line menu to display the Open dialog box. In this step, CSV (extension: *.csv) is selected as the file type.
5. After specifying the folder containing the file, select the file. Press **Open** to display the limit table on the screen.

   NOTE The limit table can be called from any trace of any channel, regardless of the channel or trace.

Limit Table Saved in CSV Format

The limit table is saved in the following format.

- On the first line, the channel number of the active channel that is valid when the saved file is the output.

- On the second line, the trace number of the active trace that is valid when the saved file is the output.
- The third line provides the header showing the items for the segments to be output on the fourth and later lines.
- Data on segments are output on the fourth and later lines.

"# Channel 1"

"# Trace 1"

Type, Begin Stimulus, End Stimulus, Begin Response, End Response

MAX, 200.0000000 MHz, 400.0000000 MHz, -100 dB, -100 dB

MAX, 490.0000000 MHz, 510.0000000 MHz, -10 dB, -10 dB

MIN, 490.0000000 MHz, 510.0000000 MHz, -20 dB, -20 dB

MIN, 600.0000000 MHz, 800.0000000 MHz, -100 dB, -100 dB

Turning the limit test ON/OFF

You can set the limit test ON/OFF for each trace individually.

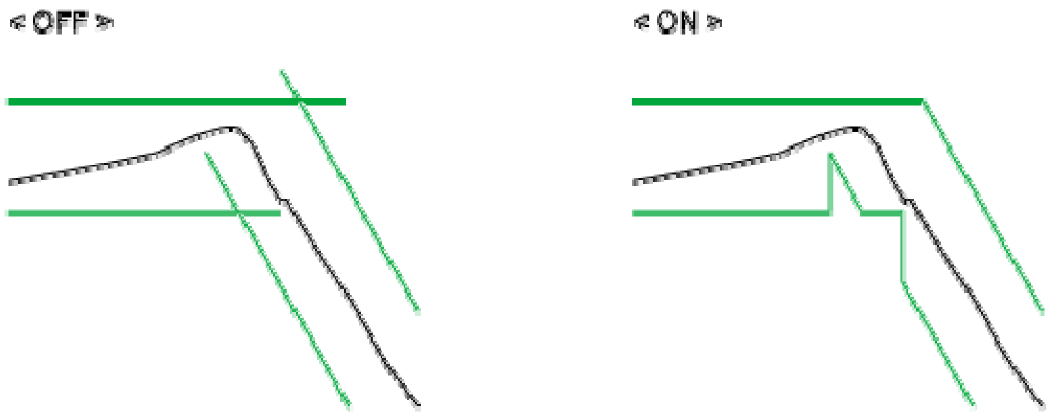
1. Press **Channel Next** (or **Channel Prev**) and **Trace Next** (or **Trace Prev**) to select the trace on which the limit test function is used.
2. Press **Analysis** > **Limit Test** to display the Limit Test menu.
3. Press **Limit Test** to set the limit test ON/OFF.
4. Press **Limit Line** to set the limit line display ON/OFF.

Changing the Limit Line Display Mode

You can specify the limit line display mode hide limit values that are not used for evaluation.
Changing the display mode:

1. Press **Channel Next** (or **Channel Prev**) and **Trace Next** (or **Trace Prev**) to activate the channel of which you want to use the limit test function.
2. Press **Analysis** > **Limit Test.**
3. Turn off **Limit Line**.
4. Press **Clip Lines** to toggle ON/OFFf.

Limit line display mode



e5061b057

Using Relative Limit Line

If the shape is more important than the amplitude, you can make the limit lines relative to the peak point of the trace using the reference tracking function.

In this function, the point to be tracked is set as the Y-axis reference value by offsetting measurement values after the sweep. Because measurement values are offset, marker values and limit test evaluation results change accordingly.

1. Press **Channel Next** (or **Channel Prev**) and **Trace Next** (or **Trace Prev**) to select the channel/trace.
2. Press **Scale** > **Reference Tracking**.
3. When you want to specify a measurement value at a frequency as the Y-axis reference value for tracking, press **Track Frequency**, then enter the frequency.
4. Press **Tracking** to select a **tracking method**.
5. **PTrk** (Track Peak) or **FTrk** (Frequency) is displayed at the trace status area.

NOTE This function is available even when the limit test function is off.

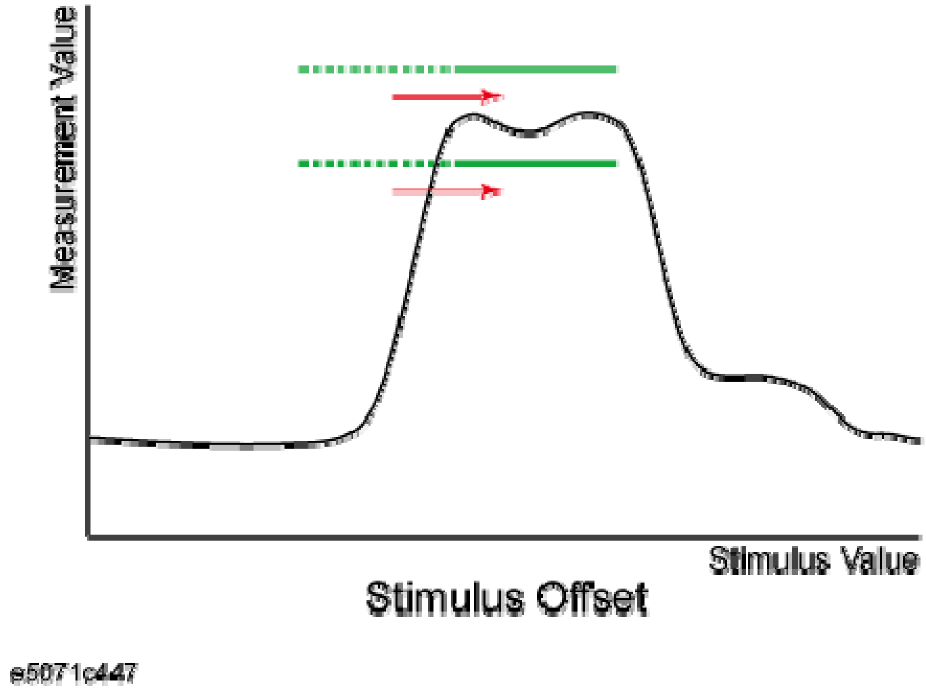
Adding Offset to Limit Line

By adding a certain offset to the limit value, you can adjust the limit line so that it conforms to the device output.

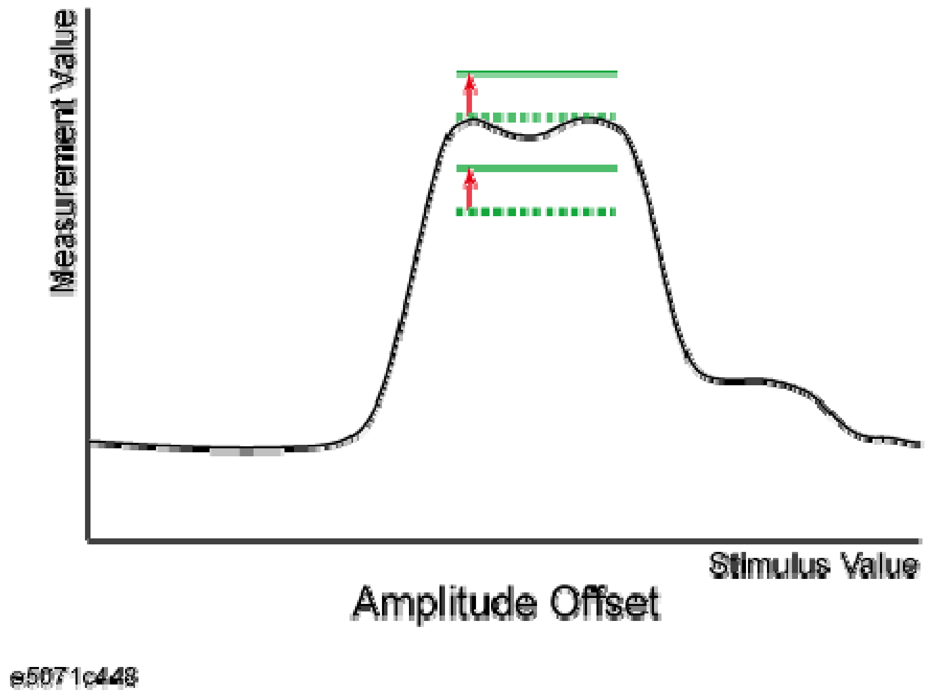
1. Press **Channel Next** (or **Channel Prev**) and **Trace Next** (or **Trace Prev**) to select the channel/trace on which the limit test function is used.
2. Press **Analysis** > **Limit Test** to display the softkeys for the limit test.

3. Click **Limit Line Offsets** to display the **limit line offset function menu**.

Stimulus offset



Amplitude offset



Initializing the Limit Table

The following operations initialize the limit table.

- At power-on
- When presetting
- When calling a limit table with zero segments
- When **Clear Limit Table** > **OK** is pressed in the Edit Limit Line menu

Using Bandwidth Test

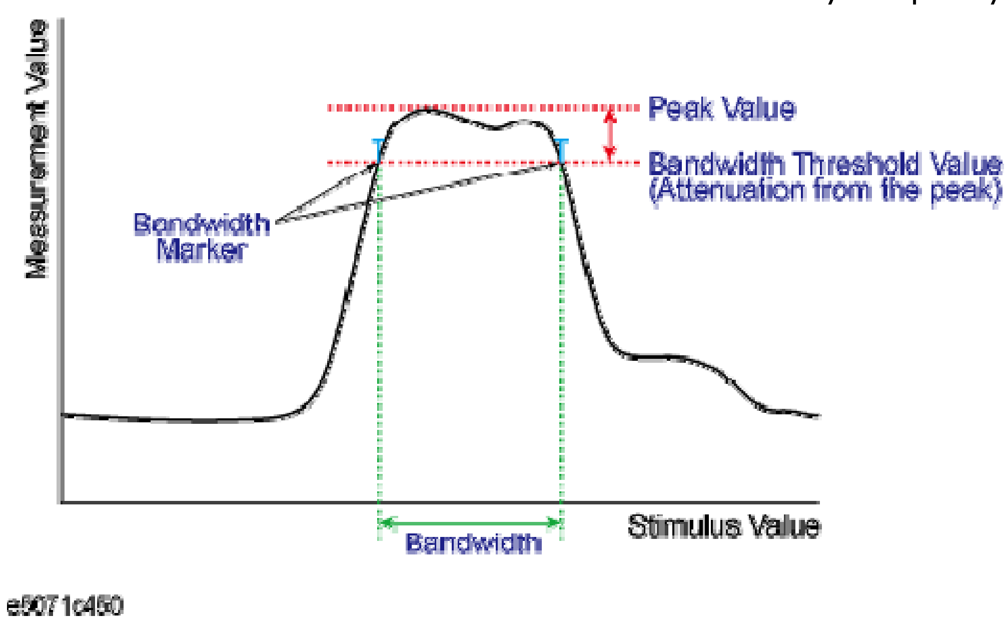
- Overview
- Displaying Bandwidth Test Results
- Bandwidth Test Setup
- Turning Bandwidth Test and Displaying Results ON/OFF

Other topics about Data Analysis

Overview

The bandwidth test function can be used to test bandwidth for the band-pass filters.

The bandwidth test find the peak of a signal in the passband and locates a point on each side of the passband at an amplitude below the peak specified in test setup. The frequency between these two points is the bandwidth of the filter. Then the obtained bandwidth is compared to minimum and maximum allowable bandwidth that you specify beforehand.

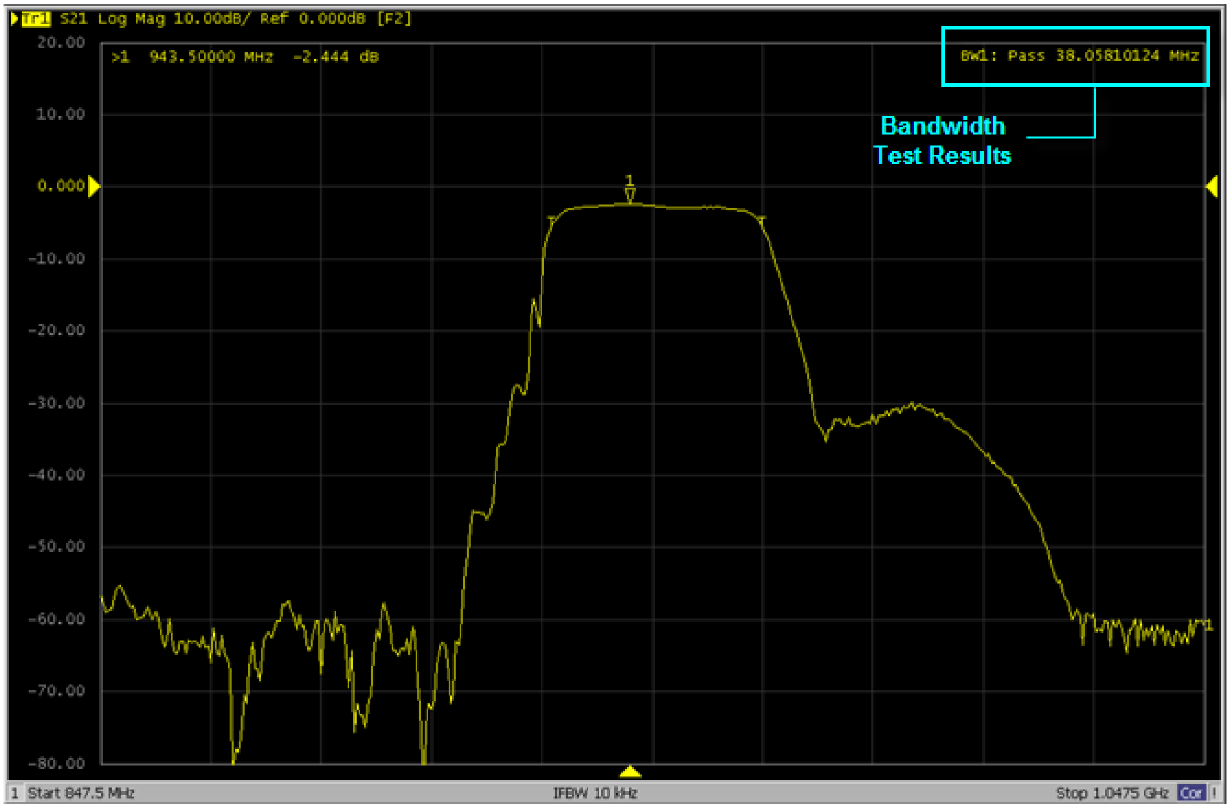


Displaying Bandwidth Test Results

Test Result for Trace

The test result of the trace is indicated in the upper-right section of the graph for each trace, following BWn:. "n" denotes the trace number. The results are shown as Pass, Wide, Narrow, or >Span (Fail). You can also choose to display the bandwidth value.

For information on how to display the results, see Turning On/Off Bandwidth Test and Displaying Results.



Test Results for Channel

If any channel is unsatisfactory, a message is displayed as shown in Judgment result of channels. (It is judged as failed if one or more failed traces are found for the limit test, ripple test, or bandwidth test within the channel.)

You can also specify this ON/OFF setting from the Fail Sign, which is provided in the limit test menu and ripple test menu. From the bandwidth test menu, follow the steps below to turn it ON/OFF.

1. Press **Analysis** key.
2. Click **Bandwidth Limit**.
3. Click **Fail Sign**. This menu toggles between ON and OFF.

In addition to the screen, the following features also let you confirm the test results:

- Beep notifying that the result is unsatisfactory
- Status register

Bandwidth Test Setup

You must set up the bandwidth threshold and the upper and lower limits before you can use the bandwidth test function. You can specify the threshold, upper limit, and lower limit for each trace.

Follow the steps below to set up the bandwidth test.

1. Press **Channel Next** (or **Channel Prev**) and **Trace Next** (or **Trace Prev**) to activate the trace for which you want to apply the bandwidth test function.
2. Press **Analysis** key to display the Analysis menu.
3. Click **Bandwidth Limit** > **N dB Points** to specify the bandwidth threshold. The unit is dB.
4. Press **Min Bandwidth** to enter the lower limit for the bandwidth. Similarly, press **Max Bandwidth** to enter the upper limit for the bandwidth. The unit is Hz for both Min and Max bandwidths.

   NOTE If the data format is Smith chart or polar, the limit test is performed for the main response value among the two marker response values.

Turning Bandwidth Test and Displaying Results ON / OFF

You can configure the ON/OFF setting of the bandwidth test function for each trace independently.

Follow the steps below to configure the ON/OFF setting of the bandwidth test.

1. Press **Channel Next** (or **Channel Prev**) and **Trace Next** (or **Trace Prev**) to activate the trace of which you want to apply the bandwidth test function.
2. Press **Analysis** key to display the Analysis menu.
3. Click **Bandwidth Limit** > **BW Test** to turn ON the bandwidth test.
4. Click **BW Marker** to turn ON so that the bandwidth marker is displayed on the screen.
5. Click **BW Display** to turn ON so that the bandwidth value is displayed.

Using Ripple Test

Other topics about Data Analysis

Overview

Independent of the limit test, you can evaluate the measurement results on a pass/fail basis by setting a limit for the ripple. This function is called the ripple test.

Concept of Ripple Test

The ripple test is a function for evaluating the results on a pass/fail basis based on the ripple limit, which is set using the ripple limit table. You can specify up to 12 frequency bands, which permits a test for each frequency band.

The ripple test judges the measurement as "Pass" when the ripple value specified with the ripple limit does not exceed any of the measurement points on the trace; otherwise, it judges the measurement as "Fail." For the measurement points in a stimulus range without a specified ripple limit, the test judges the measurement as "Pass."

> **NOTE** The measurement point alone is the target of evaluation for pass/fail. The interpolated part between measurement points is not evaluated.

The ripple limit is defined with the start point stimulus value, end point stimulus value, ripple limit value, and type (ON/OFF). For detailed information, see Configuring ripple limit.

While the ripple test function is turned ON, the measurement points corresponding to a "fail" judgment is indicated in red on the screen, and the trace's test results based on the results of each measurement point is displayed (judged as "fail" if one or more red measurement point exist on the trace). For information on how to display the results, see Turning ON/OFF ripple test and displaying results. You can also confirm the channel test results on the screen (judged as "fail" if one or more failed traces appear in the limit test, ripple test, or bandwidth test within the channel).

Displaying Ripple Test Results

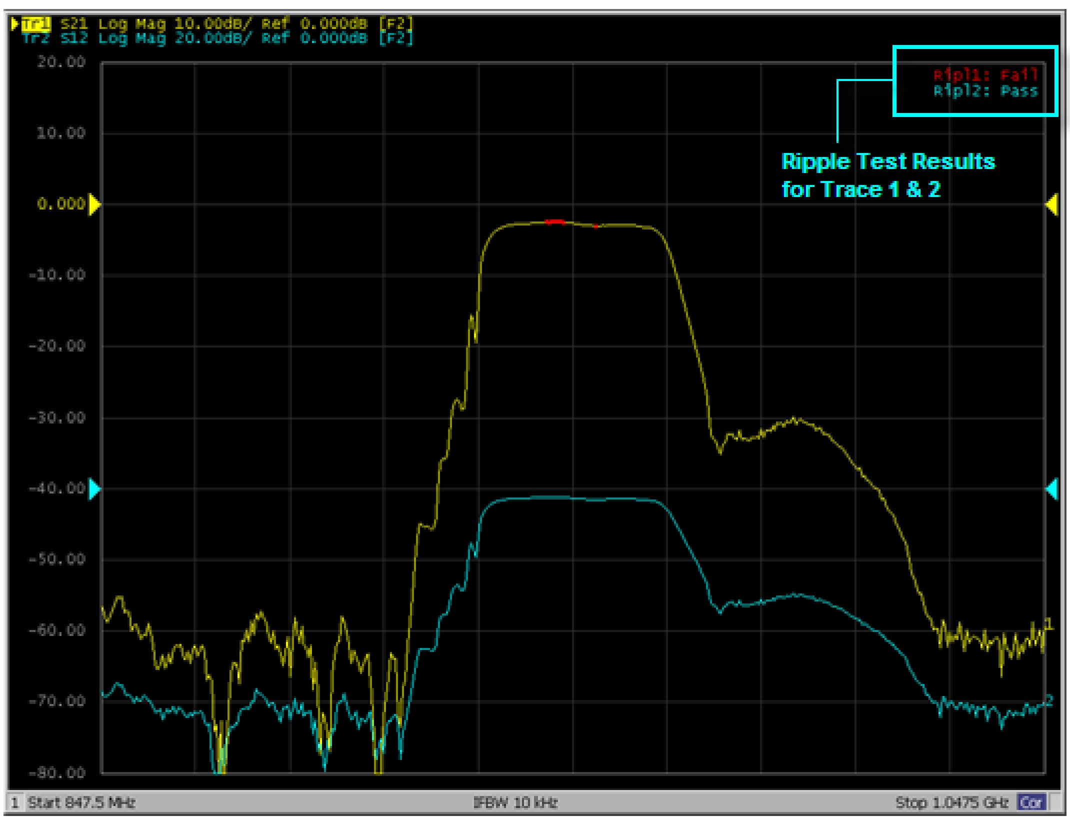
Measurement point and test results

Failed measurement points is displayed in red on the screen. The test result for the trace is indicated as Pass or Fail in the upper-right section of the graph. You can also display the ripple value at the selected frequency band.

The result is displayed as Ripln:Pass (or Fail) for each trace. n denotes the trace number. Bn is followed by the ripple value (if the ripple display is turned OFF, only Bn is displayed without the ripple value).

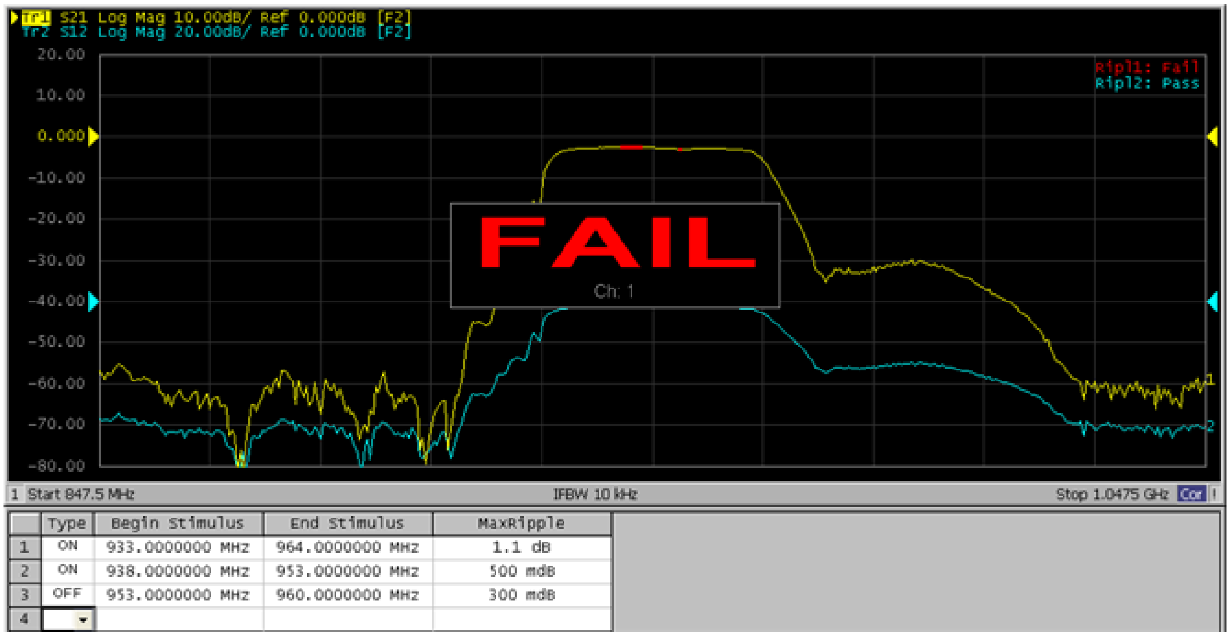
For example, in the following figure, Ripl1:Pass in the first line indicates the result for trace 1. The value following B3 is the ripple value at the third frequency band specified in the ripple test. Similarly, the second line indicates the test result for trace 2, showing the ripple value at the first frequency band.

For information on how to display the results, see Turning ON/OFF ripple test and displaying results.



Test result for channel

If a channel has a judgment result of "fail", the message is displayed as shown in Test Result for Trace. (It is judged as failed if one or more failed traces are found in the limit test, ripple test, or bandwidth test within the channel.)



You can also specify the ON/OFF setting for the Fail Sign in the limit test menu and as well as in the bandwidth test menu. From the ripple test menu, follow the steps below to turn it ON/OFF.

1. Press **Analysis** > **Ripple Limit**.
2. Click **Fail Sign**. This menu toggles between ON and OFF.

In addition to the screen, the following features also let you confirm the test results:

- Beep notifying that the result is "fail"
- Status register

Configuring Ripple Limit

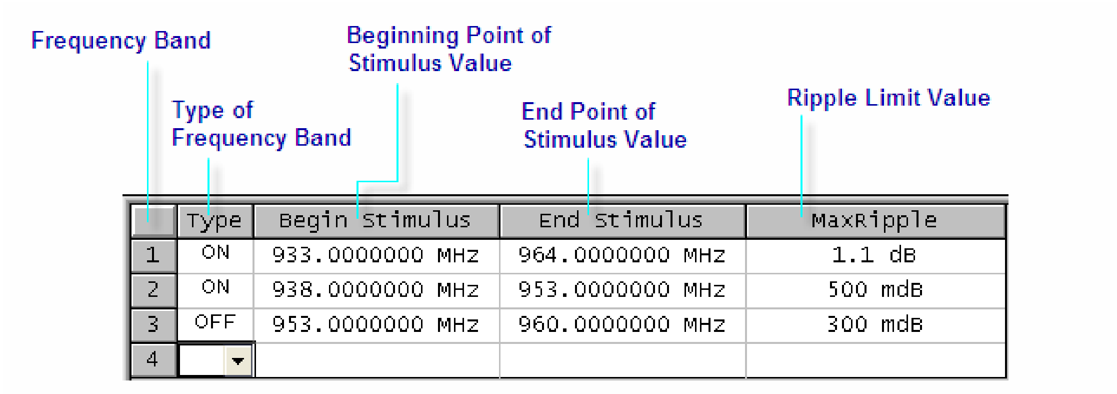
You must configure the ripple limit before you can use the ripple test function. You can specify a ripple limit table for each trace, where up to 12 ripple limit bands (frequency bands) can be configured.

Operational procedure

Follow the steps below to configure the ripple limits.

1. Press **Channel Next** (or **Channel Prev**) and **Trace Next** (or **Trace Prev**) to select the trace on which you want to apply the ripple test function.
2. Press **Analysis** > **Ripple Limit**.

3. Click **Edit Ripple Limit** to display the ripple limit table shown below.



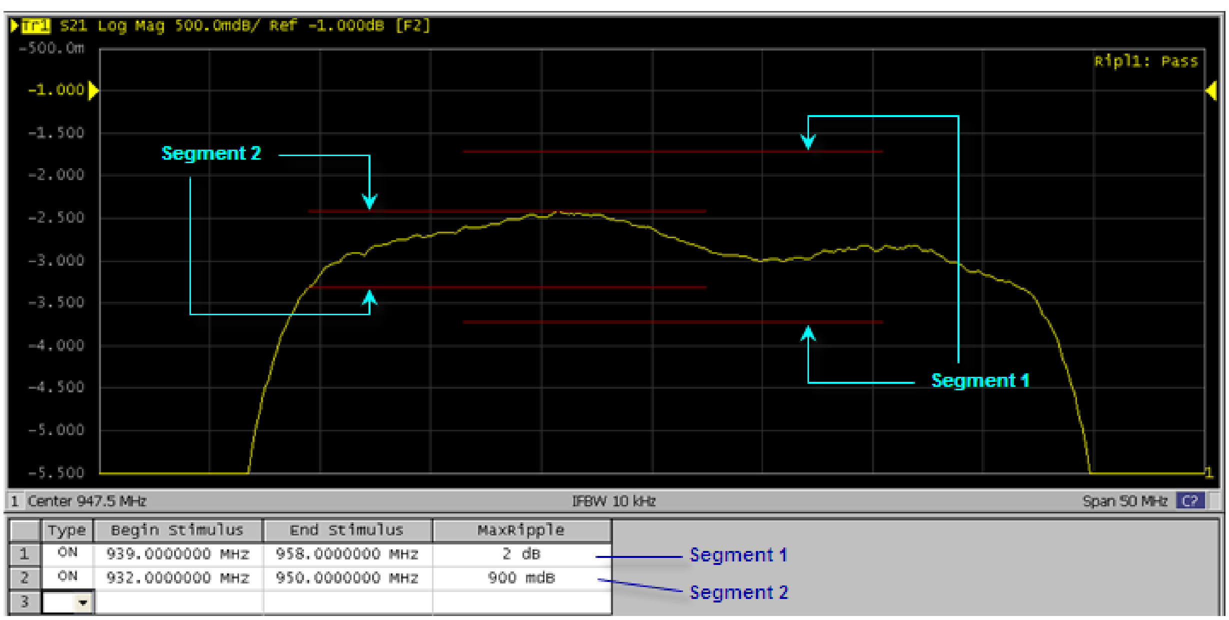
	Type	Begin Stimulus	End Stimulus	MaxRipple
1	ON	933.0000000 MHz	964.0000000 MHz	1.1 dB
2	ON	938.0000000 MHz	953.0000000 MHz	500 mdB
3	OFF	953.0000000 MHz	960.0000000 MHz	300 mdB
4				

4. Click **Add** to add a frequency band to the ripple limit table and then specify the **parameters** for the frequency band. Note that no frequency band is provided in the ripple limit table by default. Click **Delete** to delete the selected frequency band and **Clear Ripple Limit Table** to clear the table setting.

NOTE Acceptable range for the stimulus value: -500G to +500G. If any outranging value is specified, it is reset to fall within the range.

NOTE Even if the E5061B's sweep range is changed after the stimulus value has been set, the stimulus value is not susceptible.

Example of ripple limit configuration



	Type	Begin Stimulus	End Stimulus	MaxRipple
1	ON	939.0000000 MHz	958.0000000 MHz	2 dB
2	ON	932.0000000 MHz	950.0000000 MHz	900 mdB
3				

NOTE The individual frequency bands for the ripple test can overlap each other; in this case, the ripple limit test is performed for each frequency band.

NOTE Even if the E5061B's span value is set to zero, you must enter a parameter for both Begin Stimulus and End Stimulus.

NOTE If the data format is Smith chart or polar, the limit test is performed for the main response value among the two marker response values.

Saving / Recalling Ripple Limit Table

The ripple limit table can be saved in a file and recalled later for use on the screen. The file is saved in the csv format (with the extension *.csv), and values are saved as a character string with the unit. The csv formatted file can also be reused in spreadsheet software made for PCs.

Follow the steps below to save/recall the ripple limit table. This operation should be done by using the external keyboard and/or mouse.

1. Display the ripple limit table.
2. Click **Export to CSV File** from the Edit Ripple Limit menu to open the dialog box. At this time, CSV file (with the extension *.csv) is selected as the file type.
3. Specify any folder in which you want to save the file, and enter the file name. Click **Save** to save the ripple limit table displayed on the screen to a file.
4. To recall the saved ripple limit table, click **Import from CSV File** from the Edit RippleLimit menu to display the Open dialog box. At this time, CSV file (with the extension *.csv) is selected as the file type.
5. Specify the folder that contains the file, and then select the file. Click **Open** to recall the saved limit table on the screen.

   NOTE You can recall a limit table from a trace on any channel independently of the channel and trace that are active when the limit table is saved to the file.

The ripple limit table is saved in the following format:

- In the first line, the channel number for the active channel at the time of file saving will be output.
- In the second line, the trace number for the active trace at the time of file saving will be output.
- The third line is a header indicating the segment items that are output from the fourth line onwards.
- From the fourth line onwards, the segment data are output.

Operational procedure

"# Channel 1"

"# Trace 1"

Type, Begin Stimulus, End Stimulus, MaxRipple

ON, 933.0000000 MHz, 964.0000000 MHz, 1.5 dB

ON, 938.0000000 MHz, 953.0000000 MHz, 500 mdB

ON, 953.0000000 MHz, 960.0000000 MHz, 300 mdB

Turning ON / OFF Ripple Test and Result Display

You can set the limit test ON/OFF for each trace individually.

1. Press **Channel Next** (or **Channel Prev**) and **Trace Next** (or **Trace Prev**) to select the trace on which you want to apply the ripple test function.
2. Press **Analysis** > **Ripple Limit**.
3. Press **Ripple Limit Test** to set the ripple test ON.
4. Press **Ripple Limit** to display the ripple limit line.
5. Press **Ripple Value** to shows how the ripple values are displayed, then select **Absolute** (difference between maximum and minimum values within the band), **Margin** (difference between absolute value of ripple and ripple limit), or **OFF**.
6. Press **Ripple Value Band** to select the band for which you want to display the ripple value.

Initializing Limit Table

The following operations initialize the limit table.

- At power-on
- When presetting
- When calling a Ripple table with zero segments
- When **Clear Ripple Limit Table** > **OK** is clicked in the Edit Ripple Limit menu

Using Equation Editor

- Overview
- Using Equation Editor
- Equation Editor Examples
- Equation History
- Functions and Constants
- Converting S Parameters to H, Y, Z, F & T Parameters
- Operators used in Equation Editor

Other topics about Data Analysis

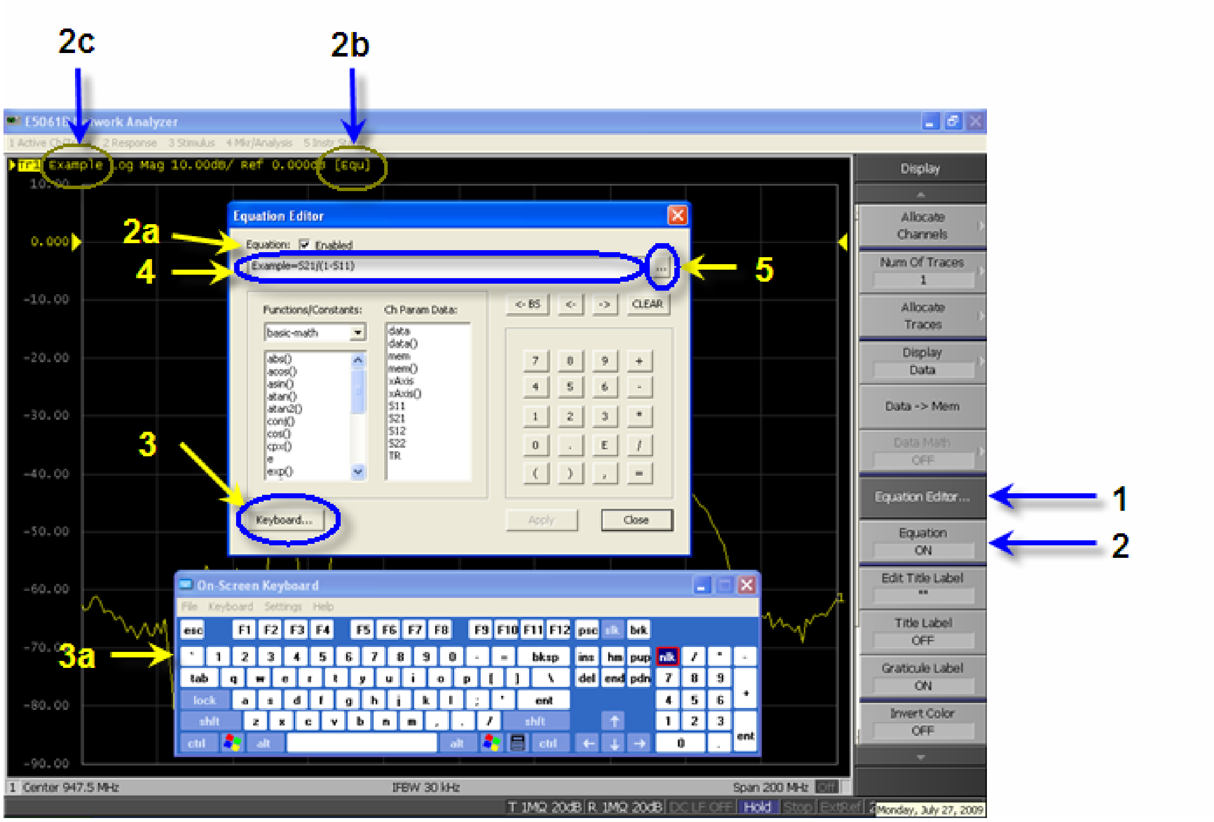
Overview

Equation Editor allows you to enter an algebraic equation of standard mathematical operators and functions, referencing data that is available in the E5061B. Once a valid equation is entered and enabled, the display of the active trace is replaced with the results of the equation, and updated in real-time as new data is acquired. For equations that can be expressed with Equation Editor's supported functions, operators, and data, there is no need for off-line processing in a separate program.

For example, on entering the equation "Example=S21/(1-S11)" in the E5061B Equation Editor (**4** in the Figure below), the resulting trace is computed as each S21 data point divided by one minus the corresponding S11 data point. For a 201 point sweep setup, the computation is repeated 201 times, once for each point.

Using Equation Editor

The step-by-step procedure of using Equation Editor is described below:



NOTE The parameter of **Z** is available in **Ch Param Data** when option 005 is installed.

Equation Editor Dialog box

1. Select a trace in which you want to enter the equation and activate the trace.

   NOTE Activating a trace is required as Equation Editor works on traces.

2. Follow the steps below to enter an equation:
   1. Press **Display**.
   2. Click **Equation Editor** (**1** in the figure above). The Equation Editor dialog box appears.
   3. Enter an equation in the equation field (**4** in the figure above).

   NOTE Referring to traces in a different channel is NOT available with Equation Editor on the ENA.

   NOTE The equation can be entered with the software keyboard enabled by selecting **Keyboard...** (**3** and **3a** in the figure above).

3. Follow the steps below to apply the defined equation. When a valid equation is entered, the Equation Enabled check box becomes available for checking.
    1. Check **Equation Enabled** check box (**2a** in the figure above).
    2. Click **Apply**. The equation becomes visible and annotation of [Equ] (**2b** in the figure above) is displayed in the trace title area.
    3. Click **Close** to hide the dialog box.

NOTE The equation can also be applied by selecting **Display** > **Equation ON**.

NOTE If error correction is not turned ON, then the raw, uncorrected data is used in the equation trace.

NOTE If an equation is NOT valid (i.e. referring to a trace that is not measured in the measured channel), annotation of [Equ!] is displayed in the trace title area instead.

Equation Editor Examples

The following examples may help you in getting started with Equation Editor. Input the equation example in the equation field (**4** in Equation Editor dialog box).0

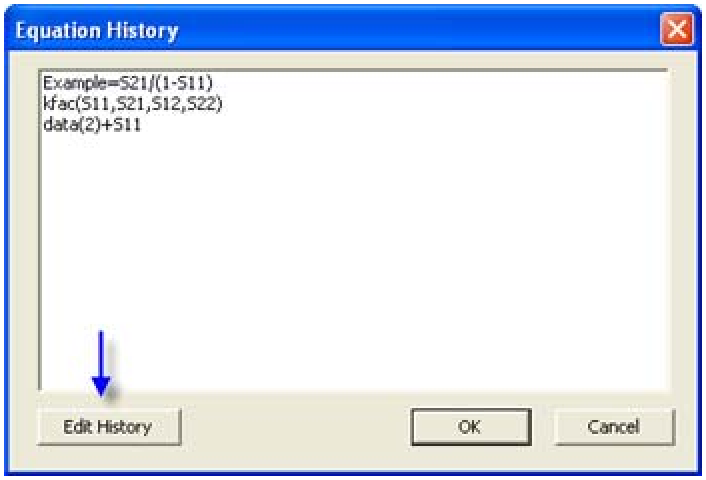
Description	Equation Example
Offset each data point in trace 2 from trace 1 by XdB	*Offset= data(1)*pow(10,X/20)*
Multiplying by X-times in log format	*XTimes= pow(data,X)*
Differential Return Loss	*Sdd11= (S11-S21-S12+S22)/2)*
CMRR of a Balanced Component	*CMRR = data(2) / data (1)*

Equation History

Equation Editor has the capability to save and recall all previously defined equations. All equations can be viewed in the Equation History dialog box.

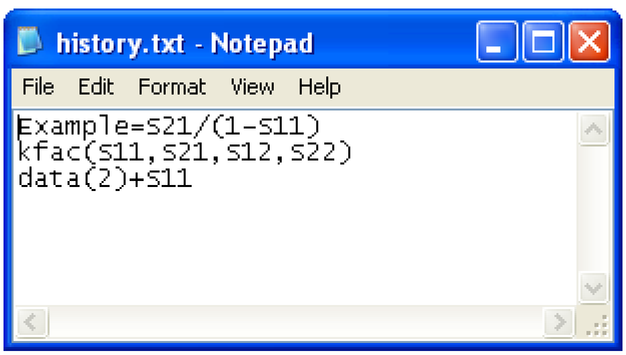
To view the equations in the list, follow this procedure:

1. Open Equation Editor by Display > **Equation Editor**
2. Enter an equation and click **Apply** in the Equation Editor dialog box to save the defined equation in the directory of the E5061B. To view a list of saved equations, click the **...** button (**5** in Equation Editor Dialog box) to open the Equation History dialog box.



NOTE To store an equation in the History List, the equation must be applied first. This can be done by clicking on the **Apply** button.

3. To edit the equations in the list, click **Edit History**. The text file of **history list** is opened with Notepad.



NOTE The **History** List is stored as a text file **D:\Agilent\Equation\history.txt** and can save a maximum of 50 lines (equations) with a maximum of 254 character per line (equation).

Functions and Constants

The following table describes the different functions and constant available in the E5061B Equation Editor. In the following table:

- Function(scalar x) means that the function requires a scalar value. If a complex value is entered, it is automatically converted to a scalar value; complex(x,y) -> scalar(x)

- Function(complex x) means that the function requires a complex value. If a scalar value is entered, it is automatically converted to a complex value; scalar(x) -> complex(x, 0)
- **a,b** are arguments that are used in the function.

Basic Math Functions

Function	Description
abs(complex **a**)	returns the sqrt(a.re^2+a.im^2)
acos(scalar **a**)	returns the arc cosine of a in radians
asin(scalar **a**)	returns the arc sine of a in radians
atan(scalar **a**)	returns the arc tangent of a in radians
atan2(complex **a**)	returns the phase of a = (re, im) in radians
atan2(scalar **a**, scalar **b**)	returns the phase of (a, b) in radians
conj(complex **a**)	returns the conjugate of a
cos(complex **a**)	takes a in radians and returns the cosine
cpx(scalar **a**, scalar **b**)	returns a complex value (a+ib) from two scalar values
exp(complex **a**)	returns the exponential of a
im(complex **a**)	returns the imaginary part of a as the scalar part of the result (zeroes the imaginary part)
ln(complex **a**)	returns the natural logarithm of a
log10(complex **a**)	returns the base 10 logarithm of a
mag(complex **a**)	returns sqrt(a.re^2+a.im^2)
phase(complex **a**)	returns atan^2(a) in degrees
pow(complex **a**,complex **b**)	returns a to the power b
re(complex **a**)	returns the scalar part of a (zeroes the imaginary part)

sin(complex **a**)	takes a in radians and returns the sine
sqrt(complex **a**)	returns the square root of a, with phase angle in the half-open interval (-π/2, π/2)
tan(complex **a**)	takes a in radians and returns the tangent
Constants	
e	2.71828182845904523536
PI	3.14159265358979323846

NOTE Mutual transformation is automatically made for scalar and complex.

scalar(x) -> complex(x, 0)
complex(x, y) -> scalar(x)

Advanced Math Functions

In the following table: **a**,**b**,**c**,**d** are arguments of complex value that are used in the function. For 2-port network measurement: **a**,**b**,**c**,**d** correspond to Sii,Sji, Sij, Sjj respectively.

NOTE The functions can also be defined by scalar arguments with port numbers of the E5061B. For example, the function, kfac(1,2) returns the K-factor of 2-port measurement between port 1 and port 2.

Z_0 refers ti the system characteristic impedance which can be accessed under Cal.

Function	Description
A(complex **a**, complex **b**, complex **c**, complex **d**)	F_{11} conversion = ((1+a)×(1-d) + b×c)/(2×b)
A(scalar i, scalar j)	returns A(Sii, Sji, Sij, Sjj)
B(complex **a**, complex **b**, complex **c**, complex **d**)	F_{12} conversion = Z_0 × ((1+a)×(1+d) - b×c)/(2×b)
B(scalar i, scalar j)	returns B(Sii, Sji, Sij, Sjj)

C(complex **a**, complex **b**, complex **c**, complex **d**)	F_{21} conversion = (1/Z_0)×((1-a)×(1-d) - b×c)/(2×b)
C(scalar i, scalar j)	returns C(Sii, Sji, Sij, Sjj)
D(complex **a**, complex **b**, complex **c**, complex **d**)	F_{22} conversion = ((1-a)×(1+d) + b×c)/(2×b)
D(scalar i, scalar j)	returns D(Sii, Sji, Sij, Sjj)
H11(complex **a**, complex **b**, complex **c**, complex **d**)	H_{11} conversion = Z_0×((1+a)×(1+d) - b×c)/((1-a)×(1+d) + b*c)
H11(scalar i, scalar j)	returns H11(Sii, Sji, Sij, Sjj)
H12(complex **a**, complex **b**, complex **c**, complex **d**)	H_{12} conversion = 2×c/((1-a)×(1+d) + b×c)
H12(scalar i, scalar j)	returns H12(Sii, Sji, Sij, Sjj)
H21(complex **a**, complex **b**, complex **c**, complex **d**)	H_{21} conversion = -2×b/((1-a)×(1+d) + b×c)
H21(scalar i, scalar j)	returns H21(Sii, Sji, Sij, Sjj)
H22(complex **a**, complex **b**, complex **c**, complex **d**)	H_{22} conversion = (1/Z_0)×((1-a)×(1-d) - b×c)/((1-a)×(1+d) + b×c)
H22(scalar i, scalar j)	returns H22(Sii, Sji, Sij, Sjj)
kfac(complex **a**, complex **b**, complex **c**, complex **d**)	k-factor = (1 - abs(a)^2 - abs(d)^2 + (abs(a×d - b×c)^2)/(2×abs(b×c))
kfac(scalar i, scalar j)	returns kfac(Sii, Sji, Sij, Sjj)
MAPG(complex **a**, complex **b**, complex **c**, complex **d**)	maximum available power gain = abs(b/c)×(kfac(a,b,c,d) - sqrt(kfac(a,b,c,d)^2 - 1))
MAPG(scalar i, scalar j)	returns MAPG(Sii, Sji, Sij, Sjj)
MSG(complex **a**, complex **b**, complex **c**, complex **d**)	maximum stable power gain = abs(b)/abs(c)

MSG(scalar i, scalar j)	returns MSG(Sii, Sji, Sij, Sjj)
mu1(complex **a**, complex **b**, complex **c**, complex **d**)	μ-factor = (1 - abs(a)^2) / (abs(d - conj(a) × (a×d-b×c)) + abs(b×c))
mu1(scalar i, scalar j)	returns mu1(Sii, Sji, Sij, Sjj)
mu2(complex **a**, complex **b**, complex **c**, complex **d**)	μ-factor = (1 - abs(d)^2) / (abs(a - conj(d) × (a×d-b×c)) + abs(b×c))
mu2(scalar i, scalar j)	returns mu2(Sii, Sji, Sij, Sjj)
T11(complex **a**, complex **b**, complex **c**, complex **d**)	T_{11} conversion = -(a×d - b×c)/b
T11(scalar i, scalar j)	returns T11(Sii, Sji, Sij, Sjj)
T12(complex **a**, complex **b**, complex **c**, complex **d**)	T_{12} conversion = a/b
T12(scalar i, scalar j)	returns T12(Sii, Sji, Sij, Sjj)
T21(complex **a**, complex **b**, complex **c**, complex **d**)	T_{21} conversion = -d/b
T21(scalar i, scalar j)	returns T21(Sii, Sji, Sij, Sjj)
T22(complex **a**, complex **b**, complex **c**, complex **d**)	T_{22} conversion = 1/b
T22(scalar i, scalar j)	returns T22(Sii, Sji, Sij, Sjj)
Y11(complex **a**, complex **b**, complex **c**, complex **d**)	Y_{11} conversion = (1/Z_0)×((1-a)×(1+d) + b×c)/((1+a)×(1+d) - b×c)
Y11(scalar i, scalar j)	returns Y11(Sii, Sji, Sij, Sjj)
Y12(complex **a**, complex **b**, complex **c**, complex **d**)	Y_{12} conversion = (1/Z_0)×(-2×c)/((1+a)×(1+d) - b×c)
Y12(scalar i, scalar j)	returns Y12(Sii, Sji, Sij, Sjj)
Y21(complex **a**, complex **b**,	Y_{21} conversion = (1/Z_0)×(-

complex **c**, complex **d**)	2×b)/((1+a)×(1+d) - b×c)
Y21(scalar i, scalar j)	returns Y21(Sii, Sji, Sij, Sjj)
Y22(complex **a**, complex **b**, complex **c**, complex **d**)	Y_{22} conversion = (1/$Z_0$0)×((1+a)×(1-d) + b×c)/((1+a)×(1+d) - b×c)
Y22(scalar i, scalar j)	returns Y22(Sii, Sji, Sij, Sjj)
Z11(complex **a**, complex **b**, complex **c**, complex **d**)	Z_{11} conversion = Z_0×((1+a)×(1-d) + b×c)/((1-a)×(1-d) - b×c)
Z11(scalar i, scalar j)	returns Z11(Sii, Sji, Sij, Sjj)
Z12(complex **a**, complex **b**, complex **c**, complex **d**)	Z_{12} conversion = Z_0×(2×c)/((1-a)×(1-d) - b×c)
Z12(scalar i, scalar j)	returns Z12(Sii, Sji, Sij, Sjj)
Z21(complex **a**, complex **b**, complex **c**, complex **d**)	Z_{21} conversion = Z_0×(2×b)/((1-a)×(1-d) - b×c)
Z21(scalar i, scalar j)	returns Z21(Sii, Sji, Sij, Sjj)
Z22(complex **a**, complex **b**, complex **c**, complex **d**)	Z_{22} conversion = Z_0×((1-a)×(1+d) + b×c)/((1-a)×(1-d) - b*c)
Z22(scalar i, scalar j)	returns Z22(Sii, Sji, Sij, Sjj)

NOTE For both mu1 and mu2, conj is the complex conjugate. For scalars a and b, conj(a+ib)=(a-ib).

Channel Parameter Data	
data	corrected data
data(scalar i)	corrected data of trace i (trace number)
mem	memory data
mem(scalar i)	memory data of trace i (trace number)
xAxis	x-axis data
xAxis(scalar i)	x-axis data of trace i (trace number)

S11 - S22	S-parameter data

NOTE When a trace number is out of range, the E5061B indicates an error message, “Equation runtime error” and equation is not reflected on the trace. The maximum number of traces can be set by **System** > **Misc Setup** > **Channel/Trace Setup.**

Converting S-Parameters to H, Y, Z, F & T-Parameters

The following section provides definition of the two port parameters in Equation Editor.

1. **Hybrid parameters (H)**

$$\begin{pmatrix} V_1 \\ I_2 \end{pmatrix} = (H)\begin{pmatrix} I_1 \\ V_2 \end{pmatrix} \qquad (H) = \begin{pmatrix} H_{11} & H_{12} \\ H_{21} & H_{22} \end{pmatrix}$$

2. **Admittance parameters (Y)**

$$\begin{pmatrix} I_1 \\ I_2 \end{pmatrix} = (Y)\begin{pmatrix} V_1 \\ V_2 \end{pmatrix} \qquad (Y) = \begin{pmatrix} Y_{11} & Y_{12} \\ Y_{21} & Y_{22} \end{pmatrix}$$

3. **Impedance parameters (Z)**

$$\begin{pmatrix} V_1 \\ V_2 \end{pmatrix} = (Z)\begin{pmatrix} I_1 \\ I_2 \end{pmatrix} \qquad (Z) = \begin{pmatrix} Z_{11} & Z_{12} \\ Z_{21} & Z_{22} \end{pmatrix}$$

Fundamental parameters (F)

$$\begin{pmatrix} V_1 \\ I_1 \end{pmatrix} = (F)\begin{pmatrix} V_2 \\ -I_2 \end{pmatrix} \qquad (F) = \begin{pmatrix} A & B \\ C & D \end{pmatrix}$$

4.

$$\begin{pmatrix} b_1 \\ b_2 \end{pmatrix} = (S)\begin{pmatrix} a_1 \\ a_2 \end{pmatrix} \qquad (S) = \begin{pmatrix} S_{11} & S_{12} \\ S_{21} & S_{22} \end{pmatrix}$$

$$a_1 = \frac{V_1 + I_1 Z_0}{2 \cdot \sqrt{Z_0}} \qquad a_2 = \frac{V_2 + I_2 Z_0}{2 \cdot \sqrt{Z_0}}$$

$$b_1 = \frac{V_1 - I_1 Z_0}{2 \cdot \sqrt{Z_0}} \qquad b_2 = \frac{V_2 - I_2 Z_0}{2 \cdot \sqrt{Z_0}}$$

5. Scattering transfer parameters (T)

$$\begin{pmatrix} b_1 \\ a_1 \end{pmatrix} = (T)\begin{pmatrix} a_2 \\ b_2 \end{pmatrix} \qquad (T) = \begin{pmatrix} T_{11} & T_{12} \\ T_{21} & T_{22} \end{pmatrix}$$

S-parameters of 2-port network can be converted to Hybrid parameters (H), Admittance parameters (Y), Impedance parameters (Z), Fundamental parameters (F) and Scattering transfer parameters (T) using the following functions:

1. **Converting S-parameters to H-parameters**

$$H_{11} = Z_0 \cdot \frac{(1+S_{11})(1+S_{22}) - S_{12}S_{21}}{(1-S_{11})(1+S_{22}) + S_{12}S_{21}}$$

$$H_{12} = \frac{2 \cdot S_{12}}{(1-S_{11})(1+S_{22}) + S_{12}S_{21}}$$

$$H_{21} = \frac{-2 \cdot S_{21}}{(1-S_{11})(1+S_{22}) + S_{12}S_{21}}$$

$$H_{22} = \frac{1}{Z_0} \cdot \frac{(1-S_{11})(1-S_{22}) - S_{12}S_{21}}{(1-S_{11})(1+S_{22}) + S_{12}S_{21}}$$

2. Converting S-parameters to Y-parameters

$$Y_{11} = \frac{1}{Z_0} \cdot \frac{(1-S_{11})(1+S_{22}) + S_{12}S_{21}}{(1+S_{11})(1+S_{22}) - S_{12}S_{21}}$$

$$Y_{12} = \frac{1}{Z_0} \cdot \frac{-2 \cdot S_{12}}{(1+S_{11})(1+S_{22}) - S_{12}S_{21}}$$

$$Y_{21} = \frac{1}{Z_0} \cdot \frac{-2 \cdot S_{21}}{(1+S_{11})(1+S_{22}) - S_{12}S_{21}}$$

$$Y_{22} = \frac{1}{Z_0} \cdot \frac{(1+S_{11})(1-S_{22}) + S_{12}S_{21}}{(1+S_{11})(1+S_{22}) - S_{12}S_{21}}$$

3. Converting S-parameters to Z-parameters

$$Z_{11} = Z_0 \cdot \frac{(1+S_{11})(1-S_{22})+S_{12}S_{21}}{(1-S_{11})(1-S_{22})-S_{12}S_{21}}$$

$$Z_{12} = Z_0 \cdot \frac{2 \cdot S_{12}}{(1-S_{11})(1-S_{22})-S_{12}S_{21}}$$

$$Z_{21} = Z_0 \cdot \frac{2 \cdot S_{21}}{(1-S_{11})(1-S_{22})-S_{12}S_{21}}$$

$$Z_{22} = Z_0 \cdot \frac{(1-S_{11})(1+S_{22})+S_{12}S_{21}}{(1-S_{11})(1-S_{22})-S_{12}S_{21}}$$

4. Converting S-parameters to F-parameters

$$A = \frac{(1+S_{11})(1-S_{22})+S_{12}S_{21}}{2 \cdot S_{21}}$$

$$B = Z_0 \cdot \frac{(1+S_{11})(1+S_{22})-S_{12}S_{21}}{2 \cdot S_{21}}$$

$$C = \frac{1}{Z_0} \cdot \frac{(1-S_{11})(1-S_{22})-S_{12}S_{21}}{2 \cdot S_{21}}$$

$$D = \frac{(1-S_{11})(1+S_{22})+S_{12}S_{21}}{2 \cdot S_{21}}$$

5. Converting S-parameters to T-parameters

$$T_{11} = \frac{-D_S}{S_{21}}$$

$$T_{12} = \frac{S_{11}}{S_{21}}$$

$$T_{21} = \frac{-S_{22}}{S_{21}}$$

$$T_{22} = \frac{1}{S_{21}}$$

Where:

$D_s = S_{11}S_{22} - S_{12}S_{21}$

Operators used in Equation Editor

Operator	Description
+	Addition
-	Subtraction
*	Multiplication
/	Division
^	Power
(	Open parenthesis
)	Close parenthesis
,	Comma - separator for arguments
=	Equal (optional)
E	Exponent (as in 23.45E6)

NOTE

Priority of operators is:

1. ^
2. *, /
3. +, -

Outputting Data

Outputting Data

- Saving and Recalling Instrument State
- Saving/Recalling Instrument State for Each Channel into/from Memory
- Compatibility of State File
- Saving Trace Data to a File
- Saving the Screen Image to a File
- Printing Displayed Screen

Saving and Recalling Instrument State

- Overview
- Saving Data
- Recall Procedure
- Recall Procedure Using "Recall by File Name"
- Priority of Recalling Configuration File at Startup

Other topics about Data Output

Overview

You can save the instrument state of the E5061B into a file on mass storage and then recall it later to reproduce that state. You can select the stored data from the following four types.

Type	Stored data and usage
State only (State Only)	Saves the setting of the E5061B and reproduces the state when it is saved by recalling it later into the E5061B.
State and calibration data (State & Cal)	Saves the setting of the E5061B and calibration data (calibration coefficient array) to reproduce the state when it was saved by recalling it later into the E5061B. At this time, you can perform error correction of measured values by using the recalled calibration data.
State and trace (State & Trace)	Saves the setting of the E5061B and traces (error-corrected data array and error-corrected memory array) to reproduce the state when it was saved by recalling it later into the E5061B. At this time, the traces are also recalled and displayed on the screen.
State, calibration data, and traces (All)	Saves the setting of the E5061B, calibration data, and traces to reproduce the state when it was saved by recalling it later into the E5061B. At this time, the calibration data and traces are also recalled.

In addition, the user-preset function is provided to allow the user to freely set up an instrument state recalled when the preset function is executed.

Saving Data

Selecting Content to be Saved

NOTE This setting takes effect both, when saving the entire instrument state into a file and when saving the instrument state for each channel into memory.

1. Press **Save/Recall** > **Save Type**.
2. Click the softkey corresponding to the content of the instrument state you want to save.

Selecting Save Target Channel/Trace

1. Press **Save/Recall** key.
2. Click **Channel/Trace** and select the save target from all channels/traces (**All**) or displayed channel/traces only (**Disp Only**).

If you specify the displayed channel/traces only as the save target, you can reduce the file size. However, for channels/traces that are not displayed, you cannot recall and reproduce the instrument state separately held for each channel/trace at a later time.

Saving Instrument State

Follow the procedure below to save internal data from the E5061B.

1. Press **Save/Recall** > **Save State.**
2. Click the softkey corresponding to the destination you want to save.

Softkey	Description
State01 to **State08**	Save the instrument state into the state number.
Autorec	Save the instrument state as the auto recall setting. The E5061B is automatically configured with this state at the startup. This key saves the state into the "**D:Autorec.sta**". When **Autorec.sta** file is found on the D: drive at startup, the E5061B is automatically configured using the saved settings. To disable the auto recall function, delete the **Autorec.sta** files.
User Pres	Save the instrument state as user preset. The user can preset the analyzer at user saved status.
File Dialog...	Save the instrument state as your desired file name. You can enter a file name using the **Input from the front panel** buttons on the dialog box when storing a file.

NOTE If **D:\Autorec.sta** is found on the system at startup, the E5061B is automatically configured using the saved settings. When the external floppy disk drive is connected as A: drive, then if **A:\Autorec.sta** is found at startup, the E5061B is also automatically configured using the saved settings. If both files are found, **A:\Autorec.sta** is recalled. To disable the auto recall function, delete the **Autorec.sta** files.

NOTE An asterisk (*) in the upper right of the softkey indicates that the corresponding file of the softkey already exists. If you save into the existing file, the existing file is copied as **backup.sta** and then overwritten.

Recall Procedure

Follow the procedure below to recall internal data from the E5061B.

NOTE If you recall a file that includes traces (its content was set to **State &Trace** or **All** when it was saved), the trigger source is automatically set to Manual.

1. Press **Save/Recall** > **Recall State**.

When you want to recall State01.sta - State08.sta, Autorec.sta

1. Press **State01** - **State08** or **Autorec**.

When you want to recall other files

1. Press **File Dialog...** to open the Open dialog box.
2. Select the folder and the file using the external keyboard and mouse.
3. Click **Open**.

The warning messages may appear when recall fails:

NOTE When a user file is used in Extending the Calibration Plane Using Network De-embedding, Determining Characteristics After Adding a Matching Circuit, or Determining the Characteristics that Result from Adding a Matching Circuit to a Differential Port and the setup status is saved, a recall error occurs if the user file is not located in the same folder as when the state is saved.

NOTE Pressing **Save/Recall** > **Explorer** executes Windows Explorer. This helps you to browse the files in the ENA hard disk drive.

Recall Procedure Using "Recall by File Name" Feature

You can use the recall feature with the **Recall by File Name** softkey for files you have named freely and save in the **D:\State** folder. This function lets you recall a file you have named freely and save by simple softkey operation, eliminating annoying operation using the Open dialog box.

NOTE Although there is no limit to the number of files saved in a folder, only up to 50 files are displayed on the softkeys. If more than 50 files are saved in a folder, they are sorted in the order of numbers 0 to 9 and alphabetic characters A to Z and the first 50 files are displayed as softkeys.

Although there is no limit to the number of characters of a file name, only up to 12 characters are displayed on the softkey. If a file name exceeds 12 characters, the first 12 characters are displayed on the softkey and the remaining characters are omitted and replaced with "...".

NOTE Different files may be displayed on softkeys with the same name or a saved file is not displayed on any softkey because of the above limitations.

1. Press **Save/Recall** > **Recall by File Name**.
2. Files that have been named and saved in the **D:\State** folder are displayed on softkeys. Press the key for the file you want to recall.

Priority of Recalling Configuration File at Startup

If several instrument configuration files exist at the startup of the E5061B, only one file is recalled and set at a time in the following order of priority.

If these files do not exist, the normal preset (factory preset) is executed.

Priority	Recalled file
1	Configuration file for the auto-recall function in the A drive (If external floppy disk drive is connected.)
2	Configuration file for the auto-recall function in the D drive
3	Configuration file for the user-preset function in the D drive. Executed when the preset operation mode is User and the file (**D:\UserPreset.sta**) exists.

Saving/Recalling Instrument State for Each Channel into/from Memory

- Overview
- Saving Instrument State for Each Channel
- Recalling Instrument State for Each Channel
- Deleting Saved Instrument State (Clearing all Registers)

Other topics about Data Output

Overview

The E5061B allows you to save/recall the instrument state for each channel independently. This function allows you to save the instrument state of the active channel independently into one of the four registers (A to D, volatile memory) and to recall the instrument state from the register to restore it as the state of the currently active channel. As in the case of saving the entire state of the instrument into a file, you can select items to be saved from four kinds.

Since you can recall the instrument state for each channel that was saved with this function from a different channel that the one used to save it, this function is very useful for copying an instrument state between channels.

NOTE Unlike when saving the entire instrument state, the instrument state for each channel is saved into volatile memory instead of a file, so if you turn off the power, this state is lost.

Saving Instrument State for Each Channel

1. Press **Channel Next** (or **Channel Prev**) to activate a channel of which the state you want to save.
2. Press **Save/Recall** > **Save Channel**.
3. Click one of **State & Cal A** to **D**, **Cal Only A** to **D** to save the (instrument state and) calibration data of the active channel to the specified register.

NOTE For registers having saved data, the * symbol is displayed to the right of their softkey label. If you specify one of these, its content is overwritten.

Recalling Instrument State for Each Channel

1. Press **Channel Next** (or **Channel Prev**) to activate a channel of which the state you want to recall and restore.
2. Press **Save/Recall** > **Recall Channel**.

3. Click the softkey of the register in which the state you want to restore is saved. This instrument state is recalled to the active channel.

Deleting Saved Instrument State (Clearing all Registers)

1. Press **Save/Recall** > **Save Channel**.
2. Click **Clear States**. The contents of all the registers are deleted.

Compatibility of State File

- Overview
- Compatibility of Files (Saving and Recalling)
- State File Converter

Other topics about Data Output

Overview

As mentioned in Saving and Recalling Instrument State, you can save the instrument state of the E5061B into a file on the mass storage and then recall it later to reproduce that state.

Alternatively, you can use the state file saved in E5061A and E5062A in a similar way.

Compatibility of Files (Saving and Recalling)

Compatibility from E5061A/E5062A and other ENA series products

State files from E5061A/E5062A can be converted to E5061B compatible state file. This can be performed by using the State File Converter Software. However, there is no compatibility of state files between E5061B and other ENA series products.

Files saved with the E5061A, E5062A and the other ENA series products cannot be recalled with the E5061B. Similarly, files saved with the E5061B cannot be recalled with the E5061A and E5062A.

Compatibility between different options

- The files saved with the E5061B without Time domain/Fault Location option can be recalled in the E5061B with Time domain/Fault Location option. The files saved with the E5061B with Time domain/Fault Location option can be recalled with the E5061B without Time domain/Fault Location option. The settings for Fault Location/Gating/SRL are not used.
- The files saved with the E5061B without Impedance Measurement option can be recalled in the E5061B with Impedance Measurement option. The files saved with the E5061B with Impedance Measurement option can be recalled with the E5061B without Impedance Measurement option.

Compatibility between different frequency models

	Recalling by E5061B									
	Opt.	3L5	23	21	13	11	23	21	13	11

Saved file by E5061B	No		5	5	5	5	7	7	7	7
	3L5	Y	N	N	N	N	N	N	N	N
	235	Y	Y	N	N	N	N	N	N	N
	215	Y	Y	Y	N	N	N	N	N	N
	135	Y	Y	N	Y	N	N	N	N	N
	115	Y	Y	Y	Y	Y	N	N	N	N
	237	N	N	N	N	N	Y	N	N	N
	217	N	N	N	N	N	Y	Y	N	N
	137	N	N	N	N	N	Y	N	Y	N
	117	N	N	N	N	N	Y	Y	Y	Y

Y: Recall is possible, N: Recall is not possible.

Compatibility when the Firmware revision is different

- Files that are saved with later firmware revisions may not be recalled by prior firmware revisions.

Compatibility when the system spec. version (available with ":SERV:SREV?") is different

- When files that are saved with different system spec. versions include calibration data, only states and trace data are recalled.

NOTE If you recall an incompatible file, an error occurs and the device recovers to the default setting.

State File Converter

The State File Converter software converts the state files from E5061A/E5062A to E5061B compatible state file.

Most state file settings from E5061A/E5062A can be converted to be used in E5061B except:

- **User calibration data**

  User calibration data cannot be converted by the State File Converter. Even if the E5061A/62A state file has calibration data, the calibration data is ignored and is not converted.

- **Multiport test set settings**

  Multiport test set settings are not converted. This is because unlike E5061A/62A, the E5061B does not support multiport test sets. Hence, the below settings are not converted:

  : SENSe [1-4]: CORRection: MULTiport: COLLect: METHod

  : SENSe [1-4]: CORRection: MULTiport: SELF [: STATe]

  : SENSe [1-4]: CORRection: MULTiport: SELF: TIMer

  : SENSe [1-4]: CORRection: MULTiport [: STATe]

  : SENSe [1-4]: MULTiport: PORT [1-2]: SELect

  : SENSe [1-4]: MULTiport: PROPerty

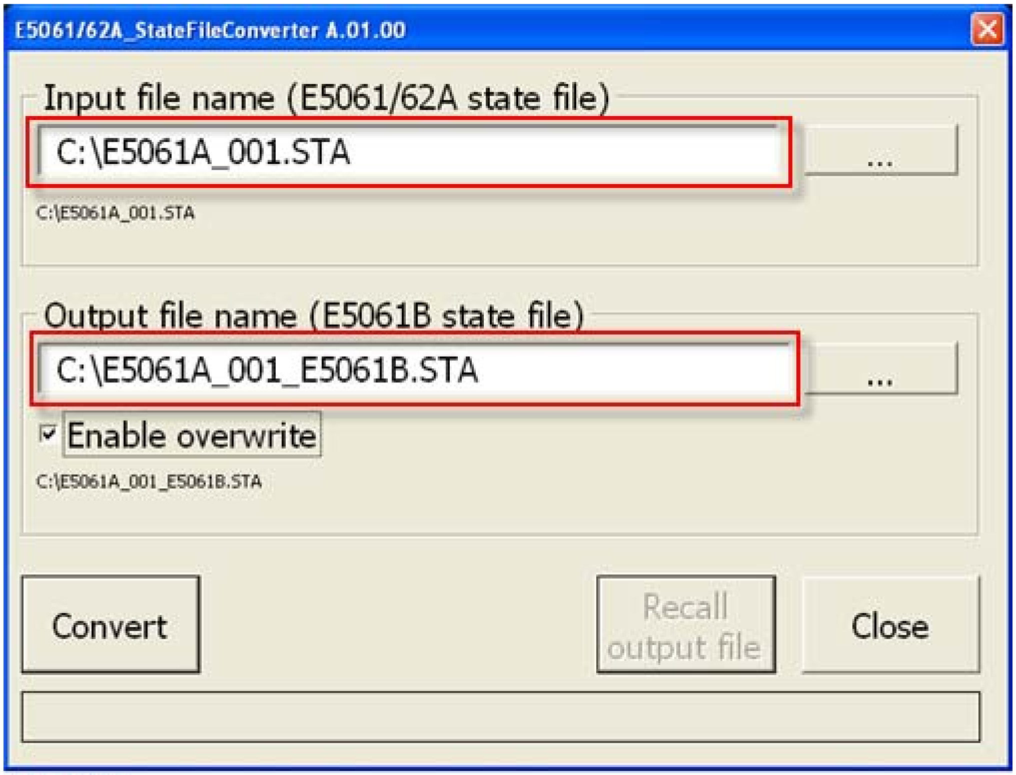
  : SENSe [1-4]: MULTiport [: STATe]

Executing State File Converter (UI)

The State File Converter is located in the E5061B at D:\Agilent\E5061A_StateFileConverter.vba. This vba software is a sort of user interface and it calls the main program named "StateFileConverter.exe" shown below.

To execute this converter file:

1. Press **Macro Setup** button.
2. Click **Load Project...** from the **Macro Setup** menu.
3. Select D:\Agilent\E5061A_StateFileConverter.vba file and click **Open.**
4. Click **Select Macro** from the **Macro Setup** menu and select **Module1 main** option.
5. The E5061A/62A State File Converter shows.



e5061b087

6. At **Input file name (E5061A/62A state file)**, select the state file from E5061A or E5062A that you want to convert.
7. At **Output file name (E5061B state file)**, enter the name of the new state file or select an existing state file to be overwritten.
8. Once both input and output file names are selected, the **Convert** button is activated.
9. Click **Convert** to execute the conversion.
10. Once the conversion process is completed, the **Recall output file** button is activated.
11. Click **Recall output file** to recall the output file. This is a shortcut method to recall the output file instead of **Save/Recall** > **Recall State** > **File Dialog**.

Executing State File Converter (.exe)

The executable file of the State File Converter is located in the E5061B at C:\Program

Files\Agilent\E5061B\StateFileConverter\StateFileConverter.exe. To execute this converter file using command prompt:

1. Go to **Command Prompt.**
2. Go to C:\Program Files\Agilent\E5061B\StateFileConverter location.
3. Type **StateFileConverter.exe -i:<input.sta> -o:<output.sta>.**

Mandatory Option

-I:<input file path> (E5061/62A state file)

-O:<output file path> (E5061B state file)

No space after the colon

Return code

0: no error

1: no error (cal data discarded)

10: read file error

20: write file error

NOTE StateFileConverter.exe should be executed from the original directory as it calls for other files to run successfully.

You can create the batch file as shown below.

Convert.bat

```
@echo off
if not exist E5061B mkdir E5061B
for %%i in (*.sta) do (
echo Converting %%~fi
"C:\Program Files\Agilent\E5061B\StateFileConverter\StateFileConverter.exe" -i:"%%~fi" -
o:"%~dp0E5061B\%%~nxi"
if errorlevel 20 (
echo     Write File Error
) else if errorlevel 10 (
echo     Read File Error
) else if errorlevel 1 (
```

```
echo    Convert Success without Cal Data
) else if errorlevel 0 (
echo    Convert Success
)
)
```

How to execute the batch file

The batch file allows you to convert multiple state files at once.

To execute the batch file:

1. Place convert.bat and all the state files in the same folder.
2. Go to **Command Prompt**.
3. Go to the location where convert.bat and state files are stored.
4. Type **convert.bat**.
5. A sub-folder named E5061B is created automatically where all the converted state files are placed.

State File Converter Messages

Below are the messages that may prompt during state file conversion:

- **"Converting..."**
- **"Conversion done"**
- **"Conversion done (CAL data discarded)"**

Conversion is successfully completed and the original CAL data is deleted.

- **"Input file does not exist"**
- **"Output file does not exist"**
- **"Output file name is an existing folder"**
- **"Input file error"**

The system cannot understand the file path as the folder or file name is in Asian characters (fonts), or the file is not a state file created by E5061A/62A.

- **"Output file error"**

The system cannot understand the file path as the folder or file name is in Asian characters (fonts).

- **"Input file name and output file name are the same"**

Saving Trace Data to a File

- [Saving Data in CSV Format]
- [Saving Data in Touchstone Format]

Other topics about Data Output

Saving Data in CSV Format

The E5061B allows the user to save data for the active trace on the active channel to a CSV file (file extension *.csv) and to load the data into PC application software for further processing.

Trace data are saved in the format shown below.

Example of saved trace data

"# Channel 1"

"# Trace 1"

Frequency, Formatted Data, Formatted Data

+3.00000000000E+005, +1.41837599227E-002, +1.43446459328E-006

+4.27985000000E+007, +1.41275293412E-002, +2.02407834551E-004

+8.52970000000E+007, +1.41334093048E-002, +4.00643331604E-004

+1.27795500000E+008, +1.41240661092E-002, +6.09250514670E-004

+1.70294000000E+008, +1.41402155348E-002, +8.05620003993E-004

The first line shows the number of the active channel at the time the data is saved.

The second line shows the number of the active trace at the time the data is saved.

The third line is a header line indicating the contents of each item of trace data written on the fourth line onward.

The fourth line onward shows the trace data. The amount of data is determined by the number of points (frequency) assigned to the trace.

Saving Trace Data

Follow the procedure below to save trace data from the E5061B.

1. Press **Channel Next** (or **Channel Prev**) and **Trace Next** (or **Trace prev**) to select the trace to be saved.
2. Press **Save/Recall** > **Save Trace Data** to open the Save As dialog box.
3. Select the destination folder and input a file name.
4. Click **Save** to save the file.

NOTE This function is inactive when the security level is set at low/high.

Saving Data in Touchstone Format

You can also save trace data of a E5061B active channel of S-parameter to a Touchstone format file.

Touchstone file data format

You can save data in "log magnitude - angle", "linear magnitude - angle", or "real number - imaginary number."

When AUTO is selected, the data format is automatically set according to the display format of the active trace. However, when the display format of the active trace is set to one other than the log magnitude format (LogMag), linear magnitude format (LinMag), or real-imaginary number format (Real/Imag), the data format is automatically set to real-imaginary number.

You can use data saved in Touchstone format for a circuit simulator such as Agilent Advanced Design System (ADS) on your PC (personal computer) or workstation. For more information on the ADS, refer to the operation manual that comes with the system.

File types of Touchstone files

File types of the E5061B Touchstone files are s1p and s2p. The file type indicates the number of ports of the data structure that is output to the Touchstone file.

Data structure in Touchstone file

Data structure of the Touchstone file consists of a header part and a data part. The contents of the file is text data, which is ready to be read with a general text editor.

The header part consists of the returned value of *IDN?, file created date, calibration state, list of all S parameters of a specified port, and format information.

The header parts of s1p and s2p are shown below.

Header of s1p

```
!Agilent Technologies,E5061B,<ID>,<FW Revision>
!Date <Date>
!Data & Calibration Information
!Freq Syy:Method(Stat)
# Hz S FMT R Z0
```

Header of s2p

```
!Agilent Technologies,E5061B,<ID>,<FW Revision>
!Date <Date>
!Data & Calibration Information
!Freq Syy:Method(Stat) Syz:Method(Stat) Szy:Method(Stat) Szz:Method(Stat)
# Hz S FMT R Z0
```

Each item has the following meaning:

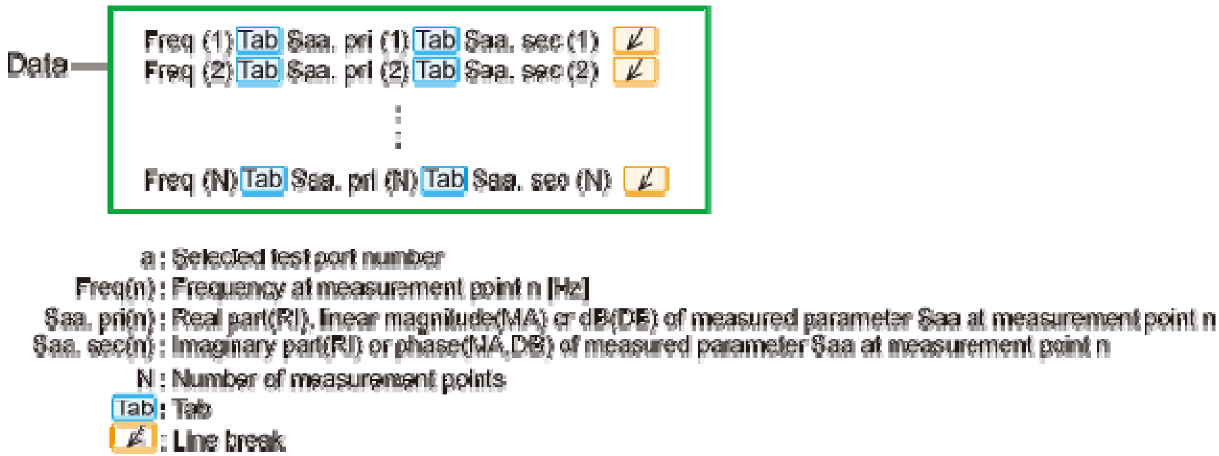
Parameter	Description
Syy to Szz	S parameters of the selected test port; corresponds in ascending order, beginning with y to z.
Method	Calibration type applied to S parameter.
Stat	State of S parameter calibration and error correction setting (ON, OFF, or --) ON = Error correction is set to ON OFF =Error correction is set to OFF -- = Calibration is not performed
FMT	Data format RI = Real number - imaginary number MA = Linear magnitude - angle DB = Log magnitude - angle
Z0	Reference impedance value

The structure of the data part depends on the combination of the selected file type and specified port.

NOTE When the S parameter of a port on which calibration is not performed is specified, if the data measured with the S parameter exists, that data are output to a Touchstone file. If no measurement data exists, 0 (for log magnitude - angle, log magnitude = -200 dB) is output in the corresponding field.

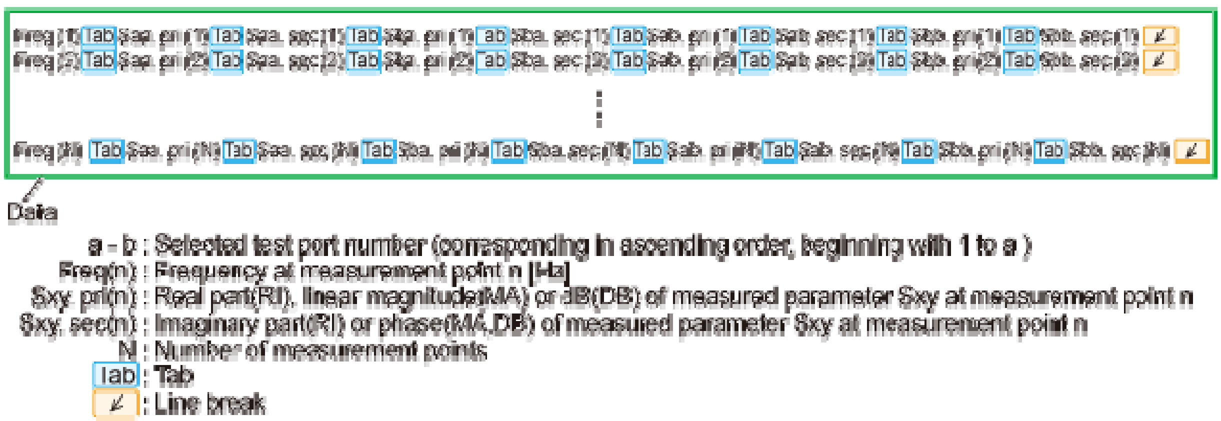
The following figures show the data structures of files saved in Touchstone format.

1-port Touchstone file



e5071c346

2-port Touchstone file



e5071c347

Restrictions when saving data in Touchstone format

The following restrictions apply in saving measurement data into Touchstone format.

- When both fixture simulation and port impedance conversion are ON, all Z0 of the ports to be saved must be set to the same value. If Z0 is different among the ports, no error occurs, but only the Z0 of the smallest port number is output to the header.
- When the time domain function is ON, the saved data are not the displayed data but the data of the S parameter before conversion.
- For data saved in touchstone format, data operation, time domain, parameter conversion, data format, electrical delay, equation editor, and smoothing are not reflected in the output data.

Saving procedure

Follow the steps below to save trace data in Touchstone format.

1. Press **Channel Next** (or **Channel Prev**) and **Trace Next** (or **Trace prev**) to select the trace to be saved.
2. Press **Save/Recall** > **SnP** > **SnP Format**.
3. Click the softkey that corresponds to the data format you want to save.

Softkey	Function
Auto	Data format is automatically set according to the display format of the active trace. When the display format of the active trace is set to one other than log magnitude format (LogMag), linear magnitude format (LinMag), or real-imaginary number format (Real/Imag), the data format is automatically set to real-imaginary number.
LogMag/Angle	Select "log magnitude - angle" data format
LinMag/Angle	Select "linear magnitude - angle" data format
Real/Imaginary	Select "real - imaginary number" data format

5. Click the softkey for **s1p** or **s2p** according to the file type you want to save.
6. Click the softkey that corresponds to the combination of ports that you want to save.
7. SaveAs dialog box opens. For its operations, use an external keyboard and mouse.
8. Specify the folder to which the file should be saved, enter a file name, and then press **Save** to save the file.

NOTE When saving data in a sweep process, the data during sweep is saved into a Touchstone file. That is, the previous sweep data is saved as data that has not been swept; or, if sweep was not performed previously, zero data might be saved. Therefore, you should set the active channel to the HOLD state when saving data into a Touchstone file.

NOTE This function is inactive when the security level is set at low/high.

Saving the Screen Image to a File

- Overview
- Saving Screen Image as File

Other topics about Data Output

Overview

Along with printing, the E5061B allows the user to save screen images as bitmap (.bmp) or portable network graphics (.png) files. Saved files can be loaded into PC application software for further processing.

Saving Screen Image as File

Follow the procedure below to save a screen image to a file.

1. Display the screen to be saved as a file. If you want to save the screen with a white background, set the display mode to inverted display before saving the screen. For details about display mode, see Setting Display Colors.
2. Press **System** key. The screen image at the time **System** key is pressed is the image that will be saved.
3. Press **Dump Screen Image** to open the Save As dialog box.
4. Select the file type from "24-Bit Bitmap (*.bmp)" or "Portable Network Graphics (*.png)".
5. Select the destination folder and type a file name. (Clicking **Input from the front panel** on the Save As dialog box activates the on screen keyboard.)
6. Press **Save** to save the screen image of E5061B to a file.

Printing Displayed Screen

- Overview
- Printed/Saved Images
- Print Procedure

Other topics about Data Output

Overview

By connecting a printer to the USB port of the E5061B, you can print the displayed screen of the E5061B.

Printed/Saved Images

The display image saved in the volatile memory (clipboard) is printed/saved. If no image is saved in the clipboard, the image displayed at the time of print execution is printed/saved.

Saving image to clipboard

The **System** key also has a screen capture feature. When you press **System** key, the image displayed on the screen immediately before pressing is saved in the clipboard.

NOTE The image in the clipboard is cleared when you execute print/save.

Print Procedure

Preparation before printing

Follow these steps to prepare for printing:

1. Turns OFF the E5061B.
2. Turn ON the printer and connect it to E5061B.
3. Turn ON the E5061B.
4. Press **System** key.
5. Press **Printer Setup**. The Printers window opens. The icons of the printers that have been connected are displayed in the window. When you connect a print for the first time, it is automatically registered and its icon is added in the window.
6. The printer with the check mark (✔) on its icon is selected as the default printer for printing. If you want to change it, select (highlight) the icon of your preferred printer in the Printers window and then click **Set as Default Printer** in the File menu.
7. Click **Printing Preferences...** in the File menu. The Printing Preferences dialog box for the selected printer appears. Set items necessary before printing such as Page Size and then click the **OK** button .

8. Click **Close** in the File menu.

Executing print

Follow these steps to print the screen information:

1. Display the screen you want to print.
2. Press **System** key to save the currently displayed screen onto the clipboard.
3. As necessary, press **Invert Image** to toggle between [OFF] for printing in colors close to the actually displayed screen and [ON] for printing in inverse colors.
4. Click **Print** to start printing.
5. To cancel the printing in progress, press **Abort Printing**.

NOTE If you start printing when the printer is not ready (for example, it is not turned ON) by mistake, the Printers Folder dialog box may appear. In this case, click Cancel to close the Printers Folder dialog box, prepare your printer, and then start printing again.

Optimizing Measurements

Optimizing Measurements

- Expanding Dynamic Range
- Reducing Trace Noise
- Improving Phase Measurement Accuracy
- Improving Measurement Throughput

Expanding Dynamic Range

- Overview
- Lowering Receiver Noise Floor

Other topics about Optimizing Measurement

Overview

The dynamic range is the finite difference between the maximum input power level and the minimum measurement power level (noise floor) of the analyzer. In evaluating a characteristic accompanied by a large change in the amplitude (the pass band and stop band of a filter, for example), it is important to increase the dynamic range.

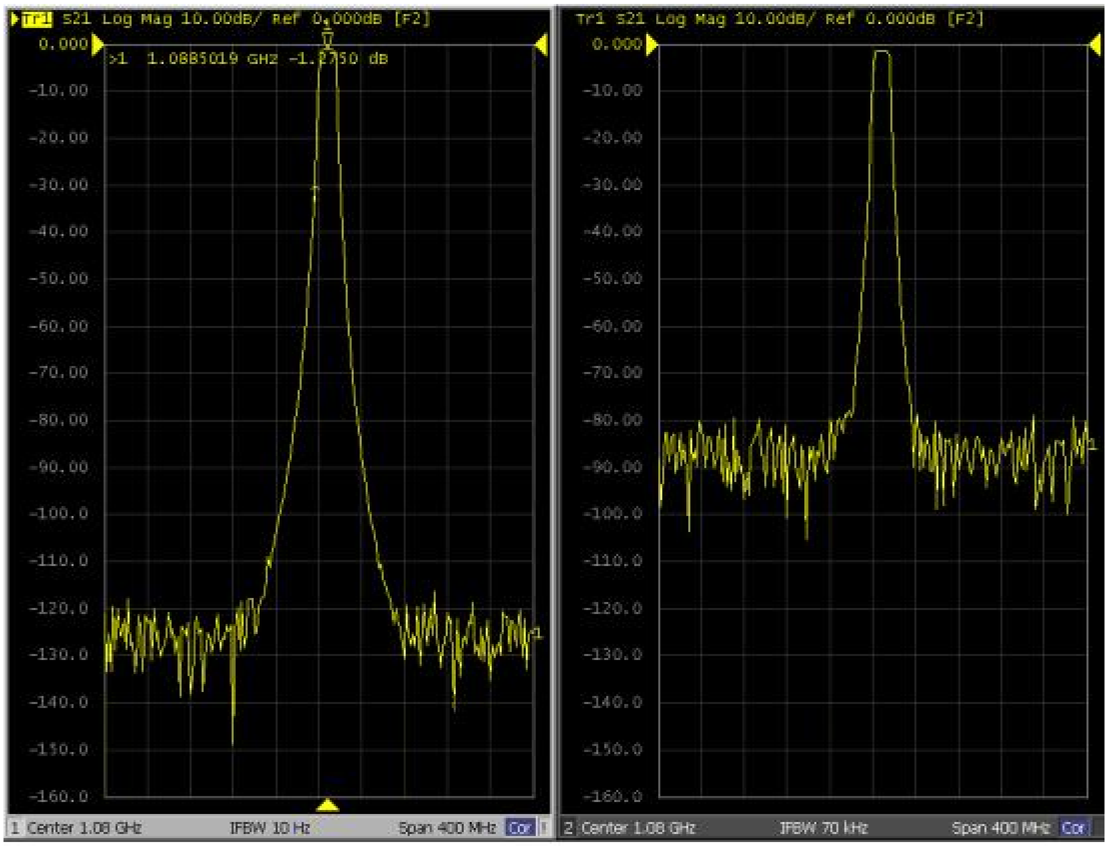
Lowering Receiver Noise Floor

Lowering the noise floor of the receiver enables you to expand the dynamic range. The following methods can be used to lower the receiver noise floor.

- Narrowing the IF bandwidth
- Turning on Sweep Averaging

Narrowing the IF bandwidth

Narrowing the receiver IF bandwidth enables you to reduce the effect of random noise on measurements. Narrowing the IF bandwidth to 1/10 the original bandwidth causes the receiver noise floor to decrease by 10 dB.



e5071c080

To specify the IF bandwidth, follow the steps described below.

1. Press **Channel Next** (or **Channel Prev**) to select a channel on which to specify the IF bandwidth.
2. Press **Avg** key.
3. Click **IF Bandwidth**, then change the IF bandwidth in the data entry area.

Setting IF Bandwidth Automatically

The IF bandwidth auto mode helps you to improve the measurement speed, when the sweep mode is set at log especially. When you make a measurement with wide span from lower frequency and the IF bandwidth is set at fixed frequency, the measurement would be slow or higher noise floor in lower frequency. This function allows you to measure narrower IF bandwidth (lower noise) in lower frequency and wider IF bandwidth (faster) in higher frequency.

1. Press **Avg** key.
2. Click **IFBW Auto** to turn ON the IF bandwidth auto mode.
3. If you want to set an upper limit of IF bandwidth in auto mode, press **IFBW Auto Limit**, and enter the upper limit with enter keys.

NOTE In IF bandwidth auto mode, the IF bandwidth is automatically set to equal to or less than 1/5 of each measurement frequency. (The E5061B does not change IFBW for each test points in IF BW auto mode. The E5061B divides its measurement span into several segments which are defined internally. The IFBW is set at equal to or less than 1/5 of the start frequency of each segment.)

NOTE IFBW Auto On may make the sweep time be longer than IFBW Auto Off. This is because IFBW auto operation requires some additional time.

Turning ON Sweep Averaging

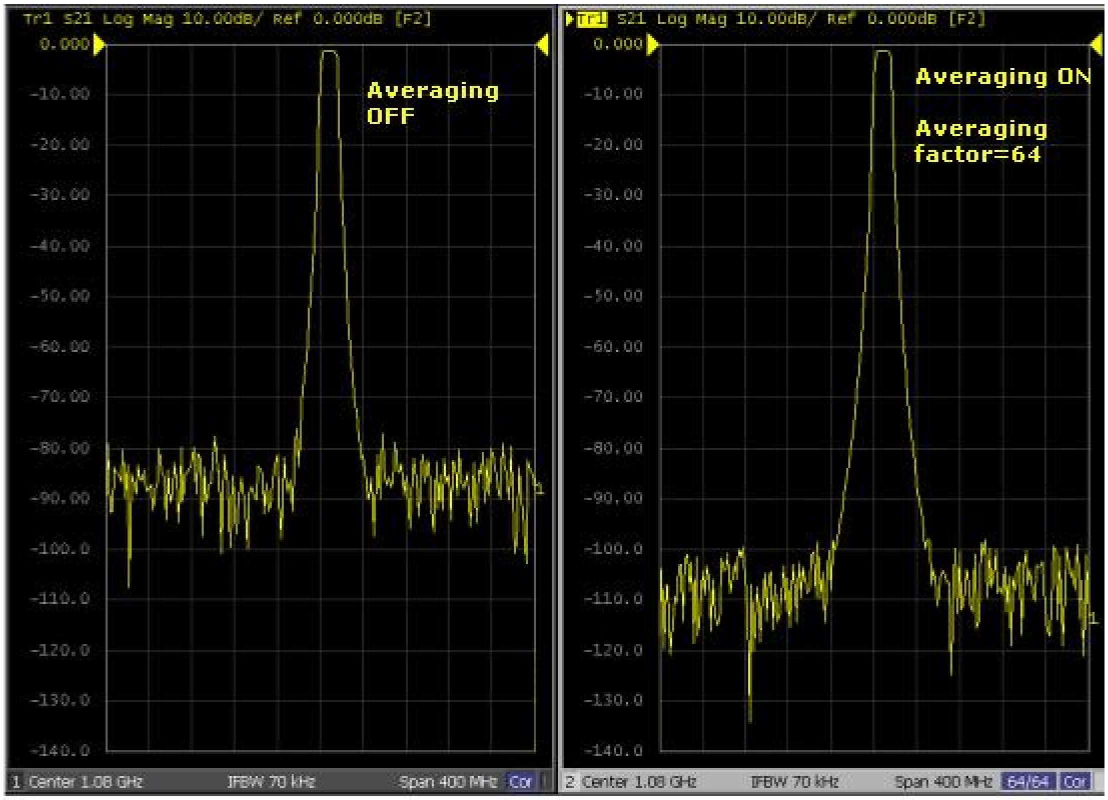
Using sweep averaging also enables you to reduce the effects of random noise on the measurements.

Sweep averaging averages data from each point (vector quantity) based on the exponential average of a continuous sweep weighted by the averaging factor specified by the user. Sweep averaging is expressed in the following equation.

$$A_n = \frac{S_n}{F} + \left(1 - \frac{1}{F}\right) \times A_{n-1}$$

where:

- A_n = Result of the calculation of sweep averaging for the nth sweep operation at the point in question (a vector quantity)
- S_n = Measurement value obtained at the nth sweep operation at the point in question (a vector quantity)
- F = Sweep averaging factor (an integer between 1 and 999)



e5071c081

Define the sweep averaging by following the steps below.

1. Press **Channel Next** (or **Channel Prev**) to select the channel on which you want to define the sweep averaging.
2. Press **Avg** > **Avg Factor**, then change the averaging factor in the data entry area.
3. Click **Averaging** to turn ON the averaging.

NOTE Clicking **Averaging Restart** resets n to 1 in Sweep Averaging equation in Turning on Sweep Averaging

Reducing Trace Noise

Any of the following methods can be used to lower the trace noise. This section provides the description of Turning on Smoothing.

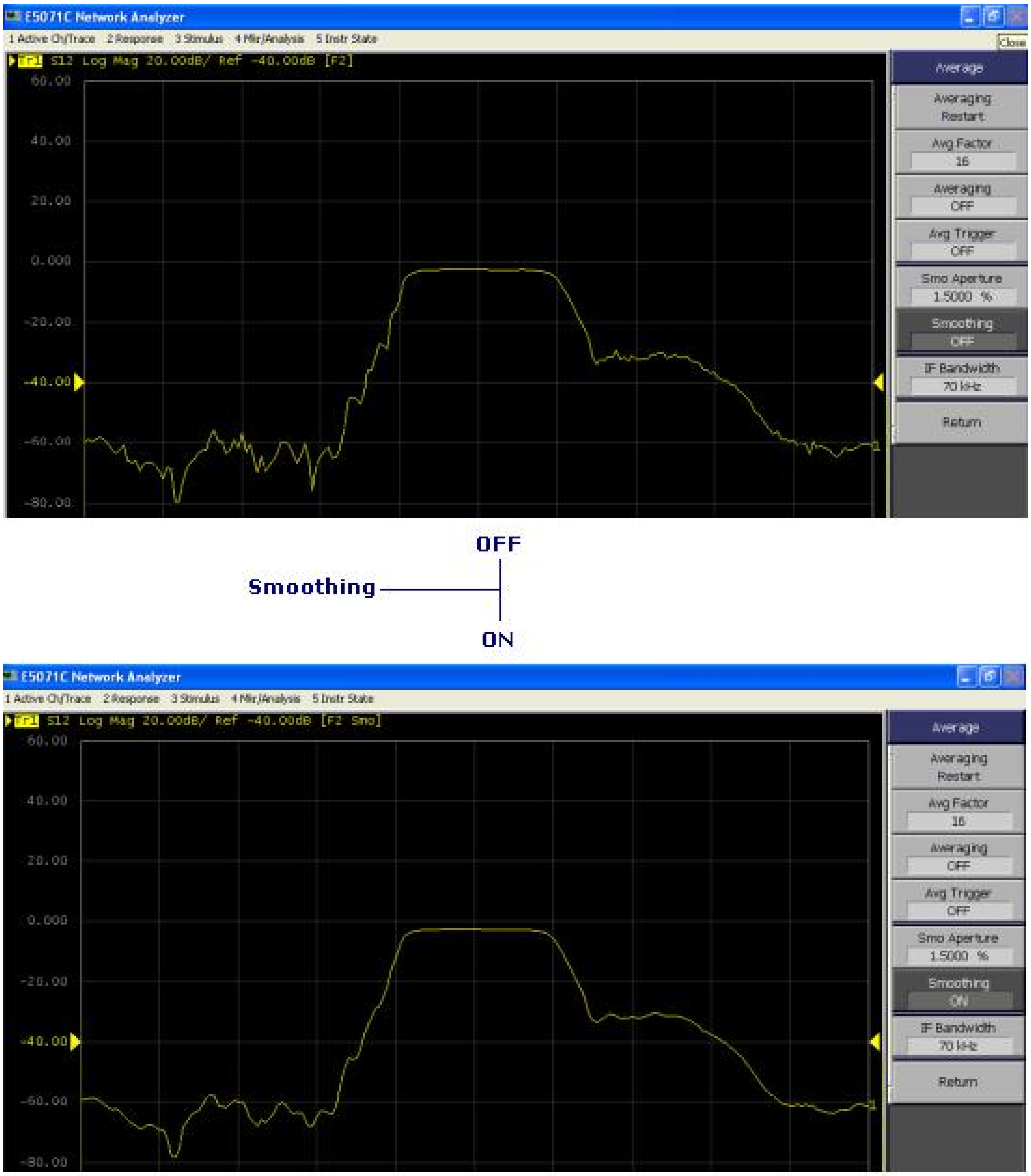
- Turning on Smoothing
- Turning on Sweep Averaging
- Narrowing IF Bandwidth

Other topics about Optimizing Measurements

Turning on Smoothing

Smoothing can be used to reduce noise that has relatively small peaks. By turning on smoothing, the value of each point on a trace is represented by the moving average over the values of several nearby points. The smoothing aperture (percentage of sweep span) defines the range of points to be included in the calculation of the moving average.

NOTE You can define the smoothing trace by trace.



e5071c075

Setting up smoothing

Set up the smoothing operation by following the steps below:

1. Press **Channel Next** (or **Channel Prev**) and **Trace Next** (or **Trace prev**) to activate the trace on which smoothing will be defined.
2. Press **Avg**.
3. Click **Smo Aperture**, then change the smoothing aperture (%) in the data entry area.
4. Click **Smoothing** to turn ON smoothing. **Smo** is displayed at the trace status area.

Improving Phase Measurement Accuracy

This section describes the following functions that can be used to improve phase measurement accuracy.

- Electrical Delay
- Velocity Factor
- Phase Offset
- Port Extensions and Loss Values

Other topics about Optimizing Measurements

Electrical Delay

Electrical Delay is a function that adds or removes a pseudo-lossless transmission line with a variable length corresponding to the receiver input. Using this function enables you to improve the resolution in phase measurement and thereby measure the deviation from the linear phase. You can specify the electrical delay trace by trace.

1. Press **Channel Next** (or **Channel Prev**) and **Trace Next** (or **Trace prev**) to activate the phase trace of which you want to specify the electrical delay.
2. Press **Scale** > **Electrical Delay**.
3. Change the electrical delay (in seconds) in the data entry area.

For how to determine the deviation from a linear phase, see Measuring the Deviation from a Linear Phase.

Procedure using marker

1. Press **Channel Next** (or **Channel Prev**) and **Trace Next** (or **Trace prev**) to activate the trace of which you want to set the electrical delay.
2. Place the active marker in an appropriate position.
3. Press **Maker Fctn**.
4. Click **Marker -> Delay** to set the electrical delay to the group delay value at the position of the active marker (a value smoothed with the aperture of 20% regardless of the smoothing setting).

Phase offset

Phase offset is a function used to add or subtract a predetermined value relative to the frequency to and from the trace. Using this function enables you to simulate the phase offset which occurs as a result of, say, adding a cable.

The phase offset can be specified from -360 ° to +360 ° .

Using the Phase Offset Function

1. Press **Channel Next** (or **Channel Prev**) and **Trace Next** (or **Trace prev**) to activate the trace of which you want to specify the phase offset.
2. Press **Scale** > **Phase Offset**, then enter the phase offset (°) in the data entry area.

Velocity factor

The velocity factor is the ratio of the propagation velocity of a signal in a coaxial cable to the propagation velocity of that signal in free space. The velocity factor for a common cable is about 0.66. The propagation velocity depends on the dielectric constant (ε_r) of the dielectric substance the cable.

$$\text{Velocity factor} = \frac{1}{\sqrt{\varepsilon_r}}$$

By specifying the velocity factor, you can match the equivalent length (in meters) appearing in the data entry area to the actual physical length when using the Electrical Delay or Setting port extensions to specify the electrical delay (in seconds).

You can define the velocity factor channel by channel.

Using the velocity factor

1. Press **Channel Next** (or **Channel Prev**) to activate the channel for which you want to specify the velocity factor.
2. Press **Cal** > **Velocity Factor**, then the velocity factor in the data entry area.

Port Extensions and Loss Values

Overview

Port Extension is a function for moving the calibration reference plane by specifying the electrical delay. This function is useful, for example, when you cannot directly perform calibration at the DUT terminal because the DUT is inside the test fixture. In such a case, this function enables you to first perform calibration at the test fixture terminal and then move the calibration plane to the DUT terminal by extending the port.

Port extension corrects the electrical delay of each test port (phase shift) only. It cannot remove errors caused by the loss in and incorrect matching of cables, adapters, or test fixtures.

In addition to port extension, you can set loss values for each port. By correcting loss due to port extension, more accurate measurement results are obtained.

There are two types of loss value settings: loss values at two frequency points for a specified port, and a DC loss value. You can make these settings at the same time for each port.

The port extension is not available for the Gain-Phase measurement.

NOTE You can set loss values channel by channel. Setting loss values for one particular channel does not affect other channels.

NOTE You can define port extension channel by channel. Setting port extension for one particular channel does not affect other channels.

Enabling port extensions and loss values

1. Press **Channel Next** (or **Channel Prev**) to activate the channel for which you want to enable port extension.
2. Press **Cal** > **Port Extensions.**
3. Turn ON **Extensions**.

Setting port extension and Loss values manually.

a. Setting electrical delay

Follow the steps below to set the delay for coaxial cable:

1. Press **Channel Next** (or **Channel Prev**) to activate the channel for which you want to set port extension.
2. Press **Cal** > **Port Extensions.**
3. Click **Extension Port 1** or **Extension Port 2** to select the port.
4. Click **Extension** to set the extension in sec.

b. Setting loss values

1. Press **Channel Next** (or **Channel Prev**) to activate the channel for which you want to set loss values.
2. Press **Cal** > **Port Extensions.**
3. Click **Extension Port 1** or **Extension Port 2** to select the port.
4. Click **Loss** to set a loss value.
5. Click **Loss1 [OFF]** to toggle to **Loss1 [ON]** (enabled), and enter a loss value (**Loss1**) and a frequency (**Freq1**).
6. If you want to set loss at two frequency points, press **Loss2 [OFF]** to toggle to **Loss [ON]** (enabled), and enter a loss value (**Loss2**) and a frequency (**Freq2**).

- Expression to calculate loss using Loss 1:

$$\mathrm{Loss(f)} = \mathrm{Loss1} \times \sqrt{\left(\frac{\mathrm{f}}{\mathrm{Freq1}}\right)}$$

- Expression to calculate loss using Loss 1 and Loss 2:

$$\mathrm{Loss(f)} = \mathrm{Loss1} \times \left(\frac{\mathrm{f}}{\mathrm{Freq1}}\right)^{n}$$

$$n = \frac{\log_{10}\left|\frac{\mathrm{Loss1}}{\mathrm{Loss2}}\right|}{\log_{10}\frac{\mathrm{Freq1}}{\mathrm{Freq2}}}$$

NOTE When you specify two frequency points, set the lower frequency to Loss1, and the higher one to Loss2.

c. Setting a DC loss value

1. Click **Cal** > **Port Extensions** > **Extension Port 1** or **Extension Port 2** to select the port.
2. Click **Loss** > **Loss1 [OFF]** to toggle to **Loss1 [ON]** (enabled).
3. Click **Loss at DC**, then enter a DC loss value.

Using the auto port extension function

The auto port extension function measures port extension and loss values for each port using the OPEN/SHORT standard connected to the port, automatically calculates and sets them.

When the auto port extension function is completed, the port extensions and loss values are updated to the calculated values.

You can use both open and short measurement values in the auto port extension function. Note that in this case, the average value of the calculation results is used for updating.

You can set the auto port extension function channel by channel. Setting the auto port extension function for one particular channel does not affect other channels.

When the sweep type is power sweep, the auto port extension is not available.

A. Selecting a port(s)

Select the port(s) of which you want to use the auto port extension function.

1. Press **Channel Next** (or **Channel Prev**) to activate the channel of which you want to set auto port extension.
2. Press **Cal** > **Port Extensions** > **Auto Port Extension** > **Select Ports** to select the port(s) of which you want to use the auto port extension function.

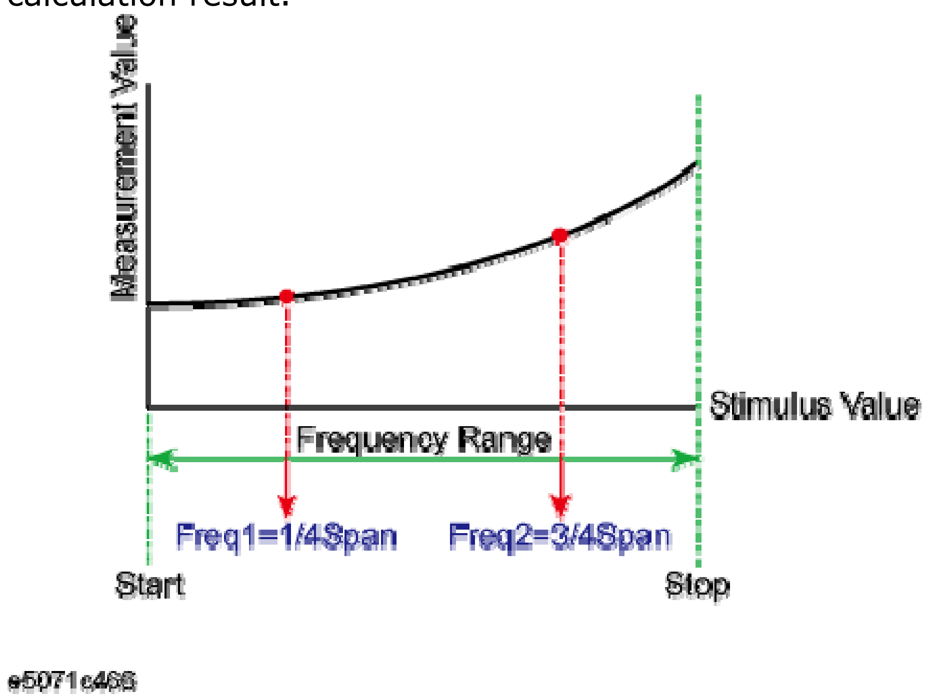
B. Setting frequencies used for calculation

Set the frequency points ofh which you want to calculate a loss value.

1. Press **Cal** > **Port Extensions** > **Auto Port Extension** > **Select Ports** > **Method** to set **the frequencies used for calculation**.
2. If you have selected **User Span**, use **User Span Start** and **User Span Stop** to set a start value and a stop value.

NOTE For **Current Span** and **User Span**, a frequency point at 1/4 of the frequency range is set to **Freq1**; a frequency point at 3/4 of the frequency range is set to **Freq2**.

NOTE If the setting is not made before starting OPEN/SHORT standard measurement, it does not affect the calculation result.



C. Specifying a loss value as a calculation target

Specify whether you want to include a loss value in the calculation result.

1. Press **Cal** > **Port Extensions** > **Auto Port Extension** > **Select Ports** > **Include Loss** to turn it ON.

   NOTE If the setting is not made before starting the measurement of the OPEN/SHORT standard, it does not affect the calculation result.

D. Specifying a DC loss value as a calculation target

Specify whether you want to include a DC loss value in the calculation result.

1. Press **Cal** > **Port Extensions** > **Auto Port Extension** > **Adjust Mismatch** to turn it ON.

   NOTE If the setting is not made before starting the measurement of the OPEN/SHORT standard, it does not affect the calculation result.

E. Measuring the OPEN/SHORT standard and executing calculation

Calculate port extensions and loss values based on the calculation results using the OPEN/SHORT standard.

1. Press **Cal** > **Port Extensions** > **Auto Port Extension**.
2. If you use the OPEN standard, click **Measure OPEN**, and select the port(s) of which you want to execute measurement. Execution is restricted to ports selected in Selecting a port(s)
3. If you use the SHORT standard, click **Measure SHORT**, and select the port(s) of which you want to execute measurement. Execution is restricted to ports selected in Selecting a port(s).

   NOTE If a port extension value or loss value has been set, the value is updated to the calculated result.

   NOTE If you execute both open measurement and short measurement, the average of the calculation results is reflected to the port extension and loss value.

F. Deleting the result of open/short measurement

When you exit from the softkey menu in the same level after open/short measurement, the measurement results are deleted. Note that you can use a GPIB command.

NOTE Port extension and loss values that have been calculated are not cleared.

Improving Measurement Throughput

This section explains the methods to improve measurement throughput.

- Turning OFF the updating of information displayed on the LCD screen

Other topics about Optimizing Measurements

Turning OFF the updating of information displayed on the LCD screen

Turning OFF the updating information feature displayed on the LCD screen eliminates the processing time required to update displays within the analyzer, hence improves the measurement throughput. If it is not necessary to check displayed information during measurements, turning OFF real-time updating is an effective means of improving throughput.

The updating of information displayed on the LCD screen can be switched using the following procedure:

Turning OFF the updating of information

1. Press **Display**.
2. Click **Update** to switch the updating of displayed information on the LCD screen ON/OFF.

When the LCD screen update is turned OFF, **Update Off** appears on Instrument Status Bar.

Measurement Examples

Measurement Examples

- Band Pass Filter Measurement (in Quick Start)
- Loop Gain Stability Measurement
- Performing Segment-by-Segment Sweep (segment sweep)

Measurement Example of a Bandpass Filter

This section describes how to measure the transmission characteristics of a 947.5 MHz bandpass filter. The measurement conditions for this measurement example are those suitable for a 947.5 MHz bandpass filter. To measure another device under test (DUT), change the measurement conditions to suit the particular DUT.

STEP 1. Determining Measurement Conditions

1. Preset the E5061B.

   Preset > **OK**

2. Set the S-parameter to S21.

   Meas > **S21**

   NOTE When measuring the reverse transmission characteristics, set the S-parameter to S12.

3. Set the data format to the log magnitude format

   Format > **Log Mag**

4. Set the center frequency to the bandpass filter center frequency. Next, specify the span frequency, which is set to 200 MHz in this measurement example.

   Center > **9** > **4** > **7** > **.** > **5** > **M/m**
   Span > **2** > **0** > **0** > **M/m**

   NOTE When entering the frequency unit using the keyboard, type "G" for GHz, "M" for MHz, and "k" for kHz.

5. Specify the number of measurement points per sweep. The number of measurement points in this measurement example is set to 401.

   Sweep Setup > **Points** > **4** > **0** > **1** > **x1**

6. Specify the power level of the signal source. The power level in this measurement example is set to -10 dBm.

   Sweep Setup > **Power** > **+/-** > **1** > **0** > **x1**

7. Specify the IF bandwidth of the receiver as necessary. In this measurement example, the IF bandwidth is set to 10 kHz because of the need to lower the noise floor.

   Avg > **IF Bandwidth** > **1** > **0** > **k/m**

STEP 2. Calibration

To turn the error correction ON, set the calibration type to the full 2-port calibration and measure the calibration data.

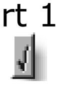
For details about calibration, see Calibration.

1. Select the calibration kit suitable for the measurement cable. In this measurement example, Calibration Kit 85032F is selected.

   Cal > **Cal Kit** > **85032F**

2. Set the calibration type to the full 2-port calibration using the test port 1 and 2.

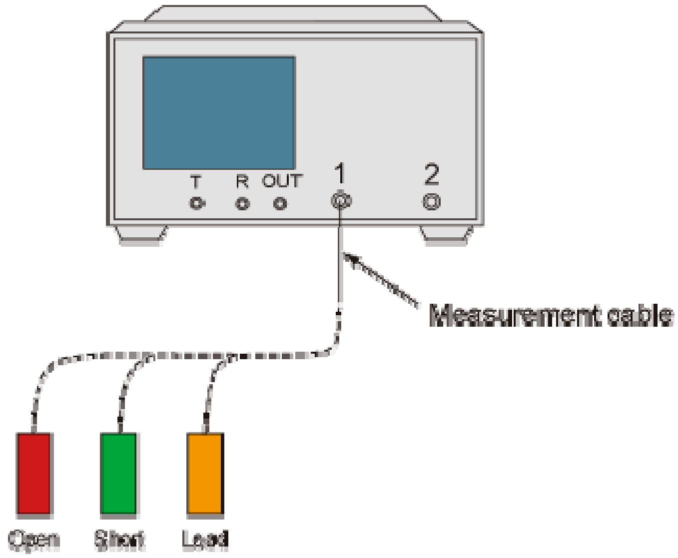
   Cal > **Calibrate** > **2-Port Cal**

3. Connect the OPEN standard (included in the calibration kit) to the other end of the measurement cable that is connected to the test port 1 as shown in the following figure, and measure the open calibration data at the test port 1. After measuring the open calibration data, a checkmark  is displayed to the left of the **Port 1 Open** menu.

   Cal > **Calibrate** > **2-Port Cal** > **Reflection** > **Port1 Open**

   In the same way, measure the calibration data for the SHORT/LOAD standards at the test port 1.

   Connecting the OPEN/SHORT/LOAD standards

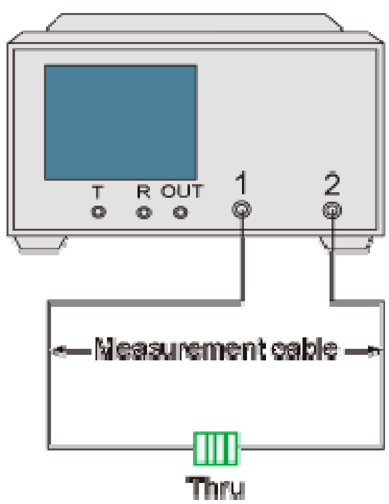


e5061b042

4. In the same way as described above, measure the calibration data for the OPEN/SHORT/LOAD standards at the test port 2.
5. Connect the measurement cables as shown in the figure below, and measure the thru calibration. After measuring the thru calibration data, a checkmark √ is displayed to the left of the **Port 1-2 Thru** button.

Cal > **Calibrate** > **2-Port Cal** > **Transmission** > **Port 1-2 Thru**

Making the through calibration



e5061b044

6. Set the full 2-port calibration measurement to DONE. The calibration factor is calculated based on the calibration data acquired, and the error correction is turned ON.

Cal > **Calibrate** > **2-Port Cal** > **Done**

7. Select the type in which the data is to be saved before saving the calibration factor (calculated based on the calibration data).

Save/Recall > **Save Type** > **State & Cal**

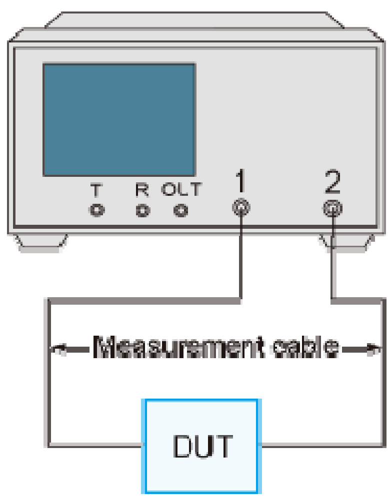
8. Store the calibration file to the disk of the E5061B. The symbol "X" appearing in the operations below represent the assigned numbers to be used when the file is saved.

Save/Recall > **Save State** > **State 0X**

STEP 3. Connecting the Device Under Test (DUT)

1. Connect to the DUT to the E5061B. (See the below figure)

Connecting the DUT



e5061b044

2. Set the appropriate scale by executing the auto scale. (See the below figure)

   Scale > **Auto Scale**

   You can also adjust the scale by entering arbitrary values in the **Scale/Div** button, **Reference Position** button, and **Reference Value**.

S21 trace after executing the auto scale

STEP 4. Analyzing Measurement Results

This section describes how to use the marker function to read out important parameters for the transmission measurement of the bandpass filter (insertion loss, -3 dB bandwidth).

Measuring the Insertion Loss

1. Display a marker.

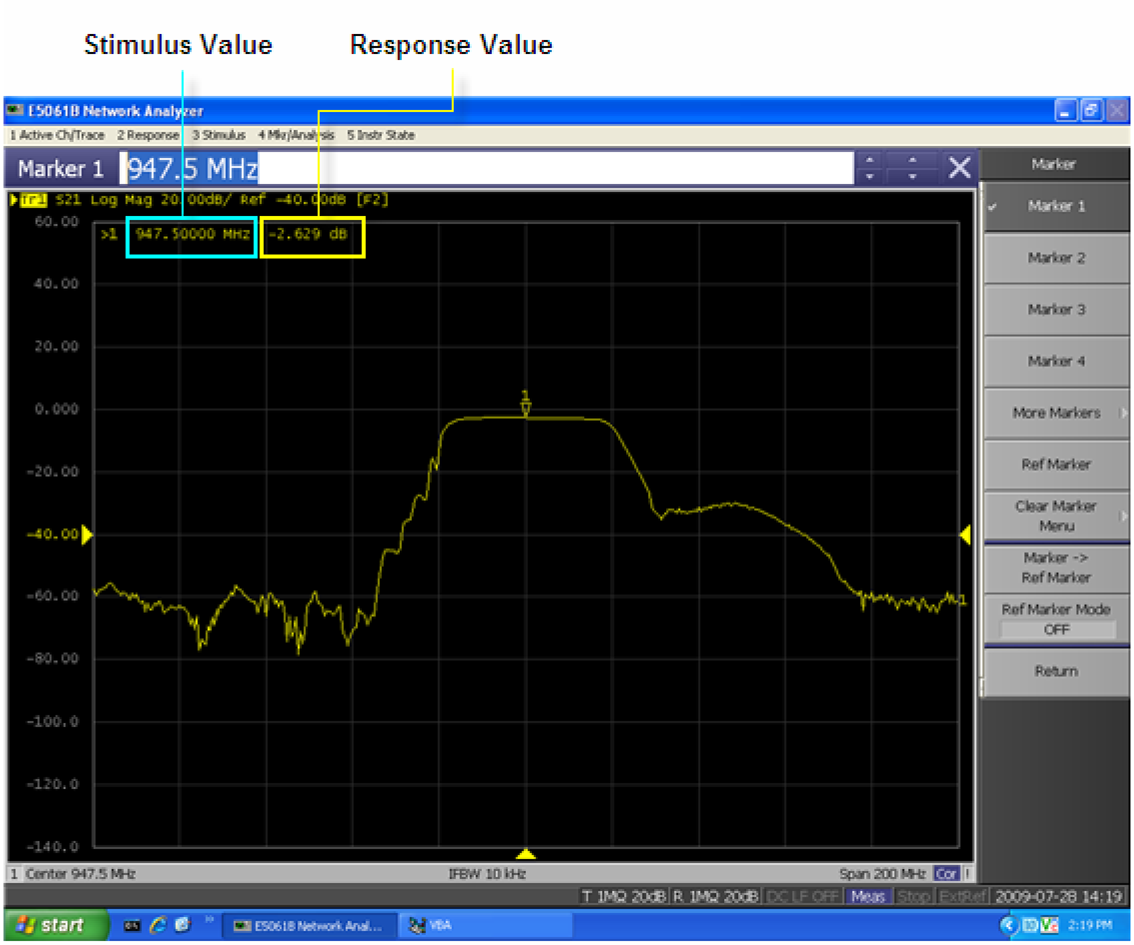
   Marker > **Marker 1**

2. Using one of the following methods to move the marker to the center frequency of the bandpass filter.

- On the entry bar, press **9**> **4** > **7** > **.** > **5** > **M/m**
- Turn the rotary knob on the front panel to set it to the center frequency (947.5 MHz).

3. Read the marker value displayed as shown in the figure below. In this example, the response value denotes the insertion loss.

Measuring an Insertion Loss



Measuring the -3 dB Bandwidth

Using the marker bandwidth search function, the bandwidth, center frequency between two cutoff frequency points, Q value, and insertion loss are all read out. These parameters are described in the following table.

NOTE If the two cutoff frequency points are not found, all data items except the insertion loss revert to zero.

Parameter	Description
BW (Bandwidth)	Stimulus width between two cutoff frequency points (low and high)
cent (Center Frequency)	Center point between cutoff frequency points (low and high)
low (Left-side Cutoff Frequency)	The lower frequency of the two cutoff frequency points
high (Right-side Cutoff	The higher frequency of the two cutoff

Frequency)	frequency points
Q (Q Value)	Q = cent/BW
loss (Insertion Loss)	The measured value of the active marker.

1. Display a marker.

Marker > **Marker 1**

2. Using one of the following methods to move the marker to the center frequency of the bandpass filter.

- On the entry bar, press **9** > **4** > **7** > **.** > **5** > **M/m**
- Turn the rotary knob on the front panel to set it to the center frequency (947.5 MHz).

3. Specify the bandwidth definition value that defines the pass band of the filter. In this measurement example, it is set to -3 dB.

Marker Search > **Bandwidth Value** > **+/-** > **3** > **x 1**

4. Set the bandwidth search function ON.

Marker Search > **Bandwidth**

5. The bandwidth data items (BW, cent, low, high, Q, loss) is displayed. (See the following figure.)

Measuring the -3 dB Bandwidth

STEP 5. Outputting Measurement Results (Save)

You can save not only the internal data but also the measurement results such as trace data and display screens to the disk.

Saving the Trace Data(in CSV format)

You can save the trace data to the disk of the E5061B in CSV file format (extension: .csv). Since the CSV-formatted data to be saved is a text file, you can analyze the data using Microsoft Excel.

Follow the step below to save the trace data:

Save/Recall > **Save Trace Data**

Saving the Display Screen

You can save the screen displayed on the E5061B to the disk of the E5061B in Windows bitmap file format (extension: .bmp) or Portable Network Graphics format (extension: .png).

Follow the step below to save the display screen:

System > **Dump Screen Image**

NOTE The image on the LCD display memorized in the volatile memory (clipboard) (the image on the LCD display when the **Capture/System** key is pressed) is saved.

Loop Gain Stability Measurement

- Overview
- Procedure

Other Measurement Examples

Overview

This section introduces an example of how to measurement loop stability for low-frequency amplifier.

Procedure

In this example, the DUT is evaluated by following the steps.

Step	Description
1. Setting the Measurement Conditions	Set the measurement conditions.
2. Executing Calibration	Execute calibration.
3. Connecting the DUT	Connect the DUT.
4. Scaling	Adjust the scale to confirm the result of measurement.

1. Setting the Measurement Conditions

The measurement conditions are defined by following the steps described below.

Setup Description	Key Operation
Presetting	**Preset** > **OK**
Setting trace number: 2	**Display** > **Num of Traces** > **2** > **Allocate Traces** > **x2**
Specifies measurement port: Gain-Phase	**Meas** > **Measurement Port** > **Gain-Phase** NOTE The measurement parameters are not specified, because the measurement parameters of both channel are T/R.

Specifies impedance of receiver port: 1 MΩ	**Meas** > **Gain-Phase Setup** > **T Input Z** > **1 MΩ** **Meas** > **Gain-Phase Setup** > **R Input Z** > **1 MΩ**
Specifies attenuator of receiver port: 20 dB	**Meas** > **Gain-Phase Setup** > **T Attenuator** > **20 dB** **Meas** > **Gain-Phase Setup** > **R Attenuator** > **20 dB**
Data format: Tr1:LogMag Tr2:Phase	(Tr1) **Format >** **Log Mag** (Tr2) **Format** > **Phase**
Sweep type: Log Frequency	**Sweep Setup** > **Sweep Type** > **Log Freq**
Start frequency: 100 Hz	**Start** > **1** > **0** > **0** > **x1**
Stop frequency: 1 MHz	**Stop** > **1** > **M/u**
IF bandwidth auto: ON	**Avg** > **IFBW Auto** (Turn it **ON**.)
IF bandwidth auto limit: 100 Hz	**Avg** > **IFBW Auto Limit** > **1** > **0** > **0** > **x1**
Power: -20 dBm	**Sweep Setup** > **Power** > **Power** > **+/-** > **2** > **0** > **x1**

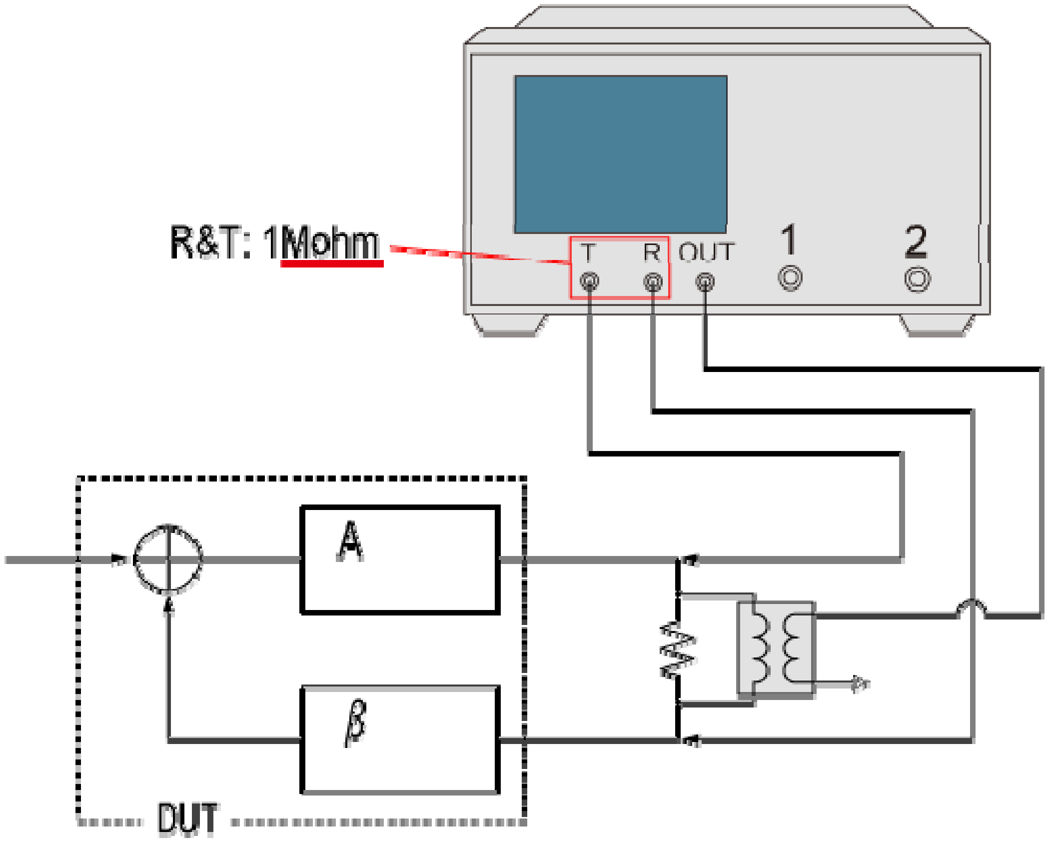
2. Executing Calibration

The THRU response calibration (Gain-Phase) is executed.

Setup Description	Key Operation or Connecting Operation
Connect THRU connection	Connect THRU connection.
Executing the THRU response	**Cal** > **Calibrate** > **Response (Thru)** > **Thru** > **Done**

3. Connecting the DUT

Connect the DUT as shown below.



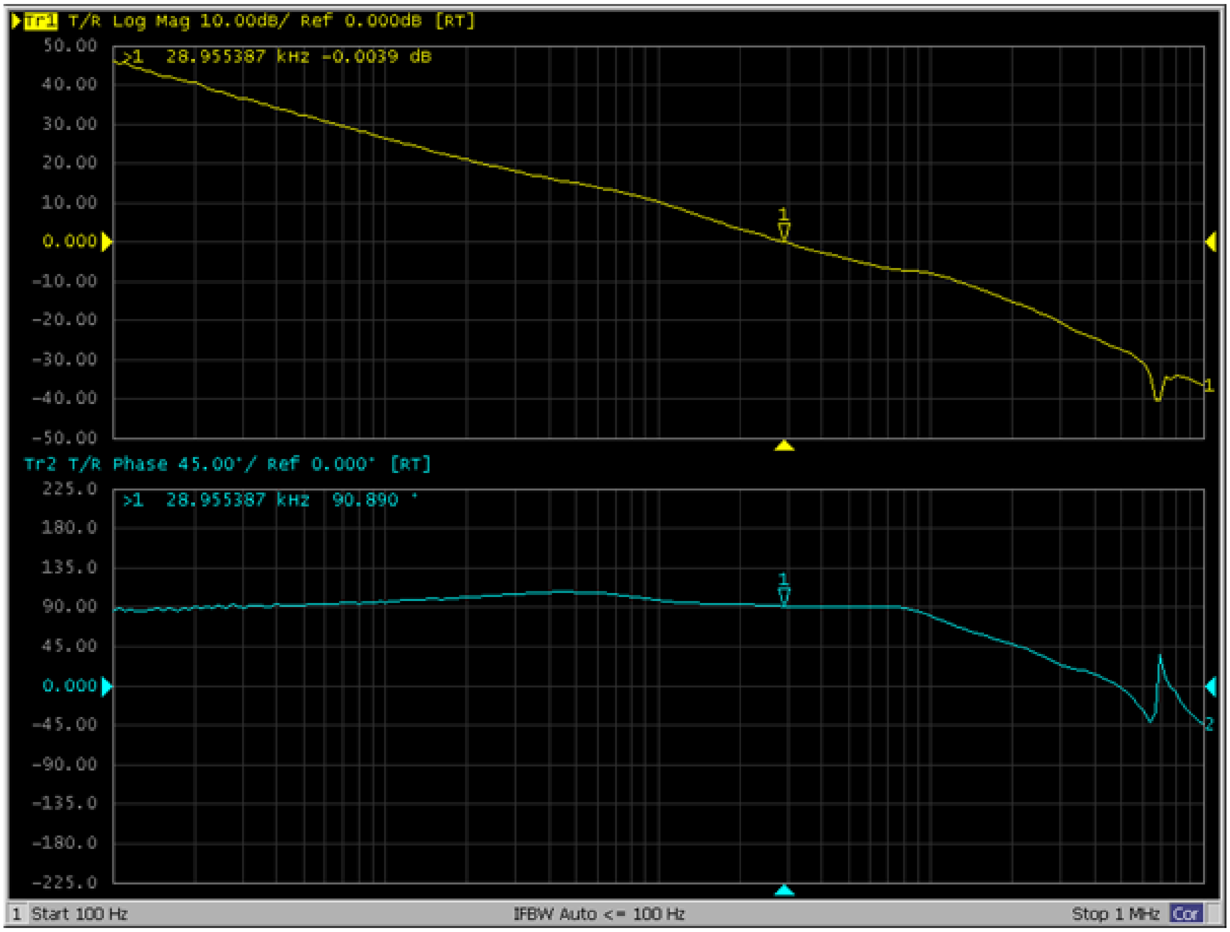
e5061b066

4. Scaling

Set each parameter for the scale setting. It is necessary to specify the trace pressing the **Trace Next** or **Trace Prev** before each parameter is specified. The following table is example of setting.

Setup Description	Key Operation
Scale divisions: 10	(Tr1) **Scale** > **Divisions** > **1** > **0** > **x1** (Tr2) **Scale** > **Divisions** > **1** > **0** > **x1**
Scale/Div: 10 dB/div (Tr1), 45 °/div (Tr2)	(Tr1) **Scale** > **Scale/Div** > **1** > **0** > **x1** (Tr2) **Scale** > **Scale/Div** > **4** > **5** > **x1**
Reference position: 5 Div	(Tr1) **Scale** > **Reference Position** > **5** > **x1** (Tr2) **Scale** > **Reference Position** > **5** > **x1**
Reference value: 0 dB (Tr1), 0 °/div (Tr2)	(Tr1) **Scale** > **Reference Value** > **0** > **x1** (Tr2) **Scale** > **Reference Value** > **0** > **x1**

Example of the loop gain stability measurement screen



Performing a Segment-by-Segment Sweep

This section describes the concept of the segment sweep and how to perform it.

- Concept of Segment Sweep
- Conditions for Setting Segment Sweep
- Items that can be set for Each Segment
- Sweep Delay/Sweep Time in Segment Sweep
- Frequency/Order Base Display

Other topics about Optimizing Measurements

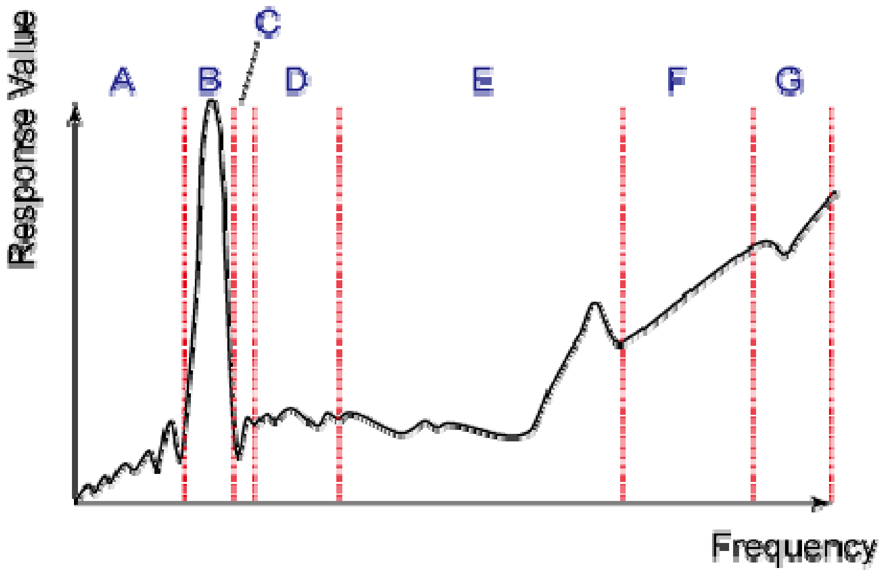
Concept of Segment Sweep

To perform a segment sweep, you must define two or more frequency ranges, called segments, and then specify the number of points, IF bandwidth, power level, sweep delay time, and sweep time for each segment. All segments are swept sequentially as if swept in one sweep operation.

NOTE By skipping the frequency range, which does not need to be measured, you can sweep and measure only the portions you need.

NOTE You can define the optimum measurement conditions for each of the segments you designate. For example, you can specify as many points as possible in a segment requiring high trace resolution and as few points as possible in a segment not requiring high resolution. This shortens the measurement time, enabling you to optimize the overall measurement throughput by not having to perform the entire operation under the same measurement conditions of a particular frequency range.

To evaluate a band pass filter that has the transmission characteristics shown in the following figure, for example, you can select the frequency ranges you need from A through G and determine the measurement conditions shown in the table below. This enables you to measure them simultaneously in one sweep operation.



e5071c393

Frequency ranges (segments) from the figure above and their measurement conditions				
	Start frequency	Stop frequency	Number of points	IF Bandwidth
A	440 MHz	915 MHz	50	50 kHz
B	915 MHz	980 MHz	130	70 kHz
C	980 MHz	1.035 GHz	60	50 kHz
E	1.07 GHz	2 GHz	100	70 kHz
G	2.6 GHz	3 GHz	40	70 kHz

Conditions for Setting Segment Sweep

The following conditions apply in setting up a segment sweep.

- The frequency range of a segment must not overlap with that of another segment. (The start frequency of a segment must be higher than the stop frequency of the immediately preceding segment).
- The start frequency of segment 1 must be greater than 5 Hz and the stop frequency of the last segment less than 3 GHz, as per the range of the Network Analyzer.

- When the start frequency and stop frequency of a segment are not the same, you can define from 2 to the maximum points in a segment.
- When the start frequency and stop frequency of a segment are the same, you can define from 1 to the maximum points in a segment.
- You can set the total number of points in the segment table from 2 to the maximum points.
- You can set the number of segments in the segment table to between 1 and 201.

Items that can be set for Each Segment

For the segment sweep, you can set the sweep range, the number of points, IF bandwidth, power level, sweep delay time, and sweep time for each segment.

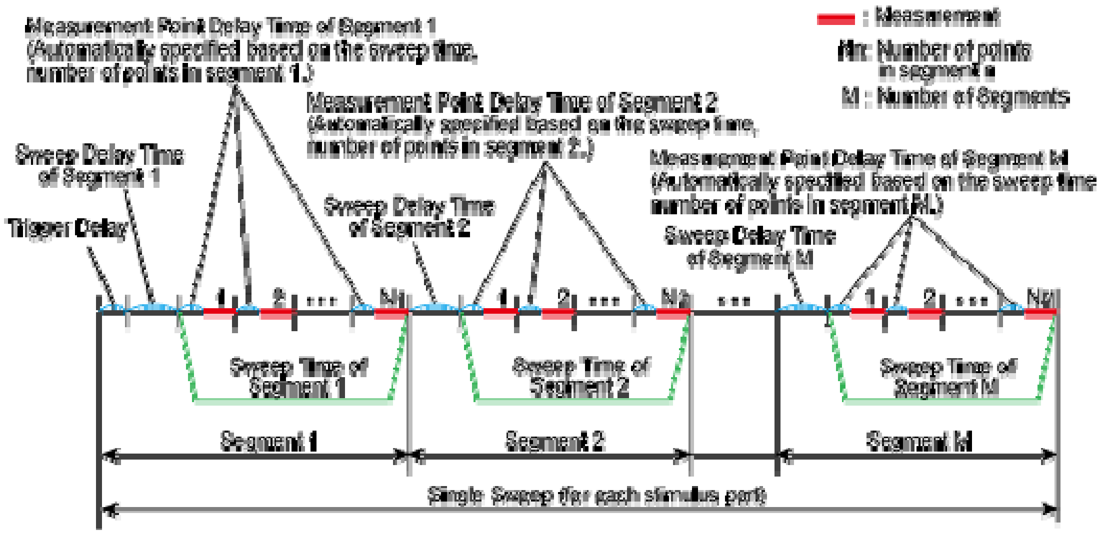
You can set the items in the following table to ON/OFF for each segment. If you enable the segment-by-segment setting, you can make the setting for each segment in the segment table; if you disable it, the setting in the following table is used.

Item	When segment-by-segment setting is disabled
IF bandwidth	For all segments, the IF bandwidth (set with **Avg** > **IF Bandwidth**) is set even if IFBW AUTO is turned ON.
Power level	For all segments, the power level for the linear/log sweep (set with **Sweep Setup** > **Power**) is set.
Sweep delay time	For all segments, 0 is set.
Sweep time	For all segments, the auto sweep time mode is set.

Sweep Delay/Sweep Time in Segment Sweep

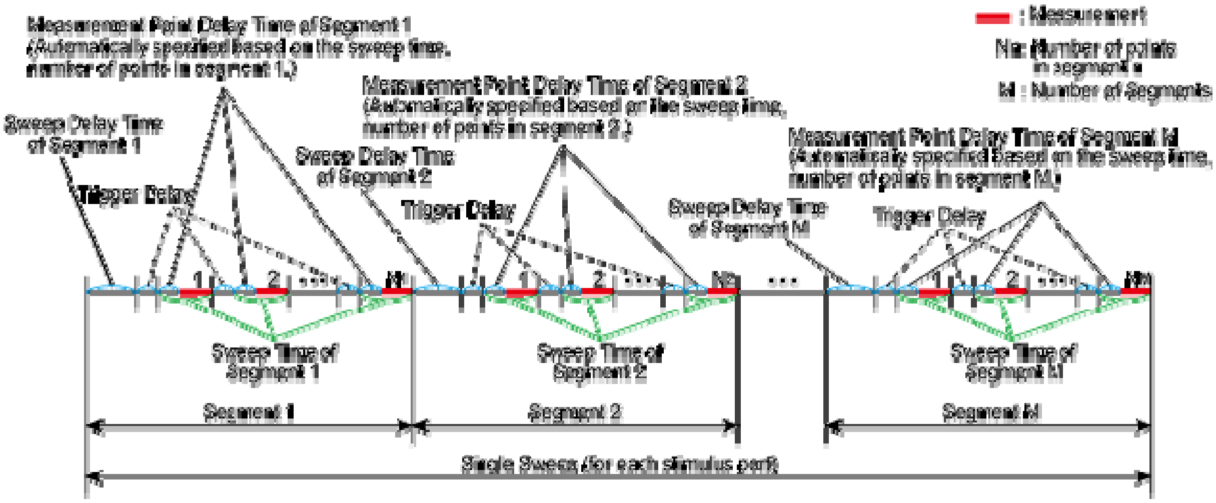
The definitions for sweep delay time and sweep time, which you can specify in the segment sweep, are shown in below figures.

The definitions for sweep delay time and sweep time when trigger mode is "On Sweep".



e5061b038

The definitions for sweep delay time and sweep time when trigger mode is "On Point".

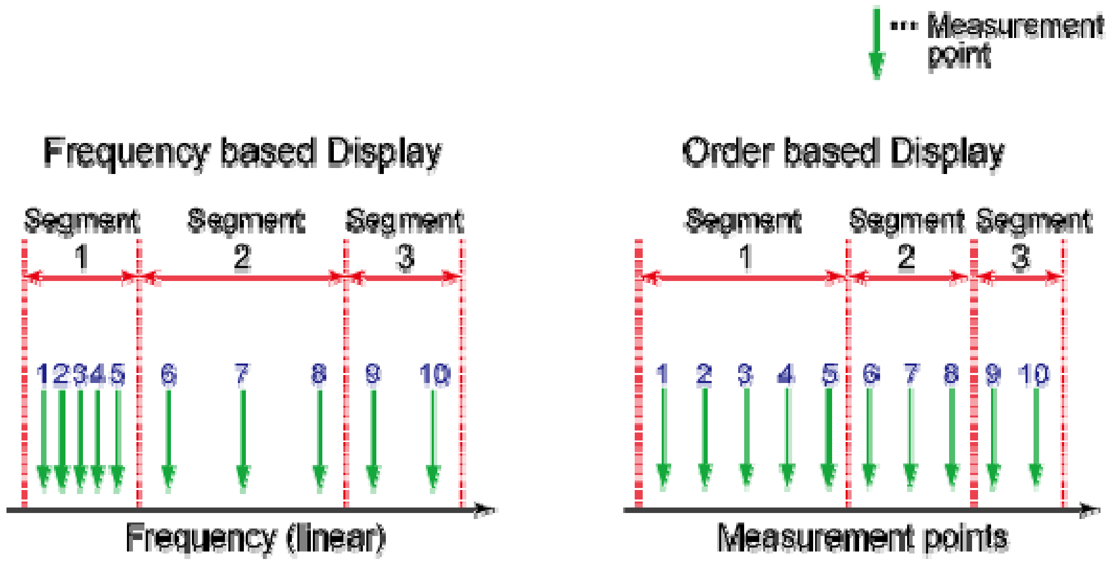


e5061b039

NOTE The segment trigger delay is not included in the segment sweep time.

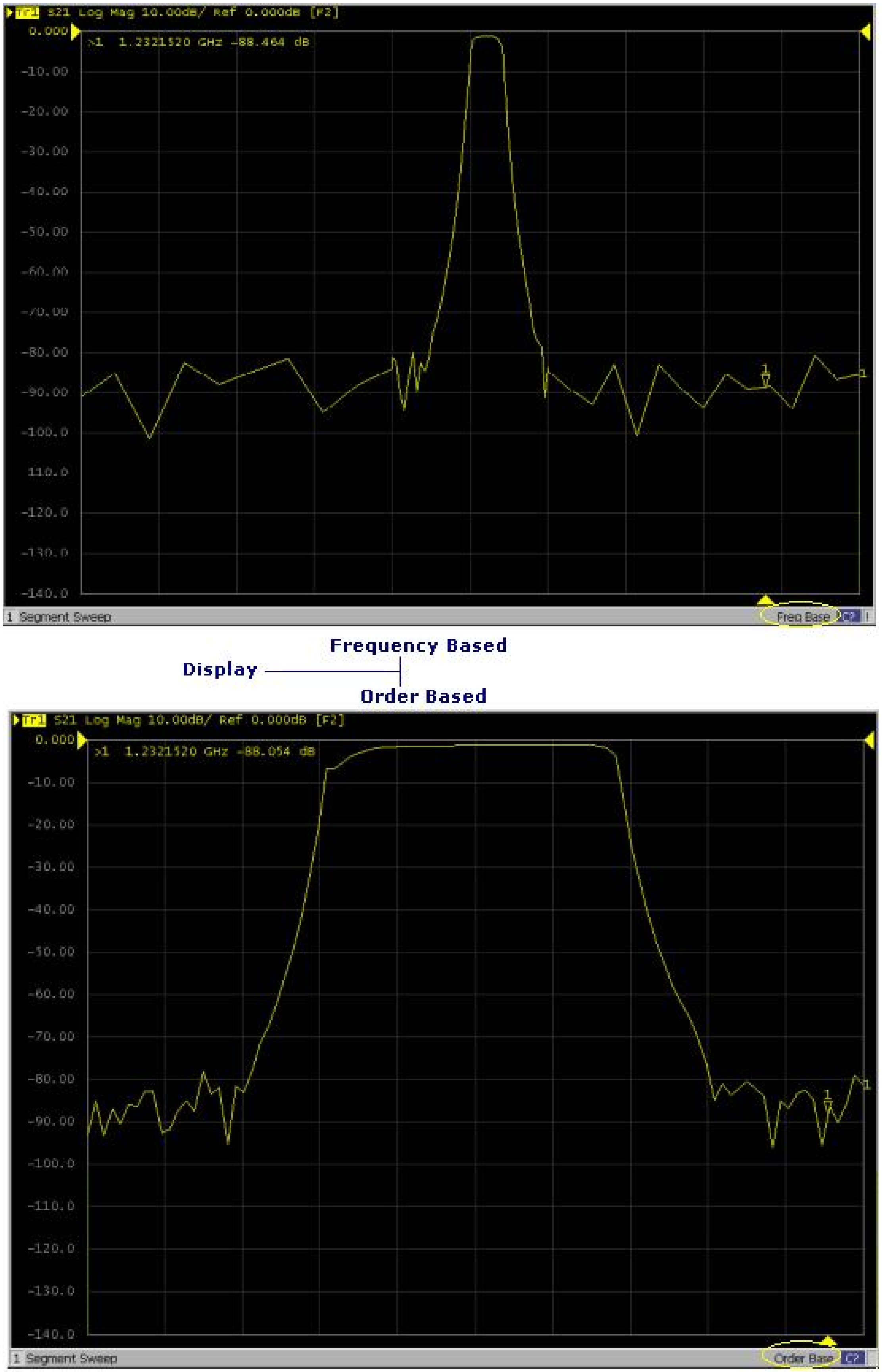
Frequency/Order Base Display

You can choose between frequency-based and order-based display as the method of displaying traces when executing a segment sweep.



Measurement point
Frequency based Display
Segment 1
Segment 2
Segment 3
1 2 3 4 5 6 7 8 9 10
Frequency (linear)
Order based Display
Segment 1
Segment 2
Segment 3
1 2 3 4 5 6 7 8 9 10
Measurement points

e5071c394


Tr1 S21 Log Mag 10.00dB/ Ref 0.000dB [F2]
>1 1.2321520 GHz -88.464 dB
1 Segment Sweep
Freq Base
Frequency Based
Display
Order Based
Tr1 S21 Log Mag 10.00dB/ Ref 0.000dB [F2]
>1 1.2321520 GHz -88.054 dB
1 Segment Sweep
Order Base

e5071c082

Procedure
Creating a segment table

1. Press **Channel Next** (or **Channel Prev**) to select the channel of which you want to create the segment table.
2. Press **Sweep Setup**.
3. Click **Edit Segment Table**. The segment table appears in the lower section of the screen.
4. Select the **softkey** to change the frequency range setting mode or to set the IF bandwidth, power level, sweep delay time, and sweep time for each segment.

   NOTE When setting the segment table using the front panel keys or keyboard, you need to place focus on (select) the operation target (segment table of softkey) first. You can change the focus by pressing **Foc** key in the **ENTRY** Block. When the focus is placed on the segment table, the window frame of the segment table is displayed as bright as the window frame of the active channel. When the focus is placed on the softkey menu, the softkey menu title area is displayed in blue.

5. Enter each item in the following table for each added segment (line) to create the segment table. To create the segment table, use the hardkeys and softkeys.

Start	Sets the start value of the sweep range
Stop	Sets the stop value of the sweep range
Center	Sets the center value of the sweep range
Span	Sets the span value of the sweep range
Points	Sets the number of points
IFBW	Sets the IF bandwidth
Power	Sets the power level; the power range is common to the settings for the linear/log sweep (**Sweep Setup** > **Power Ranges**)
Delay	Sets the sweep delay time
Time	Sets the sweep time; to specify the auto setting (**AUTO**), enter 0 as the sweep time

Useful functions when using a mouse

- You can **copy / paste / insert / delete** the cell by right-clicking on the selected cell.
- In the character-by-character edit mode, you can **undo / cut / copy / paste / delete / select all** by right-clicking in the cell.

Executing segment sweep

To execute a segment sweep by using the segment table you have created, you must specify the sweep type for that sweep operation by following the steps below.

1. Press **Channel Next** (or **Channel Prev**) to select the channel on which you execute the segment sweep operation.
2. Press **Sweep Setup** > **Sweep Type** > **Segment.**

Setting up the segment display

Define the method of displaying traces when the segment sweep is executed by following the steps described below.

1. Press **Channel Next** (or **Channel Prev**) to select the channel on which you define the segment display.
2. Press **Sweep Setup** > **Segment Display**.
3. Select the **segment display**.

Saving a newly created segment table in CSV format

As discussed in Creating a segment table, you can export the newly created segment table as a CSV (comma-separated value) formatted file (so it can be used easily in software that requires a different format).

1. Press **Sweep Setup**.
2. Click **Edit Segment Table** > **Export to CSV File** to open the Save As dialog box. Note that CSV files (***.csv**) is already selected as the file type when the dialog box first opens.
3. Type the file name in the File Name area and Click **Save** to save the segment table.

Calling a segment table saved in CSV Format

By importing a segment table file saved by E5061B, you can set up the segment table.

NOTE It is possible to recall a file from a different channel where it is saved.

1. Press **Sweep Setup**.

2. Click **Edit Segment Table** > **Import from CSV File** to open the Open dialog box. Note that CSV files (***.csv**) is already selected as the file type when the dialog box first opens.
3. Select the CSV format file to be imported, and click **Open** to call up the segment table.

   NOTE You cannot import a CSV-formatted file created/edited in spreadsheet software into the E5061B.

Measurement with Options

Measurement with Options

The following measurement is available depending on the installed option.

- Option 010 Time Domain/Fault Location Analysis
- Option 005 Impedance Analysis

Option 005 Impedance Analysis

Option 005 Impedance Analysis

- Quick Start
- Overview
- Setting Up Measurement
- Making Measurement
- Analyzing the Measurement

The option 005 can only be used with E5061B Option 3L5 and it requires the firmware revision A.02.00 and above.

Overview

Overview

- Feature
- Preparation for Measurement

Other topics about Option 005 Impedance Anaysis

Feature

- The E5061B Option 005 enables the analyzer to measure impedance parameters of electric components such as capacitors, inductors, and resonators.
- You can evaluate a broad range of components. This option supports the reflection, series-thru and shunt-thru methods using the S-parameter test port or Gain-Phase test port. These methods are suitable for low-to-middle, middle-to-high, and very low milliohm impedance range respectively.
- Agilent 7 mm and 4 TP (4-terminal pair) type test fixtures can be used with E5061B to provide a good repeatable measurement. The 7 mm type fixtures are connected to port 1 with the 16201A terminal adapter.

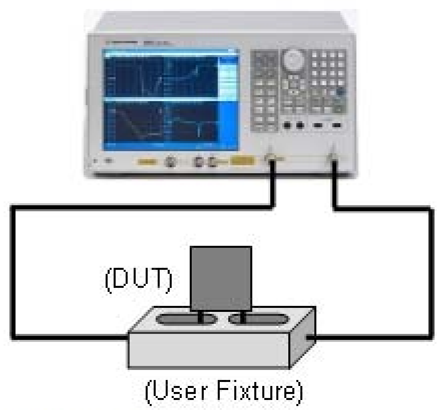
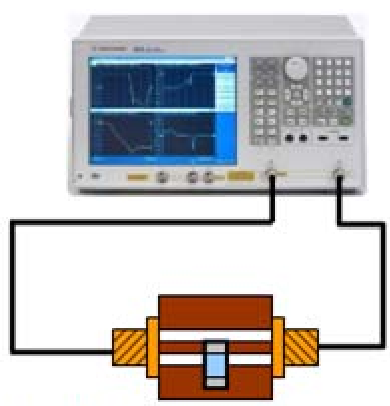
Other topics about Overview

Preparation for Measurement

- Measurement using ports 1 and 2
- Measurement using Gain-Phase Port

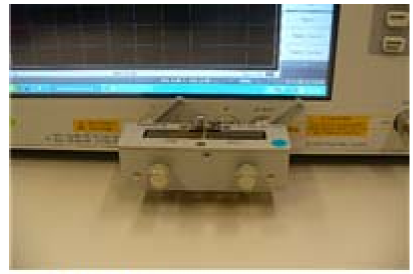
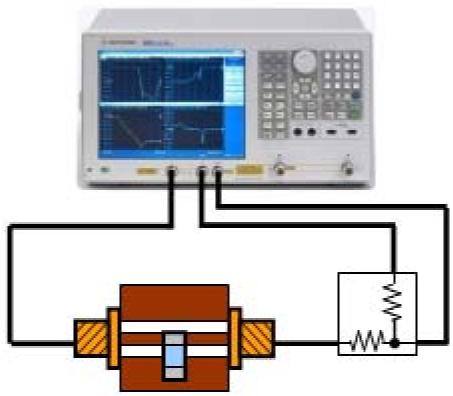
Other topics about Overview

Measurement using Ports 1 and 2

Method	Required Items	Connection
Port 1 Reflection	16201A Terminal Adapter, Agilent 7 mm type test fixture, Calibration Kit 16195B (or 85031B)	 e5061b075
Port 1-2 Series - Through	User-prepared fixture, Cables, 50 Ω precision resistor for calibration at fixture	 e5061b104
Port 1-2 Shunt - Through	User-prepared shunt-through fixture, Cables, Calibration kit	 e5061b079

Measurement using Gain-Phase Port

Method	Required Items	Connection

Gain-Phase Series - Through	Agilent 4 terminal-pair type test fixture (Recommendation 16047E for leaded DUTs, 16034E/G/H for SMDs), 50 Ω precision resistor for calibration at fixture	 e5061b111
Gain-Phase Shunt - Through	User-prepared fixture, Cables, Power splitter (11667L or equivalent)	 e5061b105

Setting Up Measurement

Setting Up Measurement

- Setting Measurement Conditions
- Performing Calibration
- Connecting Test Fixture
- Eliminating the Error Factor of Fixture

Other topics about Option 005 Impedance Analysis

Setting Measurement Condition

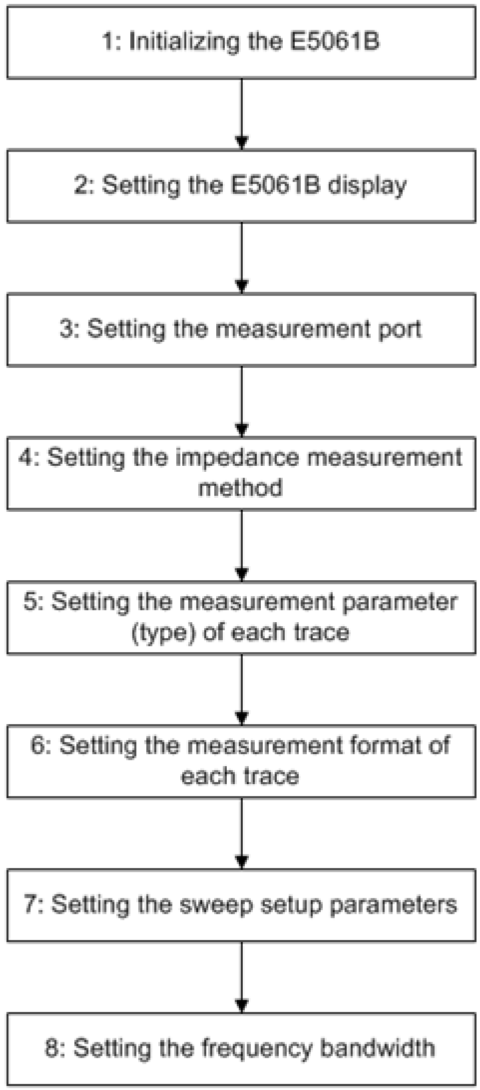
Setting Measurement Condition

- Flow of Setup
- Setting Measurement Parameters and Display Formats
- Setting Measurement Points, Sweep Parameters and Sweep Type

Other topics about Setting Up Measurement

Flow of Setup

The basic procedure of setting measurement condition is as shown in the flow chart below.



e5061b082

Other topics about Setting Measurement Condition

Setting Measurement Parameters and Display Formats

Other topics about Setting Measurement Condition

To set the measurement parameters and display formats:

Initializing the E5061B (Preset the E5061B)

1. Press **Preset** in the INSTR STATE block on the front panel.
2. Click **OK** from the menu bar.

Setting the E5061B display

1. Set the trace display (number of traces).
2. Press **Display** > **Num of Traces**.
3. Press the desired softkey to set the number of traces.
4. Set the trace layout (graph layout).
5. Press **Display** > **Allocate Traces**.
6. Select the preferred graph layout from the windows layout.

Setting the measurement port

1. Press **Meas** in the RESPONSE block on the front panel.
2. Click **Measurement Port**.
3. Select **S-Parameter** or **Gain-Phase**.

Setting the impedance measurement method

1. Press **Meas** in the RESPONSE block on the front panel.
2. Click **Impedance Analysis Menu**.
3. Click **Method** and select one of the method from the available options.

Method options	Measurement method details
Port 1 Refl	Reflection method at S-Parameter Port-1
Port 2 Refl	Reflection method at S-Parameter Port-2

Port 1-2 Series	Series-through method at S-Parameter Port-1 and 2
Port 1-2 Shunt	Shunt-through method at S-Parameter Port-1 and 2
GP Series T 50Ω, R 1 MΩ	Series-through method at Gain-phase test port with input impedance of R:1 Mohm and T:50 ohm
GP Shunt T 50Ω, R 50 Ω	Shunt-through method at Gain-phase test port with input impedance of R:50 ohm and T:50 ohm

Setting the measurement parameter (type) of each trace

1. Select the trace that you want to change its measurement parameter as the active trace.
2. To select the active trace, use the following hardkeys:

Hardkey	Function
Trace Next	Change the active trace to the next trace with the larger trace number.
Trace Prev	Change the active trace to the previous trace with the smaller trace number.

3. After selecting the active trace, press **Meas** hardkey.
4. Click **Impedance Analysis Menu**.
5. Select the measurement parameter from the available options.

Measurement parameter	Measurement parameter details
|Z|	Impedance magnitude
θz	Impedance phase in degree
|Y|	Admittance magnitude
θy	Admittance phase in degree
Cp	Parallel capacitance
Cs	Series capacitance
Lp	Parallel inductance
Ls	Series inductance

Rp	Parallel resistance
Rs	Series resistance
D	Dissipation factor
Q	Quality factor
R	Resistance
X	Reactance
G	Conductance
B	Susceptance

6. Repeat the step for other traces, if available.

Setting the measurement format of each trace

1. Select the trace that you want to change its measurement parameter as the active trace.
2. Press **Format** in the RESPONSE block on the front panel.
3. If **Impedance Analysis Menu** is selected as the measurement method, then the only available format is Expand Phase (**Exp Phase**).
4. Click the **Exp Phase** button to turn it **ON** or **OFF**.

Setting Measurement Points, Sweep Parameter and Sweep Type

- Setting the measurement points
- Setting the sweep setup parameters

Other topics about Setting Measurement Condition

To set the measurement points, sweep parameter and sweep type:

Setting the measurement points

The number of points is the number of data items collected in one sweep. It can be set to any number from 2 to 1601 for each channel independently.

1. Press **Sweep Setup** in the STIMULUS block on the front panel.
2. Click **Points** from the Sweep menu bar.
3. Then, using either the keyboard or ENTRY keys on the front panel, enter the desired number of points.

Setting the sweep setup parameters

1. Press **Sweep Setup** in the STIMULUS block on the front panel.
2. Click **Power** from the Sweep menu bar.
3. Using either the keyboard or ENTRY keys on the front panel, enter the power value.
4. To set the sweep type, click **Sweep Type** from the Sweep menu bar and select the type from the available option.

Sweep type	Sweep parameter details
Lin freq	Sweep frequencies in linear scale
Log freq	Sweep frequencies in logarithmic scale
Segment	Performs a sweep with linear sweep conditions (segments) combined
Power sweep	Sweeps power levels in linear scale
DC bias sweep	Sweeps DC bias levels

Performing Calibration

Performing Calibration

- Connecting Adapter
- Defining Calibration Kit
- Performing Impedance Calibration
- Using S-Parameter Calibration for Impedance Measurement

Other topics about Setting Up Measurement

Connecting Terminal Adapter

In the Port 1 Reflection method, the 7 mm terminal adapter can be used for connecting the Agilent's 7 mm type component test fixtures to the E5061B's Port 1.

- Required Parts
- Replacement Procedure

Other topics about Performing Calibration

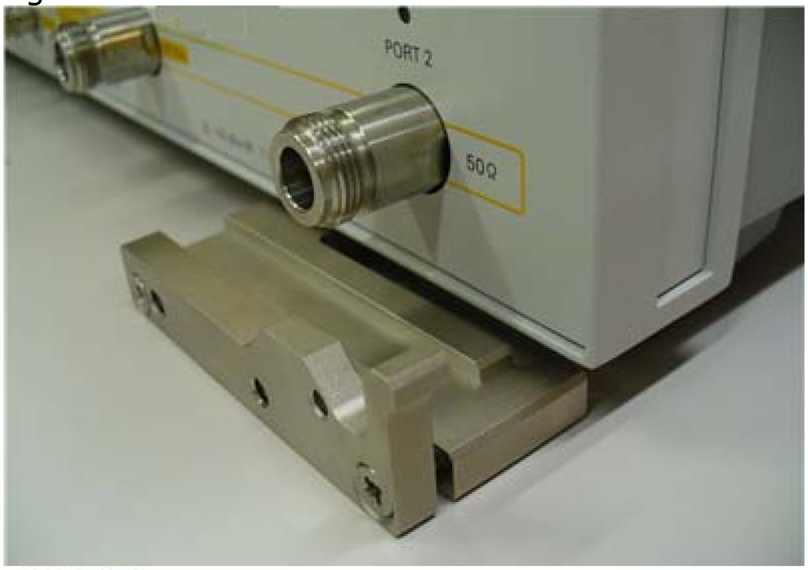
Required Parts

The following parts are furnished with the 16201A Terminal Adapter kit.

Description	Agilent Part Number
Terminal Adapter Main Unit	16201-60102
Semi-rigid Cable, N connector	16201-61601
Attachment (Front)	16201-20002
Attachment (Bottom)	16201-20001
Screws for Attachment	0515-1013 (2 ea.)

Replacement Procedure

1. Fasten the front and bottom attachments with two screws. Do not tighten the screw at this moment.

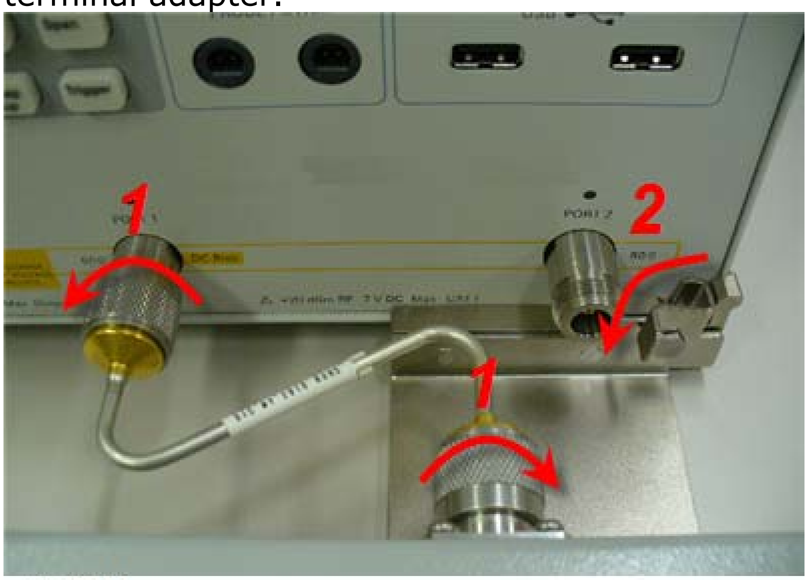


e5061b072

2. Place the attachment at the right-bottom of the E5061B. Fix it by tightening the two screws.

e5061b073

3. Place the semi-rigid cable between Port 1 on E5061B and port of the terminal adapter.



e5061b074

4. Rotate the lock so that the terminal adapter is fixed with the attachment.

Performing Impedance Calibration

- Overview
- Selecting Calibration Kit
- Performing Open/Short/Load/(Low-Loss C) Calibration
- Acquisition Status of Calibration Coefficient

Other topics about Performing Calibration

Overview

The impedance calibration is a calibration function dedicated to the impedance analysis with the E5061B option 005. The impedance calibration executes the 3-term calibration (open/short/load calibration) directly in the impedance domain instead of the S-parameter domain, after the raw S-parameter measurement data or the gain-phase T/R measurement data is converted to the impedance data. Therefore, you need to select your desired impedance measurement method before performing the impedance calibration. The impedance calibration basically consists of the open, short, and load calibrations. You can perform the optional low-loss capacitor calibrations in addition to the open/short/load calibrations. The impedance calibration is applied only to the impedance parameters (see Data Processing). It is not applied to the S-parameter and Gain-Phase T/R ratio measurements.

The following table shows the recommended calibration for each method. Note that the calibrations for network analysis such as the 2 port full and response through are used in the Shunt-Thru methods, instead of the impedance calibration.

Recommended Calibration

Method	Configuration	Calibration Device	Calibration	Calibration Plane	Fixture Compensation
Port 1 Refl	16201A Terminal Adapter + Agilent 7mm Fixture	7 mm calibration kit	Impedance Calibration (open/short/load/[low-loss C])	7 mm connector on the terminal adapter	Required (open/short and electrical length)
Port 1 Refl / Port 2 Refl	User-prepared fixture	50 Ω resistor	Impedance Calibration (open/short/load)	The measurement terminals of Fixture	Not required
Port 1-2 Series	User-prepared fixture	50 Ω resistor	Impedance Calibration (open/short/load)	The measurement terminals of	Not required

				Fixture	
Port 1-2 Shunt	User-prepared fixture	Coaxial calibration kit	2 Port Full calibration and Port extension.	End of test cable	Not required
GP Series T 50 Ω, R 1 MΩ	Agilent 4 terminal-pair fixtures (2 Terminal contact Type)	50 Ω resistor	Impedance Calibration (open/short/load)	The measurement terminals of Fixture	Not required
	User-prepared fixture	50 Ω resistor	Impedance Calibration (open/short/load)	The measurement terminals of Fixture	Not required
GP Shunt T 50 Ω, R 50 Ω	User-prepared fixture	User-prepared through device or user prepared open, short and load	Response Calibration (Through) or Impedance Calibration (open/short/load)	The measurement terminals of Fixture	Not required

Difference of Calibration (for S-parameter) and Impedance Calibration

- The impedance calibration executes the 3-term calibration in the impedance domain after the S-parameter or Gain-Phase T/R ratio is converted to the impedance.
- The impedance calibration supports the low-loss capacitor calibration.
- The impedance calibration is not applied to the network analysis data (S-parameters and Gain-Phase T/R ratio).

The impedance calibration and the calibrations for network analysis, such as the 1-port full, 2-port full, and response through cannot be performed at the same time. If you perform the impedance calibration after performing these network-analysis calibrations in the same channel, the data of network-analysis calibrations is overwritten and deleted.

Low-Loss Capacitor Calibration

When you measure Q, D, and ESR of RF capacitors and RF inductors by using the reflection method with the 16201A terminal adapter and Agilent’s 7 mm test fixtures, performing the low-loss capacitor (LLC) calibration in addition to the open/short/load improves the accuracy of Q, D, and ESR measurements at high frequencies over 1GHz. The LLC provides a reference for calibration with respect to the 90°-phase component of impedance. Agilent 16195B calibration kit provides a low-loss capacitor termination in addition to the open/short/load terminations

Selecting Calibration Kit for Impedance Calibration

You need to select calibration kit before you perform the impedance calibration.

Method	Configuration	Required Calibration Kit
Port 1 Refl	Terminal Adapter + 7 mm test fixture	85031B (open/short/load only), or 16195B (open/short/load, plus low-loss-C)
Others	Test fixture for SMD	SMD 50ohm
	Test fixture for leaded DUT	Leaded 50ohm

1. Press **Cal** > **Cal Kit.**
2. Select your calibration kit.

Performing Open/Short/Load/(Low-Loss C) Calibration (Impedance Calibration)

Impedance calibration should be performed at the following location.

Configuration	Calibration Plane
16201A Terminal Adapter + 7 mm test fixture	7mm Connector on the terminal adapter
Test fixture for SMD	Measurement terminals of the fixture
Test fixture for leaded DUT	Measurement terminals of the fixture

1. Press **Channel Next** (or **Channel Prev**) to select the channel for which you want to calibrate (If you are using multiple channels).
2. Confirm if the desired method is selected (**Meas** > **Method**).
3. Press **Cal** > **Calibrate** > **Impedance Calibration.**
4. Connect the open termination, or leave the fixture's measurement terminals open.
5. Click **Open** to execute open calibration (After the calibration is performed, a check mark is displayed at the left of the softkey).
6. Connect the short termination to the location of the open termination, or short the measurement terminals of the fixture.

7. Click **Short** to execute short calibration.
8. Connect the load termination in place of the short termination, or connect the 50 Ω resistor to the fixture.
9. Click **Load** to execute load calibration.
10. If you use the 16201A terminal adapter and the 16195B 7 mm calibration kit, you can perform the optional low-loss capacitor calibration here. Follow the procedure below:
    a. Connect the Low-Low capacitor termination instead of the load termination.
    b. Click **Low-Loss C (Optional)** to execute the low loss capacitor calibration.
11. Click **Done** to complete the impedance calibration process. Upon pressing this key, calibration coefficients are calculated and saved. The error correction function is also enabled automatically.
12. **Cor** is displayed at the Channel Window.
    (If there is no impedance parameters in the channel, **Cor** is not displayed. For example, when only S11 is displayed at the trace, **Cor** is not displayed even after impedance calibration. Changing the parameter to |Z| displays **Cor**.)

Acquisition Status of Calibration Coefficient

As in the S-Parameter measurement, the calibration property (**Cal** > **Property**) shows the acquisition status of Impedance calibration.

Method	S	R	Property Value
Port 1 Refl	1	1	G
Port 2 Refl	2	2	G
Port 1-2 Series	1	2	S
Port 1-2 Shunt	1	2	P
GP Series T 50 Ω, R 1 MΩ	G	G	S
GP Shunt T 50 Ω, R 50 Ω	G	G	P

The following example shows that impedance calibration performed with Port 1-2 Series method.

		S	
	1	2	G
1	-	-	-

R2 S- -
G- - -

Defining Calibration Kit

- Overview
- Procedure Defining Calibration Kit
- Preset of Calibration Kit Definition

Other topics about Performing Calibration

Overview

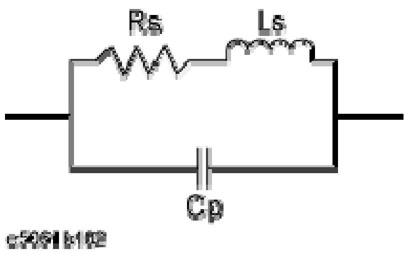
When option 005 impedance analysis is installed, the calibration kit can be defined by the following three standard models. The equivalent circuit and table models are available only when option 005 is installed.

Polynomial

This model is the same as the calibration kit definition of S-Parameter measurements. See Modifying Calibration Kit Definition.

Equivalent Circuit

This model allows you to define the calibration kit by the following model. When you use the equivalent circuit model, one of either Open, Short, Load or Arbitrary should be selected as the Standard Type.



Table

You can specify the frequency characteristic of impedance for the termination. The definition cannot be set using the softkey. You need to specify it by using the commands. See the following command to specify the calibration kit.

- SCPI.SENSe(Ch).CORRection.COLLect.CKIT.STAN(Std).TABLe / :SENS:CORR:COLL:CKIT:STAN:TABL

When you use the table model, either one of these: Open, Short, Load or Arbitrary should be selected as the Standard Type.

Procedure of Defining Calibration Kit

This section provides the procedure to change the definition of a calibration kit.

Polynomial

See Defining Parameters for Standards.

Equivalent Circuit/Table

a. Select and define a calibration kit

1. Press **Cal** > **Cal Kit**, then select the calibration kit to be redefined.
2. Click **Modify Kit**.
3. If necessary ,click **Label Kit** and type a new label for the calibration kit.

b. Select the standard type and define standard coefficient

4. Click **Define STDs** and select the standard number to be redefined.
5. If necessary, click **Label**, then type your desired name for the selected standard.
6. Click **STD Type**, then select the **type of standard**.
7. Click **STD Model** > **Equiv Ckt** or **Table**
8. When **STD Model** is set at **Equiv Ckt**, set the **standard coefficient**.
9. Repeat steps 4 to 8 to redefine all the standards for which changes are necessary, then click **Return**.

Standard Class allows you to use different standards for each port. Generally, this is not used in the impedance calibration.

Preset of Calibration Kit Definition

See Preset the definition for calibration kits

Using S-Parameter Calibration for Impedance Measurement

It is not possible to apply both the network-analysis calibration and impedance calibration to the same channel of the E5061B. Also, the impedance calibration is not applied to the S-Parameter measurement traces. For example, if you want to measure the circuit's input/output impedance with both S11 and impedance parameter traces, 1 port full calibration should be performed instead of the impedance calibration, because the impedance calibration is not applied to the S11 traces.

Connecting Test Fixture

Connecting Test Fixture

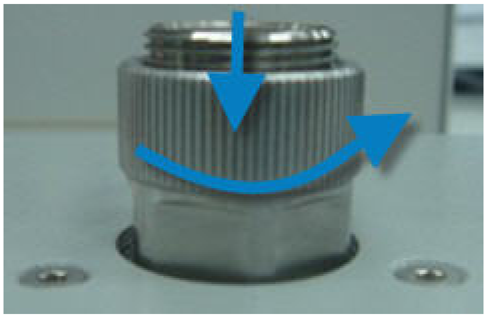
- Connecting Fixture (S-Parameter)
- Connecting Fixture (Gain-Phase)

Other topics about Setting Up Measurement

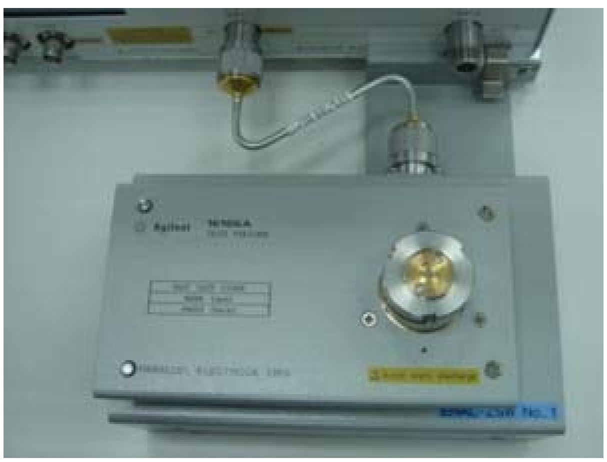
Connecting Fixture for Port 1 Reflection Method

In the Port 1 reflection method, Agilent's 7 mm-type component test fixtures can be connected to the E5061B's Port 1 via the 16201A terminal adapter. If you perform impedance calibration using the 7 mm calibration kit, the test fixture should be connected after the calibration is completed. Once the fixture is connected, open/short compensation should be performed at the measurement terminals of the fixture. Or, if you perform impedance calibration at the measurement terminals of the fixture by using the leaded or SMD 50 Ω resistors, the fixture should be connected before performing the calibration.

1. Turn the 7-mm connector nut of the test head counterclockwise until the connector sleeve is fully retracted.



2. Set the test fixture on top of the terminal adapter with the 7-mm connectors aligned. Ensure that the two holes on the test fixture is mounted on the two mount posts of the terminal adapter.
3. Turn the 7-mm connector nut of the terminal adapter counterclockwise to secure the connection, as shown in the following figure. Use both hands to turn the connector nut because the space between the terminal adapter and test fixture is narrow.



Connecting Fixture for Gain-Phase Series-through Method

You can use the Agilent's 4-terminal-pair type fixtures for the Gain-Phase series-through method. Although the E5061B does not have a 4-terminal pair configuration, you can perform the impedance measurement by directly connecting the 4-terminal-pair fixture to the Gain-Phase test port and performing impedance calibration (open/short/load calibration) at the fixture's measurement terminals. Note that the measurement technique of this configuration is not an auto balancing bridge method but a simpler VNA-based series-through method..

- Available Test Fixtures
- Connecting Procedure

Other topics about Connecting Test Fixture

Available Test Fixtures

4-terminal-pair fixture for the Gain-Phase series-through method should be a 2-terminal contact type, and the fixture's measurement terminals should have good contact repeatability so that the 50 Ω resistor is tightly fixed when performing the impedance calibration.

The following fixtures are recommended:

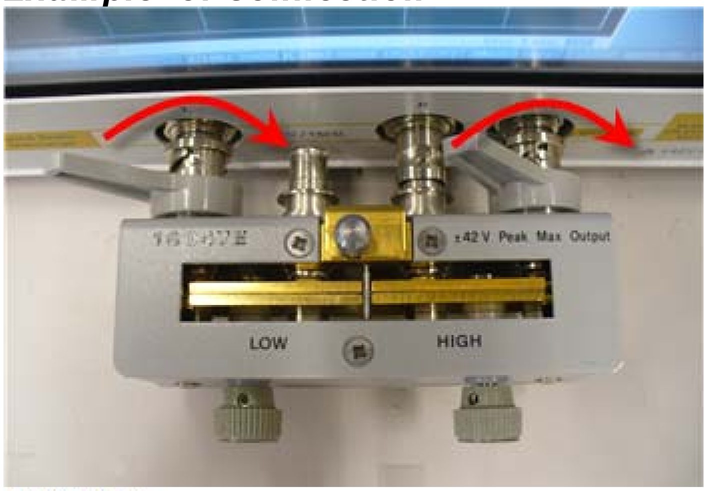
- 16047E (for leaded DUTs)
- 16034E/G/H (for SMDs)

The 4-terminal contact (Kelvin contact) type fixtures such as 16044A are not recommended, because the contacting method to the DUT will not be a 4-terminal contact when used with the E5061B's Gain-Phase series-thru method and there is no advantage in using these fixtures.

Connecting Procedure

1. Connect the fixture on the Ports T, R and LF Out. The second connector (for Lpot) from the left of fixture is not connected.
2. Turn the knobs (of the fixture) to fix the fixture.

Example of Connection



e5061b107

Eliminating Error Factor of Fixture

Eliminating Error Factor of Fixture (Fixture Compensation)

The following function allows you to eliminate the error factor of fixture and cable.

- Open/Short/Load Compensation
- Z Port Extension (Fixture Electrical Length)

Generally, these functions are used in one of these ways.

- Open/Short Compensation and Z Port Extension (Fixture Electrical Length)
- Open/Short/Load Compensation

Other topics about Setting Up Measurement

Other topics about Eliminating Error Factor of Fixture

Overview

When you measure the impedance by using the Agilent 16201A terminal adapter and performing the impedance calibration at the adapter's 7 mm plane, the error factors of the 7 mm fixture mounted on the adapter can be eliminated with the fixture compensation. The recommended fixture compensation method up to a few hundreds of MHz is the open/short compensation. For the measurement at higher frequencies over a few hundreds of MHz, it is recommended to perform the combination of the fixture electrical length compensation and the open/short compensation.

The open/short compensation eliminates the error factors of the fixture's non-coaxial section. The open admittance and the short impedance of the fixture can be eliminated by measuring the impedance with the fixture's measurement terminals opened and shorted, respectively.

The open/short compensation is applicable only to the impedance measurement data, and is not applicable to the S-parameter and gain-phase T/R measurement data.

The fixture electrical length compensation eliminates the measurement error induced by the phase shift in the fixture's coaxial section. The fixture electrical length compensation eliminates the phase-shift error by selecting the fixture's model number, or by manually entering the fixture's electrical length in meters.

Another method for eliminating the phase-shift error is to define the delay with the Z Port Extension function. The Z Port Extension is the port extension function dedicated to the impedance measurement of the E5061B option 005. For example, if you connect an extension coaxial cable between the 7 mm calibration plane and the fixture, the phase-shift error that occurs in the cable can be eliminated with the Z Port Extension.

Z Port Extension and Fixture Electrical length compensation is the same idea and the only difference is the unit. Z Port Extension is expressed in a delay time in seconds. On the other hand, the Fixture Electrical length is expressed in meters (mili-meter).

Electrical Length =Port Extension × (velocity of the signal in free space: 3×10^8) .

The fixture electrical length compensation and the Z port extension are applicable only to the impedance measurement data, and are not applicable to the S-parameter and gain-phase T/R measurements. (See the Data Processing)

Open/Short Compensation

Open/Short compensation provides the same functionality as the one in Agilent RF Impedance Analyzer such as the E4991A.

1. Set the open state of the fixture.
2. Press **Cal** > **Fixture Compen** > **Compensate** > **Open.**
3. Set the short state of the fixture.
4. Click **Short**.
5. Click **Done**.
6. Open/Short compensations are turned ON automatically. **Comp OS** is displayed at the Channel Window.

If you use the fixture electrical length compensation or the Z Port Extension in addition to the open/short compensation, the fixture electrical length or the Z Port Extension values must be set before performing the open/short compensation.

Z Port Extension and Fixture Electrical Length

Z Port Extension

Z Port Extension provides the same functionality as the one in Agilent RF Impedance Analyzer such as the E4991A. This is used when the coaxial cable is connected from where the calibration is performed. In Z port extension, loss is not compensated. Z Port Extension is applied even if **Cal** > **Fixture Compen** is OFF.

1. Press **Cal** > **Fixture Compen** > **Z Port Extension**
2. Enter the port extension value, then press **Enter**.
3. **ZExt** is displayed at the Channel Window.

Fixture electrical length

Fixture Electrical length compensation provides the same functionality as the one in Agilent RF Impedance Analyzer such as the E4991A. Since the electrical length of an exclusive-use test fixture is registered in the E5061B, the necessary electrical length can be set by simply selecting the model number of the test fixture used. It is also possible to input the specified electrical length value. Fixture Electrical length is applied even if **Cal** > **Fixture Compen** is OFF.

1. Press **Cal** > **Fixture Compen** > **Fixture.**
2. Select the fixture used.
3. **Fixtr {Fixture Model Number}** is displayed at the Channel Window.

Using User Fixture

If you use the fixture whose electrical length is not pre-defined in the firmware menu, you can manually define its electrical length. If you use the 16092A fixture, for example, define its electrical length of 3.4 mm by using the following procedure:

1. Press **Cal** > **Fixture Compen** > **Modify User Fixt.** > **User Fixture**
2. Enter the electrical length value of your fixture, then press **Enter.**
3. Press **Cal** > **Fixture Compen** > **Fixture** > **User.**
4. **Fixtr User** is displayed at the Channel Window.

When the electrical length value of user fixture is set at 0, **Fixtr User** is not displayed at the Channel Window even if **User** is selected at **Cal** > **Fixture Compen** > **Fixture** > **User.**

Other topics about Eliminating Error Factor of Fixture

Overview

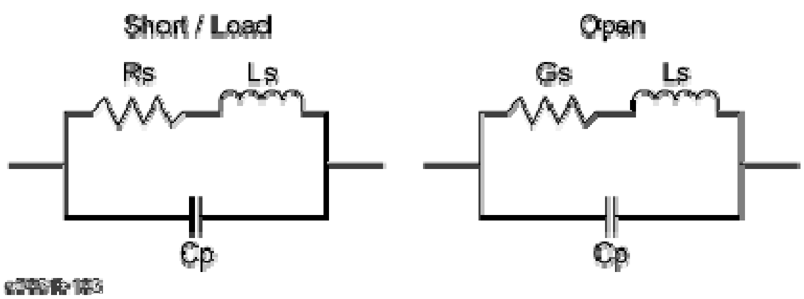
The open/short/load compensation gives the same calibration result as the open/short/load calibration at the fixture. In the impedance measurement methods such as the Gain-phase series-thru method, which require the open/short/load calibration at the fixture, you can use the open/short/load compensation as an alternative method to the open/short/load calibration. Similarly to the open/short/load calibration at the fixture, you can use the SMD or leaded 50 Ω resistor as a load device.

The difference from Impedance Calibration

- Open/Short/Load can be turn ON/OFF independently
- Definition of standard

Method	Standard Model
Impedance Calibration	Polynomial, Equivalent Circuit and Table
Compensation	Equivalent Circuit only

In the fixture compensation, you can define the open Gs value by the Admittance.



Defining Compensation Kit

1. Press **Cal** > **Fixture Compen** > **Compen STDs**.
2. Click the parameter which you want to define.
3. Enter the value.

Open/Short/Load Compensation

1. Set the open state of the fixture.
2. Press **Cal** > **Fixture Compen** > **Compensate** > **Open.**
3. Set the short state of the fixture.
4. Click **Short.**
5. Set the load state of the fixture.
6. Click **Load**.
7. Click **Done**.
8. Open/Short/Load compensations are turned ON automatically. **Comp OSL** is displayed at the Channel Window.

Making Measurement

Making Measurement

- Connecting DUT to Test Fixture
- Making a Trigger
- Scaling
- Having Stable Measurement Result
- Changing Sweep Conditions
- Measuring Other DUTs

Other topics about Option 005 Impedance Analysis

Connecting DUT to Test Fixture

The method to connect DUT to test fixture differs depending on the test fixture being used. Refer to the Operation and Service Manual of the respective test fixture to learn more.

Making a Trigger

To perform a measurement, it is necessary to generate a trigger. The trigger source may either be internal, external, manual or bus.

Scaling

- Scale Type
- Auto Scaling
- Manual Scaling

Other topics about Making Measurement

Scale Type

To make the measurement, the scale of the impedance measurement, |Z| trace should be set to log scale:

To set the scale of the |Z| trace to log scale, perform the following procedure:

1. Select the trace as an active trace.
2. Press **Scale** in the RESPONSE block on the front panel.
3. Click **Y-Axis**.
4. Click **Log**.

When the scale of the Y-axis is changed from Linear to Log, the trace parameter display at upper-left of the trace window changes.

The setting of Log and Linear scale is independent, hence:

- When Y-Log is selected, Y-Linear Scale Division and Reference Position values are not used. Instead, Top/ Bottom scale values are used.
- Top/Bottom scale values are not used in Y-Linear.

Top/Bottom

Top/bottom value has the following limitation:

$$2 \leq \text{Top/Bottom} \leq 10^{24}$$

If one of these values (Top or Bottom) exceeds the limit, the other value changes automatically so that the Top/Bottom value is within the limit.

Auto Scaling

Auto scale sets the appropriate scale for Y-axis of all the traces. In the case of Log scale, auto scale sets the top and bottom of the waveform within 80% of the full scale.

To set the scale automatically, perform the following procedure:

1. Press **Scale** in the RESPONSE block on the front panel.
2. Click **Auto Scale** to perform the auto scale function on a specific trace.
3. OR click **Auto Scale All** to perform the auto scale function on all traces within a channel.

Manual Scaling

To adjust the scale manually, perform the following procedure:

1. Press **Scale** in the RESPONSE block on the front panel.
2. Click **Log Y-Axis Top/Bottom.**
3. At Log Y-Axis Top/Bottom menu bar, using either the keyboard or ENTRY keys on the front panel, enter the Top value and Bottom value.

Having Stable Measurement Result

For more stable measurement you can choose to set the followings:

- Setting IF Bandwidth
- Setting Averaging
- Setting Smoothing

Other topics about Making Measurement

Setting IF Bandwidth

To set the IF bandwidth perform the following procedure:

1. Press **Avg** in the RESPONSE block on the front panel.
2. Click **IF Bandwidth**.
3. Using either the keyboard or ENTRY keys on the front panel, enter the bandwidth value.

Alternatively, you can set the IF bandwidth automatically. The procedure is as follows:

1. Press **Avg** in the RESPONSE block on the front panel.
2. Click **IFBW Auto** to turn ON the IF bandwidth auto mode.

If you want to set an upper limit of IF bandwidth in auto mode:

1. Click **IFBW Auto Limit**.
2. Using either the keyboard or ENTRY keys on the front panel, enter the upper limit of IF bandwidth.

Setting Averaging

To turn ON the averaging, perform the following procedure:

1. Press **Avg** in the RESPONSE block on the front panel.
2. Click **Averaging**.
3. Click **ON**.

Setting Smoothing

To turn ON the smoothing, perform the following procedure:

1. Press **Avg** in the RESPONSE block on the front panel.
2. Click **Smoothing**.
3. Click **ON**.

Changing Sweep Conditions

When the below conditions change:

- IF bandwidth
- Power level
- Power range
- Sweep time / Sweep delay time
- Sweep type
- Number of points

When these conditions change, interpolation is executed. Error correction status symbol, **C?** is displayed at the channel status bar at the lower part of the window. This indicates that the changes in the conditions are within an acceptable range. At this point, calibration is advisable but not mandatory.

When the below conditions change:

- Start/Stop
- Centre/Span

When these conditions change and the sweep span is extended, the extrapolation is executed. Error correction status symbol, Hence **C!** is displayed at the channel status bar at the lower part of the window. This indicates that the changes in the conditions are far from the acceptable range. At this point, calibration is mandatory.

Changing Measurement Points

To change the number of measurement points, refer to Setting the measurement points.

Changing Sweep Parameter

To change the sweep parameter, refer to Setting the sweep parameter.

Measuring Other DUTs

If you measure another DUT of the same type and size as the one used in the previous measurement, start measurement with Connecting DUT to Test Fixture. If you use the same test fixture to measure a DUT of a different type and size, start measurement with calibration or fixture compensation. When using a different test fixture, start measurement with Connecting Test Fixture.

When measuring a DUT in the initial state after turning the power ON, start measurement with Setting Measurement Conditions.

Analyzing the Measurement

Analyzing the Measurement

- Using Equivalent Circuit Analysis
- Other Analyzing Functions

Other topics about Option 005 Impedance Analysis

Using Equivalent Circuit Analysis

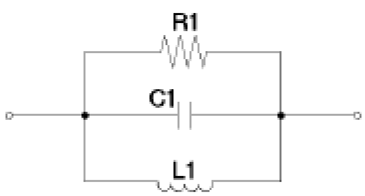
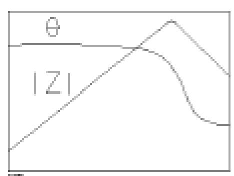
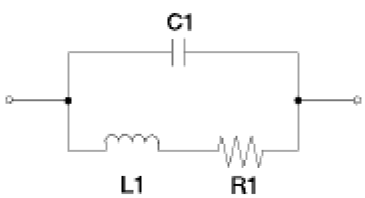
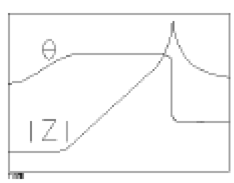
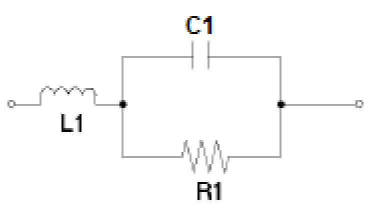
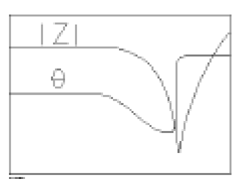
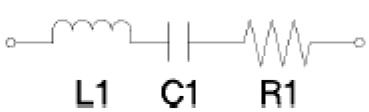
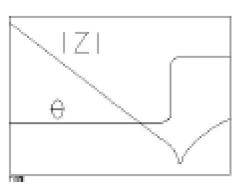
- Overview
- Equivalent Circuit Menu
- Performing Equivalent Circuit Analysis

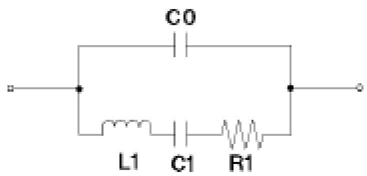
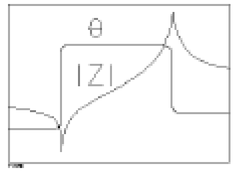
Other topics about Analyzing the Measurement

Overview

The E5061B Option 005 provides five types of equivalent circuit models; four types of 3-element equivalent circuits and one type of 4-element equivalent circuits. These circuits can be used to calculate the approximate values of the equivalent circuit parameters from the measurement data as well as to display the frequency characteristics on the screen based on the input equivalent circuit parameter.

The following table shows the selection of equivalent circuit:

Equivalent Circuit Model		Typical Frequency Characteristics	DUT Example
A		 *1	Inductor with high core loss
B		 *1	Inductor Resistor
C		 *1	High-value resistor
D		 *1	Capacitor

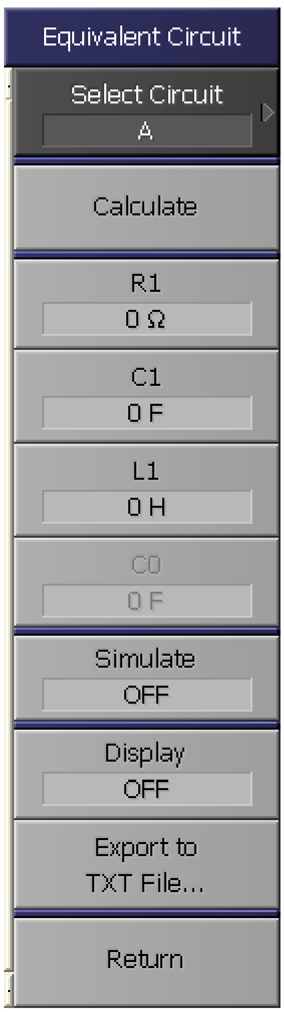
E		 *2	Resonator

*1. Measurement parameter: |Z| - θ, Sweep type: log, Vertical axis: |Z| is log and θ is linear.

*2. Measurement parameter: |Z| - θ, Sweep type: linear (or log), Vertical axis: |Z| is log and θ is linear.

Equivalent Circuit Menu

The equivalent circuit analysis function is only available when option 005 Impedance Analysis is installed and impedance measurement is selected (**Meas**> **Impedance Analysis Menu**).



e5061b129

Select Circuit

Shows the menu bar to select one of the five equivalent circuit model:

Model A - Generally suited to analyze inductors with high core loss.

Model B - Generally suited to analyze general inductors and resistors.

Model C - Generally suited to analyze resistors with high resistance.

Model D - Generally suited to analyze capacitors.

Model E - Generally suited to analyze resonators and oscillators.

Calculate

Calculates the equivalent circuit parameters based on the measurement results and the selected equivalent circuit model.

R1

Allows you to enter the R1 parameter for the equivalent circuit model selected at Select Circuit. Also displays the equivalent R1 parameter calculated when Calculate is selected.

C1

Allows you to enter the C1 parameter for the equivalent circuit model selected at Select Circuit. Also displays the equivalent C1 parameter calculated when Calculate is selected.

L1

Allows you to enter the L1 parameter for the equivalent circuit model selected at Select Circuit. Also displays the equivalent L1 parameter calculated when Calculate is selected.

C0

Allows you to enter the C0 parameter for the equivalent circuit model selected at Select Circuit. When equivalent circuit model E is not selected, C0 is disabled. Also displays the equivalent C0 parameter calculated when Calculate is selected.

Simulate

Simulates the selected equivalent circuit model frequency characterization based on the equivalent circuit parameter entered or calculated by the **Calculate** button. It is performed for all traces. The simulated results are stored into the memory trace and displayed on screen. When the Simulate is ON, Memory Trace is updated automatically.

Display

Displays the equivalent circuit model in schematic and the value of each equivalent parameter on the bottom-left of the channel window.

Export to TXT File...

Saves the equivalent circuit parameters in text file at user-defined location.

Performing Equivalent Circuit Analysis

Step 1: Performing impedance measurement

Step 2: Selecting equivalent circuit

1. Press **Analysis** in the MKR/ANALYSIS block on the front panel.
2. Click **Equivalent Circuit**.
3. Click **Select Circuit**.
4. Select the desired equivalent circuit model from the five equivalent circuit models.

Step 3: Calculating equivalent circuit

1. Click **Calculate** to execute calculation of the equivalent circuit parameter.

The calculated equivalent circuit parameters are displayed in each box of **R1**, **C1**, **L1** and **C0**.

Step 4: Simulating equivalent circuit

1. Click **Simulate**.

The selected equivalent circuit model frequency characterization is simulated based on the equivalent circuit parameter entered or calculated by the **Calculate** button. The simulated results are stored into the memory trace and displayed on screen.

Step 5: Saving equivalent circuit

1. Click **Export to TXT File...**.
2. **Save As** dialog box shows.
3. Select the desired file name and location. The default file name is D:\EqvCkt01.txt.
4. Click **Save**.

Other Analyzing Functions

Besides Equivalent Circuit Analysis, other available analyzing functions are:

Marker

- Reference Marker
- Clear Marker

Marker Search

- Max
- Min
- Peak
- Target
- Multi Peak
- Multi Target
- Tracking
- Search Range

Marker Function

- Marker -> Start
- Marker -> Stop
- Marker -> Center
- Marker - Reference
- Discrete
- Couple
- Marker Table
- Statistics
- Annotation Options
  - Marker Info
  - Align
  - Active Only

Option 010 Time Domain/Fault Location Analysis

Option 010 Time Domain/Fault Location Analysis

- Fault Location Analysis
- Structural Return Loss Measurement

Fault Location Analysis

Fault Location Analysis

- Theory
- Making Fault Location Measurement
- Deleting Unnecessary Data in Fault Location (gating)

Other topics about Option 010 Time Domain/Fault Location Analysis

Theory

This section describes basic fault location measurement theory, how the analyzer converts frequency-domain data to distance-domain data, and the relationship between start distance, stop distance and frequency span.

- Overview
- How the Analyzer Converts Frequency Data to Distance Data
- Start/Stop Distance and Frequency Span Explanation

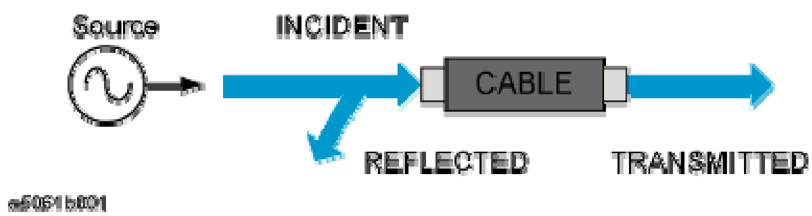
Other topics about Fault Location Analysis

Overview

Fault location measurements are designed to quickly and easily locate faults, or discontinuities, in either 50 Ω or 75 Ω transmission lines.

The network analyzer has an RF signal source that produces an incident signal that is used as a stimulus to locate and measure discontinuities in your transmission line or cable. Each fault or discontinuity responds by reflecting a portion of the incident signal and transmitting the remaining signal.

The analyzer measures the frequency response of the cable and then transforms the frequency data to distance data.



Typically, fault location measurement results are expressed in one of the four ways:

Format	Description
Return Loss (RL)	The number of dB that shows the reflected signal is below the incident signal. Its relationship to the reflection coefficient (ρ) is described by the following formula: RL = -20 log ρ.
Reflection Coefficient (ρ)	The ratio of the reflected voltage wave to the incident voltage wave.
Standing Wave Ratio (SWR)	Any two waves traveling in opposite directions (the incident and reflected for example) cause a "standing wave" to be formed on the transmission line. SWR is defined as the

	maximum voltage over the minimum voltage of the standing wave. SWR can also be mathematically derived from the reflection coefficient (ρ) with the following formula: *SWR=(1+ρ)/(1-ρ)*
Impedance Magnitude	The magnitude of the complex impedance at each measurement point. $\lvert Z \rvert = \sqrt{(Z_{real}^2 + Z_{imaginary}^2)}$

How the Analyzer Converts Frequency Data to Distance Data

Fault-location measurements are single-ended measurements, meaning that only one end of a cable under test needs to be connected to the analyzer's RF OUT test port.

This type of measurement is generally called a reflection measurement and typically displays a response commonly known as return loss.

The analyzer performs swept-frequency measurements of return loss versus frequency, then uses the Fourier transform to convert the response-versus-frequency to a response-versus-distance. The analyzer's internal computer makes the calculation by using either the inverse discrete Fourier transform (inverse FFT) technique or the chirp-Z Fourier transform technique.

The Fourier transform technique is essentially a process of adding the signals measured by the analyzer in the frequency domain and combining them to create the fault-location response in the time domain.

The resulting measurement is an error-corrected fault-location response of the cable under test.

Start/Stop Distance and Frequency Span Explanation

When the analyzer is set up for a fault location measurement, you can determine the center frequency (when in band pass mode), and start and stop distances for the measurement. The distance range (start distance - stop distance) determines the frequency span, which in turn determines the start and stop frequencies.

NOTE In band pass mode (as opposed to low pass mode), you can select center frequency. Changes to the distance range do not affect the user-chosen center frequency.

The analyzer attempts to set the frequency span to the setting required for the distance range. The maximum setting for the frequency span cannot exceed the analyzer's frequency capability. For instance, the start frequency cannot be lower than the analyzer's low frequency limit, and the stop frequency cannot be higher than the analyzer's high frequency limit.

Making Fault Location Measurement

- Basic Measurement Procedures

  The basic measurement procedures of fault location measurement consists of the following steps:

  1. Enabling the fault location function
  2. Selecting the transformation types
  3. Calculating the measurement conditions
  4. Setting the window
  5. Setting the frequency range and the number of points
  6. Setting the velocity factor
  7. Setting the cable loss
  8. Setting the display range
  9. Calibrate the analyzer
  10. Connect the cable under test

- Fault Location Measurement Setup using the VBA Utility Program

  The VBA utility program which facilitates measurement setup is available.

Other topics about Fault Location Analysis

Enabling the fault location function

1. Press **Channel Next** (or **Channel Prev**) and **Trace Next** (or **Trace Prev**) to activate a trace for which you want to use the conversion type.
2. Press **Analysis** > **Fault Location** to display the Fault Location menu.
3. Press **Fault Location** to enable the conversion feature (ON).

   NOTE To enable the conversion feature, the following conditions must be met. Otherwise, an error occurs.

   - The sweep type is linear sweep.
   - The number of measurement points is 3 or more.

Selecting the transformation type

Select the conversion type. The E5061B simulates the response from the DUT of two types of stimulus signals: impulse signal and step signal. The impulse signal is a pulse-shaped signal in which the voltage rises from 0 to a certain value and returns to 0 again. The pulse width depends on the frequency sweep range. The step signal is a signal in which the voltage rises from 0 to a certain value. The rise time depends on the maximum frequency within the frequency sweep range.

1. Press **Channel Next** (or **Channel Prev**) and **Trace Next** (or **Trace Prev**) to activate a trace for which you want to set the conversion type.
2. Press **Analysis** > **Fault Location** to display the Fault Location menu.
3. Press **Type** and then select softkeys to specify the **type**.

Calculating Measurement Conditions

To use the transformation function efficiently, you need to make the following two settings appropriately.

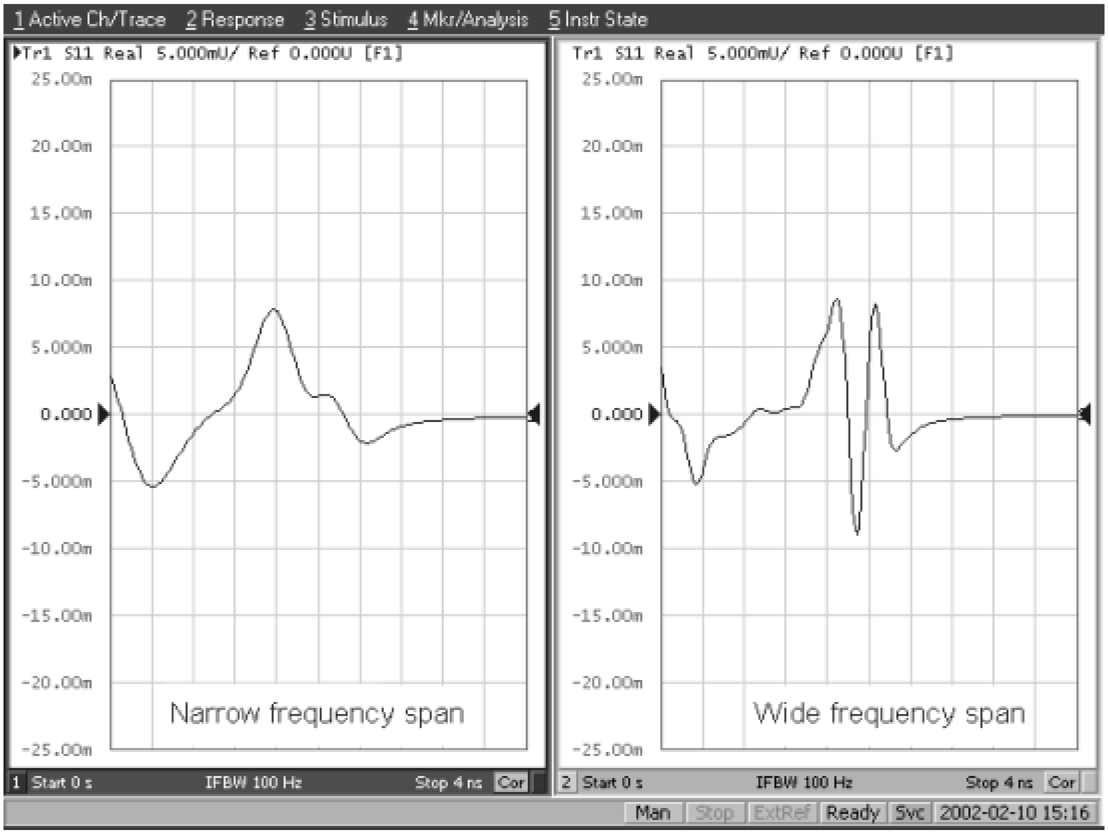
- Window
- Sweep conditions: frequency range and number of points

> NOTE The VBA utility program for the fault location measurement, calculates and sets up frequency sweep range to get the highest resolution available for the required display distance range. See Fault Location Measurement Setup using the VBA Utility Program.

Effect of frequency sweep range on response resolution

The following figure shows an example when measuring the same cable while changing the sweep span. When measured in a narrower sweep range, the overlap between 2 peaks is larger than when measured in a wider sweep range. By performing measurement in a wider sweep range, adjacent peaks can be clearly separated, which means that the response resolution is smaller.

Effect of frequency sweep range on resolution

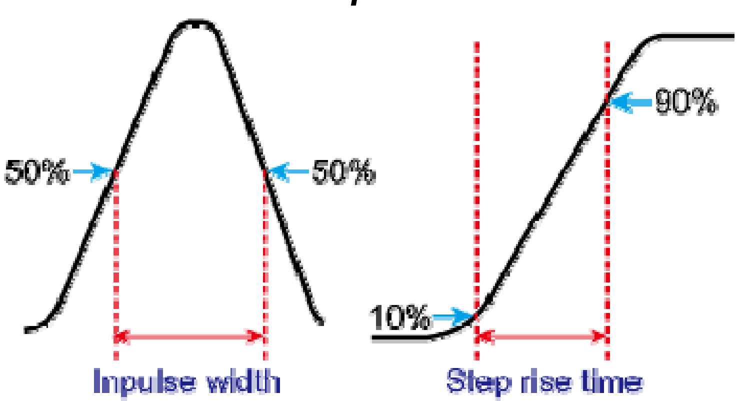


e5070aue208

The sweep range affects the width of the impulse signal and the rise time of the step signal. The width of the impulse signal and the rise time of the step signal are inversely proportional to the sweep range. Therefore, the wider the sweep range is, the shorter these times are.

The resolution is equal to the width defined at the point of 50 % of the impulse signal or the rise time defined at the points of 10 % and 90 % of the step signal.

Definition of the impulse width and the step rise time



e5071c418

Effect of the window function on the response resolution

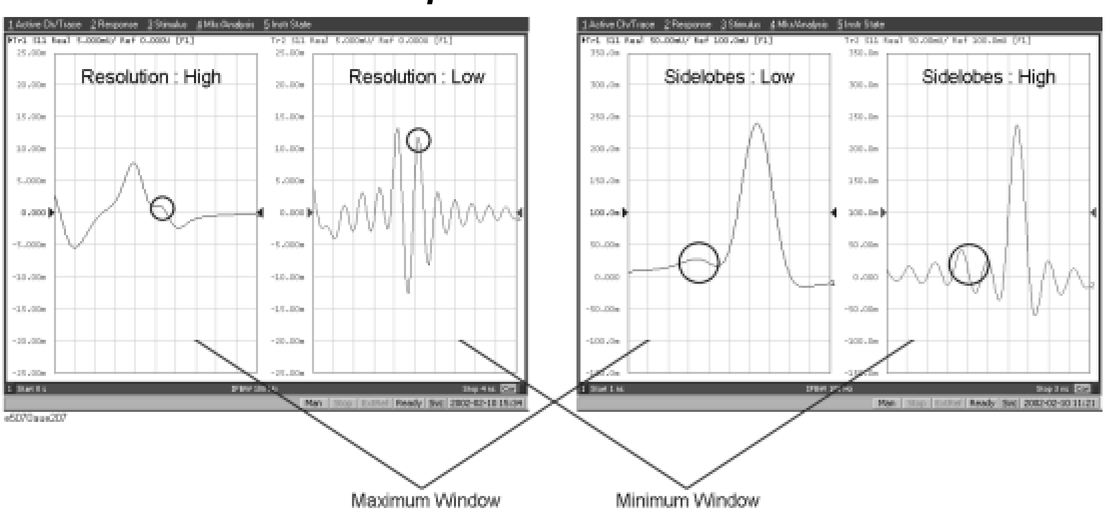
Lowering the sidelobe level with the window function elongates the width of the impulse signal and the rise time of the step signal. As described in Effect of frequency sweep range on response resolution, because the response resolution is equal to the width of the impulse signal and the rise time of the step signal, lowering the sidelobe level enlarges the response resolution. The following table shows the relation between the approximate response resolution and the window setting.

The shape of window and response resolution

Window	Low pass step	Low pass impulse	Band pass
Minimum	0.45/stop frequency	0.60/stop frequency	1.20/frequency span
Normal	0.99/stop frequency	0.98/stop frequency	1.95/frequency span
Maximum	1.46/stop frequency	1.39/stop frequency	2.78/frequency span

The following figure shows how the response changes when changing the window shape. You can see that, if the magnitudes of adjacent peaks are comparable, you need to make the resolution higher and, if they differ significantly, you need to set the window so that smaller peaks with lower sidelobes appear.

Effect of window on response resolution



Effect of the transformation type on the response resolution

Although both transformation types, band pass and low pass impulse, simulate the response of the impulse signal, the impulse width in the low pass impulse mode is half the width in the band pass mode. Therefore, the resolution is better in the low pass mode. If the DUT can be measured in the low pass mode, response data with better resolution is obtained in the low pass mode.

Measurement range

In the fault location function, the measurement range means the range within which the response can be measured without repetition. The repetition of the response occurs because measurement in frequency domain is performed discretely instead of continuously. The measurement range is inversely proportional to the frequency difference between adjacent measurement points.

The frequency difference between measurement points ΔF is expressed as follows using the span of the sweep frequency F_{span} and the number of points N_{meas}.

$$\Delta F = \frac{F_{span}}{N_{meas} - 1}$$

Therefore, the measurement range is proportional to (the number of points- 1) and inversely proportional to the span of the sweep range. To enlarge the measurement range, use one of the following methods:

- Increase the number of points.
- Narrow the span of the sweep range.

NOTE When you change the above settings after performing calibration, you need to perform calibration again.

The sweep range is expressed as time or distance. The time of the measurement range T_{span} is as follows:

$T_{span} = 1/\Delta F$

The distance of the measurement range L_{span} is expressed as follows using the velocity factor V and the speed of light in a vacuum C (3×10^8 m/s).

$L_{span} = V_c/\Delta F$

NOTE The maximum length of the DUT that can be measured in the transmission measurement is L_{span}. On the other hand, in the reflection measurement, because the signal goes and returns, it is 1/2 of L_{span}.

The velocity factor varies depending on the material through which the signal propagates. For polyethylene, it is 0.66; for Teflon, it is 0.7.

The change of the setting and the change of the response

The following table shows the effect of the change of the measurement conditions on the response resolution and the measurement range.

Change of setting	Response resolution	Measurement range	Sidelobe
Widen the sweep range	Becomes smaller	Becomes narrower	Does not change
Sets the window type to maximum	Becomes larger	Does not change	Becomes lower
Increase the number of points	Does not change	Becomes wider	Does not change

Setting Window

Because the E5061B transforms data within a finite frequency domain to data in distance or time domain, unnatural change of data at the end points within the frequency domain occurs. For this reason, the following phenomena occur.

- The width of the impulse signal and the rise time of the step signal

  The time width occurs in the impulse signal and the rise time occurs in the step signal.
- Sidelobe

  Sidelobes (small peaks around the maximum peak) occur in the impulse signal and the step signal. Ringing occurs on the trace due to sidelobes, which reduces the dynamic range.

By using the window function, you can lower the level of sidelobes. However, the width of the impulse and the rise time of the step become larger as a penalty. You can select from 3 types of windows: maximum, normal, and minimum. The following table shows the relationship between the window and the sidelobe/impulse width.

Characteristics of window

Window	Sidelobe level of	Width of the impulse (50%	Sidelobe level of	Rise time of the step

	the impulse signal	in low pass mode)	the step signal	signal (10 - 90%)
Minimum	-13 dB	0.60/frequency span	-21 dB	0.45/frequency span
Normal	-44 dB	0.98/frequency span	-60 dB	0.99/frequency span
Maximum	-75 dB	1.39/frequency span	-70 dB	1.48/frequency span

The window function is available only when the response in time domain is displayed. It does not have any effect when the response in frequency domain is displayed.

Procedure

1. Press **Channel Next** (or **Channel Prev**) and **Trace Next** (or **Trace Prev**) to activate a trace of which you want to set the window.
2. Press **Analysis** > **Fault Location** to display the Fault Location menu.
3. Press **Window** and then select a **window type**.

   NOTE By specifying Kaiser Beta, Impulse Width, or Step Rise, you can specify a window that is not classified into the three window types. When you specify a window type, these values are set automatically.

Setting Frequency Range and Number of Points

Set the sweep range and the number of points.

Procedure

1. Press **Channel Next** (or **Channel Prev**) to activate a channel you want to set.

   The frequency range and the number of points are common to all the traces in the channel. If you want to use different settings, use another channel.
2. Press **Sweep Setup** > **Sweep Type** > **Lin Freq** to set the sweep type to linear sweep.

   When the sweep type is set to other than the linear sweep, the fault location feature is not available.
3. Use the **Start**/**Stop** (or **Center**/**Span**) to set the sweep range.

4. Press **Sweep Setup** > **Points** and enter the number of measurement points in the data entry bar in the upper part of the screen.
5. When performing measurement in the low pass mode, press **Analysis** > **Fault Location** > **Set Freq Low Pass** to adjust the frequency range so that it is appropriate for the low pass mode. The start/stop frequencies are set as shown below.

Condition of the stop frequency	Frequency setting
> 5 Hz × the number of points	Start frequency = stop frequency/number of points
< 5 Hz × the number of points	Start frequency = 5 Hz Stop frequency = 5 Hz × number of points

NOTE If the above condition is met, the Set Freq Low Pass softkey is displayed in gray.

Setting the Velocity Factor

NOTE The velocity factor setting affects the cable loss setting and the display range setting. Thus it is recommended to set the velocity factor prior to the cable loss and display range.

Procedure

1. Press **Channel Next** (or **Channel Prev**) and **Trace Next** (or **Trace Prev**) to activate a trace of which you want to set the cable loss value.
2. Press **Cal** > **Velocity** (need to scroll to display this softkey).
3. Enter the value in the data entry bar in the upper part of the screen.

Setting the cable loss

Procedure

1. Press **Channel Next** (or **Channel Prev**) and **Trace Next** (or **Trace Prev**) to activate a trace of which you want to set the cable loss value.
2. Press **Analysis** > **Fault Location** > **Cable Loss**.
3. Enter the cable loss value in the data entry bar in the upper part of the screen. The unit differs depending on the display unit: dB/us for the display unit of time (second), dB/100 m for distance (m), and dB/100 Ft for distance (Ft). If the display unit is changed after entry, the cable loss value also changes appropriately for the display unit.

Setting Display Range

Set the range displayed on the graph. The displayed range can be set not only by time but also by distance. The number of response points displayed on the graph is the same as the number of points regardless of the response resolution. Note that, for reflection measurement, the E5061B lets you set the values on the horizontal axis for one-way data or round-trip data.

Procedure

1. Press **Channel Next** (or **Channel Prev**) and **Trace Next** (or **Trace Prev**) to activate a trace of which you want to set the display range.
2. Press **Analysis** > **Fault Location** > **Unit**.
3. Select a unit to set the display range from the following. The unit selected here determines the type of display: in time (**Seconds**) or in distance (**Meters**/**Feet**).
4. Use the **Start**/**Stop** (or **Center**/**Span**) to enter the display range in the data entry bar in the upper part of the screen. The data entry bar displays the distance (time when setting distance) corresponding to the set time (or set distance) next to the setting value.

   NOTE You cannot use the stimulus setting hardkeys (**Start**/**Stop**/**Center**/**Span**) to set the display range.

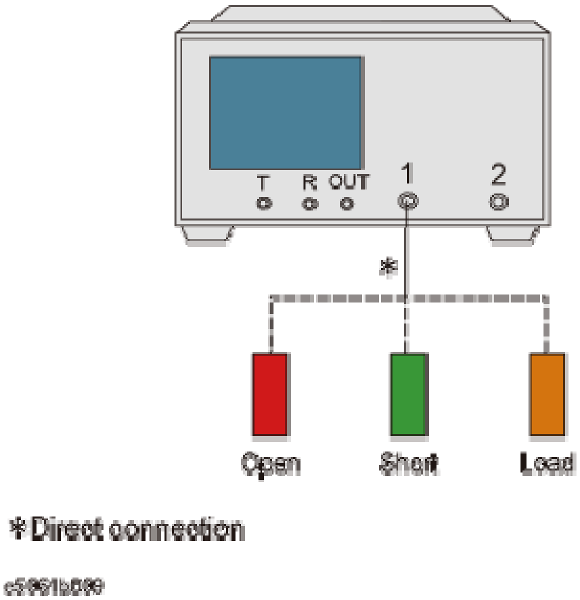
5. Press **Reflection Type** to select the type of the values on the horizontal axis in reflection measurement from one-way or round-trip.

Calibrate the Analyzer

In practical, a calibration should be done at the measurement reference plane using open, short, and load calibration standards to correct the instrument and optimize accuracy. Refer to calibration procedures.

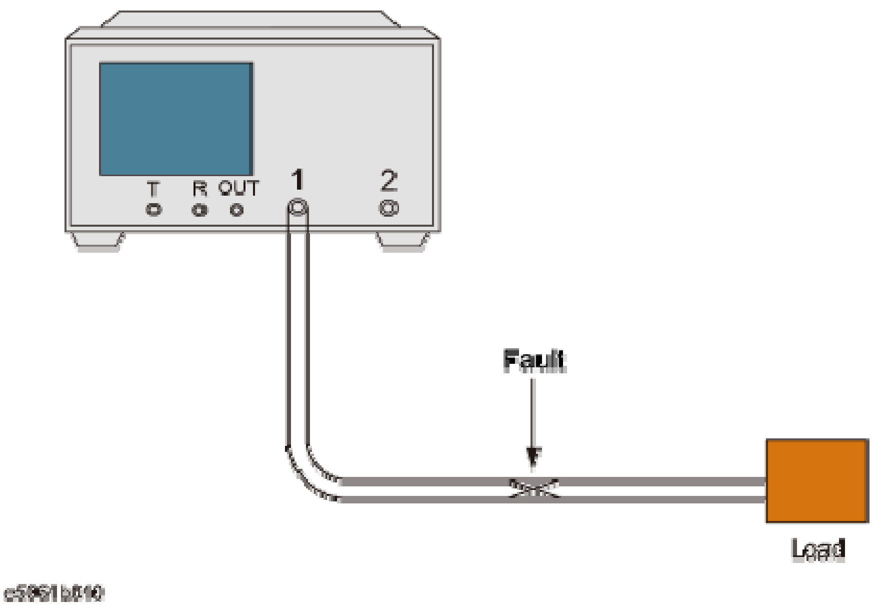
NOTE Most fault location measurements are made by connecting the cable under test directly to the analyzer's Port 1 test port. In this case the measurement reference plane would be the analyzer's port and you would connect calibration standards to the test port as shown in the following figure. Fault location measurements may also be made using a test lead cable. If this is the case, the measurement reference plane would be the end of the test lead cable, and calibration standards would be connected to the end of the test lead cable.

Calibrate the Instrument



Connect the Cable Under Test

The basic equipment setup for fault location measurements is shown below.



Fault Location Measurement Setup using the VBA Utility Program

The E5061B provides a macro program called **flt_util.vba**, which facilitates the measurement setup for fault location analysis. The utility program calculates and sets up frequency sweep range to get the highest resolution available for the required display distance range. (See Calculating Measurement Conditions for more information on frequency range calculation.) The utility program sets up following measurement conditions as well as the frequency sweep range so that you can start fault location analysis from the preset condition by using the utility program:

- Sweep type - Linear Frequency
- Unit - meter
- Reflection Type - One Way
- Fault Location- ON

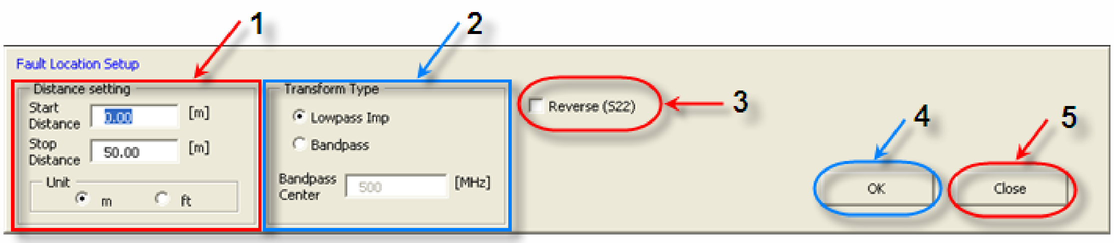
You need to set up the following parameters manually according to your measurement requirements:

- Number of points
- Cable loss
- Velocity factor
- Window

Procedure

1. Press **Channel Next** (or **Channel Prev**) and **Trace Next** (or **Trace Prev**) to activate a trace for which you want to perform fault location analysis.
2. Press **Macro Setup** > **Load Project**.
3. From the open dialog box, select the VBA project file **D:\Agilent\flt_util.vba**, and press **Open**.
4. Press **Select Macro** > **Module1 main** to execute the macro program.
5. The Fault Location Setup window appears in the lower part of the screen. Following procedures is performed with this setup window.

Fault Location Setup Window



1. Select the unit of distance (meter or feet), then enter the start and stop distance values.
2. Select the transformation type: Lowpass impulse or Bandpass. In the case of bandpass mode, enter the center frequency.

NOTE Center frequency can be from 1.3 MHz to the analyzer's highest frequency minus 1 MHz.

3. If you measure S22 (instead of S11) check the Reverse (S22) check box.
4. Press **OK** to setup the analyzer using the parameters entered in the setup window.

   NOTE The velocity factor and number of points need to be properly set before this step because these parameters are used to calculate the frequency sweep range. If you change these values after this step, the analyzer setup can be invalid.

5. Press **Close**, once you have completed the setup, to close the fault location setup window.

Deleting Unnecessary Data in Fault Location (Gating)

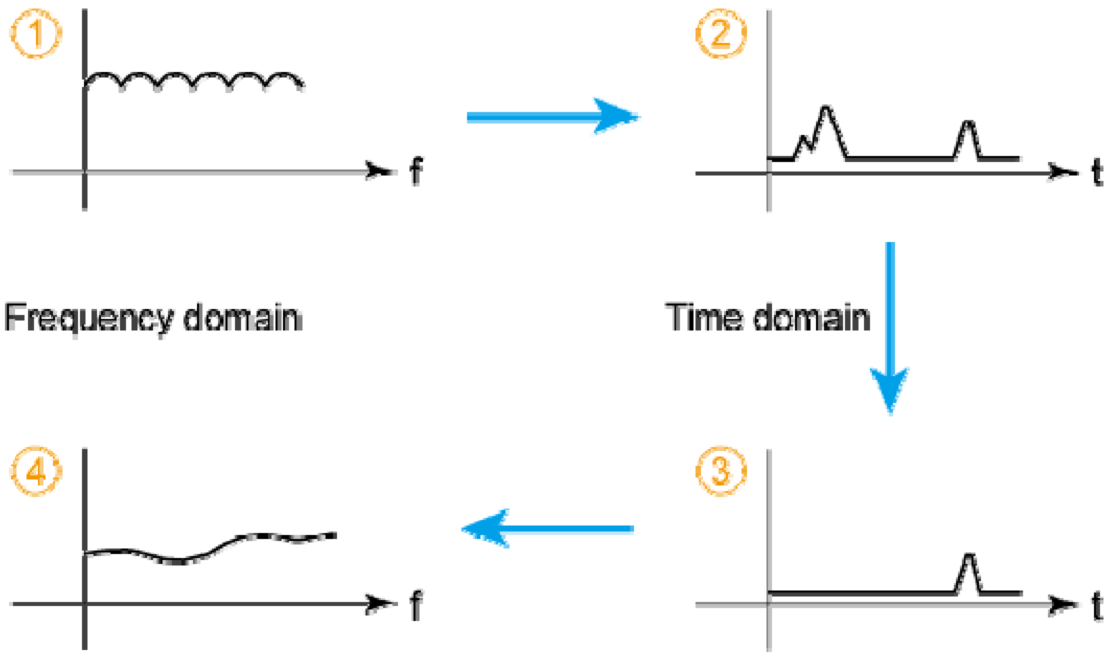
- Measurement Flow
- Setting Gate Type
- Setting Gate Shape
- Setting Unit
- Setting Gate Range
- Enabling Gating Function

Other topics about Time Domain Analysis

Measurement Flow

Item	Description
1. Measurement in frequency domain	Executes measurement in frequency domain
2. Transformation to time domain	Enables transformation function and transforms measurement data to data in time domain
3. Setting the gate	Makes the following settings of the gate to select the necessary domain: • Gate type • Gate shape • Gate range
4. Transformation back to frequency domain	Disables transformation function and displays response in frequency domain corresponding to the data selected with the gate

The following figure shows the change in the waveform at each step of the flow.



e5071c414

Setting Gate Type

The E5061B lets you choose from the following two gate types:

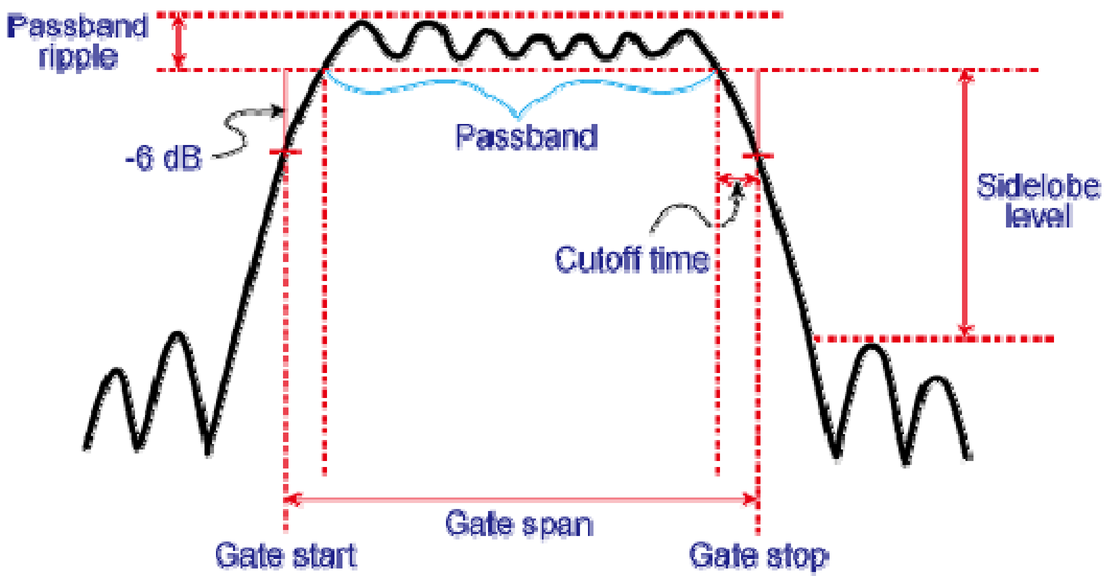
Gate type	Description
Band pass	Deletes response outside the gate range
Notch	Deletes response inside the gate range

Operational procedure

1. Press **Channel Next** (or **Channel Prev**) and **Trace Next** (or **Trace Prev**) to activate the trace of which you want to set the gate type.
2. Press **Analysis** > **Gating**.
3. Click **Type** to toggle between band pass (**Bandpass**) and notch (**Notch**).

Setting Gate Shape

The gate is a filter whose shape looks like a band pass filter. There are several parameters that indicate the gate shape. The following figure shows the definition of the gate shape parameters.



e5071c416

The following table compares the characteristics according to the gate shape. When the shape is "minimum," the cutoff time is shorter and the response is deleted abruptly, but the sidelobe level and band pass ripples become larger. When it is "maximum," cutoff is gentler, but the sidelobe level and the band pass ripple become smaller. The minimum gate span in the following table is the minimum gate range you can set. This value is defined as the minimum gate span necessary for the existence of the pass band and is equal to 2 times the cutoff time.

Gate shape	Passband ripple	Sidelobe level	Cutoff time	Minimum gate span
Minimum	± 0.13 dB	- 48 dB	1.4/frequency span	2.8/frequency span
Normal	± 0.01 dB	- 68 dB	2.8/frequency span	5.6/frequency span
Wide	± 0.01 dB	- 57 dB	4.4/frequency span	8.8/frequency span
Maximum	± 0.01 dB	- 70 dB	12.7/frequency span	25.4/frequency span

Procedure

1. Press **Channel Next** (or **Channel Prev**) and **Trace Next** (or **Trace Prev**) to activate a trace of which you want to set the gate shape.
2. Press **Analysis** > **Gating**.
3. Press **Shape** and then select the gate shape.

Setting Unit

Setting Gate Range

Specify the gate range in time. The ends of the range are defined as the -6 dB attenuation points shown in the figure above. You can set the gate range by specifying the start and stop times or the center and span. The E5061B has the following limitations on the gate range you can set.

Lower limit	$-T_{span}$
Upper limit	T_{span}

T_{span} is the measurement range expressed in time obtained in Measurement range.

Procedure

1. Press **Channel Next** (or **Channel Prev**) and **Trace Next** (or **Trace Prev**) to activate the trace of which you want to set the gate range.
2. Press **Analysis** > **Gating**.
3. Press each of the **Start**/**Stop** (or **Center**/**Span**) softkeys to specify the gate range. Distance corresponding to the setting time is displayed at the side of the set value area in the data entry bar. The displayed distance is a value that takes the velocity factor into consideration.

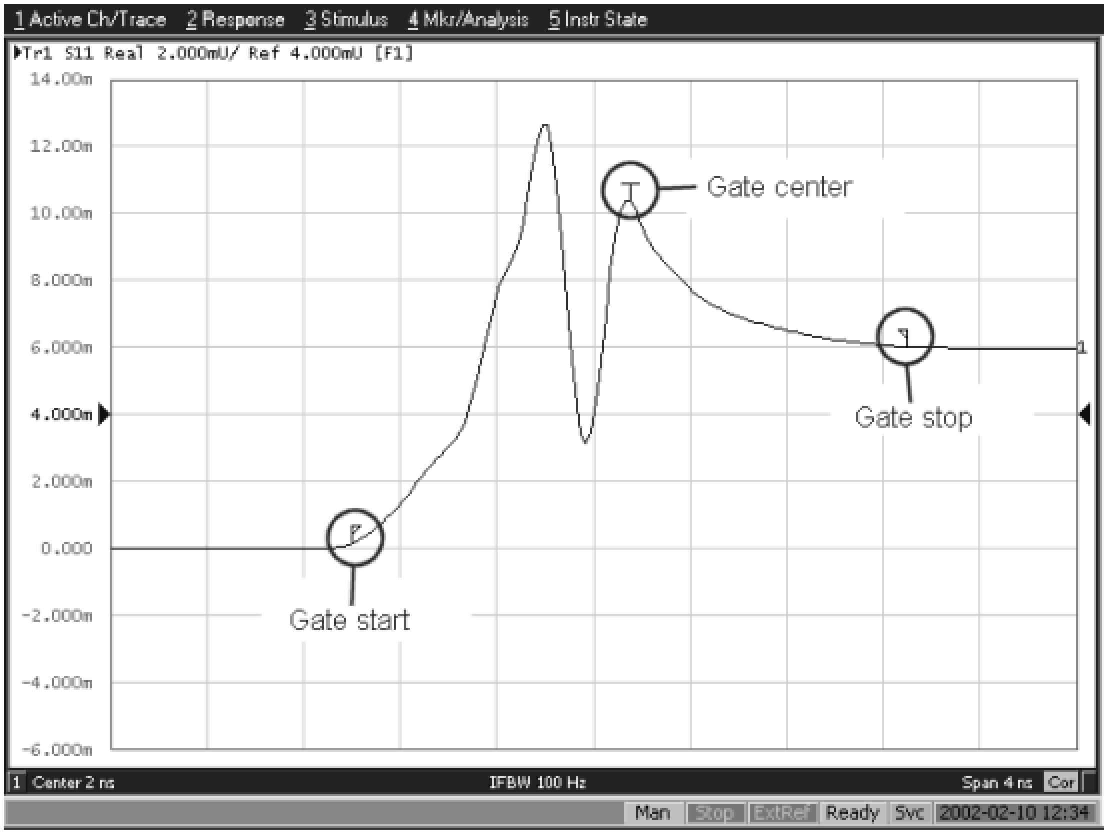
   NOTE You can not set this by **Start**/**Stop**/**Center**/**Span** hardkeys. The hardkeys are dedicated to setting the sweep range.

   NOTE You can set the center and span by dragging and dropping flags indicating the gate range.

Enabling Gating Function

When you enable the gating function, data within the specified range is deleted. When the transformation function is enabled, the flags indicating the gate range is displayed as shown in the following figure.

NOTE In the figure, the gate type is set to band pass. When it is set to notch, the directions of the flags indicating the ends of the gate range are reversed.



e5070aue205

Procedure

1. Press **Channel Next** (or **Channel Prev**) and **Trace Next** (or **Trace Prev**) to activate the trace of which you want to use the gate function.
2. Press **Analysis** > **Gating**.
3. Use **Gating** to enable (**ON**) the gate function.

Structural Return Loss Measurement

Structural Return Loss Measurement

- Theory
- Cable Preparation
- Making SRL Measurement

Other topics about Option 010 Time Domain/Fault Location Analysis

Theory

The SRL feature is designed to measure cable impedance and structural return loss. Cable impedance is the ratio of voltage to current of a signal traveling in one direction down the cable. Structural return loss is the ratio of incident signal to reflected signal in a cable, referenced to the cable's impedance.

The network analyzer uses a synthesized RF signal source to produce an incident signal as a stimulus. A reflection measurement is made and then used to compute the cable impedance. The structural return loss measurement is displayed referenced to the measured cable impedance.

For CATV cable, the cable is measured from 5 MHz to 1000 MHz at narrow frequency resolutions down to 125 kHz. The analyzer, with a furnished VBA utility program, automatically scans the cable, then reports the worst-case responses.

The following items are described in this topic.

- Cable Impedance
- SRL and Periodic Cable Faults
- SRL and Discrete Cable Faults
- Techniques for Removing Connector Effects
- Measurement Uncertainties

Other topics about Structural Return Loss Measurement

Cable Impedance

The analyzer automatically computes the cable impedance (Z). However, if you wish to turn OFF this "auto Z" function and input your own value of impedance, you can. See Connector Model for Short Cables.

In coaxial cable, the value of the impedance depends upon the ratio of the inner and outer conductor diameters, and the dielectric constant of the material between the inner and outer conductors. The cable impedance is also affected by changes in conductivity. These changes are a natural consequence of RF currents that flow near the surface of a conductor. This effect is known as the "skin effect." Also, the construction of the cable can change along the length of the cable, with differences in conductor thickness, dielectric material and outer conductor diameter changing due to limitations in manufacturing. Thus the cable impedance may vary along the length of the cable.

The extent to which manufacturing imperfections degrade cable performance is characterized by a specification called structural return loss (SRL). SRL is the ratio of incident signal to reflected signal in a cable. This definition implies a known incident and reflected signal. In practice, the SRL is loosely defined as the reflection coefficient of a cable referenced to the cable's impedance. The reflection seen at the input of a cable, which contributes to SRL, is the sum of all the tiny reflections along the length of the cable. In terms of cable impedance, the SRL can be defined mathematically as:

Equation 1

$$Z_{SRL}(\omega) = \frac{Z_{in}(\omega) - Z_{cable}}{Z_{in}(\omega) + Z_{cable}}$$

Z_{in} is the impedance seen at the input of the cable, and Z_{cable} is the nominal cable impedance.

Cable impedance is a specification that is defined only at a discrete point along the cable, and at a discrete frequency. However, when commonly referred to, the impedance of the cable is some average of the impedance over the frequency of interest. Structural return loss, on the other hand, is the cumulative result of reflections along a cable as seen from the input of the cable. The above definitions need to be expressed in a more rigorous form in order to apply a measurement methodology.

Defining Cable Impedance

Following are three common methods of defining cable impedance. Although all three methods may be commonly used in your industry, your network analyzer uses the third method (Z-average normalization) to define cable impedance.

Method 1

One definition of cable impedance is that impedance which results in minimum measured values for SRL reflections over the frequency of interest. This is equivalent to measuring a cable with a return loss bridge that can vary its reference impedance. The value of reference impedance that results in minimum reflection, where minimum must now be defined in some sense, is the cable impedance. Mathematically, this is equivalent to finding a cable impedance such that:

Equation 2

$$\frac{\partial[\bar{\rho}(\omega, Z_{cable})]}{\partial(Z_{cable})} = 0$$

where ρ is some mean reflection coefficient. Thus, cable impedance and SRL are somewhat inter-related: the value of SRL depends upon the cable impedance, and the cable impedance is chosen to give a minimum SRL value.

Method 2

An alternate definition of cable impedance is the average impedance presented at the input of the cable over a desired span. This can be represented as:

Equation 3

$$Z_{avg} = \frac{\int_{F_{min}}^{F_{max}} Z_{in}(F)\, dF}{(F_{max} - F_{min})}$$

The value found for Z_{avg} would be substituted for Z_{cable} in Equation 1 to obtain the structural return loss from the cable impedance measurement.

Method 3 (Z-average normalization)

The mathematics for the Z-average normalization as performed by the analyzer are shown below.

Equation 4

$$Z_{in}(\omega) = Z_0 \times \frac{(1 + \rho(\omega))}{(1 - \rho(\omega))}$$

Z_0 = system impedance, 50 Ω.

Equation 5

$$Z_{cable} = \frac{\sum_{n=1}^{N} \left| Z_{in}(\omega_n) \right|}{N}$$

Equation 6

$$\rho_{SRL}(\omega) = \frac{Z_{in}(\omega) - Z_{cable}}{Z_{in}(\omega) + Z_{cable}}$$

In Equation 4, ρ(ω) is the reflection coefficient from the analyzer measured at each frequency and Z_{in}(ω) is the impedance of the cable for that measured reflection coefficient.

1. The calculation of Z_{cable}, described in Equation 5, is the Z-average impedance of the cable over the number of frequency points (N). The

default frequency range is approximately 5 MHz to 200 MHz. This frequency range is chosen because mismatch effect of the input connector is small. High quality connectors must be used if the average impedance is calculated over a wider span. The frequency range for this calculation can be modified by using the **Z Cutoff Freq.** softkey in the connector model menu to change the cable impedance cutoff frequency.

Equation 6 is the structural return loss for the cable. This calculation can be done by the analyzer or an external computer.

SRL and Periodic Cable Faults

SRL is the measurement of the reflection of incident energy that is caused by imperfections or disturbances (bumps) in the cable which are distributed throughout the cable length. These bumps may take in the form of a small dent, or a change in diameter of the cable. These bumps are caused by periodic effect on the cable during the manufacturing process. For example, consider a turn-around wheel with a rough spot on a bearing. The rough spot can cause a slight tug for each rotation of the wheel. As the cable is passed around the wheel, a small imperfection can be created periodically corresponding to the tug from the bad bearing.

Each of these small variations within the cable causes a small amount of energy to reflect back to the source due to the non-uniformity of the cable diameter. Each bump reflects so little energy that it is too small to observe with fault location techniques. However, reflections from the individual bumps can sum up and reflect enough energy to be detected as SRL. As the bumps get larger and larger, or as more of them are present, the SRL values will also increase. The energy reflected by these bumps can appear in the return loss measurement as a reflection spike at the frequency that corresponds to the spacing of the bumps. The spacing between the bumps is one half the wavelength of the reflection spike and is described by equations 7 and 8.

Equation 7

$$wavelength \approx \frac{c}{f}$$

$$c = \text{speed of light} \qquad f = \text{frequency}$$

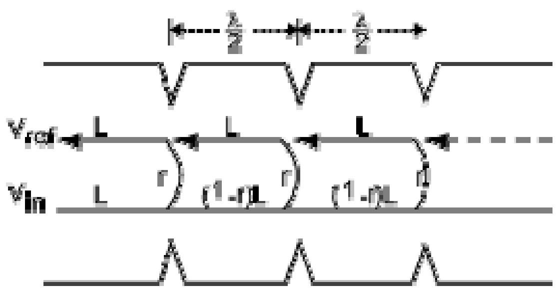
Equation 8

$$\frac{wavelength}{2} = \text{spacing between the bump}$$

The wavelength/2 spacing corresponds to the frequency at which down and back reflections add coherently (in-phase). The reflections produce a very narrow response on the analyzer display that is directly related to the spacing of the bumps. The amount of reflected energy is observed as return loss. When this return loss measurement is normalized to the cable impedance, the return loss becomes structural return loss.

The following figure diagrams reflections from bumps in a cable. We can combine the energy reflected by each bump in a cable and make a few basic assumptions, to mathematically describe SRL with the series shown in Equation 9.

Periodic Bumps in a Cable



e5061b002

Equation 9

$$V_{ref} = [V_{in}L\Gamma L] + [V_{in}L(1-\Gamma)L\Gamma LL] + [V_{in}L(1-\Gamma)L(1-\Gamma)L\Gamma LLL] + \ldots$$

V_{ref} = reflected Voltage

V_{in} = incident Voltage

L = cable loss

Γ = reflection coefficient of the bumps

The bumps are assumed to be uniform in reflection and spaced by a wavelength/2 separation.

The series may be reduced to a simple form to leave us with the relationship shown in Equation 10. The term L is a function of the loss of the cable at a specific frequency and the wavelength at that frequency.

Equation 10

$$\mathrm{SRL} = \frac{Vref}{Vin} = \Gamma\left(\frac{\mathrm{L}^2}{1-\mathrm{L}^2}\right)$$

The term $(L^2)/(1-L^2)$ can be thought of as the number of bumps that are contributing to SRL. It represents a balance between the contribution of loss in a single bump and further bumps in the cable for the specified frequency and cable loss. Calculate the distance into the cable by multiplying the term $(L^2)/(1-L^2)$ by the distance between bumps.

The following table illustrates some calculated values for a typical trunk cable. From the table, bumps spaced 1.5 meters apart out to 307 meters contributes to SRL.

SRL Equation Constant

Frequency	Spacing (λ/2) m	Loss (dB)/m	dB/bump	bumps $(L^2)/(1-L^2)$	Distance (m)
100 MHz	1.5	0.014	-0.02	205	307
500 MHz	0.3	0.033	-0.01	433	129
1 GHz	0.15	0.05	-0.0075	554	83

How to Use Table

Refer to Periodic Bumps in a Cable and Equation 10 for the following discussion.

- Γ = the reflection coefficient of each bump ($V_{reflected}/V_{incident}$)
- L = the cable loss between bumps ($V_{transmitted}/V_{incident}$)
- The distance between bumps equals λ/2 (1/2 wavelength).

Typical values:

- Γ << 1
- L ≤ 1 for low loss cable

Derivation of L

In Equation 10, L is the cable loss for a 1/2 wavelength length of cable, expressed in linear.

1. Find the cable loss from a spec sheet. Cable loss is typically expressed in loss per foot.
2. Convert loss per foot to loss per meter.
3. Find the 1/2 wavelength in meters. This will be the spacing between bumps.
4. Multiply loss per meter × 1/2 wavelength to get dB loss per bump.

5. Convert dB loss per bump to linear.

Example

1. A spec sheet states that the cable loss spec at 300 MHz is 1 dB per 100 feet.
2. Convert loss per foot to loss per meter:
   1 dB/100 ft ≈ 1 dB/30 m ≈ 0.033 dB/m
   This is the Loss (dB)/m column in SRL Equation Constant.
3. Find the 1/2 wavelength in meters:
   1/2 wavelength at 300 MHz ≈ 0.5 meters
   This is the Spacing (λ/2)m column in SRL Equation Constant.
4. Multiply loss/meter × 1/2 wavelength:
   0.033dB/meter × 0.5 meters = 0.0165 dB = L_{dB}
   This is the dB/bump column in SRL Equation Constant.
5. Convert L_{dB} (loss) in dB to linear:
   20 log (L_{dB}) = -0.0165 L = $10^{(-0.016/20)}$ = 0.998
6. $(L^2)/(1-L^2) \approx 262$
7. There are approximately 262 bumps contributing to SRL at 300 MHz.
8. 262 × 0.5 = 131. The distance into the cable for 262 bumps is 131 meters.

In actual cables, the reflections from the bumps and the spacing of the bumps may vary widely. The best case for a minimum SRL, is that the bumps are totally random and very small. Real world examples are somewhere in between the uniform bumps and scattered case. As the sizes of the bumps, their spacing, and the number of bumps vary within the manufacturing process, varying amounts of SRL are observed.

SRL and Discrete Cable Faults

In addition to a set of periodic bumps, a cable can also contain one or more discrete faults. For this discussion, discrete imperfections is referred to as “faults,” and periodic imperfections is referred to as "bumps."

Reflections from discrete faults within the cable also increases the level of SRL measured. The energy reflected from a fault sums with the energy reflected from the individual bumps and provide a higher reflection level at the measurement interface. Examining the cable for faults before the SRL measurement is a worthwhile procedure. The time required to perform the fault location measurement is small compared to the time spent in performing an SRL measurement scan.

A fault within the cable provides the same type of effect as a bad connector. If the fault is present within the end of the cable nearest to the analyzer, the effect is noticed throughout the entire frequency range. As the fault is located further into the cable, the cable attenuation reduces the effect at higher frequencies. The reflected energy travels further through the cable at lower frequencies where the cable attenuation per unit distance is lower.

Techniques for Removing Connector Effects

Connector Effects on SRL

To remove the unwanted effects of worn connectors, the SRL measurement uses a built-in connector model. The connector model consists of compensation for connector length and compensation for connector capacitance (connector C).

The "connector C" compensation emulates the C trim value of a variable impedance bridge.

The connector length is used to compensate for the effects of an electrically long connector and extends the calibration reference plane.

NOTE A calibration reference plane is established at the point where the short, open, and load standards are measured.

The analyzer can automatically measure the optimum values for your connector model, or you may enter them manually.

The default values for the connector model are 0.00 mm length, and 0.00 pF capacitance (no compensation).

When measuring spools of the cable, typically two connectors are used: the test-lead connector and the termination connector. (See the following figure.) These connectors provide the cable interface and are measured as part of the cable data.

Basic SRL Measurement Setup and Connections

Often, slight changes in the test-lead connector can cause significant changes in the values of structural return loss measured at high frequencies. This is because the reflection from a connector increases for high frequencies. In fact, the return loss of a test-lead connector can dominate the SRL response at frequencies above 500 MHz. This is where training, good measurement practices, and precision cable connectors are needed, especially for measurements up to 1 GHz. Precision connectors are required to provide repeatability over multiple connections. Slip-on connectors are used to provide rapid connections to the cables, but require careful attention in obtaining good measurement data. Repeatability of measurement data is directly affected by the connector's ability to provide a consistently good connection. This is the major cause of repeatability problems in SRL measurements.

Effects of the test-lead connector at the measurement interface are observed as a slope in the noise floor at higher frequencies.

By observing the SRL measurement display and slightly moving the connector, the effects of the connection can be observed at the higher frequencies. The test-lead connector should be positioned to obtain the lowest possible signal level and the flattest display versus frequency. The mechanical interface typically provides an increasing slope with frequency and flattens out as the connection is made better.

The termination connector may also affect the SRL measurement if the cable termination connector and load provide a significant amount of reflection and the cable is short enough. As longer lengths of cable are measured, the cable attenuation provides isolation from the termination on the far end. Use a fault location measurement technique to observe the reflection from the termination at the far end of the cable. If the termination is shown as a fault, the reflection from the terminating connector is contributing to the reflection from the cable. A more suitable termination is required or a longer section of cable must be measured. The cable must provide sufficient attenuation to remove the effects of the connector and load for a good SRL measurement. Performing a good measurement on a short length of cable is quite difficult and requires connectors with very low reflections to be effective.

Fixed Bridge with Connector Compensation

The analyzer employs the fixed-bridge method and instrument software to emulate the traditional variable-bridge method. Vector error correction is used to provide the most accurate measurements up to the calibration plane defined by the calibration standards. Additional corrections can also be used to minimize the effects of the test-lead connector on the measured SRL response.

The error corrections done for a fixed bridge can also include connector compensation. The fixed bridge method with connector compensation technique mathematically removes the effects of the test-lead connector by compensating the predicted connector response given by a connector model.

Shunt C Connector Model

One model that can be used for the cable connector is the shunt C connector model. With this model, the adjustment of the C value given in a variable impedance bridge can be emulated. The shunt C connector model assumes the discontinuity at the interface is abrupt and much smaller than a half wavelength of the highest frequency of measurement. With this assumption, the discontinuity can be modeled as a single-shunt twisted pair, where $C = C_0$ + second and third order terms.

Intuitively this is the right model to choose because the effect of a typical poor connector on structural return loss measurement is an upward sloping response, typically worst at the high frequencies.

Using a shunt C to model the connector, a value of the susceptance, -C, may be chosen by the network analyzer to cancel the equivalent C of the connector and mathematically minimize the effect of the connector on the response measurement.

The equations for computing structural return loss and the average cable impedance with capacitive compensation are described next.

Equation 11

$$Z'_{in}(\omega) = \frac{Z_{in}(\omega) \cdot \frac{1}{jwC}}{Z_{in}(\omega) + \frac{1}{jwC}}$$

Equation 12

$$Z'_{cable} = \frac{\sum \left| Z'_{in}(\omega) \right|}{N}$$

Equation 13

$$\rho'_{SRL}(\omega) = \frac{Z'_{in}(\omega) - Z'_{cable}}{Z'_{in}(\omega) + Z'_{cable}}$$

In Equation 11, Z_{in}(w) is calculated from the measured return loss as described in Equation 4, previously. The primed values are the new calculation values using the capacitive compensation. With these equations, the network analyzer can compute values for the cable impedance and mathematically compensate for the connector mismatch with a given value of C connector compensation.

Connector Length

The shunt C connector model can be improved with the addition of connector length. Connector length is used to compensate the phase shift caused by the electrical length within the connector. The calibration plane can be moved from one side of the cable connector to the other side, so that the shunt C is placed exactly at the discontinuity of the connector and cable under test.

Measurement Uncertainties

In any comparison of cable impedance or structural return loss data, it is important to understand the measurement uncertainty involved in each type of measurement. This is critical for manufacturers, who often use the most sophisticated techniques to reduce manufacturing guard bands. It is also important in field measurements that users choose the proper equipment for their needs, and understand the differences that can occur between manufacturers' data and field data. Also, note that measurement uncertainty is usually quoted as the worst-case result if the sources of error are at some maximum value. This is not the same as error in the measurement, but rather a way to determine measurement guard band, and to understand how closely to expect measurements to compare on objects measured on different systems.

The errors that can occur in a reflection measurement are reflection tracking (or frequency response), *T*, source match, Γ_M, and directivity, *D*. The total error in a measurement can be shown to be

Equation 14

$$\Gamma_{MEAS} = T \cdot \left[D + \frac{(\Gamma_{DUT})}{(1 - \Gamma_M \Gamma_{DUT})} \right]$$

where Γ_{DUT} is the reflection response of the DUT.

Error correction techniques can effectively remove the effects of tracking. Also, source match effects are small if Γ_{DUT} is small. This leaves directivity as the largest error term in the reflection measurement. The causes and effects of these error terms will be described for each of the measurement methodologies.

For variable bridge measurements, the directivity of the bridge is the major error term. One-port vector error correction reduces the effects of tracking and source match, and improves directivity. The directivity after error correction is set by the return loss of the precision load, specified to be better than 49 dB at 1 GHz. However, the directivity is only well known at the nominal impedance of the system, and the directivity at other impedances should be assumed to be that specified by the manufacturer. For best performance, the bridge should be connected directly to the cable connector, with no intervening cable in between.

The directivity of the bridge could be determined at impedances other than 75 ohms, by changing the impedance and measuring the resulting values. This can be done by changing the reference impedance to the new value, say 76 ohms, changing the bridge to that value, and measuring the impedance on a Smith chart display. The difference from exactly 75 ohms represents the directivity at that impedance.

For fixed bridge methods, the reflection port is often connected to the cable connector through a length of test lead. A one-port calibration is performed at the end of the test lead. The directivity is again set by the load, but any change in return loss of the test lead due to flexing degrades the directivity of the measurement system. In both fixed and variable bridge measurements, the repeatability and noise floor of the analyzer may limit the system measurement. A convenient way to determine the limitation of the measurement system is to perform a calibration, make the desired measurement, then re-connect the load to check the effective directivity. A very good result is better than -80 dB return loss of the load. Typically, flexure in the test leads, connector repeatability, or noise floor in the network analyzer limits the result to between -60 to -40 dB. If the result is better than -49 dB, then the system repeats better than the load specification for the best available 75 Ω loads. Thus, the effective directivity should be taken to be the load spec of -49 dB. It is possible to reduce this limitation by having loads certified for better return loss.

Measurement Uncertainty for Impedance Measurements

The fixed bridge method calculates the cable impedance by averaging the impedance of the cable over frequency. The variable bridge uses a reading of the impedance from the dial on the bridge. The directivity at any impedance can be determined, as stated earlier, but only to the limit of the return loss of the load, and the system repeatability.

Any connectors and adapters used to connect the test-lead cable to the cable under test can have a significant effect on the impedance measurement. With the variable bridge method, the operator determines the appropriate setting, taking into account the capacitive tuning adjustment. With the fixed bridge method, it is also possible to compensate somewhat for the connector. However, it is often the case that the cable impedance is determined by the low frequency response, up to perhaps 200 MHz to 500 MHz, where the connector mismatch effect is still small. The choice of frequency span to measure cable impedance can itself affect the value obtained for cable impedance. In general, as the connector return loss becomes worse, it has a greater effect on the resulting impedance measurement. The uncertainty caused by the connector is difficult to predict, but large errors could occur if the low frequency return loss is compromised to achieve better high frequency structural return loss.

Finally, note that since both methods average, in some way, the measurement over the entire frequency range, it is probable that the worst case error will never occur at all frequencies, and with the same phase. In fact, it is more likely that the errors will cancel to some extent in cable impedance measurements. Also, the loads that are used will invariably be somewhat better than specified, especially over the low frequency range.

Measurement Uncertainty for Structural Return Loss

The same factors that affect cable impedance - directivity, system and test lead stability, and cable connector mismatch - also affect structural return loss. However, since structural return loss is measured at all frequencies, it is much more likely that a worst case condition can occur at any one frequency. For that reason, the measurement uncertainty must include the full effect of the above listed errors.

Cable Preparation

Cable preparation (for slip-on connectors) can be critical for some SRL measurements, especially when measuring mainline cables with an SRL of -30 dB or lower. An improperly prepared cable can degrade the cable/connector response which may affect the measurement enough to make a "good" cable fail.

This section describes the most common cable preparation problems that should be avoided in order to obtain good measurements.

- Cable Preparation Problems
- Recommended Tools and Cables

Other topics about Fault Location Analysis

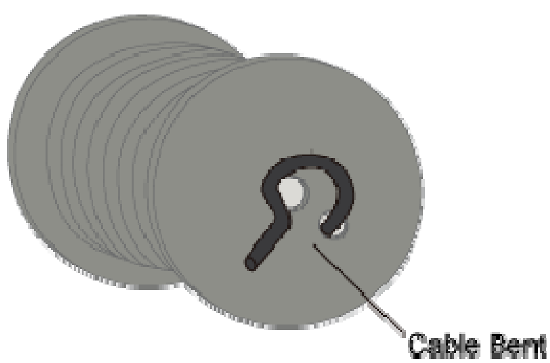
Cable Preparation Problems

Follow the preparation instructions provided by the connector manufacturer and take great care to avoid the following cable preparation problems:

- bent cable
- deformed cable
- contaminated dielectric
- damaged outer conductor
- non-flush cut

Bent Cable

Poor measurement results can occur if the cable is bent or kinked near the end of the spool. The bend should be removed before proceeding with the SRL measurement.

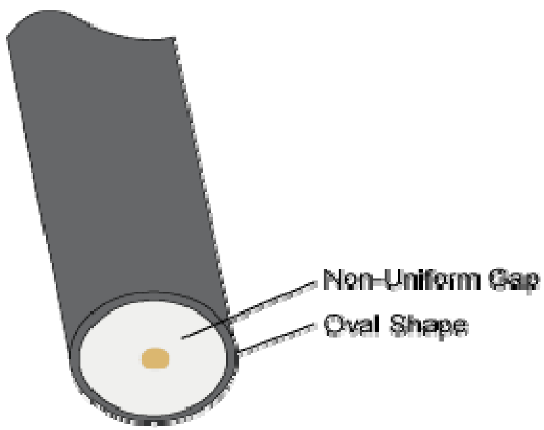


Bent/Kinked Cable

NOTE The built-in connector modeling will attempt to remove the effects of the connector. The connector response is shown at 0.0 ft. on the bottom trace. The extent to which the effects of the connector can be removed may depend on the quality of the cable preparation as well as the connector.

Deformed Cable

Compressing the dielectric (the gap) produces egg-shaped or oval deformations which can cause impedance mismatches and affect the quality of the connector model compensation. This can easily happen when using diagonal cutters to cut the cable.



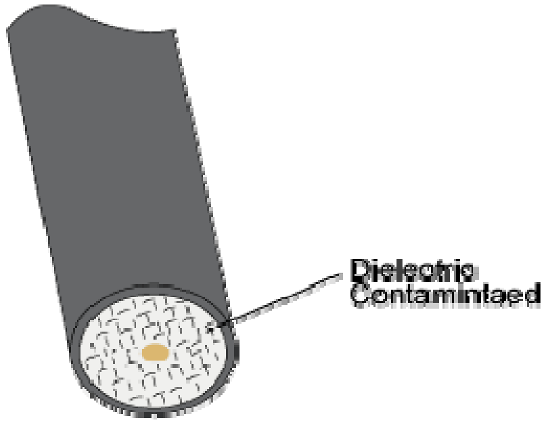
e5061b095

Deformed Cable (cut with diagonal cutters)

NOTE The built-in connector modeling will attempt to remove the effects of the connector. However, the modeling cannot remove the effects of the cable bend.

Contaminated Dielectric

When a cable is cut, contamination of the dielectric can occur from cuttings or shrapnel from the outer or inner conductor. This type of contamination can cause problems and change the connector model compensation needed.



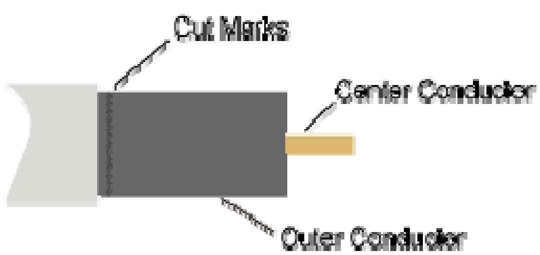
e5061b006

Contaminated Cable Dielectric

NOTE The built-in connector modeling will attempt to remove the effects of the connector. The extent to which the effects of the connector can be removed may depend on the quality of the cable preparation as well as the connector.

Damaged Outer Conductor

The outer conductor may be cut or dented when the outer insulation is removed. This can cause a close-in fault which cannot be compensated by the connector model.



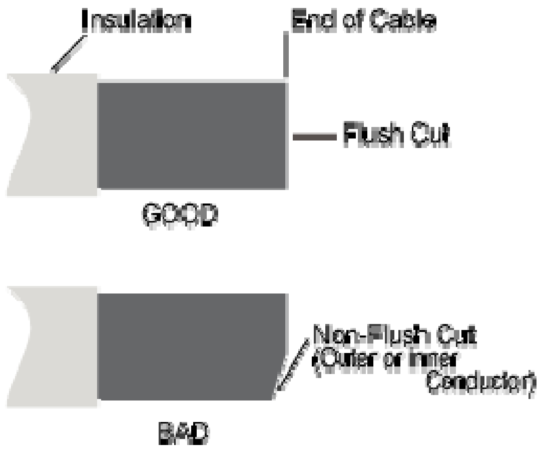
e5061b007

Scarred Outer Conductor of Cable

NOTE The built-in connector modeling will attempt to remove the effects of the connector at 0.0 ft. However, the modeling may not remove the effects of the outer conductor damage (which is a few inches into the cable).

Non-Flush Cut

Cables which require a flush cut, such as for GTC-XXX-TX-N ("Pogo") connectors, might not actually be cut in such a way. This can cause an inconsistent connection or poor repeatability of the SRL measurement.



e5061b008

Cable Cut Flush (good) and Non-Flush (bad)

NOTE The built-in connector modeling will attempt to remove the effects of the connector. The extent to which the effects of the connector can be removed may depend on the quality of the cable preparation as well as the connector.

Recommended Tools and Cables

Recommended Tools

For connectors such as the GTC-XXX-TX-GHZ-N ("GHZ") connector, cable prep tools similar to CableMatic Model SST-A (Ripley Company) are recommended.

Recommended Cables

The following table lists the recommended test lead cables for use in cable testing applications.

Cable Description	Part Number
75 Ω Type-N 10 ft. (m-m)	8120-6737
75 Ω Type-N 10 ft. (m-f)	8120-6740
75 Ω Type-N 15 ft. (m-m)	8120-6738
75 Ω Type-N 15 ft. (m-f)	8120-6741
75 Ω Type-N 30 ft. (m-m)	8120-6739
75 Ω Type-N 30 ft. (m-f)	8120-6742

Making SRL Measurement

A typical SRL measurement consists of the following steps:

1. Setting the sweep type, the sweep range, and the number of points
2. Enabling the SRL function.
3. Setting the average impedance.
4. Calibrate the analyzer.
5. Connect the cable under test
6. Determine the connector model.
7. Perform the SRL Cable Scan
8. Interpret the SRL Measurement

Other topics about Structural Return Loss Measurement

Setting Sweep Type, Sweep Range, and Number of Points

Set the sweep range and the number of points. If you perform the SRL cable scan, use the VBA utility program to set up the analyzer as described in SRL Cable Scan Setup using the VBA Utility Program instead of the following manual procedures.

NOTE The SRL utility program not only sets sweep conditions but also enables the SRL function and sets the average impedance described in the following sections.

Procedure

1. Press **Channel Next** (or **Channel Prev**) to activate a channel you want to set.

   NOTE The frequency range and the number of points are common to all the traces in the channel. If you want to use different settings, make them on another channel.

2. Press **Sweep Setup** > **Sweep Type** and select a sweep type with the softkeys.
3. Use the **Start**/**Stop** (or **Center**/**Span**) to set the sweep range.
4. Press **Sweep Setup** > **Points** and enter the number of measurement points in the data entry bar in the upper part of the screen.

Enabling SRL Function

NOTE For channels of which SRL is enabled, it affects the calculation of the reflection coefficient and does not affect the transmission coefficient.

Procedure

1. Press **Channel Next** (or **Channel Prev**) to activate the channel of which you want to enable the SRL feature.
2. Press **Analysis** > **SRL**.
3. Press **SRL** to enable the SRL feature (ON).

Setting Average Impedance

The E5061B lets you select manual entry or auto calculation for the average cable impedance.

Procedure

1. Press **Channel Next** (or **Channel Prev**) to activate a channel of which you want to set the average impedance.
2. Press **Analysis** > **SRL**.
3. When calculating the average impedance automatically (the preset value has been set automatically), press **Z Cutoff Freq.** to specify the cutoff frequency to calculate the average impedance. If **Z Cutoff Freq.** is not available, the instrument is in manual entry mode, so press **Auto Z** to turn it ON and specify **Z Cutoff Freq.** again.

   NOTE If you want to enter the average impedance value manually, set **Auto Z** to OFF to switch to manual entry mode, in which you can enter **Manual Z**. Press **Manual Z** and specify the average impedance value.

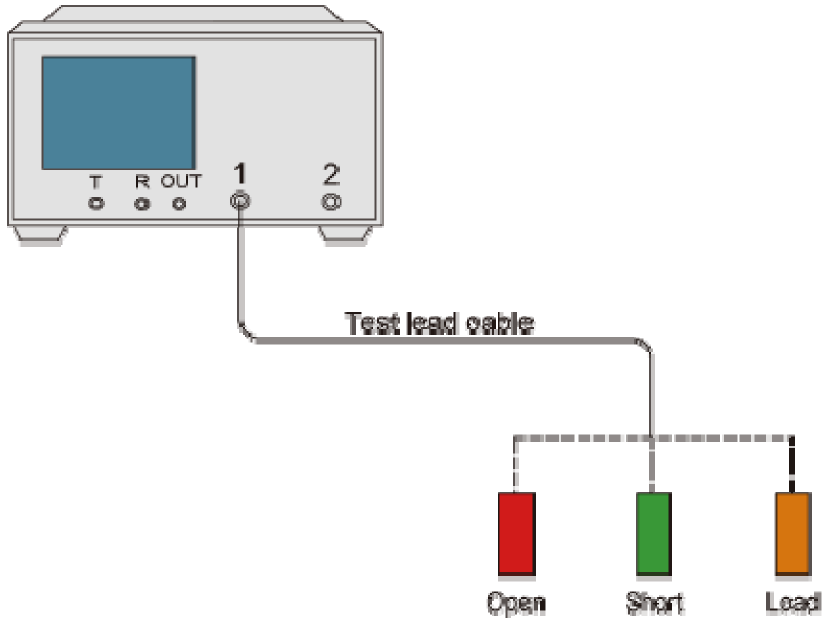
   Note that, if there is no measurement value from the start frequency to the cutoff frequency, the value for manual entry is used as the average impedance value even in mode to calculate the average impedance automatically.

Calibrate the Analyzer

In practical, a calibration should be done at the measurement reference plane using open, short, and load calibration standards. Refer to calibration procedures.

NOTE Most SRL measurements are made using a test lead cable. If this is the case, the measurement reference plane would be the end of the test lead cable as shown in the following figure. If you are testing cables by connecting them directly to the analyzer's test port, you should perform the calibration at the analyzer's port.

Calibrate the Instrument for an SRL Measurement



e5061b003

Verifying the Calibration

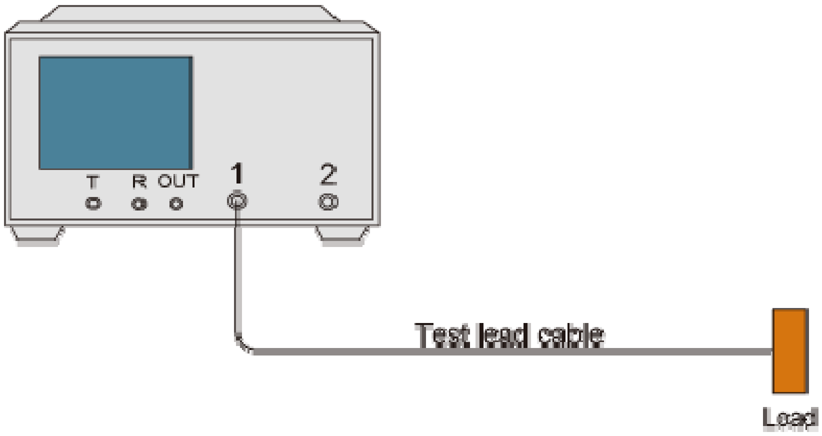
After calibrating, it is important to verify that the calibration is good. Always determine your system directivity and verify the quality of your test lead cable after performing a calibration.

When verifying the calibration and the quality of your test lead cable, you should look for a combination of good system directivity (<-50 dB, but acceptable up to -40 dB) and small variations in peak amplitudes (<10 dB) when the test lead cable is wiggled or moved.

Determine System Directivity

1. Determine the system directivity by connecting the load standard to the end of the test lead cable as shown in the following figure. (Or, if your reference plane is the analyzer's RF OUT (or PORT 1) test port, connect the load directly to that front panel connector.)

Connect the load



c5061b011

2. Observe the magnitude of the response on measurement channel 1. The highest peak response on channel 1 is the system directivity. If the peak response on channel 1 is <-50 dB, the calibration is good. If the peak response is > -40 dB, you should recalibrate the analyzer.

   NOTE Measurement quality is related to the system directivity. For the highest quality measurements, system directivity should be < -50 dB, but measurement quality is acceptable up to -40 dB. See Measurement Uncertainties.

Determine the Quality of the Test Lead Cable

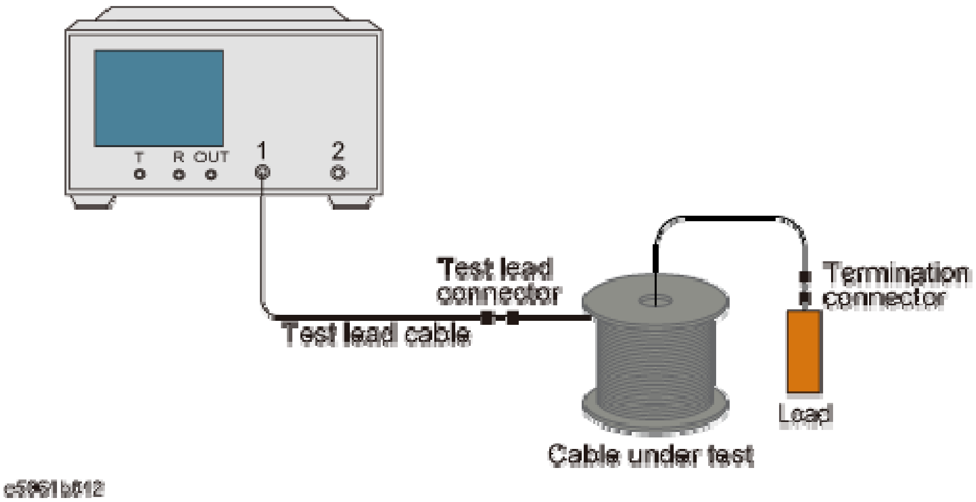
1. Leave the load connected to the end of the test lead cable and note the level of the peak response on measurement channel 1 (the system directivity).
2. Wiggle the test lead cable while observing the response on the analyzer's display.
   a. If the measurement trace is relatively stable, the test lead cable is of good quality.
   b. If you observe significant movement in the peaks of the measurement trace when wiggling the cable (>10 dB), the test lead cable may need to be replaced.

   NOTE Variation in the system directivity that occurs as a result of test lead cable movement degrades the quality and repeatability of SRL measurements. Take precautions to protect your test lead cables from mishandle or abuse. Do not step on or drive vehicles over test lead cables.

Connect the Cable Under Test

The basic equipment setup for SRL measurements is illustrated in the figure below.

Basic SRL Measurement Setup



Determine the Connector Model

After connecting the cable under test, you should determine the connector model for the best response. The connector model may need to be determined each time a new cable is tested.

When using connectors that have very consistent interfaces, modeling the connector for each new connection to a cable may not be required. When using connectors that do not have a repeatable interface contact, modeling the connector for each new connection to a cable is necessary.

For some SRL measurements, the response of the connector can be critical for obtaining a true measurement of structural return loss. For example, a connector with a return loss of 30 dB swamps out SRL responses less than about -20 dB. A connector with a 40 dB return loss provides a more accurate measurement of the -20 dB responses.

The following table shows the effects of a connector mismatch on the measurement of a -35 dB SRL spike.

Corrected Connector Return Loss	SRL	Total Measured
-53 dB	-35 dB	-34 dB
-42 dB	-35 dB	-31.8 dB
-35 dB	-35 dB	-29 dB

The best true SRL measurement is made when the contribution of the connector is minimized by

- a good calibration
- a high-quality connector and connection (see Cable Preparation)

- a connector model which provides the lowest corrected connector response

The effect of the connector response can be minimized with the built-in connector model and the corrected connector response can be measured while the SRL measurement is being made. For some connectors, a response correction of up to 15 dB or more improvement is possible with the built-in connector model.

NOTE The maximum extent to which the effects of the connector response can be removed is to the accuracy and repeatability of the analyzer system (including the effects of test lead cable stability and quality). The accuracy of the system is given by the system directivity of the analyzer (which can be determined from the trace with a load connected after calibration).

Connector Model for Long Cables

If a long cable is being measured, you can use the Measure Connector feature to automatically determine the L and C values. (A long cable is defined as approximately 300 m (1000 ft)).

1. Press **Analysis** > **SRL**.
2. Press **Port1 Connector** (or **Port2 Connector**).
3. Connect the terminated cable, then press **Measure Connector** to set the length and C values automatically.

Connector Model for Short Cables

If you are measuring a short cable, or if you have very large mismatches in the cable under test, you may need to manually set the L and C values.

1. Press **Analysis** > **SRL**.
2. Press **Portx Connecto**r (x is the port to which the cable is connected).
3. Press **Length** and **Capacitance** to specify the connector length and the connector capacitance.
4. Observe the SRL measurement trace while adjusting the connector length and C values for the best (lowest overall) response.
5. Some cables may be best measured by adjusting only the connector length, other cables may require a connector C adjustment, and some others may require a combination of connector length and C values.

   NOTE When manually adjusting the connector length or connector C values, be sure to wait for the analyzer to complete a sweep and update the display before trying another value.

6. You may need to measure the connector using a Smith chart to get the best connector model:

Press **Format** > **Smith** > **Real/Imag**

Observe the display while adjusting the connector C and connector length parameters. The best response is obtained when the Smith chart response has been most compacted by the connector C and connector length adjustments.

NOTE If you cannot obtain a low enough response by adjusting the connector length and/or connector C values, you should perform a fault location measurement on the connector and the cable under test. (See Making Fault Location Measurement) Be sure to determine the quality of the connector being used; some cable connectors degrade rapidly with use. The response of a bad connector is often large enough to swamp out the response from cable SRL.

Connector L and C Values

The following table shows some typical values for two types of slip-on connectors for mainline cable:

Connector	L Value	C Value
GTC-700-TX-GHZ-N ("GHz")	40 to 80 mm	0 to 0.15 pF
GTC-700-TX-N ("Pogo")	-12 to 12 mm	0 to 0.125 pF

For these connectors, use of values within these ranges should be optimum for the best corrected connector fault response and lowest SRL spikes. Values far outside this range usually indicate a bad calibration, a poor connector or connection, or a close-in cable fault which cannot be compensated by the connector model.

NOTE The optimum calculated value for the connector lengths of "Gilbert Pogo" connectors may be slightly negative. This is a normal value and should not be a cause for concern.

For type-F connectors, which are typically used to measure 75 ohm drop cable, the range of connector L and C values varies widely and depends greatly on the quality of the type-F connector.

Perform the SRL Cable Scan

Once the connector model has been established for the best response, the cable should be scanned at narrow frequency resolution to look for narrow response spikes that are characteristic of periodic defects in the cable. The SRL cable scan is required to determine the cable's SRL with 125 kHz resolution.

By taking five sweeps of 1601 points each at slightly different frequency ranges (see Table 4-3), the analyzer can obtain 8005 distinct frequency points to achieve the desired frequency resolution of 125 kHz.

The resolution of the SRL measurement is determined by the following formula:

$Resolution = (F_{stop}-F_{start})/N$

where N is the number of measurement points. See the table below.

	F_{start}	F_{stop}	N	Resolution
No Cable Scan	5 MHz	1000 MHz	201	4.95 MHz
	5 MHz	1000 MHz	1601	612 kHz
Using Cable Scan	5 MHz	1000 MHz	8005	125 kHz

SRL Cable Scan Frequency Sweeps

Sweep Number	Start Frequency (MHz)	Stop Frequency (MHz)
1	5.000	999.500
2	5.125	999.625
3	5.250	999.750
4	5.375	999.875
5	5.500	1000.000

SRL Cable Scan Setup using the VBA Utility Program

The E5061B provides a macro program called **srl_util.vba**, which facilitates the measurement setup for SRL cable scan.

Procedure

1. Press **Channel Next** (or **Channel Prev**) to activate a trace of which you want to set up.
2. Press **Macro Setup** > **Load Project**.
3. From the open dialog box, select the VBA project file **D:\Agilent\srl_util.vba**, and press **Open**.
4. Press **Select Macro** > **Module1 main** to execute the macro program.

5. The SRL Setup window appears in the lower part of the screen. Click on the following 2 settings to check-mark them with the mouse as necessary and press **OK**.
    - Connector Fault - Splits the screen vertically into 2 sections and displays trace 2 with fault location on. The display range of trace 2 is set to 0 to 5 m.
    - Reverse(S22) - Sets the measurement parameter to S22.

NOTE Executing this macro program automatically makes the following settings in addition to the above.

Parameter	Value
Start frequency	5 MHz
Stop frequency	1 GHz (the display range of trace 1)
Sweep type	Linear Frequency
Number of points	1601
SRL	ON

Perform the SRL Cable Scan Using the VBA Utility Program

The E5061B provides a macro program called **srl_util.vba**, which performs SRL cable scanning.

NOTE It is recommended to do calibration before scanning a cable.

Procedure

1. Press **Channel Next** (or **Channel Prev**) to activate a trace of which you want to perform the cable scan.
2. Press **Macro Setup** > **Load Project**.
3. From the open dialog box, select the VBA project file **D:\Agilent\srl_util.vba**, and press **Open**.
4. Press **Select Macro** > **CableScan main** to execute the macro program.
5. The following processing is automatically performed.
    i. Set the following settings: SRL - ON, sweep type - Linear Frequency, number of points - 1601, display format - LogMag.
    ii. Set the measurement parameter of the active trace to S22 (or S11) and enter into reflection measurement status.

iii. Sweep five times with the frequency settings listed in SRL Cable Scan Frequency Sweeps.

iv. Perform the measurement again that gives the highest SRL value of the 5 sweeps.

v. Move the active marker to the maximum value.

Interpret the SRL Measurement

Periodically spaced SRL response bumps cause frequency spikes at a frequency given by the following formulas:

wavelength ≈ c/f

Where c= speed of light and f = frequency

wavelength/2 = spacing between the bumps

The bumps may be located near one end of the cable or somewhere in the middle. Although the bumps from individual defects may be small, fault location measurements may be useful to determine the location(s) of the cable's defect(s). See SRL and Periodic Cable Faults.

Setting Control Functions

Remote Control

- Setting the GPIB
- Remote Control Using HTTP

Display

- Turning off the Date/Time Display
- Turning off the LCD Screen Backlight
- Calibration of the Touch Screen

Others

- Initial Source Port Control function
- Band Wait Function
- Checking the product information
- Activating Software Option
- Locking the Front Keys, Keyboard, and/or Mouse (Touch Screen)
- Setting the Beeper (Built-in Speaker)
- Setting the preset function
- Exit/Restart E5061B Measurement Application
- Overload Detection and Power Trip Function

Setting the GPIB

- Setting talker/listener GPIB address of E5061B
- Setting system controller (USB/GPIB interface)

Other topics about Setting Control Functions

This section describes how to set the interface necessary to use the GPIB (General Purpose Interface Bus) of the E5061B.

Setting talker/listener GPIB address of E5061B

When controlling the E5061B using GPIB commands from the external controller connected to the GPIB connector, you need to set the talker/listener GPIB address of the E5061B.

Follow these steps to make this setting:

1. Press **System** > **Misc Setup** > **GPIB Setup**> **Talker/Listener Address**.
2. Enter the address using the ENTRY block keys on the front panel.

Setting system controller (USB/GPIB interface)

When controlling an external device from the E5061B, connect the USB port of the E5061B and the GPIB port of the external device through the USB/GPIB interface.

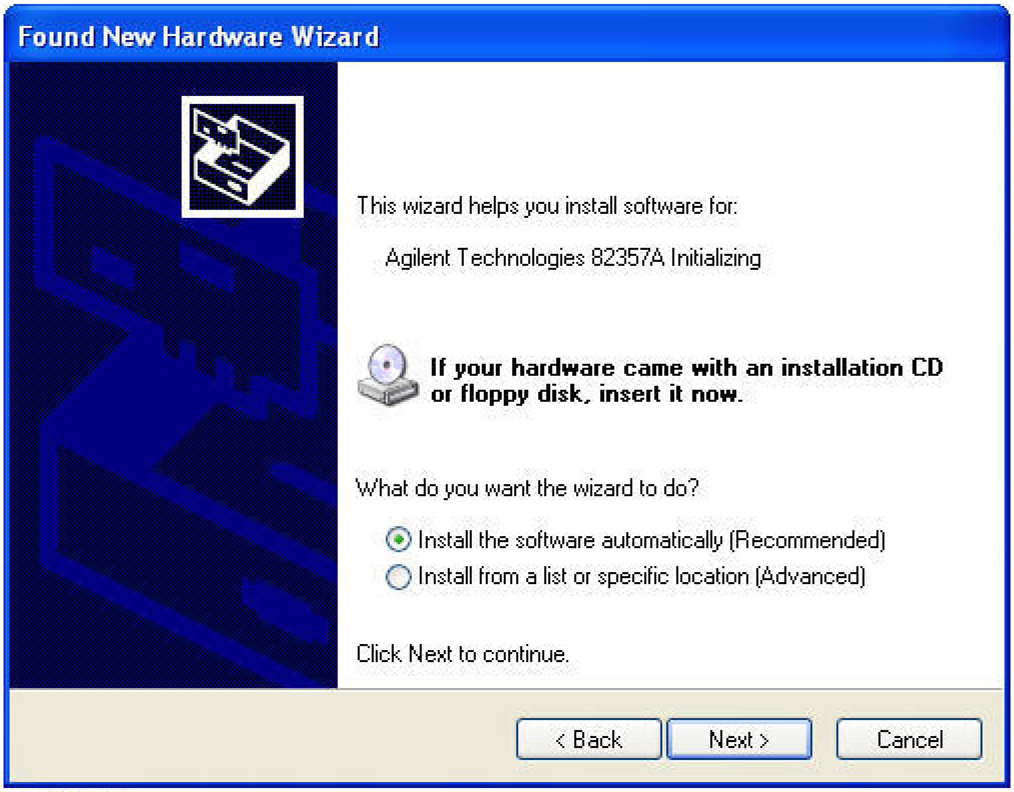
NOTE Do not connect two or more USB/GPIB interfaces.

Follow these steps to set the USB/GPIB interface:

1. Connect the USB port of the E5061B to the USB/GPIB interface.

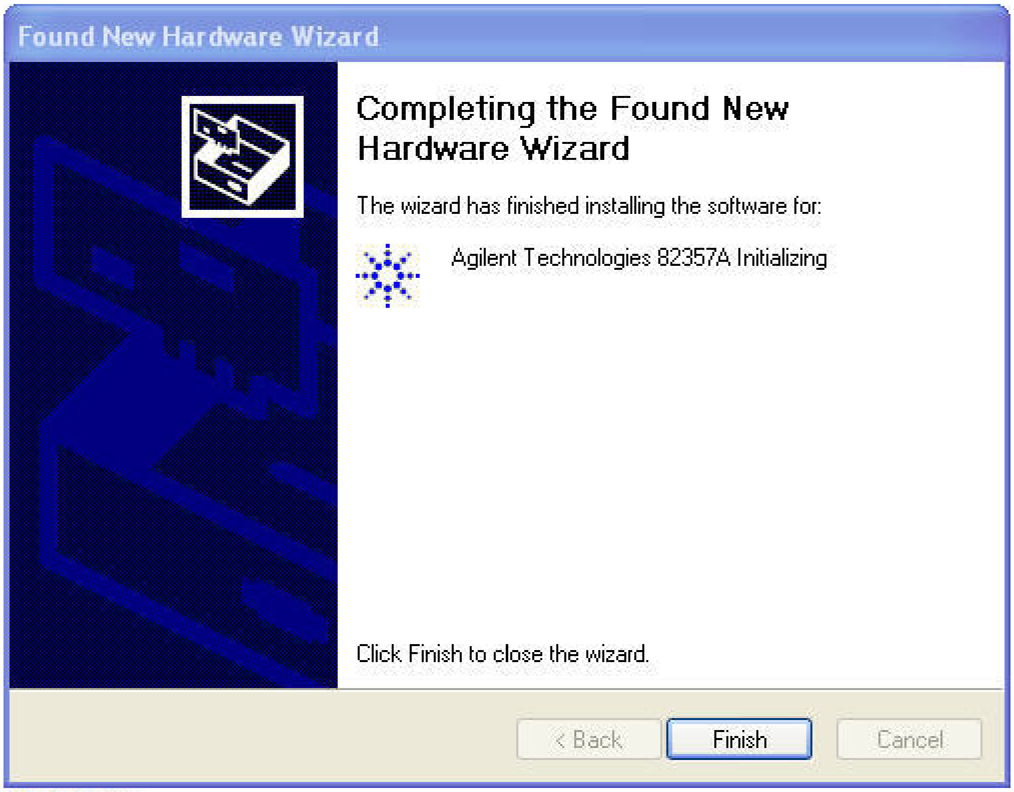
2. Select **No, not this time**, then click **Next >**.

3. Select **Install the software automatically (Recommended)**, then click **Next >**.



e5071c161

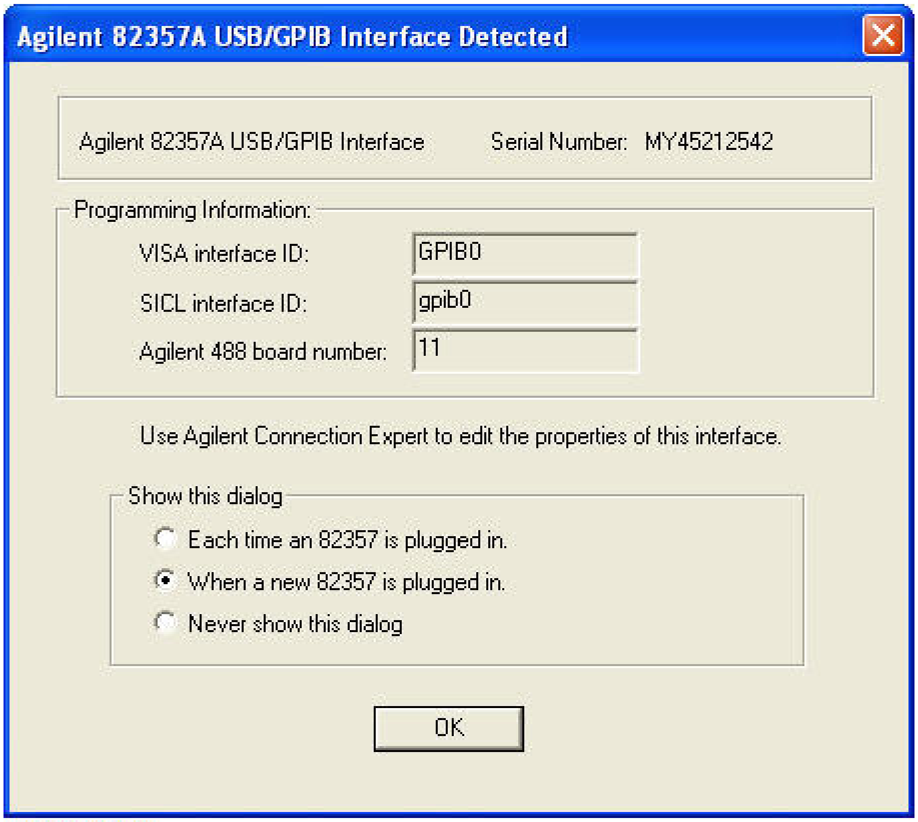
4. Click **Finish.**



e5071c163

5. The **Found New Hardware Wizard** appears again. Repeat step 2 to 4.

6. Select **When a new 82357 is plugged in**, then click **OK.**



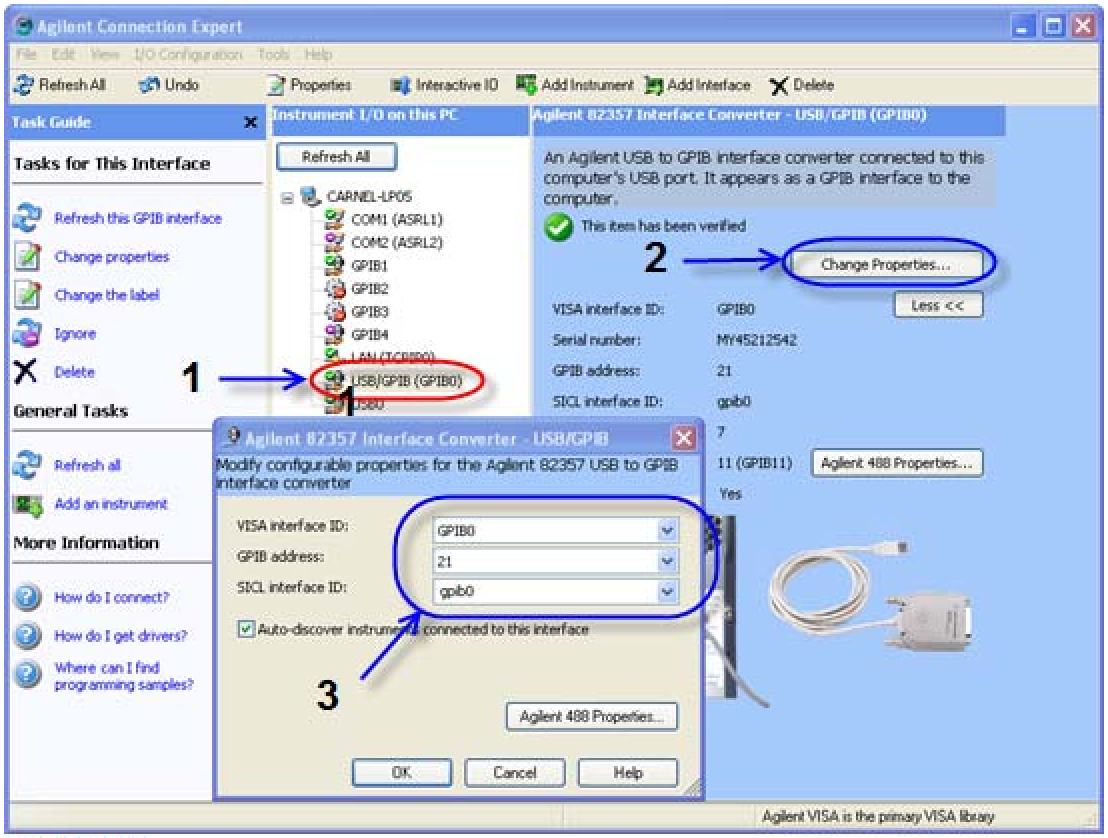
e5071c0169

NOTE Select **When a new <USB device> is plugged in.** or **Never show this dialog** instead of **Each time a USB devices is plugged in.**.

Changing the setting of the USB/GPIB interface

If you need to check/change the setting of the USB/GPIB interface after connecting the USB/GPIB interface, follow these steps:

1. Press **System** > **Misc Setup** > **GPIB Setup** > **System Controller Configuration.**
2. The Agilent Connection Expert appears. (You can also execute Agilent Connection Expert from Task bar or Start menu in Windows.)



e5071c172

3. Select USB/GPIB (GPIBx)
4. Click **Change Propetries...**.
5. Change the setting of USB/GPIB interface.

Remote Control Using HTTP

- Enabling Web Server
- Browser Web Control

Other topics about Setting Control Functions

You can access the web page installed in the E5061B by using the hypertext transfer protocol (http) and the E5061B's IP address from the external PC's web browser. This function is called web-enabled analyzer. Through the built-in web page, you can control the E5061B remotely and display the measurement screen on external PCs.

The following browsers are recommended:

- Internet Explorer 6.0 and later

Enabling Web Server

Network Configuration

To use web server, you have to configure the E5061B's network correctly. For detailed information on configuration and notes, see Configuring the Network.

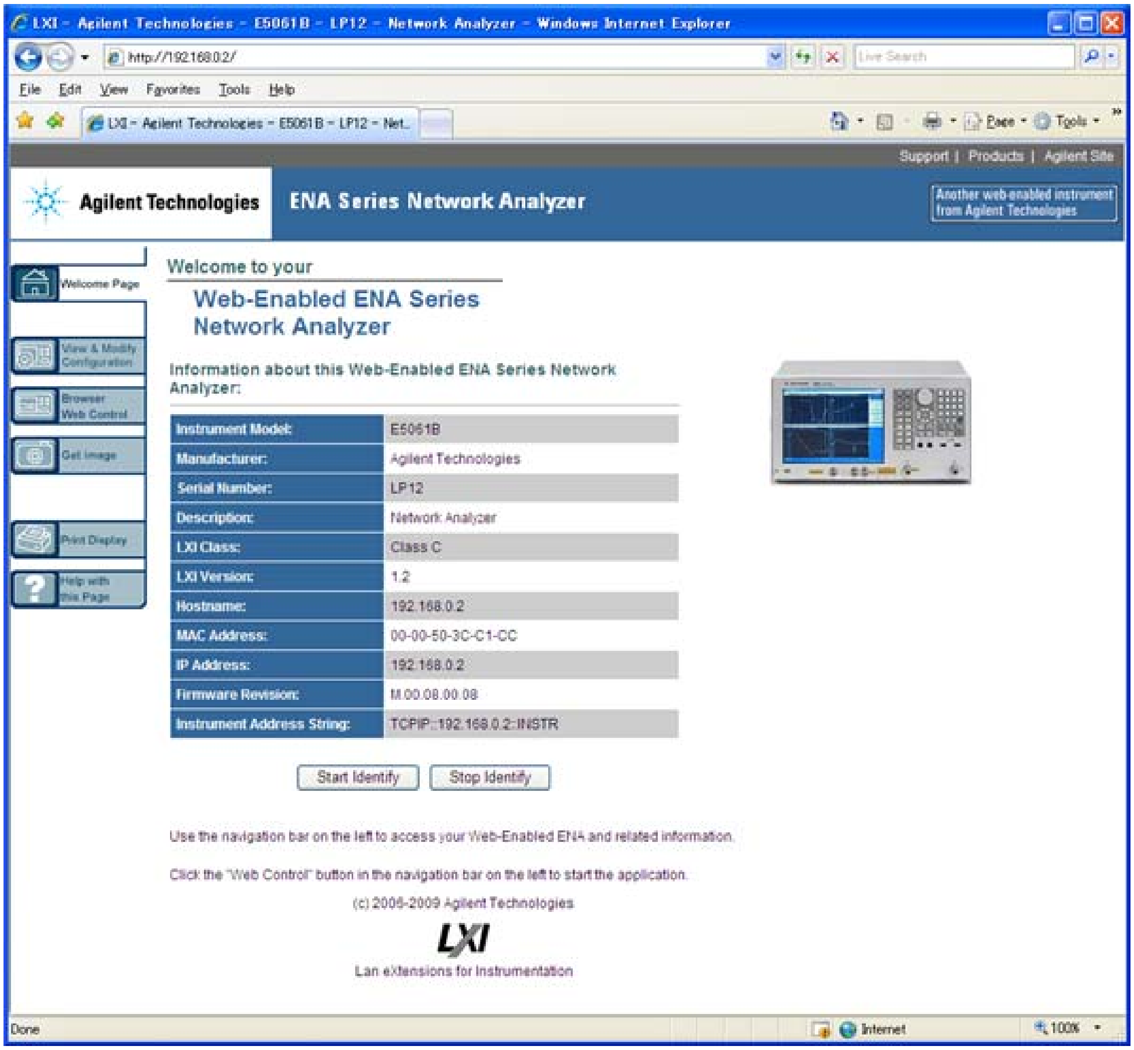
Enabling Web Server

Enable the web server for the E5061B so that it may allow access from an external PC. Follow these steps:

1. Press **System** > **Misc Setup** > **Network Setup**.
2. Click **Web Server** to turn it ON.

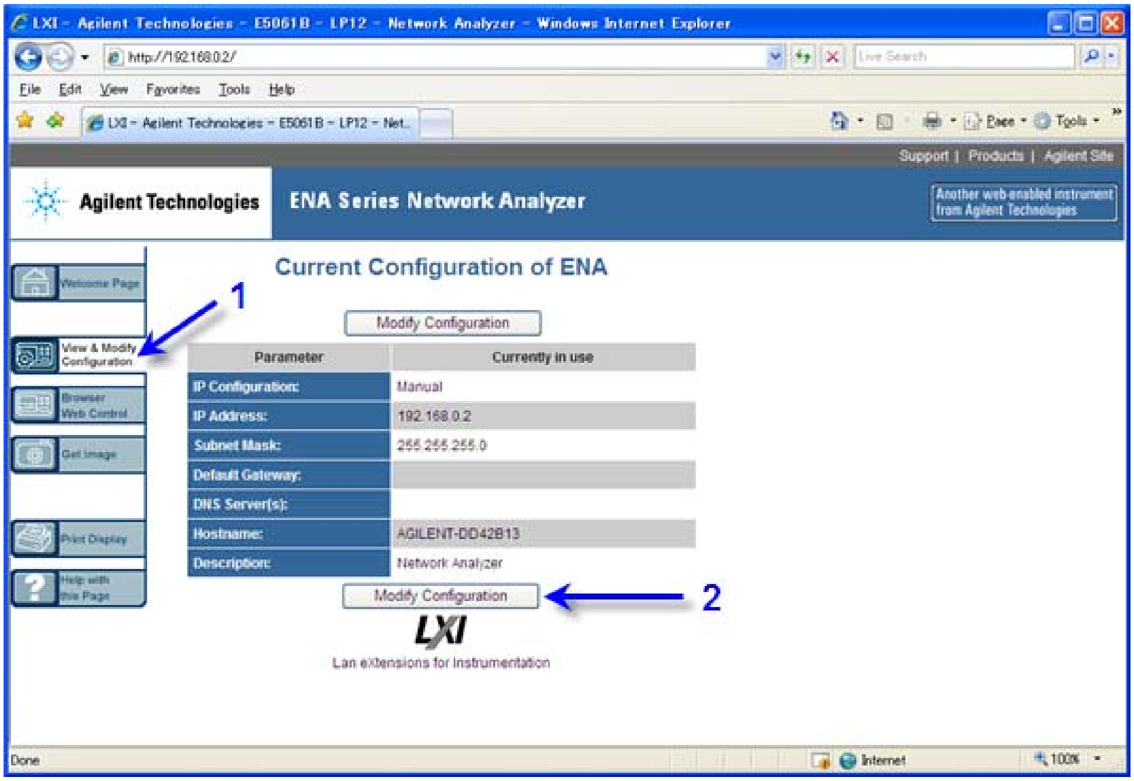
Access from an external PC

1. Execute web browser on your PC.
2. Check IP address of the E5061B.
3. Enter IP address of the E5061B in the address bar and press Enter. The following screen appears:

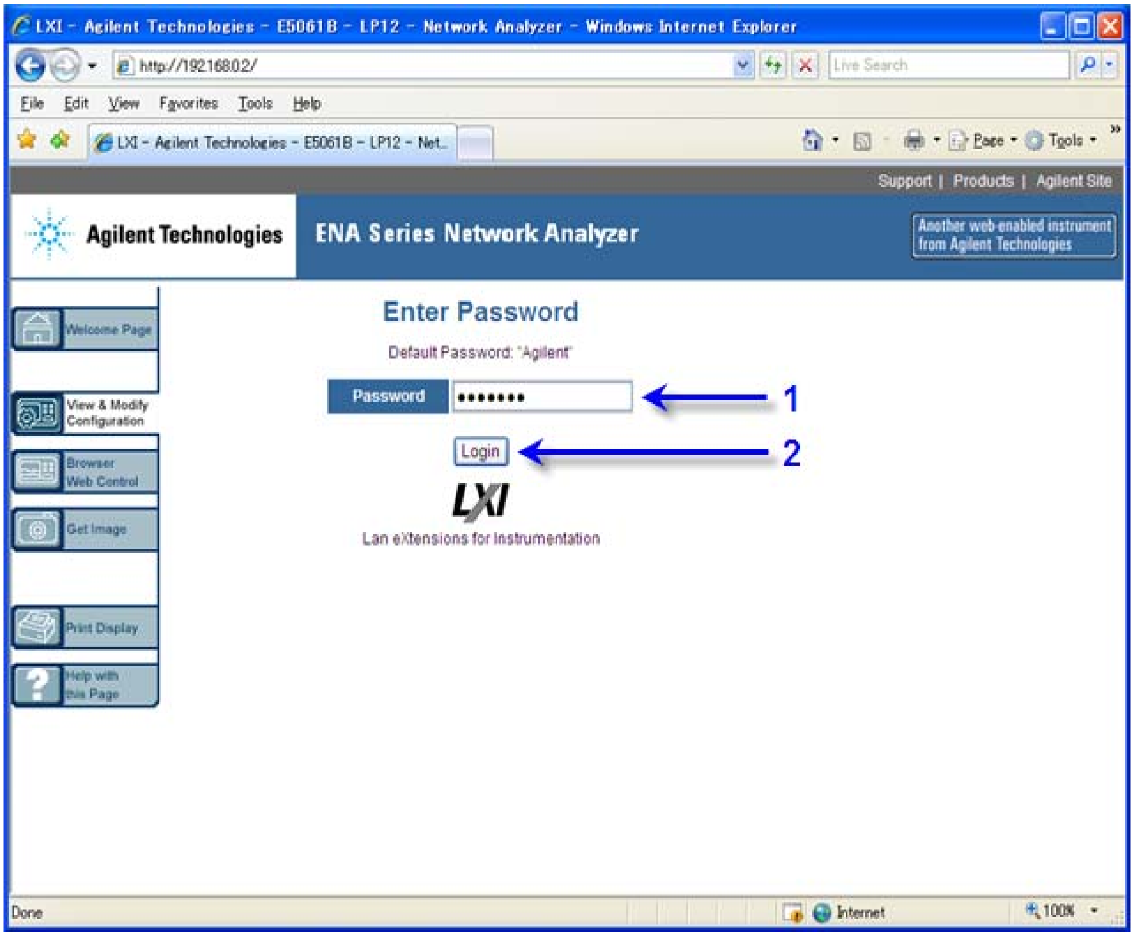


To change LAN (TCP/IP) configration of you ENA.

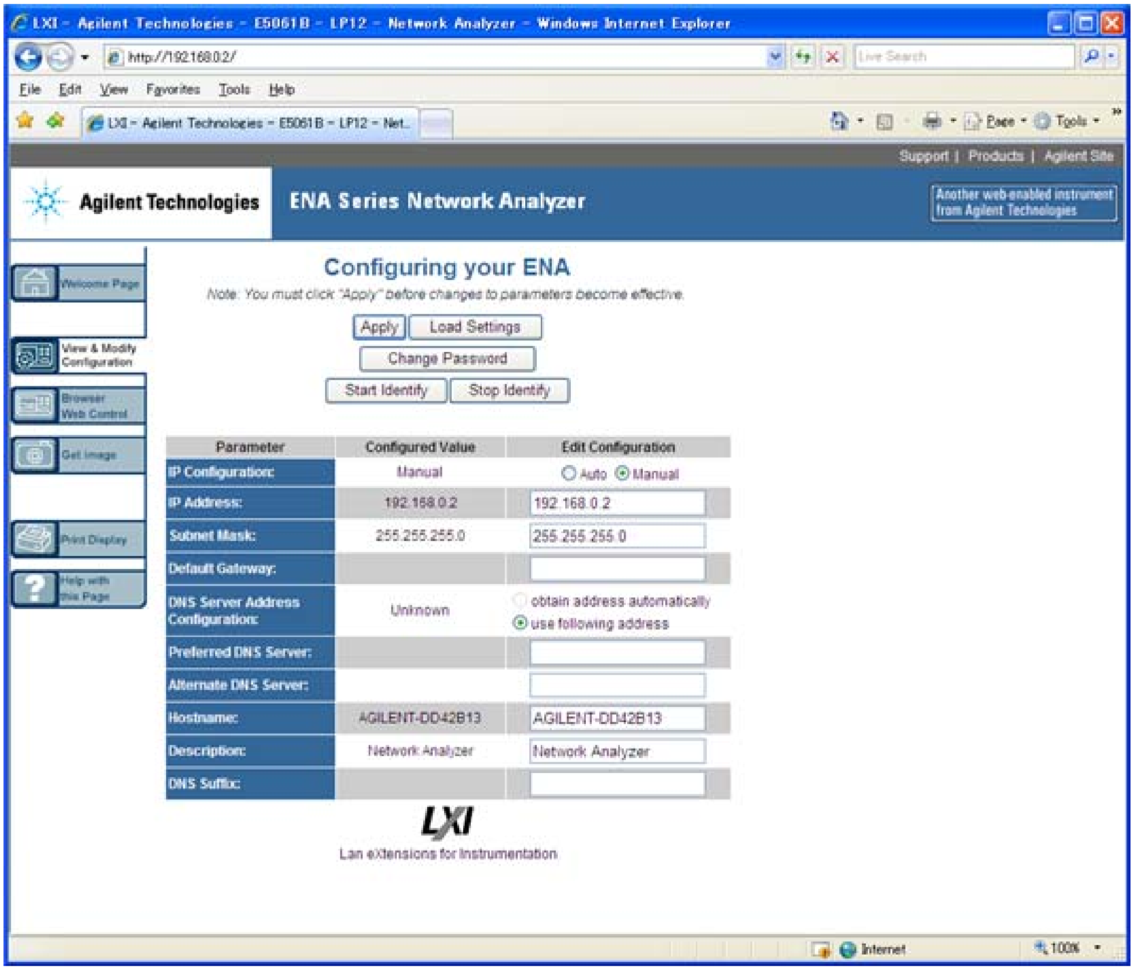
1. Click **View & Modify Configuration** (**1** in the Figure below). The following screen appears:



2. Click **Modify Configuration** (**2** in the Figure above). Modifying this setup affects the Windows Internet Protocol (TCP/IP) property. The following screen appears:



3. Enter the password (Default: "Agilent") in the Password field (**1** in the figure above) and click **Login** (**2** in the figure above). The following screen appears:



Browser Web Control

Browser Web Control function allows you to control your E5061B from web browser. This function is executed by the VNC server.

NOTE The external PC must have the Java Runtime Environment installed otherwise the Browser Web Control function might not work properly. To install Java Runtime Environment, see http://www.java.com.

The following is a description of how to start the VNC server configuration. Visit the web site at http://www.realvnc.com for information on the password setting procedure and VNC server.

1. Press **System** > **Misc Setup** > **Network Setup.**
2. Click **VNC Server Configuration** to start the VNC Server Properties.

NOTE To restrict external access, set a password for the VNC server configuration. The default password at factory shipment is blank.

Turning on/off the Date/Time Display

The date/time display in the instrument status bar can be switched ON/OFF using the following procedure.

1. Press **System** > **Misc Setup** > **Clock Setup**.
2. Click **Show Clock** to toggle the date/time display ON/OFF.

Other topics about Setting Control Functions

Turning off the LCD Screen Backlight

You can switch OFF the backlight (illumination) of the LCD screen of the E5061B. This extends the life of the backlight when it is used continuously over a long period.

Turning OFF the LCD Screen Backlight

1. Press **System** > **Backlight** to switch the backlight ON/OFF.
2. Switching OFF the backlight causes the indications on the LCD screen to be almost invisible.
3. The backlight that has been switched OFF can be turned ON again by pressing any key from the front panel.

Other topics about Setting Control Functions

Calibration of the Touch Screen

When you have executed system recovery on the E5061B, you have to calibrate the touch screen. Follow the procedure described below to calibrate the touch screen.

1. Press **System** > **Service Menu** > **Test Menu** > **Adjust Touch Screen.**
2. The touch screen calibration screen appears.
3. Touch the x mark on the upper left with your finger. The mark x also appears on the lower left, upper right, and lower right. Touch the x marks in that order with your finger.
4. Touching all the four locations described above with your finger automatically concludes the touch screen calibration.

NOTE With no operation on the touch screen calibration screen for a preset time, it automatically closes and the previous measurement screen reappears.

Other topics about Setting Control Functions

Initial Source Control Function

The E5061B stops to output RF/LF signal source and DC Bias while the measurement is hold. (when Initial Source Control is ON, default setting). This function allows you to output signal source and DC Bias even while the measurement status is in hold.

- RF/LF Signal
    1. Press **System** > **Service Menu** > **Init Src Ctrl** > **RF Out** to turn OFF.
- DC Bias
    1. Press **System** > **Service Menu** > **Init Src Ctrl** > **DC Bias** to turn OFF.

Other topics about Setting Control Functions

Band Wait Function

This function can set a wait time for each band in a sweep. This function allows you to make more stability measurement when the DUT need a time to stabilize.

Press **System** > **Service Menu** > **Band Wait** , then enter the required time.

Checking the product information

- Checking the serial number and options
- Checking the Firmware Revision
- Checking the HDD Revision

Other topics about Setting Control Functions

Checking the serial number and options

The serial number and software options of the E5061B can be checked using the following procedure. See Options for the option displayed in this dialog.

1. Press **System** > **Firmware Revision**.
2. The Firmware Revision dialog is displayed. It provides information about the installed Options, Serial Number, IP Address, Mac Address and USB ID.

Checking the Firmware Revision

The revision number of the firmware installed in the E5061B can be checked using the following procedure.

1. Press **System** > **Firmware Revision**.
2. The Firmware Revision dialog is displayed.

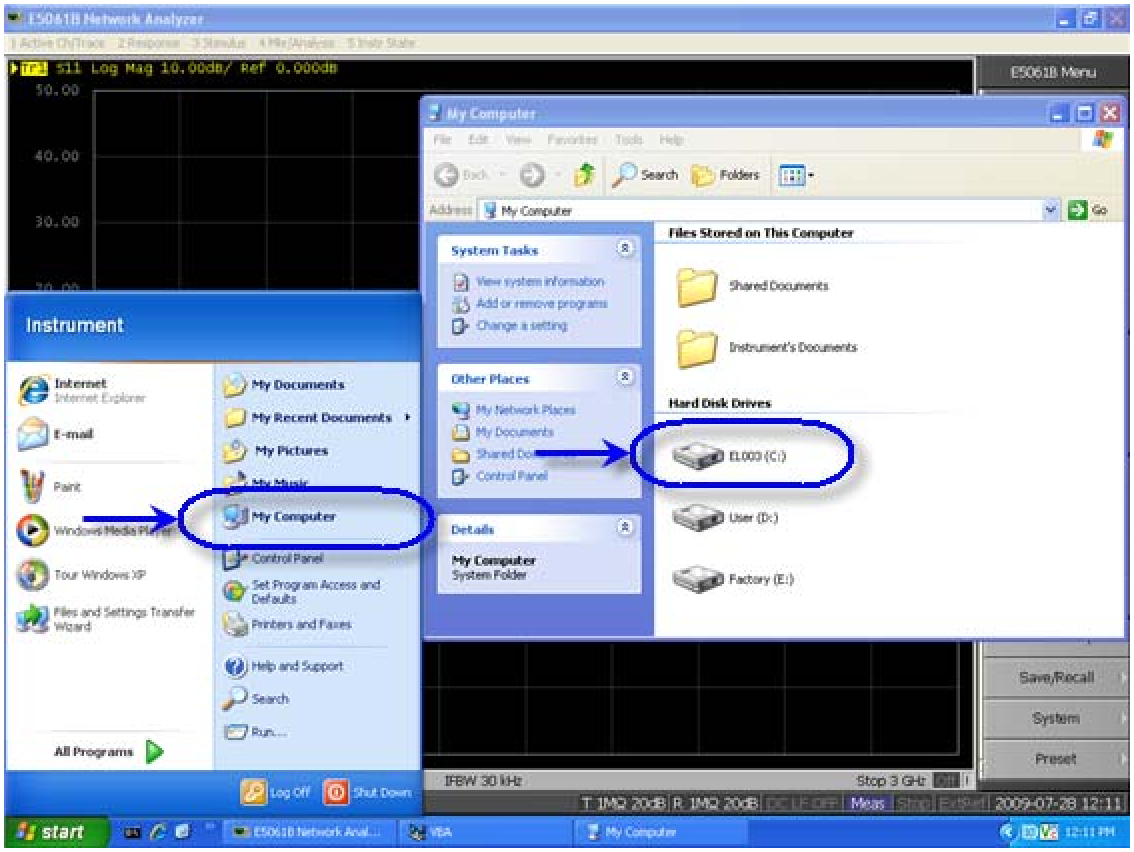
Checking the HDD Revision

1. Press **System** > **Firmware Revision**.
2. The Firmware Revision dialog is displayed.
3. HDD revision is shown at HDD image.

For more information, refer to HDD Revision History.

At the firmware revision A.1.xx, the HDD revision is not displayed at the Firmware Revision dialog box.

The HDD revision for the revision A.1.xx can be checked using **Start** > **My Computer** > **C:** drive label. For example, the HDD version of the E5061B in the figure below is EL003.


E5061B Network Analyzer
Tr1 S11 Log Mag 10.00dB/ Ref 0.000dB
50.00
40.00
30.00
E5061B Menu
My Computer
Instrument
Internet
Internet Explorer
E-mail
Paint
Windows Media Player
Tour Windows XP
Files and Settings Transfer Wizard
All Programs
My Documents
My Recent Documents
My Pictures
My Music
My Computer
Control Panel
Set Program Access and Defaults
Printers and Faxes
Help and Support
Search
Run...
Log Off
Shut Down
Search
Folders
Address
My Computer
Go
System Tasks
View system information
Add or remove programs
Change a setting
Other Places
My Network Places
My Documents
Control Panel
Details
My Computer
System Folder
Files Stored on This Computer
Shared Documents
Instrument's Documents
Hard Disk Drives
E1000 (C:)
User (D:)
Factory (E:)
Save/Recall
System
Preset
IFBW 30 kHz
Stop 3 GHz
Meas
2009-07-28 12:11
start
E5061B Network Anal...
VBA
My Computer
12:11 PM

Activating Software Option

- Activating Option
- Backing Up License Key File

Other topics about Setting Control Functions

Activating Option

The software options can be purchased separately to enhance the E5061B measurement functionality. When you purchase the software option upgrade kit, Agilent provides the software entitlement certificate.

1. Get the license number at http://www.agilent.com/find/softwarelicense.
2. Press **System** > **Service Menu** > **Enable Options** and then select option which you want to activate.
3. Type the relevant 12 character long license key sent by Agilent in the Key Code entry dialog box, then click Enter.
4. Check the installed option according to Checking the product information.

Backing Up License Key File

The license keys are kept in a text file (.lic) located at **RECOVERY(E):\LICENSE\gen.lic**. Even if you lost the file, you can activate the option again if you have the license number. You can re-create the license number at http://www.agilent.com/find/softwarelicense with your software entitlement certificate.

Locking the Front Keys, Keyboard, and/or Mouse (Touch Screen)

You can lock (disable) the front keys, keyboard, and/or mouse (touch screen). This feature prevents erroneous operation caused by inadvertently touching any of these devices.

Locking the Front Keys, Keyboard, and/or Mouse

1. Press **System** > **Misc Setup** > **Key Lock.**
2. Click the corresponding key to switch the lock ON/OFF.

Softkey	Function
Front Panel & Keyboard Lock	Switches the lock of the front panel keys and keyboard ON/OFF.
Touch Screen & Mouse Lock	Switches the lock ofr the touch screen and mouse ON/OFF.

NOTE You cannot use a locked device to unlock that same device. To unlock the front panel keys, keyboard, touch screen and mouse that have been locked, press the Standby switch to turn OFF the power supply and then turn it ON again. When setting at power-on, the front panel keys, keyboard, touch screen and mouse are all in unlocked condition.

Other topics about Setting Control Functions

Setting the Beeper (Built-in Speaker)

- Setting the Operation Complete Beeper
- Setting the Warning Beeper

Other topics about Setting Control Functions

The E5061B has a built-in speaker that emits a beep tone. The beeper allows you to make two types of settings.

Type	Function
Operation complete beeper	Emits a beep tone to inform the user that operations have completed. • When calibration data measurements are done • When data storage has completed
Warning beeper	Emits a beep tone to prompt the user to use caution. • When an instrument error occurs (An error message appears at the same time.) • When a limit test fails

The operations complete beeper emits slightly longer than the warning beeper.

Setting the Operation Complete Beeper

1. Press **System** > **Misc Setup** > **Beeper** > **Beep Complete** to switch the operation complete beeper ON/OFF.
2. Clicking **Test Beep Complete** allows you to hear and check the beep tone of the operation complete beeper.

Setting the Warning Beeper

1. Press **System** > **Misc Setup** > **Beeper** > **Beep Warning** to switch the warning beeper ON/OFF.
2. Clicking **Test Beep Warning** allows you to hear and check the beep tone of the warning beeper.

Setting the preset function

- Showing/hiding the confirmation buttons when presetting
- Setting the user preset function
- Saving a user-preset instrument state

Other topics about Setting Control Functions

Showing/hiding the confirmation buttons when presetting

The preset function can be executed without displaying the **OK** and **Cancel** softkey buttons when pressing the preset button of the E5061B.

1. Press **System** > **Misc Setup** > **Preset Setup**.
2. **Confirm** to toggle ON (show)/OFF (hide) the confirmation buttons.

Setting the user preset function

You can save the instrument state of the E5061B into a file in the mass storage, and then recall it with the preset function to reproduce that state.

If no user preset instrument state is stored, you cannot set the user preset function.

1. Press **System** > **Misc Setup** > **Preset Setup** > **State**.
2. Use one of the following keys for the desired setting.

Softkey	Function
Factory	Specifies the normal preset function.
User	Specifies the user-preset function.
Cancel	Returns to the softkey display in one upper level.

Saving a user-preset instrument state

To execute the user-preset function, you must have a preset setting file that has been saved. Follow these steps to save a preset instrument state of the E5061B.

1. Press **Save/Recall** > **Save State** > **User Pres**.

Exit / Restart E5061B Measurement Application

Pressing **System** > **Service Menu** > **Exit** shuts down the firmware of E5061B.

Pressing **System** > **Service Menu** > **Restart Firmware** restarts the firmware of the E5061B.

Other topics about Setting Control Functions

Overload Detection and Power Trip Function

An overload detection and Power Trip function is a safety feature to protect your E5061B from over-input. When an overload is detected, the measurement is aborted and the settings are changed as shown below, then the "Ovld Protection" is displayed on the instrument status bar.

- RF Out : OFF
- Port Coupling: AC
- DC Bias: OFF
- Gain-Phase Input Attenuator: 20 dB
- Gain-Phase Input Impedance: 1 MΩ

The softkey labels which setting have been affected have **(Protected)** sign (e.g. **RF OUT** / **OFF (Protected)**).

The overload occurs when DC current is over:

- Ports 1 and 2: 40 mAdc
- Ports R and T (Gain-Phase): 100 mAdc

When the overload occurs, remove the cause of the over-input. Then press **System** > **Overload Recovery** > **Clear Overload Protection**. This softkey is not only to clear the protection status but also to set the former setting before you have overload.

The softkeys under **System** > **Overload Recovery** menu is identical with the same label softkeys under the other menu. For example, **System** > **Overload Recovery** > **RF Out** is identical with **Sweep Setup** > **Power** > **RF Out**. These softkeys help you to see the setting at a glance.

If changing the setting, such as attenuator: 0dB, causes overload, you can observe which setting have been changed and change them quickly at the softkeys under **System** > **Overload Recovery**. Changing the setting also clears the protection status.

Other topics about Setting Control Functions

Product Information

Product Information

- Options
- Documentations
- Specifications
- Customer Contacts
- Error Messages (Warning Message)
- Default Conditions
- Troubleshooting
- Maintenance
- Measurement Accessories
- General Principles of Operation

Options for E5061B

The following list shows available options. Some options can be retrofitted on your E5061B. For upgrade (retrofit) kits, refer to http://www.agilent.com/find/ena_support.

- Test Set Options
- Software Options
- Hard Disk Drive Options
- Timebase Options
- Accessary Options
- Calibration Option
- System Rack Options

Other topics about Product Information

Test Set Options

Option Number	Description
3L5	LF-RF Network Analyzer with DC bias, 5 Hz to 3 GHz.
215	Network Analyzer 100 kHz to 1.5 GHz, S-parameter test set, 50 Ω system impedance
235	Network Analyzer 100 kHz to 3 GHz, S-parameter test set, 50 Ω system impedance
115	Network Analyzer 100 kHz to 1.5 GHz, T/R test set, 50 Ω system impedance
135	Network Analyzer 100 kHz to 3 GHz, T/R test set, 50 Ω system impedance
217	Network Analyzer 100 kHz to 1.5 GHz, S-parameter test set, 75 Ω system impedance
237	Network Analyzer 100 kHz to 3 GHz, S-parameter test set, 75 Ω system impedance
117	Network Analyzer 100 kHz to 1.5 GHz, T/R test set, 75 Ω system impedance

137	Network Analyzer 100 kHz to 3 GHz, T/R test set, 75 Ω system impedance

These options are displayed at the option information.

Software Options

Option Number	Description
005	Impedance Analysis for LF-RF VNA
010	Time Domain/Fault Location Analysis

NOTE Option 005 can be installed with Option 3L5.

These options are displayed at the option information. To activate the software option, refer to Activating Software Option.

Hard Disk Drive Options

Option Number	Description
019	Standard (non-removable) HDD [Discontinued]
020	Standard HDD

These options are displayed at the option information.

Time Base Options

Option Number	Description
1E5	High Stability Time base

These options are displayed at the option information.

Accessary Options

Option Number	Description
720	50 Ω resistor set
810	Add Keyboard

820	Add Mouse

Calibration Option

Option Number	Description
A6J	ANSI Z540 Compliant Calibration
1A7	ISO 17025 Compliant Calibration

System Rack Options

Option Number	Description	Equivalent Agilent Part Number
1CM	Rack Mount Kit	5063-9216
1CN	Front Handle Kit	5063-9229
1CP	Rack Mount and Front Handle Kit	5188-4430

Documentations for E5061B

- Manuals
- Sales Literature

Other topics about Product Information

Manuals

The following documentations are provided with the E5061B.

Name	Description
Help (This file)	Provides the information about the measurement operation, programming, built-in VBA, I/O interface.
Installation Guide	Provides information about start up setup and system recovery information when the Windows cannot be boot up. This is furnished with the E5061B as a hardcopy manual.
Service Guide	Provides information about the parts, troubleshooting, performance test, adjustment and service menu.

Both Installation Guide and Service Guide can be downloaded from http://www.agilent.com/find/e5061b-manual. The latest revision of Help System, Help in PDF and WebHelp formats are also available at the site.

Sales Literature

The following sales literatures are available on http://www.agilent.com/find/ena-lf.

- Brochure
- Data sheet
- Configuration Guide (Ordering information)
- Application notes

Specifications

The Data Sheet (PDF format) shows the E5061B specifications. This is for option 3L5.

The information for options 115/117/135/137/215/217/235/237 is available at http://cp.literature.agilent.com/litweb/pdf/5990-4392EN.pdf

For its history, see the Data Sheet Revision History.

Other topics about Product Information

Customer Contacts

For assistance on E5061B, refer to http://www.agilent.com/find/assist for your regional customer contacts. Click **Select a Country or Area** on the upper right of the web page to select your region.

Other topics about Product Information

Error Messages

Error Messages

An error message is displayed against a red background in the instrument message/warning area in the lower left part of the screen. Pushing a front panel key or executing :DISP:CCL command clears the error message. Errors caused by the operation of a front panel key simply appear on the display. They are not stored in the error queue with some exceptions.

An error with a positive error number is uniquely defined for this instrument. On the other hand, an error with a negative error number is basically defined for common GPIB devices in IEEE488.2.

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

A

No.	Message	Description
20	Additional standard needed	The GPIB command that turns ON the calibration function has been sent before to all of the data measurements needed to calculate the calibration factor that has been completed. In 1-port calibration, for example, when measurements are completed for OPEN and SHORT standards but not yet for LOAD standard: Try to turn ON the calibration function by calculating 1-port calibration coefficient using SENS:CORR:COLL:SAVE. Be sure to measure all necessary calibration data before sending the commands. This error is not generated by front key operations.
58	Auto port extension not allowed	This message appears when you try to execute the auto port extension when the sweep type is power or DC bias sweep.

B

No.	Message	Description
-168	Block data not allowed	An block-data element has been received at a position where this instrument does not accept one.

C

No.	Message	Description
97	Calculate equivalent parameter failed	This error occurs when the equivalent circuit analysis command SCPI.CALCulate(1-4).EPARameters.EXECute is executed and one or more of the parameters (R1, C1, L1, C0) could not be calculated.
430	Calibration below 100kHz not allowed with AC-couple mode	The calibration cannot be performed. The calibration for s-parameter measurement with AC couple should be performed when the minimum frequency is 100 kHz or above. Check if the frequency setting is correct.
240	Calibration data lost	This error occurs when a file containing the system calibration data is not found or in a damaged state at time of the startup of this instrument, indicating a failure of this instrument. Contact the Agilent Technology sales office or the company from which you bought the instrument.
22	Calibration method not selected	This error occurs when the command for validating the calibration data, SENS:CORR:COLL:SAVE is executed before the command for selecting a calibration type, SENS:CORR:COLL:METH:xxxx, is executed. This error is not generated by front key operations.
-148	Character data not allowed	A character data element (not violating the standard) has been received at a position where this instrument does not accept one. Double quotes (") are omitted where it is necessary to place a parameter in double quotes ("), for example.
-100	Command error	A comprehensive syntax error has occurred showing that this instrument cannot detect a more detailed error. This code simply shows that a command error defined in 11.5.1.1.4, IEEE488.2 has occurred.
90	Compensation required	Compensation is required.
460	Continuous switching may damage source	This error occurs when different power ranges are selected in multiple channel measurement settings to avoid source attenuator damage.

	switch	

D

No.	Message	Description
-222	Data out of range	A data element (not violating the standard) outside the range defined by this instrument has been received. This error occurs when an integer-based command for which the parameter can be rounded exceeds the range of -65536 to +65536 or when a real-number-based command for which the parameter can be rounded exceeds the range of -9.9e37 to +9.9e37, for example. This error occurs also when a numeric value other than a specified one is entered into a command in which the "port number" and "CalKit number" are specified as parameters and hence the parameters are not rounded. Such commands are, for example, SENS:CORR:COLL:ACQ:OPEN, SENS:CORR:COLL:ECAL:SOLT2, SENS:CORR:COLL:CKIT:ORD:LOAD, etc.
-104	Data type error	The parser has recognized a data element that must not exist. Block data is sent instead of numeric value data or character string data that had been expected, for example.
441	DC Bias not allowed (ECal Module Connected)	When an ECal is connected on the E5061B USB port, DC Bias cannot be turned ON. Disconnect the ECal.
440	DC Bias not allowed on Port 1 (S-Param couple AC)	When the port coupling is set to AC, DC bias cannot be turned ON.

E

No.	Message	Description
32	ECal module not in appropriate RF path	This error occurs when an ECal command, SENS:CORR:COLL:ECAL:SOLTn, is executed with the port on the ECal module not connected correctly to the instrument. When the auto-detect function is turned OFF, (SENS:CORR:COLL:ECAL:ORI OFF), however, this

		error does not occur even when the port on the ECal module is not connected correctly to the instrument.
502	Equation runtime error	This error occurs under the following conditions: • The trace number in data(tr)/mem(tr)/xAxis(tr) is out of range # of trace available depends on the maximum number of channel/traces • The port number in Advanced math function is out of range # of port available depends on the model option
95	Equivalent circuit analysis not allowed	The equivalent circuit analysis is only allowed when Option 005 is installed and impedance measurement is selected. Else, the analysis is not allowed, hence error occurs.
-200	Execution error	An error associated with execution has been generated for which this instrument cannot specify the error message. This code shows that an error associated with execution defined in 11.5.1.1.5, IEEE488.2 has occurred. This error also occurs when a calibration measurement is aborted.
-123	Exponent too large	The absolute value of the exponent exceeds 32,000 (see 7.7.2.4.1, IEEE488.2).
-178	Expression data not allowed	An expression-data element has been received at a position where this instrument does not accept one.
-170	Expression error	When the expression data is put to syntactic analysis, an error not corresponding to one of Error Numbers -171 through -179 occurs.

F

No.	Message	Description
31	Failed to configure ECal module	This error occurs when the control of the ECal module fails at the time of executing an ECal command, SENS:CORR:COLL:ECAL:SOLTn. The failure results are from the failure to connect the ECal module to the USB port, failure of the ECal module, etc.

102	Failed to copy file	This error occurs when copying a file (MMEM:COPY command) fails.
104	Failed to create directory	This error occurs when creating a directory (MMEM:MDIR command) fails.
103	Failed to delete file	This error occurs when deleting a file (MMEM:DEL command) fails.
100	Failed to read file	This error occurs when the formatted data array (MMEM:LOAD:FDAT command) and limit table (MMEM:STOR:LIM command) for the active trace on the active channel, segment sweep table (MMEM:LOAD:SEGM command) for the active channel, a VBA project file (MMEM:LOAD:PROG command), etc. cannot be read normally.
101	Failed to write file	This error occurs when the formatted data array (MMEM:STOR:FDATcommand) and limit table (MMEM:STOR:LIM command) for the active trace on the active channel, segment sweep table (MMEM:STOR:SEGM command) for the active channel, display image (MMEM:STOR:IMAG command) for the LCD screen, a VBA project file (MMEM:STOR:PROG command), etc. cannot be written normally.
54	Fault location, Gate not allowed	This message appears when you turn ON the fault location/gating function of the time domain feature, while measurement points are set to two, or sweep type is set to other than linear sweep Set the measurement points to three or more and the sweep type to linear sweep, and then turn ON the fault location and/or gating conversion function.
-257	File name error	A file name error. This message appears when an error exists in the file name and hence a command is not executed correctly. This error occurs when you try to copy to an unsuitable file name, for example.
-256	File name not found	The specified file name is not found and hence the command is not executed correctly. This error occurs when you try to read a file that does not exist in a disk or a disk is not correctly inserted into the drive to read or write a file, for example.

107	File transfer failed	This error occurs when writing data into or reading data from a file (MMEM:TRAN command) fails.

G

No.	Message	Description
431	Gain-Phase calibration not allowed at f>30MHz	The calibration cannot be performed. The calibration for gain-phase measurement should be performed when the maximum frequency is 30 MHz and below. Check if the frequency setting is correct.
432	Gain-Phase calibration not allowed with Port 1 DC Bias	The calibration cannot be performed. The calibration for gain-phase measurement should be performed when DC bias is selected at LF port. Select the LF port for DC bias.
421	Gain-Phase measurement not allowed at f>30MHz	The measurement cannot be performed. The maximum frequency for the Gain-Phase measurement is 30 MHz and below. Check if the frequency setting is correct. When this error occurs, the trigger is held for the channel.
422	Gain-Phase measurement not allowed with Port 1 DC Bias	The measurement cannot be performed. When the Gain-Phase measurement is performed, you can not select the port1 for DC bias. Select the LF port for DC bias. When this error occurs, the trigger is held for the channel.
702	Gain-Phase R ch IF overload	The excessive signal has been input on port R. When this error occurs, the attenuator and input impedance settings are set at 20 dB and 1 MΩ, respectively and signal source and DC bias is turned OFF. Check the input signal level on port R.
722	Gain-Phase R ch overload	The excessive signal has been input on port R. When this error occurs, the attenuator and input impedance settings are set at 20 dB and 1 MΩ, respectively and signal source and DC bias is turned OFF. Check the input signal level on port R.
703	Gain-Phase T ch IF overload	The excessive signal has been input on port T. When this error occurs, the attenuator and input impedance settings are set at 20 dB and 1 MΩ, respectively and signal source and DC bias is turned OFF. Check the input signal level on port T.

723	Gain-Phase T ch overload	The excessive signal has been input on port T. When this error occurs, the attenuator and input impedance settings are set at 20 dB and 1 MΩ, respectively and signal source and DC bias is turned OFF. Check the input signal level on port T.
-105	GET not allowed	A group execution trigger (GET) has been received in the program message (see 7.7, IEEE488.2).

H

No.	Message	Description
-240	Hardware error	When hardware error occurs, this error is displayed.
-114	Header suffix out of range	The unit of the header is outside the range. The header is invalid in the unit for numeric parameters following a SCPI command.

I

No.	Message	Description
-224	Illegal parameter value	The parameter value is not suitable. This error occurs when the CALC:PAR:DEF command is used to specify an S-parameter that does not exist in the model (specify S44), for example.
-282	Illegal program name	This error occurs when a nonexistent VBA program name is specified by the PROG:SEL:NAME command.
-213	Init ignored	Because another measurement is in progress, the request for initiating a measurement ("INIT" command) is ignored.
35	Insufficient ECal module memory	This error occurs when the embedded memory is insufficient to save the user property in ECal module during the user definition ECal processing.
-161	Invalid block data	Block data has been expected, but the block data that appears is invalid for some reason (see 7.7.6.2, IEEE488.2). The END message is received before the length of block data has been filled, for example.
28	Invalid calibration	If the type of calibration is not specified or not correct when partial overwrite is executed with the GPIB command, this error occurs. This error does not occur

	method	for operation with front keys.
-101	Invalid character	An invalid character exists in the program message character string.
-141	Invalid character data	An invalid character is found in the character data element, or the parameter received is not valid.
-121	Invalid character in number	A character that is invalid for the data type subject to syntactic analysis has been received. For example, a letter is found in a decimal numeric value or a numeric character "9" in octal data.
500	Invalid equation expression	The equation expression used in Equation Editor is not valid.
501	Invalid equation label	The equation label used in Equation Editor is not valid.
-171	Invalid expression	The expression-data element is invalid (see 7.7.7.2, IEEE488.2). For example, parentheses are not paired, or illegal characters are used.
-232	Invalid Format	A legal program data element was parsed but could not be executed because the data format or structure is inappropriate. For example when loading memory tables or when sending a SYSTem:SET parameter from an unknown instrument.
-103	Invalid separator	The parser (a syntactic analysis program) had been expecting a delimiter, but a character that is not a delimiter has been sent.
-151	Invalid string data	Character string data has been expected, but the character string data that appears is invalid for some reason (see 7.7.5.2, IEEE488.2). The END message is received before the ending quotation mark character appears, for example.
-131	Invalid suffix	The suffix does not comply with the syntax defined in 7.7.3.2, IEEE488.2. Or it does not suit E5061B.

L

No.	Message	Description

713	LF Out source current overload	The current of LF Out signal exceeds the limit. LF Out has been turned OFF automatically.
712	LF Out source voltage overload	The voltage of LF Out signal exceeds the limit. LF Out has been turned OFF automatically.
53	Log sweep requires 2 octave minimum span	The span of sweep range did not satisfy the requirement for logarithmic sweep. The sweep type is automatically changed to linear sweep when this error occurs. For example, this error occurs when, with the start and stop frequency are set 1 MHz and 2 MHz respectively, the sweep type is changed to logarithmic sweep. Set the stop frequency to more than four times as many as the start frequency. And then select logarithmic sweep.

M

No.	Message	Description
420	Measurement below 100kHz not allowed with AC-couple mode	The measurement can not be performed. The minimum frequency for the S-parameter measurement with AC couple is 100 kHz and above. Check if the frequency setting is correct. When this error occurs, the trigger is held for the channel.
-109	Missing parameter	The number of parameters is less than that required for the command, or the parameter has not been entered. For example, the command SCPI.SENSe(Ch).SWEep.POINts (SENS:SWE:POIN) requires one more parameter. Therefore, when a message "SENS:SWE:POIN" is sent, this error is returned.
96	Must be more than 2 points for analysis	The number of points are not sufficient. At least 2 points are required to perform the equivalent circuit analysis.

N

No.	Message	Description

52	No valid memory trace	This error occurs when you have executed either DISP:WIND:TRAC:MEM ON command to display memory trace, or any other command to enable data calculation using memory trace (CALC:MATH:FUNC command with other than NORM is specified), though no valid data exists in memory trace. This error is not generated by front key operations.
80	Not enough points for connector model	Insufficient points for the connector model
-120	Numeric data error	An error resulting from the numeric value data (including numeric value data having no decimal point representation) has occurred. A numeric value error other than Errors -121 through -129 has occurred.
-128	Numeric data not allowed	An numeric-value-data element (not violating the standard) has been received at a position where this instrument does not accept one.

O

No.	Message	Description
200	Option not installed	The received command has been ignored because of the mismatch between the contents of an option for this instrument and the command. It also occurs when you try to enable the fault location function in a model not having the fault location option. This holds true for the SRL (Structural Return Loss) option. This error is not generated by front key operations.
-225	Out of memory	Insufficient memory is available in this instrument to perform the required operation.

P

No.	Message	Description
-220	Parameter error	When a parameter-related error other than Errors -221 through -229 occurs, that error is displayed.
-108	Parameter not allowed	The number of parameters exceeds the one required for the command.

		For instance, when a program message ":SENS1:SWE:TYPE LIN, SEGM" is sent instead of a correct program message with a command ":SENS1:SWE:TYPE LIN" which requires a parameter, the instrument receives the message as the number of parameters is invalid. See the command reference to confirm the required number of parameters.
41	Peak not found	This error occurs when, after specifying a peak and executing the CALC:MARK:FUNC:EXEC and CALC:FUNC:EXEC commands, the specified peak is not found in the marker search analysis.
220	Phase lock loop unlocked	This error occurs when the PLL circuit of this instrument becomes unlocked while the measurement is in progress. The measurement value is not correct. This error may occur when an external reference out of specification is connected to this instrument. Should an error occur with an external reference not connected, this instrument is faulty. Contact an Agilent Technology sales office or the company from which you bought the instrument.
400	Port 1 DC Bias not enabled	Port 1 DC BIAS can not be enabled
700	Port 1 receiver IF overload	The excessive signal has been input on port 1. When this error occurs, signal source and DC is turned OFF. Check the input signal level on port 1.
720	Port 1 receiver overload	The excessive signal has been input on port 1. When this error occurs, signal source and DC is turned OFF. Check the input signal level on port 1.
711	Port 1 source current overload	The current of Port 1 signal exceeds the limit. Signal source and DC is turned OFF automatically.
710	Port 1 source voltage overload	The voltage of Port 1 signal exceeds the limit. Signal source and DC is turned OFF automatically.
701	Port 2 receiver IF overload	The excessive signal has been input on port 2. When this error occurs, signal source is turned OFF. Check the input signal level on port 2.

721	Port 2 receiver overload	The excessive signal has been input on port 2. When this error occurs, signal source is turned OFF. Check the input signal level on port 2.
241	Power on test failed	This error occurs when the power-on test fails, indicating a failure of this instrument. Contact an Agilent Technology sales office or the company from which you bought the instrument.
61	Power unleveled	The out power level exceeds available range. For example, if the level after correction exceeds the power level that can be outputted when correcting the power level with the power slope feature, this error occurs. Check that the power level is set correctly and the correction value of the power slope is set correctly.
121	Print failed	This error occurs when printing fails for reasons other than Error 120, Printer error.
120	Printer error	This error occurs when the previous printing is still in progress or the printer fails (offline, short of paper, etc.) at the time of outputting the display image on the LCD screen to the printer (SCPI.HCOPy.IMMediate command).
-284	Program currently running	This error occurs when the PROG:SEL:STAT RUN command is executed with the VBA program in the Run state.
-112	Program mnemonic too long	The length of the header exceeds 12 characters (see 7.6.1.4.1, IEEE488.2).
-286	Program runtime error	An error occurring when VBA is executed.

Q

No.	Message	Description
-430	Query DEADLOCKED	The state that generates a "DEADLOCKED" Query error (see 6.3.1.7, IEEE488.2). This error occurs when both input and output buffers have become full, preventing the instrument from continuing processing, for example.

-400	Query error	A comprehensive query error has occurred showing that this instrument cannot detect a more detailed error. This code simply shows that a query error defined in 11.5.1.1.7 and 6.3, IEEE488.2 has occurred.
-410	Query INTERRUPTED	The state that generates a "INTERRUPTED" Query error (see 6.3.2.3, IEEE488.2). This error occurs when data bytes (DAB) or GET are received before the transmission of the response when a query is not completed, for example.
-420	Query UNTERMINATED	The state that generates an "UNTERMINATED" Query error (see 6.3.2, IEEE488.2). This error occurs when this instrument is designated as the talker and an incomplete program message is received, for example.
-440	Query UNTERMINATED after indefinite response	After a query asking for an indefinite response has been run, another query is received in the same program message (See 6.5.7.5.7, IEEE488.2).
-350	Queue overflow	The queue contains a specific code in place of the code which caused this error. The code indicates that the error occurred because of no space is available in the queue, but the error is not recorded.

R

No.	Message	Description
105	Recall failed	This error occurs when reading an instrument status file (State01.sta, etc.) (MMEM:LOAD:STAT command) fails.
45	Reference tracking not allowed with Log-Y scale	Reference tracking is not allowed with Log-Y scale.

S

No.	Message	Description
106	Save failed	This error occurs when writing an instrument status file (State01.sta, etc.) (MMEM:STOR:STAT command)

		fails.
33	Selected parameter not valid for confidence check	This error occurs when the parameter except S-parameter has been selected, while using the confidence check function for calibration coefficient. Select S-parameter (S11, S21, S12 or S22).
57	SnP request not valid	This error occurs when you try to save data to a Touchstone file but no measurement has been executed. This error also occurs when you try to save a Touchstone file with power sweep measurement specified.
50	Specified channel hidden	This error occurs when an attempt is made to activate a channel not on display using the DISP:WIND:ACT command. This error is not generated by front key operations.
23	Specified error term does not exist	The error occurs when the calibration coefficient type of the parameter specified in the read/write command of the calibration coefficient (SENS:CORR:COEF) is invalid for the calibration type selected by the calibration type selection command (SENS:CORR:COEF:METH:xxxx.) This error is not generated by front key operations.
21	Specified ports overlapped	This error occurs when a port number is duplicated in a command requiring two or more port numbers as parameters. Such commands are, for example, SENS:CORR:COLL:THRU 2,2. Specify port setup correctly to avoid duplication of ports. This error is not generated by front key operations.
51	Specified trace dose not exist	This error occurs when CALC:PAR:SEL command is executed to activate more traces than specified by CALC:PAR:COUN command. This error is not generated by front key operations.
455	Specified Z Conversion not allowed with Gain-Phase Port	This error occurs when SCPI.SENSe(1-4).Z.METHod command is executed and the selected method does not match the defined Gain-Phase port.
456	Specified Z	This error occurs when SCPI.SENSe(1-4).Z.METHod

	Conversion not allowed with S-Param Port	command is executed and the selected method does not match the defined S-Parameter port.
-150	String data error	When a character-string-data element is put to syntactic analysis, an error not corresponding to one of Error Numbers -151 through -159 occurs.
-158	String data not allowed	A character-string-data element has been received at a position where this instrument does not accept one.
-138	Suffix not allowed	A suffix is attached to a numeric value element to which a suffix is not allowed to be attached.
-134	Suffix too long	The unit is too long. The unit is expressed in 12 or more characters (see 7.7.3.4, IEEE488.2).
-102	Syntax error	A command or data type that is not recognized exists.
-310	System error	One of the errors designated as "system errors" in this instrument has occurred.

T

No.	Message	Description
40	Target value not found	This error occurs when the target is not found during the marker search analysis after specifying the target and executing the CALC:MARK:FUNC:EXEC and CALC:FUNC:EXEC commands. This error occurs also when the bandwidth is not found after executing the bandwidth marker command, CALC:MARK:BWID:DATA?
-124	Too many digits	The number of digits of the argument of the decimal numeric-value-data element exceeds 255 with the preceding 0 removed (see 7.7.2.4.1, IEEE488.2).
-223	Too much data	The block-, expression-, or character-string-type program data that has been received conforms with the standard. But it exceeds the amount that can be processed under the condition of the memory or conditions specific to memory-related devices. In this instrument, this error occurs when the number of characters exceeds 254 in a character-string parameter.

-211	Trigger ignored	This instrument receives and detects a trigger command ("TRIG") or an external trigger signal. But it is ignored due to the timing condition (This instrument is not in the wait-for-trigger state, for example). Change the setup so that a trigger command or an external trigger signal can be sent after the instrument has entered the wait-for-trigger state.

U

No.	Message	Description
-113	Undefined header	A command not defined in this instrument, though not illegal in the syntactic structure, has been received. For example, when a message ":DISP:WIND1:TABL:MEM ON" is sent to a correct program message ":DISP:WIND1:TRAC1:MEM ON," the message sent is received as an undefined command by this instrument. See the command reference and use correct commands. This error occurs also when a port that does not exist in this model is specified in a command specifying a port number as an index. Such commands are SENS:CORR:EXT:AUTO:PORTn, SENS:CORR:COLL:ADAPn:xxxx, CALC:SRL:CONNn:xxxx, and SENS:CORR:EXT:PORTn:xxxx; they include "n" as a part.
34	User characterization not found in module	This error occurs when the selected user profile is not detected in the ECal memory, while reading it from the ECal module, written by the user definition ECal.

V

No.	Message	Description
30	Valid Ecal module not found	This error occurs when a required ECal module is not connected on the E5061B. This error occurs, for example, when the Ecal characterization is executed with 4 port Ecal. Use 2 port Ecal in this case.

Warning Message

A warning message is displayed in the instrument message/Warning area in the lower left part of the display against a gray background. Pushing a front panel key or executing :DISP:CCL command clears the message.

This message simply appears on the display, being not known to a remote environment such as a GPIB. This message is not displayed when another error (against a red background) is already displayed in the instrument message/Warning area.

Warning Messages during Measurement

Messages	Description
Calibration data lost ([parameter])/	One or more of the below calibration data is lost. The lost parameter is indicated in the message, for example, calibration data lost (band information): (nominal value), (band information), (frequency reference), (synthesizer gain), (source output power), (DC bias level), (mixer offset), (receiver IF range), (port characteristics), (gain phase ratio), (absolute gain), (distortion), (src dc offset), (constants for sweep controller)
Invalid key code	The key code entered to enable or disable an option is incorrect. Enter the correct key code to proceed.
Load VBA project error	This warning message is displayed when the attempt to upload a VBA project fails. Check the validity of the VBA project file.
Peak not found	This warning message is displayed when, with the tracking turned ON, the peak specified by the marker search has not been found by the time the sweep is finished (with the tracking executed).
Target value not found	This warning message is displayed when, with the tracking turned ON, the target specified by the marker search has not been found by the time the sweep is finished (with the tracking executed). This warning message is displayed also when, with the bandwidth marker displayed, the setting for the bandwidth marker is changed at the end of the sweep, or when, with the active marker changed or moved, the bandwidth is not found.

Unable to find help file	This warning message appears when Help file could not be executed when you press the **Help** key. The file is either corrupted or unavailable.
Unable to find help id file	This warning message appears when Help file could not be executed when you click a button at the menu bar and press the **Help** key to execute a help topic related to the button. The help id file is either corrupted or unavailable.

Default Conditions

Analysis

Key Operation			Default Value	Preset	Backup	Save/Recall
Gating	**Gating**		OFF	←		√
	Start		-10 ns	←		√
	Stop		10 ns	←		√
	Center		0 s	←		√
	Span		20 ns	←		√
	Type		Bandpass	←		√
	Shape		Normal	←		√
	Unit		Seconds	←		√
Fault Location	**Fault Location**		OFF	←		√
	Start		-10 ns	←		√
	Stop		10 ns	←		√
	Center		0 s	←		√
	Span		20 ns	←		√
	Type		Bandpass	←		√
	Window		Normal	←		√
		Impulse Width	650.3957 psec	←		√
		Step Rise	328.6776 psec	←		√

		Kaiser Bata	6	←		√
	Reflection Type		Round Trip	←		√
	Cable Loss		0 dB/μs	←		√
	Unit		Seconds	←		√
SRL	**SRL**		OFF	←		√
	Port 1/2 Connector	**Length**	0 m	←		√
		Capacitance	0 F	←		√
	Auto Z		ON	←		√
	Z Cutoff Freq		210 MHz	←		√
	Manual Z		50 Ω	←		√
Conversion	**Conversion**		OFF	←		√
	Function		Z:Reflection	←		√
Limit Test	**Limit Test**		OFF	←		√
	Limit Line		OFF	←		√
	Clip Lines		ON	←		√
	Limit Line Offsets	**Stimulus Offset**	0 Hz	←		√
		Amplitude Offset	0 dB	←		√
	Fail Sign		ON	←		√

Ripple Limit	**Ripple Limit Test**	OFF	←		√
	Ripple Limit	OFF	←		√
	Ripple Value	Off	←		√
	Ripple Band	1	←		√
	Fail Sign	ON	←		√
Bandwidth Limit	**BW Test**	OFF	←		√
	BW Display	OFF	←		√
	BW Marker	OFF	←		√
	N dB Points	3 dB	←		√
	Min Bandwidth	10 kHz	←		√
	Max Bandwidth	300 kHz	←		√
	Fail Sign	ON	←		√
Equivalent Circuit	**Select Circuit**	A	←		
	Display	OFF	←		

Avg

Key Operation	Default Value	Preset	Backup	Save / Recall
Avg Factor	16	←		√
Averaging	OFF	←		√
Avg Trigger	OFF	←		√
SMO Aperture	1.5 %	←		√
Smoothing	OFF	←		√
IF Bandwidth	30 kHz	←		√
IFBW Auto	OFF	←		√
IFBW Auto Limit	30 kHz	←		√

Cal

Key Operation				Default Value	Preset	Backup	Save / Recall
Correction				OFF	←		√
Calibrate	**Response (Open)**	**Select Port**		1	←		
	Response (Short)	**Select Port**		1	←		
	Response (Thru)	**Select Ports**		2-1 (S21)	←		
	Enhanced Response	**Select Ports**		2-1 (S21)	←		
	1-Port Cal	**Select Port**		1	←		
	Adapter Removal	**Port 1/2**	**Length**	Auto (0 s)	←		
		Cal Kit		85032B /E	←		
	Impedance Calibration	**Open**		Open (f)	←		√
		Short		Short (f)	←		
		Load		Broad Band	←		
ECal	**Enhanced Response**			2-1 (S21 S11)	←		√
	Isolation			OFF	←		√
	Characterization			Factory	←		√
	Orientation			Auto	←		√
Property				OFF	←		√
Cal Kit				85032B /E	←		√
Modify Cal Kit	**Define STDs**			Define STDs	←		√

	Label Kit			85032B /E			
Port Extensions	**Extensions**			OFF	←		√
	Auto Port Extension	**Select Ports (Port 1/Port 2)**		ON	←		√
		Method		Current Span	←		√
		Method	**User Span Start**	5 Hz	←		√
		Method	**User Span Stop**	3 GHz	←		√
		Include Loss		OFF	←		√
		Adjust Mismatch		OFF	←		√
	Extension Port 1/Port 2	**Extension**		0 s	←		√
		Loss 1		OFF	←		√
		Loss 1		0 dB	←		√
		Freq1		1 GHz	←		√
		Loss 2		OFF	←		√
		Loss 2		0 dB	←		√
		Freq2		1 GHz	←		√
		Loss at DC		0 dB	←		√
Velocity Factor				1	←		√
Set Z0				50 Ω	←		√
Fixture Compensation	**Fixture Compensation**			OFF	←		√
		Compen Open		OFF	←		√

		Compen Short	OFF	←		√
		Compen Load	OFF	←		√
	Compen STDs	**Open Ls**	0	←		√
		Open Gs	0	←		√
		Open Cp	0	←		√
		Short Ls	0	←		√
		Short Rs	0	←		√
		Short Cp	0	←		√
		Load Ls	0	←		√
		Load Rs	50	←		√
		Load Cp	0	←		√
	Fixture		None	←		
	Modify User Fxt.		0	←		
	Z Port Extension		0	←		

Center

Key Operation	Default Value	Preset	Backup	Save/Recall
	1.50005 GHz	←		√

Display

Key Operation	Default Value	Preset	Backup	Save/Recall
Allocate Channels	x1	←		√
Number of Traces	1	←		√
Allocate Traces	x1	←		√
Display	Data	←		√
Data Math	OFF	←		√
Equation	OFF	←		√
Edit Title Label	""	←		√
Title Label	""	←		√
Graticule Label	ON	←		√
Invert Color	OFF	←		√
Frequency	ON	←		√
Update	ON	←		√

Format

Key Operation	Default value	Preset	Backup	Save/Recall
	Log Mag	←		√

Macro Setup

Key Operation	Default Value	Preset	Backup	Save / Recall
Echo Window	OFF	←		√

Marker Fctn

Key Operation		Default Value	Preset	Backup	Save / Recall
Discrete		OFF	←		√
Couple		ON	←		√
Marker Table		OFF	←		√
Statistics		OFF	←		√
Flatness		OFF	←		√
RF Filter Status		OFF	←		√
Annotation Options	Marker Info X Pos	1 %	←		√
	Marker Info Y Pos	1 %	←		√
	Align	ON	←		√
	Active Only	ON	←		√

Marker Search

Key Operation		Default Value	Preset	Backup	Save/Recall
Peak	**Peak Excursion**	3 dB	←		√
	Peak Polarity	Positive	←		√
Target	**Target Value**	0 dB	←		√
	Target Transition	Both	←		√
Multi Peak	**Peak Excursion**	3 dB	←		√
	Peak Polarity	Positive	←		√
Multi Target	**Target Value**	0 dB	←		√
	Target Transition	Both	←		√
Tracking		OFF	←		√
Search Range	**Search Range**	OFF	←		√
	Start	0 Hz	←		√
	Stop	0 Hz	←		√
	Couple	ON	←		√
Bandwidth		OFF	←		√
Bandwidth Value		-3 dB	←		√
Notch		OFF	←		√
Notch Value		-3 dB	←		√

Marker

Key Operation	Default Value	Preset	Backup	Save / Recall
Marker1	OFF (Marker 1 is turned ON immediately after the marker softkey menu is displayed)	←		√
Ref Marker Mode	OFF	←		√

Meas

Key Operation			Default value	Preset	Backup	Save / Recall
Measurement Port			S-Param	←		√
			S11	←		√
Gain-Phase			T/R	←		√
Gain-Phase Setup	**T Input Z / R Input Z**		1M Ω	←		√
	T Attenuator / R Attenuator		20 dB	←		√
DC Monitor Setup	**Function**		Vdc Bias	←		√
	DC Monitor On Sweep End	**Monitor**	OFF	←		√

Save/Recall

Key Operation		Default Value	Preset	Backup	Save/Recall
Save Type		State & Cal	←		√
Channel/Trace		Disp Only	←		√
Save SnP	**SnP Format**	Auto	←		√

Scale

Key Operation		Default value	Preset	Backup	Save / Recall
Divisions		10	←		√
Scale/Div		10.000 dB/div	←		√
Reference Position		5 Div	←		√
Reference Value		0 dB	←		√
Reference Tracking	**Tracking**	Off	←		√
	Track Frequency	0 Hz	←		√
Electrical Delay		0 s	←		√
Phase Offset		0 °	←		√
Y-Axis		Linear	←		√
Log Y-Axis	**Top Value**	1000 dB	←		√
	Bottom Value	0.001 dB	←		√

Span

Key Operation	Default Value	Preset	Backup	Save / Recall
	2.9999 GHz	←		√

Start

Key Operation	Default Value	Preset	Backup	Save / Recall
	100 kHz	←		√

Stop

Key Operation	Default Value	Preset	Backup	Save / Recall
	3 GHz	←		√

Sweep Setup

Key Operation			Default Value	Preset	Backup	Save/Recall
Power	**Power**		0 dBm	←		√
	Port Couple		ON	←		√
	Port Power	**Port 1 Power**	0 dBm	←		√
		Port 2 Power	0 dBm	←		√
		LF OUT Power	0 dBm	←		√
	Slope [xx dB/GHz]		0 dB/GHz	←		√
	Slope [ON/OFF]		OFF	←		√
	RF Out		ON	←		√
CW Freq			100 kHz	←		√
Sweep Time AUTO			ON	←		√
Sweep Delay			0 s	←		√
Points			201	←		√
Sweep Type			Lin Freq	←		√
Edit Segment Table	**Freq Mode**		Start/Stop	←		√
	List IFBW		OFF	←		√
	List Power		OFF	←		√
	List Delay		OFF	←		√
	List Time		OFF	←		√
Segment Display			Order Base	←		√
DC Bias			OFF	←		
DC Bias Port			LF Out	←		√
DC Bias Level			0 V	←		√

System

Key Operation					Default Value	Preset	Back up	Save / Recall
Invert Image					ON	←		√
Misc Setup	**Beeper**	**Beep Complete**			ON	←		√
		Beep Warning			ON	←		√
	GPIB Setup	**Talker/Listener Address**			17	Non-changing	√	
	Network Setup	**Telnet Server**			OFF	Non-changing	√	
		SICL-LAN Server			OFF	Non-changing	√	
		SICN-LAN Address			17	Non-changing	√	
		Web Server			ON	Non-changing	√	
	Clock Setup	**Show Clock**			ON	←		√
	Key Lock	**Front Panel & Keyboard Lock**			OFF	←		
		Touch Screen & Mouse Lock			OFF	←		
	Display Setup	**Color Setup**	**Normal** (**Invert** is the same)	**Data Trace 1**	Red:5 Green:5 Blue:0	←		√
				Data Trace 2	Red:0 Green:5 Blue:5	←		√

				Data Trace 3	Red:5 Green:0 Blue:5	←		√
				Data Trace 4	Red:0 Green:5 Blue:0	←		√
				Mem Trace 1	Red:3 Green:3 Blue:0	←		√
				Mem Trace 2	Red:0 Green:3 Blue:3	←		√
				Mem Trace 3	Red:3 Green:0 Blue:3	←		√
				Mem Trace 4	Red:0 Green:3 Blue:0	←		√
				Graticule Main	Red:3 Green:3 Blue:3	←		√
				Graticule Sub	Red:1 Green:1 Blue:1	←		√
				Limit Fail	Red:5 Green:0 Blue:0	←		√
				Limit Line	Red:3 Green:0 Blue:0	←		√
				Backgrou nd	Red:0 Green:0 Blue:0	←		√
		Magnification			Normal		√	
	Preset Setup	**State**			Factory	Non-changing	√	
		Confirm			ON	Non-changing	√	
Backlight					ON	←		

S-Param Port Couple				DC	←		
Overload Recovery	**RF Out**			ON	←		√
	DC Bias			OFF	←		
	S-Param Port Couple			DC	←		√
	T Input Z			1M Ω	←		√
	R Input Z			1M Ω	←		√
	T Attenuator			20 dB	←		√
	R Attenuator			20 dB	←		√
Service Menu	**DC Bias Range**			Auto	←		√
	Init Src Ctrl	**RF Out**		ON	←		√
		DC Bias		ON	←		√
	Security Level			None	←		√
	Band Wait			0 sec	←		√
	Service Functions	**Distortion cal**		ON	←		

Trigger

Key Operation	Default Value	Preset	Backup	Save/Recall
Continuous	Continuous (Ch1) Hold (Other Ch)	←		√
Trigger Source	Internal	←		√
Trigger Event	On Sweep	←		√
Trigger Scope	All Channel	←		√
Ext Trig Input	Negative Edge	←		√
Trigger Delay	0 sec	←		√
Ext Trig Output	OFF	←		√
Polarity	Positive Pulse	←		√
Position	After Point	←		√
Pulse Width	1 μs	←		√

Troubleshooting

Troubleshooting

This section describes the steps you should take when you believe that the Agilent E5061B is operating improperly. The results of these simple investigative procedures may help you avoid the down-time and inconvenience of repair service. The troubleshooting instructions are divided into three categories.

When all troubleshooting measures are taken but it does not work. Contact Agilent Technology's Customer Contact.

- Troubleshooting during Startup
- Troubleshooting during Operation
- Troubleshooting for External Devices

Troubleshooting during Startup

Symptom	Solution
Turning ON (|) the standby switch does not start up the system.	• Confirm that the power cable is properly plugged in. • Confirm that the line switch on the rear panel is turned ON.
Standby switch color is red and does not start up the system.	There is some trouble on the fan. Turn OFF and ON the line switch once, then turn ON this standby switch.
The system starts up, but it automatically shuts down immediately.	Execute the system recovery.
The system starts up, but it enters the service mode (The instrument status bar in the lower right part of the screen displays SVC in red).	Execute the system recovery.
The measurement screen appears after startup, but the date and time displayed on the instrument status bar in the lower right part of the screen differ greatly from the previous settings.	Execute the system recovery.
The measurement screen appears after startup, but the power-on test fails, with Error Message 241 appearing against the red background in the instrument message/warning area in the lower left part of the screen.	Execute the system recovery.

Troubleshooting during Operation

Symptom	Solution
The Error Message "Port N receiver overload" (N denotes a port number) is Displayed. During the measurement of an amplifier, Error Messages 720 through 721 "Port N receiver overload" (N denotes a port number) are displayed. This error occurs when the input to a test port exceeds the maximum input level in the measurement of an amplifier. The measurement value obtained in such cases is not correct. In the worst case, a failure (damage to the receiver) may occur.	• Disconnect any cable and adapter in the test ports. • Change the measurement condition so that the input to the test port does not exceed the maximum input level.
A clearly abnormal measurement value The measurement value is not reproducible, or clearly abnormal.	• Confirm that the DUT, connection cables, and other parts are connected correctly. • Confirm that the connectors and cables used to connect the DUT are free from damage and poor contact. • Confirm that the calibration has been executed correctly. If you have not acquired a correct error correction factor, you cannot obtain a correct measurement value. • Confirm that the calibration kit is selected correctly. • Confirm that the calibration kit is defined correctly. • Confirm that if the stimulus signal output is turned ON.
The system cannot operate manually (Front Panel Keys, Keyboard, Touch Screen and Mouse) The keyboard or mouse becomes inoperable.	• Confirm that the keyboard or mouse is connected correctly. When it is connected correctly, turn OFF the power once, and restart the system.
The front panel key or keyboard becomes inoperable.	• Using the mouse, turn **System > Key Lock > Front Panel & Keyboard Lock OFF.**
The touch screen becomes inoperable.	• Using the front panel keys, press **System** > **Key Lock** > **Touch Screen & Mouse Lock**

	OFF to turn OFF the lock. • Execute the calibration of the touch screen.
The mouse becomes inoperable.	• Using the front panel keys, press **System** > **Key Lock** > **Touch Screen & Mouse Lock** to turn OFF the lock.
All of the front panel keys, keyboard, and mouse become inoperable.	• Confirm that the keyboard or mouse is connected correctly. When it is connected correctly, turn OFF the power once, and restart the system.
The keyboard and mouse have been connected after power-on.	• Turn OFF the power once, and restart the system. When taking all these measures does not recover operability, there is a possibility of a failure.
The screen freezes and all operations become impossible. The measurement in progress or screen update is stalled and all of the front panel keys, keyboard, mouse, and touch screen are inoperable.	• Press the standby switch to turn OFF the power once, and restart the system.
The system freezes while in operation.	• Press the standby switch to turn OFF the power once, and restart the system.
The rear cooling fan does not operate.	• There is a possibility of a failure.
The sweep action stops during measurement or is not executed. An error or warning message appears.	• There is a possibility of a failure.
An error or warning message is displayed on the instrument message/warning area in the lower part of the screen	• Refer to Error Messages and Warning Messages.

Troubleshooting for External Devices

Symptom	Solution
Cannot output to a printer Cannot output a measurement screen or	• Confirm that the power to the printer is turned ON and that the line cable

data to a printer. Attempting to output to a printer causes Error Messages 120 and 121 to appear.	is connected correctly. • Confirm that the connector cable of the printer is connected correctly. • Confirm that the printer is online. • Confirm that the printer has not run out of paper. • Confirm that the printer has not run out of ink.
Does not respond to an external controller/fails to function normally A GPIB device does not respond to the external controller, or fails to function normally.	• Confirm that the GPIB address is defined correctly. • Confirm that the GPIB cable is connected. • Confirm that another instrument connected by the GPIB cable has the same GPIB address. • Confirm that the GPIB cable connection forms a loop.

Maintenance

Maintenance

- Backing Up the Data
- Cautions Applicable to Requesting Repair, Replacement, Regular Calibration, etc.
- Cleaning this Instrument
- Replacement of Parts with Limited Service Life
- System Recovery
- Updating Firmware
- Service Functions
- Removing Log Data

Backing Up the Data

Be sure to regularly back up your important data (including program) files in this instrument to a CD-R or other backup medium. Agilent Technologies shall not be liable for any data damages caused by improper functin of this instrument.

Making Backup Files

You can make backup files on the hard disk of an external PC using following methods.

You can access to drive D: of the E5061B from an external PC via LAN, and copy your important data files on the drive D: to the external PC. See Accessing Hard Disk of E5061B from External PC for details.

You can transfer your important data files on the drive D: of the E5061B to the external PC using :MMEM:TRAN command via GPIB.

CAUTION Do not modify any files and folders in drives other than drive D:. Doing so will cause malfunction.

Other topics about Maintenance

Cautions Applicable to Requesting Repair, Replacement, Regular Calibration, etc.

- Backing Up Data in the Hard Disk
- Devices to be Sent Back for Repair or Regular Calibration

Other topics about Maintenance

Backing Up Data in the Hard Disk

The user is requested to back up the stored programs and data into external media by using the instrument's storing function before requesting the Company's Service Center to repair the instrument or replace hard disks.

See Making Backup Files for how to make backup files.

Please take note that the Company will not be held liable to any extent for potential erasure or change of stored programs or data due to the repair or replacement of hard disks performed by the Company. When a hard disk itself fails, the programs and data stored in it cannot be recovered.

Devices to be Sent Back for Repair or Regular Calibration

If it is necessary to send the unit to the Service Center of Agilent Technologies for repair or regular calibration, please follow the instructions below.

Equipment to be Sent

When requesting repair or regular calibration of the unit by our Service Center, send only the E5061B main unit without any installed option you may have ordered. Unless specifically instructed, it is not necessary to send accessories and calibration kits.

Packing

Use the original package and shock absorbers, or equivalent anti-static packing materials, when sending the unit.

Shipping Address

For the location of the nearest Agilent Technologies Service Center, contact the Customer Contact.

Recommended Calibration Period

The recommended calibration period for this instrument is one year. The user is recommended to request the Company's Service Center to perform regular calibration every year.

Cleaning this Instrument

- Cleaning an LCD
- Maintenance of Test Ports and Other Connectors/Ports
- Cleaning Parts Other than the LCD, Test Ports, and Other Connectors/Ports

Other topics about Maintenance

This section describes how to clean the instrument.

To protect yourself from electrical shock, be sure to unplug the power cable from the outlet before cleaning the instrument.

Never clean the internal components of the instrument.

Cleaning an LCD

Use one of the following methods to clean the display surface regularly.

- For normal cleaning, rub the surface gently with a dry, soft cloth.
- When stains are difficult to remove, gently wipe the surface with cloth damped with a small amount of dehydrated ethanol.
  You can clean the standard LCD (no touch screen function) with a cloth dipped in water and then wrung tightly.

NOTE Do not use chemicals other than dehydrated ethanol to wet the cleaning cloth. To clean the touch screen LCD, do not wet the cloth with water.

Maintenance of Test Ports and Other Connectors/Ports

The ports of the E5061B are fitted with BNC and N Types connectors. Stains or other damage to these connectors would significantly affect the accuracy in measurements. Always pay attention to the following precautions.

- Always keep the connectors free from stains and dust.
- Do not touch the contact surface on the connectors.
- Do not plug damaged or scratched connectors into the test ports.
- Use compressed air for cleaning connectors. Do not use abrasives under any circumstance.

The above precautions must also be observed in maintaining connectors and ports other than these test ports.

Cleaning Parts Other than the LCD, Test Ports, and Other Connectors/Ports

To remove stains on parts other than the LCD, test ports, and other connectors/ports of the instrument, wipe them gently with a soft cloth that is dry or wetted with a small amount of water and wrung tightly.

Replacement of Parts with Limited Service Life

This instrument incorporates parts with limited service life as shown in the following table. Using the recommended replacement time as a guide, request the Agilent Service Center to replace these parts. However, a part may need to be replaced at an earlier time than that listed in the table, depending on conditions such as location, frequency of use, and where it is stored.

NOTE Each service life and recommended replacement time listed below is for reference only and does not imply a guarantee of the part's service life.

Part Name	Service Life (Parts supplier reference value)	Recommended replacement time
Hard Disk Drive (HDD)	5 years or 20,000 operating hours, whichever comes earlier. Exchanging hard disk drives causes the contents written after shipment from the factory (LAN setup, etc.)to be initialized to the state at the time of shipment. The programs and data stored in Drive D (user directory) are erased.	3 years
Main fan	50,000 operating hours. The service life may be significantly shorter when used in a dusty and dirty environment.	5 years
CPU fan		
Battery on Mother board	50,000 operating hours. The service life may be shorter if E5061B power has not been turned on for long time.	5 years
Power supply	50,000 operating hours (Depends on the service life of the power supply cooling fun) The service life may be significantly shorter when used in a dusty and dirty environment.	5 years
LCD screen backlight	50,000 operating hours. When the unit is used for automatic measurements in a production line and the on-screen information is not required, the life of the LCD backlight can be saved by turning it OFF. As for the method of turning the backlight OFF, refer to Turning OFF the LCD Screen Backlight.	5 years
Touch screen	One million times (dotting life)	5 years

(function)		
USB receptacle	1,500 cycles insertion/extraction. The service life may be shorter when used in a dusty and dirty environment. In case that the insertion/extraction is in heavy usage such as Ecal in the production line, using USB extension cable may save the USB receptacle life.	N/A

Other topics about Maintenance

System Recovery

By executing system recovery, you can return the Windows operating system of the E5061B to the factory state or the user state at the setting the user performed save user state.

The procedure of system recovery is described in both Installation Guide and Service Guide.

Other topics about Maintenance

Updating Firmware

- Overview
- Procedure

Other topics about Maintenance

Overview

User can update E5061B firmware by themselves. The latest firmware can be downloaded from the http://www.agilent.com/find/ena_support.

Updating firmware does not include the following software update.

- Windows Operating System
- VBA (Visual Basic for Application)
- Driver for Windows
- Calibration Constant Data

Procedure

NOTE User should log in as "Instruments", and user should have administrator authority to perform firmware update.

1. Download the latest firmware from the download site. It is prepared as execution file (**E5061B_xx.xx.exe**).
2. Run the **E5061B_xx.xx.exe** to extract the **E5061B.msi**.

   If you use your local PC to download, save the file to a USB mass storage device in order to move it to E5061B, then connect the USB mass storage device into the front USB port of the E5061B.
3. Press **System** > **Service Menu** > **Update Firmware**. Then, the password dialog box appears.
4. Type **e506xa** as the password, then click **OK**.
5. In Open dialog box, select **E5061.msi**, then press **Open**. Then the windows installer appears.
6. Follow the instruction of windows installer.
7. After the installation is finished, the instrument restarts.
8. Press **System** > **Firmware Revision** to confirm the firmware revision you have just installed.

Service Functions

This menu (**System** > **Service Menu** > **Test** and **Service Functions**) provides information about various test related to the E5061B.

For more information about any test related to the E5061B, like System Test, please refer to the E5061B Service manual which is available at www.agilent.com/find/e5061b-manual.

Under the service menu, the following topic is described in this help.

Other topics about Maintenance

Removing Log Data

The E5061B creates automatic log of data for troubleshooting purpose. For security reasons, if this data needs to be deleted, then SCPI.SERVice.LOGGing.CLEar command can be used to clear the log recorded by the E5061B.

The log file stores data related to:

- Power ON time
- Number of times of power ON
- Result of power ON test
- Number of switching times of internal mechanical relay
- Number of times of overload
- Event Log

Measurement Accessories

Measurement Accessories

- Calibration Kits
- System Accessories

Agilent Technologies provides various probes, cables and adapters. Refer to http://www.agilent.com/find/na_accessories and E5061B Configuration Guide for more information.

Calibration Kits

- Overview
- Mechanical Calibration Kit
- ECal

Other topics about Measurement Accessories

Overview

Calibration kits are used to improve the accuracy of the analyzer in various measurements.

The electronic calibration kit reduces the time required for calibration, misconnections, and wear and tear on connectors since it requires fewer changes of connection than the mechanical type.

Specifications for calibration kits and the availability of particular calibration kits are subject to change without prior notice.

Refer to http://www.agilent.com/find/na_accessories for more information.

Mechanical Calibration Kit

Model Name	Description	Connector Type	Frequency Range
85032B/E	Calibration Kit	Type-N, 50 Ω	DC to 6 GHz
85032F	Mechanical Calibration Kit	Type-N, 50 Ω	DC to 9 GHz
85033D	Calibration Kit	3.5 mm	DC to 6 GHz
85033E	Mechanical Calibration Kit	3.5 mm	DC to 9 GHz
85036B/E	Calibration Kit (Economy Calibration Kit)	Type-N, 75 Ω	DC to 3 GHz
85038A/F/M	Calibration Kit	7-16	DC to 7.5 GHz
85039B	Calibration Kit	Type F , 75 Ω	DC to 3 GHz
85052D	Economy Mechanical Calibration Kit	3.5 mm	DC to 26.5 GHz

For more information on the definition of calibration kit, refer to http://na.tm.agilent.com/pna/caldefs/stddefs.html

ECal (electronic calibration) kit

The ECal supported by E5061B are defined below:

Model Name	Connector Type	Frequency Range
85091B	7 mm	300 kHz to 9 GHz
85091C	7 mm	300 kHz to 9 GHz
85092B	Type N	300 kHz to 9 GHz
85092C	Type N	300 kHz to 9 GHz
85093B	3.5 mm	300 kHz to 9 GHz
85093C	3.5 mm	300 kHz to 9 GHz
85096B	Type N, 75 Ω	300 kHz to 3 GHz
85096C	Type N, 75 Ω	300 kHz to 3 GHz
85098B	7-16	300 kHz to 7.5 GHz
85098C	7-16	300 kHz to 7.5 GHz
85099B	Type F	300 kHz to 3 GHz
85099C	Type F	300 kHz to 3 GHz
N4431A	3.5 mm, Type N	300 kHz to 9 GHz
N4431B	3.5 mm, Type N	9 kHz to 13.5 GHz
N4432A	Type N	300 kHz to 18 GHz
N4433A	3.5 mm	300 kHz to 20 GHz
N4690A	Type N	10 MHz to 18 GHz
N4690B	Type N	300 kHz to 18 GHz
N4691A	3.5 mm	10 MHz to 26.5 GHz
N4691B	3.5 mm	300 kHz to 26.5 GHz
N4692A	2.92 mm	10 MHz to 40 GHz
N4696A	7 mm	10 MHz to 18 GHz
N4696B	7 mm	300 kHz to 18 GHz

NOTE The 4-ports Ecal module such as the N4431A/B, N4432A, and N4433A does not support the user-characterized Ecal VBA macro program. When the user-characterized ECal program is executed, use the 2-ports ECal module.

System Accessories

- USB/GPIB Interface
- GPIB Cables
- Agilent IO Libraries

Other topics about Measurement Accessories

82357A/B USB/GPIB Interface

The 82357A/B can be used to control external GPIB devices by the E5061B. It can also be used to control E5061B by PC with USB. See "82357B USB/GPIB Interface High-Speed USB 2.0" in http://www.agilent.com/find/gpib for more information.

GPIB Cables

The following GPIB cables can be used to connect the analyzer with an external device such as a computer.

Product Number	Length
10833A	1.0 m (3.3 ft)
10833B	2.0 m (6.6 ft)
10833C	3.0 m (9.9 ft)
10833D	0.5 m (1.6 ft)

Agilent IO Libraries

Agilent IO Libraries is a collection of libraries that gives you the ability to use your instruments as a test and measurement program, and utilities that help you quickly and easily connect your instruments to your PC.

See the IO libraries manual for more information. The latest revision of IO Libraries can be downloaded from http://www.agilent.com/find/iolib.

CAUTION Do not update Agilent IO Library on E5061B besides Agilent recommends to do so. Refer to http://www.agilent.com/find/ena_support for more information.

General Principles of Operation

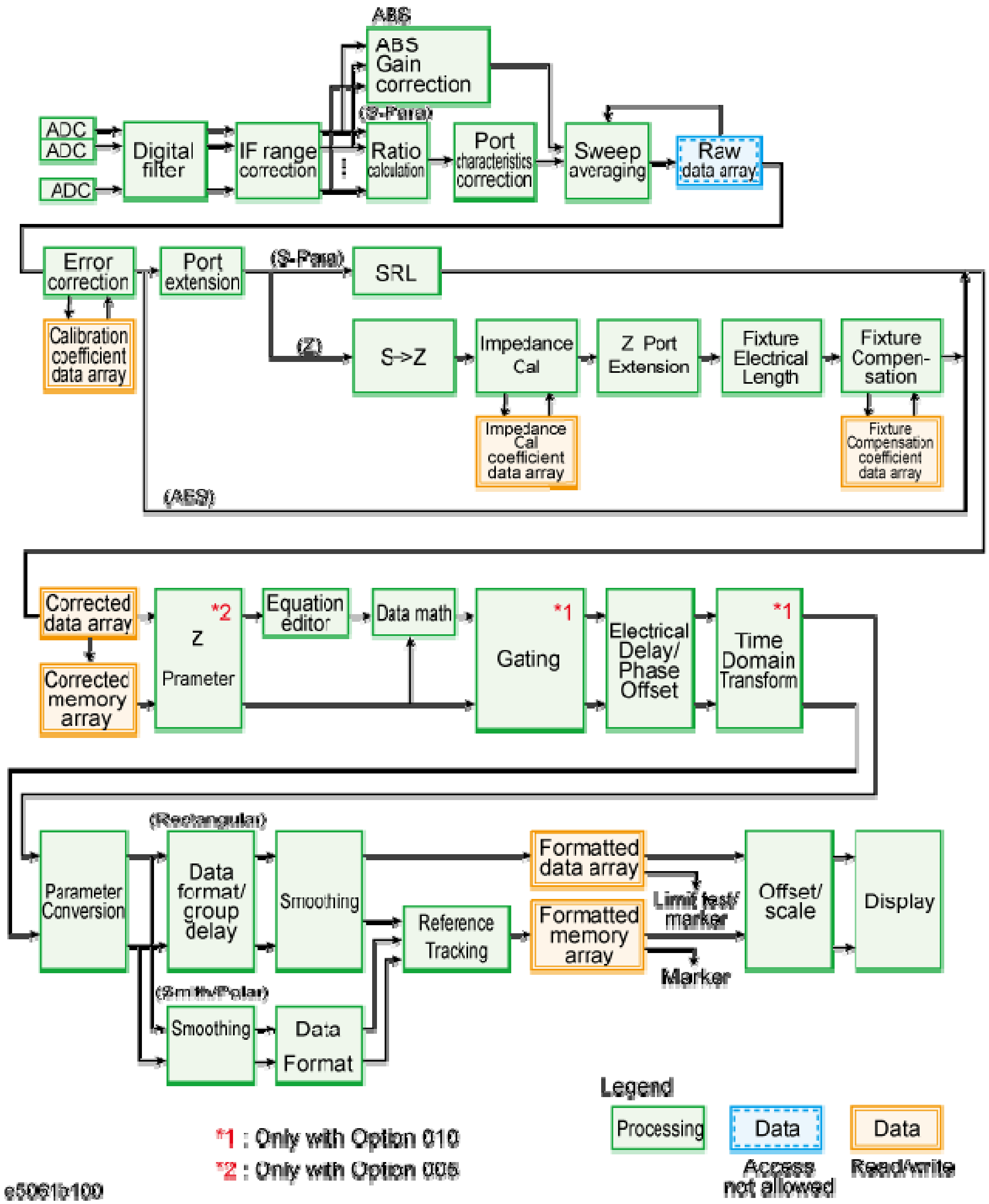
General Principles of Operation

- Data Processing
- System Description

Data Processing

The internal data processing flowchart for the E5061B is shown in the following figure.

Data Processing Flowchart



ADC

The ADCs (analog-to-digital converters) convert analog signals. One ADC is available for each signal and the conversion takes place simultaneously.

Digital Filter

The digital filter performs a discrete Fourier transformation (DFT) and picks up IF signals. Each IF signal is then converted into a complex number that has a real part and an imaginary part. The IF bandwidth of the analyzer is equivalent to the bandwidth of the DFT filter.

IF Range Correction

Input signals that went through ranging at the receiver are reverted (corrected) to previous values before the ranging.

Ratio Calculation

The ratio between two signals is determined by performing divisions on complex numbers. In the case of absolute measurements, the ratio of complex number cannot be calculated.

Port Characteristics Correction

The equivalent source match error, the directivity error, and the tracking error of each test port bridge are corrected. In the case of absolute measurements, the gain of each test port is corrected.

Sweep Averaging

The average of complex indices is determined based on data obtained from multiple sweep measurements. Sweep averaging is effective in reducing random noise in measurements.

Raw Data Array

The results from all data processing done up to this point are stored in this array as raw data. All prior data processing is performed as each sweep takes place. The user is not allowed to access (read/write) this raw data array.

Error Correction / Calibration Coefficient Data Array

When error correction is enabled, the process eliminates the system errors that are reproducible and stored in the calibration coefficient data array. It accommodates everything from the simple vector normalization to the full 12-term error correction. The user is allowed to access (read/write) this calibration coefficient data array. Gain correction is performed in absolute measurement.

Port Extension

This process carries out a simulation of adding or eliminating a variable length transmission path on each test port so that the reference plane of calibration is moved. The port extension is defined by an electrical delay (sec).

SRL

This process calculates Structural Return Loss. This is available only when option 010 is installed.

S -> Z

This process converts from S-parameter to impedance

Impedance Calibration / Impedance Calibration Coefficient Data Array

Impedance calibration provides the error correction for the impedance parameters. The impedance calibration does not apply for S-Parameter. The impedance calibration and error correction can not be applied in the same time.

Z Port Extension

This is basically same as the port extension. The Z port extension can not have loss.

Fixture Electrical Length

Fixture electrical length is another compensation method to eliminates a length transmission path of the coaxial cable. The electrical length is defined by meter.

Fixture Compensation / Fixture Compensation Coefficient Data Array

Fixture compensation eliminates the error factor of fixture. Open, short and load compensation can be performed independently.

Corrected Data Array

Unlike the raw data array, this array stores the results obtained after error corrections, port extensions, or the fixture simulator functions are applied. The user is allowed to read/write data from/to the corrected data array.

Corrected Memory Array

By pressing **Display** > **Data -> Mem**, the contents of the corrected data array is copied to this array. The user is allowed to read/write data from/to the corrected memory array.

Z Parameter

In option 005 Impedance Analysis, impedance parameter such as |Z| and Cp is calculated from impedance complex data.

Equation Editor

Equation Editor allows users to use a custom equation to display data in the E5061B. Equation Editor can be accessed through **Display** > **Equation Editor**.

Data Math

Data processing is carried out using the corrected data array and the corrected memory array. Four types of data processing addition, subtraction, multiplication, and division are available.

Electrical Delay / Phase Offset

An electrical delay and a phase offset are applied to each trace. By setting an electrical delay, a linear phase that is proportional to the frequency will be added or subtracted. On the other hand, setting a phase offset adds or subtracts a phase that is constant throughout the frequency range. Incidentally, data processing performed from this point on in the flowchart is applied to both the data array and the memory array.

Data Format / Group Delay

Complex data consisting of the real parts and the imaginary parts are converted into scalar data according to the data format of user's choice. Group delays are also calculated here.

Smoothing

By enabling the smoothing function, each point in a sweep measurement is replaced by a moving average of several measurement points nearby. The number of points used in calculating a moving average is determined by the smoothing aperture set by the user. The smoothing aperture is defined by a percentage against the sweep span.

Formatted Data Array / Formatted Memory Array

All results from data processing are stored in the formatted data array and the formatted memory array. The marker functions are applied to these arrays. The limit test is applied to the formatted data array. The user is allowed to read/write data from/to these arrays.

Offset / Scale

Each set of data is processed so that traces can be drawn on the screen. Particular scaling depending on the data format is applied using the position of the reference line, the value of the reference line, and the scale/graticule settings.

Display

The results obtained after data processing are displayed on the screen as traces.

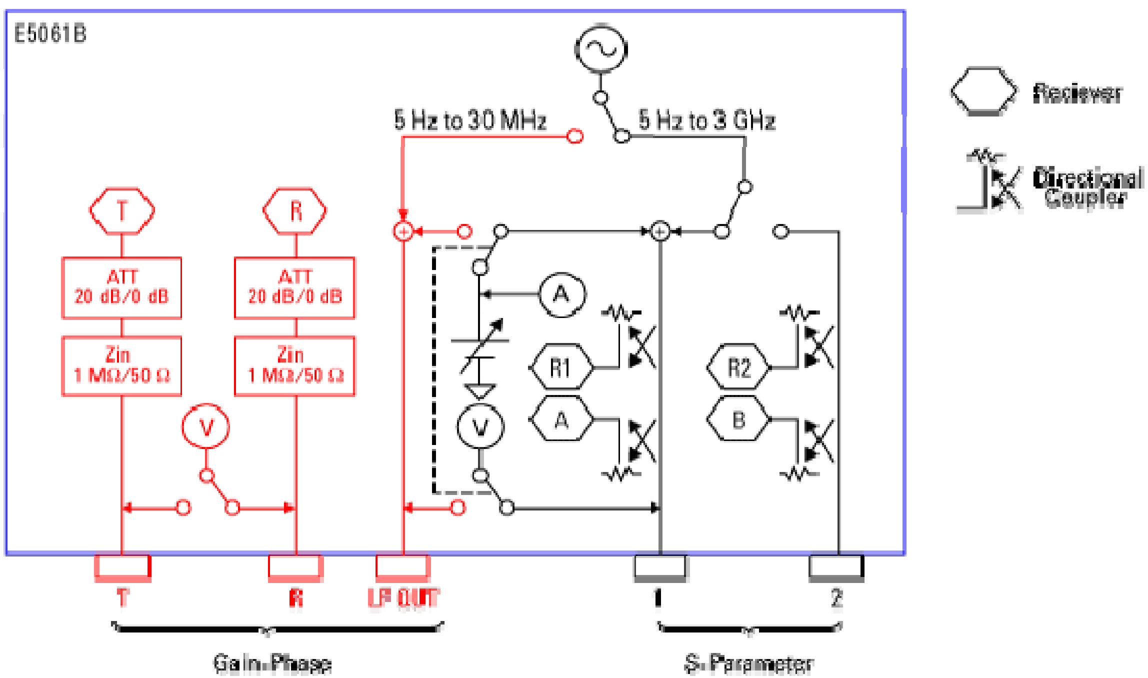
System Description

- Synthesized Source
- Source Switcher
- Signal Separator
- Receiver
- DC Bias Source
- DC Monitor

Other topics about General Principles of Operation

A network analyzer supplies a sweep signal to a DUT, measures its transmission and reflection, and displays the results as ratios against the input signal from the signal source. The E5061B network analyzer consists of the circuit modules shown in the following figure.

System Diagram for the E5061B Network Analyzer (Optoin 3L5)



e5061b030

System Diagram for the E5061B Network Analyzer (Besides Option 3L5)

Synthesized Source

The synthesized source generates a sweep signal in the specified frequency range.

The signal source is phase-locked to a highly reliable quartz crystal oscillator to maintain a high level of accuracy in its frequency as well as to achieve precise phase measurements.

Source Switcher

The source switchers are used to switch test ports to which the signal is supplied from the source. The source switcher always set at the port1 in T/R test set option (options 115, 135, 117 and 137)

Signal Separator

The signal separator consists of directivity couplers that detect input and output signals at the test ports. On a test port to which a signal is output, the output signal and the reflection from the DUT are detected as the reference signal (R1) and the test signal (A), respectively. On the other ports, the signal that is transmitted through the DUT is detected as the test signal (B). All signals are then sent to the receiver.

Receiver

Each signal that is sent to the receiver is first converted into an IF signal by a mixer and then converted into a digital signal by an ADC (analog to digital converter). These processes are applied to each signal independently. The digital data is then analyzed by a micro processor and measurement results are displayed on the screen. The receiver R2 is not available in T/R test set option (options 115, 135, 117 and 137)

DC Bias Source (Option 3L5)

Built in DC bias source can apply -40 V to +40 V DC on the signal of LF out and Port 1. The switcher for DC bias can be changed independently from the Source switcher.

DC Monitor (Option 3L5)

Voltage and current meters are equipped to monitor DC on received and DC biased source. DC voltage can be measured at LF out, R, T and Port 1. DC current can be measured at LF out and Port 1 .

Revision History

Revision History

- Firmware Revision History
- HDD Revision History
- Data Sheet Revision History

Other topics about Product Information

Firmware Revision History

- A.02.0x
- A.01.1x
- A.01.00

Other topics about Revision History

A.02.0x

- Options 115, 135, 215, 235, 117, 137, 217 and 237 support
- Impedance Analysis (Option 005) support
- State File Converter
- The windows icon for minimize, maximize and close are replaced with resize.
- Log for Service Support

A.01.1x

- Segment Sweep: IFBW Auto function does not work in the Segment Table even if IFBW AUTO is ON.
- Initial Source Port Control function: The DC bias and RF/LF source can be set on/off independently.
- Overload Recovery: **Clear Overload Protection** function is added. The measurement is aborted when overload is detected.
- SCPI.SYSTem.ISPControl.STAT: This commands turn on/off the initial source port contron for RF signal only.
- The following commands have been added:
  - SCPI.SERVice.POWer.OVERload.PROTect.CLEar
  - SCPI.SERVice.POWer.OVERload.PROTect.STATe
  - SCPI.SYSTem.ISPControl.DCBias.STATe

A.01.00

The first revision of E5061B firmware. This revision is for DEMO unit only.

HDD Revision History

HDD revision is based upon a number of factors such as windows OS and driver upgrade patch which are installed at the factory shipment. Firmware revision denotes E5061B measurement software. To know about the E5061B HDD revision, refer to Checking the product information. ELxxx is shipped only with E5061B option 3L5. KYxxx is shipped with both E5061B options 3L5 and 115/135/117/137/215/235/217/237.

- KY20x
- EL20x
- EL120/140
- EL110
- EL100

Other topics about Revision History

KY20x

- Windows License is changed from Windows Vista Business to Windows XP Pro for Embedded Systems (Windows task bar is not displayed).
- System Recovery procedure is changed.

EL20x

- LXI is changed to version 1.3.
- Internet Explore is changed to version 8 .
- Windows Media Player is changed to version 11.

EL140/EL120

- No functionally change

EL110

- The E5061 Network Analyzer Service in Windows Firewall is on at factory shipment

EL100

The first revision. This revision is for DEMO unit only.

Data Sheet Revision History

The data sheet revision history is shown below. The print date is used to define the revision. See the last page of the data sheet for its print date.

- November 6, 2009
- September 11, 2009

Other topics about Revision History

November 6, 2009

The foot note 2 of System dynamic range has been changed.

September 11, 2009

The first edition

Using Windows

- Windows Consideration
- Windows License
- Change Date/Time Settings
- User Account and Password
- On-Screen Keyboard
- Configuring Network
- Accessing Hard Disk of E5061B from External PC
- Windows Firewall
- Connecting External Accessories

Windows Consideration

- Using USB
- Plug & Play Stability and Security
- LAN Connections
- Single and Double Click option
- Printing

Other topics about Using Windows

Using USB

The E5061B has six USB ports for connecting devices: two in the front panel and four on the rear panel. The main advantages of USB are instant connect and disconnect, and faster data transfer speed. Electronic Calibration modules are now available with USB connections.

The first time you plug a device into a USB port there is some wait time. Windows reports that it is identifying the hardware, then searching for the correct driver, then installing the driver (if it is found).

Connecting that same device back into that same port later is quick and easy, but if you move the device to a different USB port, you will have to wait through the hardware ID and driver search again.

NOTE Certain USB devices (such as ECAL modules) require you to be logged on with Administrator's privileges, the first time you plug them into the E5061B. This must be done for each serial number. Click **Next** to choose the default settings when installing new USB devices.

NOTE Some USB memory may not operate on the E5061B. In that case, try the other kind of USB memory. There is some USB memories which do not work on the front panel USB port, but work on the rear panel USB port.

NOTE Agilent Technologies shall not be responsible for, nor assume any liability for data loss in your USB memory device after using it with the E5061B.

Plug & Play Stability and Security

Plug & Play capabilities provide both a stable and secured operating environment. You may also notice that it greatly reduces the number of required reboots.

LAN Connections

Windows supports DHCP and fixed IP addressing. Also, instant connect and disconnect of the LAN cable, as well as a visual indicator of LAN status in system tray area, makes LAN connections more intuitive. In addition, the Hardware Wizard helps users with system hardware configuration.

Single and Double Click option

By default, Windows allows a double-click method of launching icons. To revert to single-clicking, see the Windows help.

Printing

Adding a printer should be done as Windows operation. See Connecting Printer.

Windows License

There are differences in Windows display for firmware A.02.0x and above compared to previous revisions. The differences in Windows display is due to the Windows licenses used with the firmware.

E5061B Option 3L5 with firmware revision A.02.0x and above as well as E5061B (all others options except Option 3L5):

- Labelled as Windows XP Pro for Embedded Systems (at rear panel)
- The Windows taskbar is no longer shown at the bottom of the screen.
- The Windows resizing buttons no longer appear at the upper-right of the Windows titlebar. It is replaced with E5061B resize button at the upper-right of the channel titlebar, as shown in the following figure:

E5061B Option 3L5 with firmware older than revision A.02.0x (for example, A.01.1x):

- Labelled as Windows Vista Business (at rear panel)
- The Windows task bar is shown at the bottom of the screen.

- The Windows resizing buttons are shown at the upper-right of the Windows titlebar. There is no E5061B resize button at the upper-right of the channel titlebar.

However, if you had the earlier firmware version and have upgraded to revision A.02.0x:

- The Windows taskbar is still shown at the bottom of the screen.
- However, the Windows resizing buttons no longer appear at the upper-right of the Windows titlebar. It is replaced with E5061B resize button at the upper-right of the channel titlebar.

Change Date / Time Settings

The E5061B has the built-in clock that keeps track of the date and time. This clock is used for the following functions.

- To display the current date and time in the instrument status bar at the lower part of the screen
- To write date and time information when saving internal data or a VBA program

NOTE The **Date & Time** of E5061B clock can be changed by double-clicking the clock at the bottom right of the taskbar.

Other topics about Using Windows

User Account and Password

- To Change Password
- To Add New User

Other topics about Using Windows

To Change Password

At the factory default setting, the account name is **Instrument**, the password is **measure4u**, and Auto Log On is activated. Once you change the password, Auto Log On is not available.

1. Open **User Accounts** in Control Panel.
2. Click **Change an account**.
3. Select the icon of your account name.
4. Click **Change the password**.
5. Enter **new password**, **new password again to confirm**, and **password hint**.
6. Click **Change Password** button.

NOTE When the password of the account being logged in is changed, the above procedure is little different.

To require users to press CTRL+ALT+DELETE before logging on

For added security, you can require users to press CTRL+ALT+DELETE before logging on.

1. Open **User Accounts** in Control Panel.
2. Click **Change the way users log on or off**.
3. Check the **Use the Welcome screen** checkbox.
4. Click **Apply Options**.
5. Log off Windows.

NOTE The E5061B must be removed from the domain for the procedure to work. When shipped, the E5061B is configured as a member of a workgroup. If the configuration of the E5061B is later changed so as to be a member of domain, then name/password must always be entered at startup. This is a security requirement of Windows.

NOTE If the E5061B is assigned to a domain, then **Change the way users log on or off** is not available in **User Accounts** dialogue box.

To Add New User

The login accounts must be as the computer administrator if you make measurements.

1. Open **User Accounts** in Control Panel.
2. Click **Create a new account.**
3. Enter the new account name, then click **Next**.
4. Select **Computer Administrator.**
5. Click **Create Account** button.
6. Select the icon of your account name.
7. Click **Change password**.
8. Enter **new password**, **new password again to confirm**, and **password hint**.
9. Click **Change Password** button.

NOTE The administrator authority is necessary for some of function in E5061B. Select computer administrator when you make the user account.

On-Screen Keyboard

- On-Screen Keyboard overview
- To open Windows On-Screen Keyboard

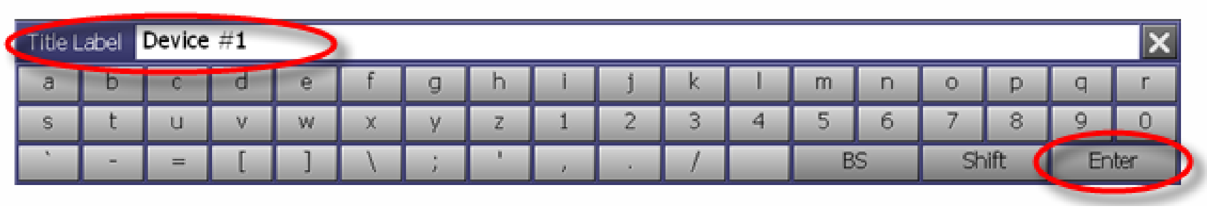
Other topics about Using Windows

On-Screen Keyboard overview

On-Screen Keyboard is a utility provided by Windows that displays a virtual keyboard on the screen. It allows the users of the E5061B to input characters without the need of a keyboard.

NOTE The E5061B does not requires an external keyboard for its operation. Users can input characters using an on-screen keyboard in-built with the E5061B firmware.

On-screen keyboard in-built with the E5061B firmware



To open Windows On-Screen Keyboard

To open On-Screen Keyboard: click **Start** > **All Programs** > **Accessories** > **Accessibility** > **On-Screen Keyboard**.

On-screen keyboard provided by windows OS

Configuring Network

NOTE When you use the E5061B by connecting it to your LAN, consult your network administrator and make the setting of the LAN correctly.

This section describes how to set the following basic items necessary to connect the E5061B to the LAN (Local Area Network).

- Enabling/Disabling network
- Check/Reset IP Address
- Setting IP address
- Checking computer name

If you need detail network settings, consult your network administrator and perform operation in the same way as the Windows PC.

Other topics about Using Windows

Enabling/Disabling Network

You can enable/disable the network connection function of the E5061B.

To enable/disable the network connection function

1. Use the LAN cable to connect the E5061B to the LAN.
2. Press System > **Misc Setup** > **Network Setup** > **Network Configuration** to open **Network Connections** window.
3. Perform either of the following:

- When switching from disable to enable:

Double-click the Local Area Connection icon in the **Network Connections** window to enable the network connection function.

- When switching from enable to disable:

Double-click the Local Area Connection icon in the **Network Connections** window. The **Local Area Connection Status** screen appears. Click the **Disable** button to disable the network connection function.

4. Close **Network Connections** window.

Check/Reset IP Address

The IP address of the E5061B can be checked/reset by System > **Misc Setup** > **Network Setup** > **LAN Dialog**.

Setting IP address

Follow these steps to set the IP address:

1. Press **System** > **Misc Setup** > **Network Setup** > **Network Configuration**.
2. Double-click the Local Area Connection icon in the **Network Connections** window. The **Local Area Connection Status** screen appears.
3. Click **Properties**. The **Local Area Connection Properties** screen appears.
4. Select (highlight) **Internet Protocol (TCP/IP),** and then click **Properties**.
5. The **Internet Protocol (TCP/IP) Properties** appears. Click (select) **Use the following IP address** and then enter the **IP address**, **subnet mask** and **gateway address**.
6. If the IP address can be obtained automatically (if the DHCP server can be used), click (select) **Obtain an IP address automatically**.
7. In **Internet Protocol (TCP/IP) Properties**, click **OK**.
8. In **Local Area Connection Properties**, click **OK**.
9. In **Local Area Connection Status**, click **Close**.
10. Close **Network Connections** window.

Checking Computer Name

Follow these steps to check the computer name:

1. Press **System** > **Misc Setup** > **Network Setup** > **Network Identification**.
2. See the desired computer name in Computer description in **Computer Name** tab.

Accessing Hard Disk of E5061B from External PC

If you connect the E5061B to LAN, you can access the hard disk (D drive) in the E5061B as a network drive from an external PC connected to the same LAN.

- Enabling the access from an external PC
- Accessing hard disk of E5061B from external PC
- E5061B Hard Disk Drive

Other topics about Using Windows

Enabling the access from an external PC

This section shows the procedure to enable the access from the external PC.

1. Press **Save/Recall** > **Explorer** which opens the explorer.
2. Select (highlight) **USER (D:)** and then click **Properties** in the File menu. The USERS(D:) Properties dialog box appears.
3. Select the **Sharing** tab.
4. Click **New Share.** The New Share dialog box appears.
5. Enter the share name (name used when accessed from the external PC) in the **Share Name** box and click **OK** .
6. In the USERS(D:) Properties dialog box, click the **OK** button.

Accessing hard disk of E5061B from external PC

This section describes the procedure to connect to the hard disk (D drive) in the E5061B to which access has been made possible according to the procedure described in Enabling the access form the external PC from the external PC, taking Windows as an example.

NOTE For information on connection, see your PC's operation manual.

1. Right-click the start menu, then select **Explorer** to start the explorer.
2. From the Explorer's menu, click **Tools** > **Map Network Drive..** The Map Network Drive dialog box appears.
3. Select an appropriate drive, enter **\\C_NAME\S_NAME** as the network path and then click the **OK** button.
4. The dialog box to enter the user name and the password appears. Enter an appropriate user name and password and then click the **OK** button.

NOTE **C_NAME** in the network path is the computer name of the E5061B and **S_NAME** is the share name of the D drive.

NOTE The user name and password differ depending on the setting made when enabling access from the external PC. When you have set them according to Enabling the access from an external PC you can make connection using the user name, instrument, without the password.

E5061B Hard Disk Drive

The E5061B Hard disk contains several memory partitions. The following table explains the different partitions of the E5061B.

Drive	Description
ELxxx or KYxxx (C:)	The system drive is replaced with the original image when system recovery is executed.
USER (D:)	User can keep their files, in this drive. This drive is not replaced even when system recovery is executed.
Factory (E:)	This drive contains license information and back up data for syscal. Also,user can keep their files, in this drive. The drive is not replaced even when system recovery is executed.

Windows Firewall

- To turn Windows Firewall on or off
- To Unblock the Remote I/O server

Other topics about Using Windows

To turn Windows Firewall on or off

1. Open Windows Firewall. To open Windows Firewall, open Control Panel, and then double-click **Windows Firewall**.
2. On the **General** tab, click one of the following:
   - **ON (recommended)**. This is the setting that you should normally use.
   - **OFF (not recommended)**. Turning OFF Windows Firewall might make your computer (and your network, if you have one) more vulnerable to damage from viruses.

To Unblock the Remote I/O server

To use SICL-LAN, the Agilent Remote I/O Server and Agilent Remote I/O Port Mapper Utility must be unblocked. To use telnet and socket servers, the E5061 Network Analyzer must be unblock.

NOTE At the firmware revision A.02.00 and above, the following settings are unblocked at the default.

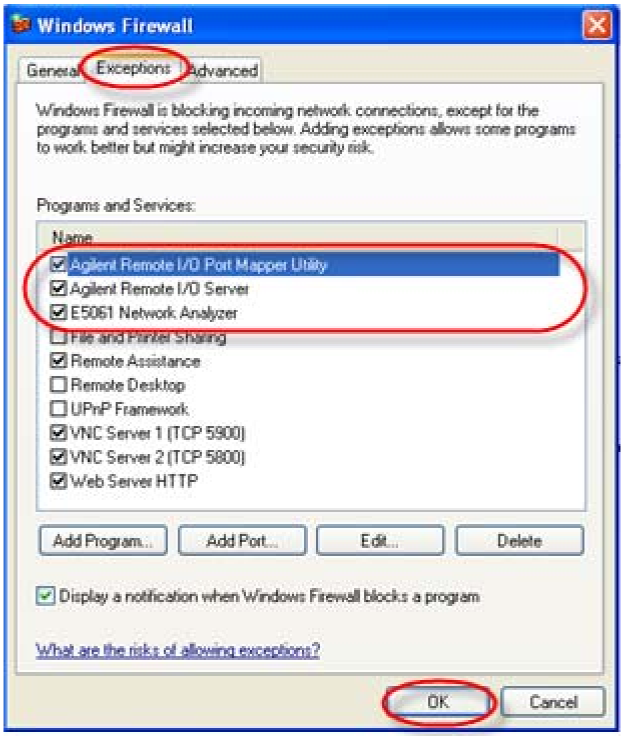
To use SICL-LAN:

1. Check the **Agilent Remote I/O Server** and **Agilent Remote I/O Port Mapper Utility** on the **Exceptions** tab in Windows Firewall.
2. Click the **OK** button to apply the setting of exceptions.

To use telnet and socket servers:

1. Check the **E5061 Network Analyzer** on the **Exceptions** tab in Windows Firewall.
2. Click the **OK** button to apply the setting of exceptions.

Exceptions tab in Windows Firewall window



e5061b071

NOTE **Agilent Remote I/O Server**, **Agilent Remote I/O Port Mapper Utility**, and **E5061 Network Analyzer** are not on the list above at the factory shipment setting (They are blocked). They appear after **System** > **Misc Setup** > **Network Setup** > **Telnet Server [ON]** and **SICL-LAN Server [ON]** are clicked.

Connecting External Accessories

The E5061B can be connected to external input/output devices such as printer, monitor, keyboard and mouse.

- Connecting a Printer
- Connecting a Mouse/Keyboard
- Connecting a Monitor

Other topics about Using Windows

Connecting a Printer

A printer can be connected to the E5061B through any USB ports on the front or rear panels. Its driver may be required to install.

Connecting a Mouse/Keyboard

A USB mouse and USB keyboard can be connected to the E5061B through any USB ports on the front or rear panels.

Connecting a Monitor

A monitor can be connected to the E5061B using the External Monitor Output Terminal (Video) located in the rear panel of the E5061B.

NOTE An external monitor needs to be connected to the analyzer and turned ON before the analyzer is turned ON so that the analyzer recognizes the monitor properly.

Programming

Programming

Remote Control

- Overview
- Setting up Analyzer
- Performing Calibration
- Making Measurement
- Reading-Writing Measurement Data
- Analyzing Data
- Saving and Recalling
- Communication with External Devices
- Status Reporting System
- Sample Programs

VBA Programming (Embedded VBA)

- Introduction to VBA Programming
- Operation Basics
- Controlling E5061B
- Controlling Peripherals
- Application Programs
- Complex Operation Library
- Waveform Analysis Library

Command Reference

- Notational Conventions
- COM Object Model
- Command Finder

Remote Control

Remote Control

- Overview
- Setting up Analyzer
- Performing Calibration
- Making Measurement
- Reading-Writing Measurement Data
- Analyzing Data
- Saving and Recalling
- Communication with External Devices
- Status Reporting System
- Sample Programs

Overview

Overview

- Types of remote control system
- GPIB remote control system
- LAN remote control system
- USB Remote Control System
- Sending SCPI command messages

Types of remote control system

Depending on the system controller and the interface, you can configure 5 types of remote control system as shown in the table below.

System controller	Interface	Overview
External controller (external computer such as PC and workstation)	GPIB (talker/listener mode)	System to control the E5061B and other devices connected via GPIB from the external controller. For more information, refer to GPIB remote control system.
	LAN	System to control the E5061B and other devices connected via LAN from the external controller. For more information, refer to LAN remote control system.
	USB	System to control the E5061B and other devices connected via USB from the external controller. For more information, refer to USB Remote Control System.
E5061B	None (Internal Connection)	System to control the E5061B itself using built-in E5061B VBA.
	GPIB (system controller mode)	System to control the E5061B itself and external devices connected via GPIB using built-in E5061B VBA.

Other topics about Overview

GPIB remote control system

- About GPIB
- System Configuration

Other topics about Overview

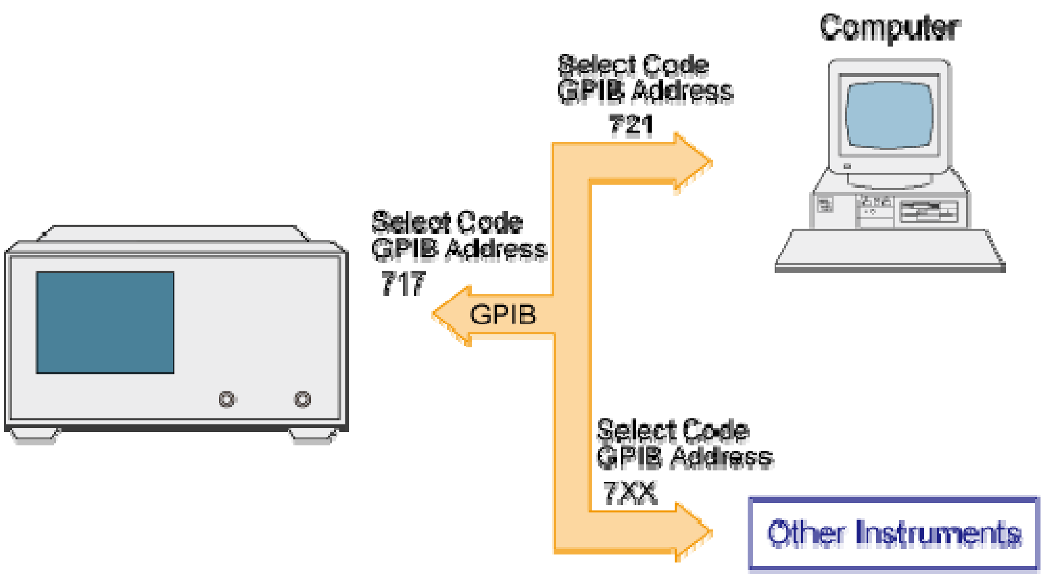
About GPIB

GPIB (General Purpose Interface Bus) is an interface standard for connecting computers and peripherals, which supports the following international standards: IEEE 488.1, IEC-625, IEEE 488.2, and JIS-C1901. The GPIB interface allows you to control the Agilent E5061B from an external computer. The computer sends commands and instructions to the E5061B and receives data sent from the E5061B via GPIB.

System Configuration

Use GPIB cables to connect between the E5061B, the external controller (computer), and peripherals. The following figure shows the overview of the system configuration of the GPIB remote control system.

Configuration of the GPIB remote control system



e5061b064

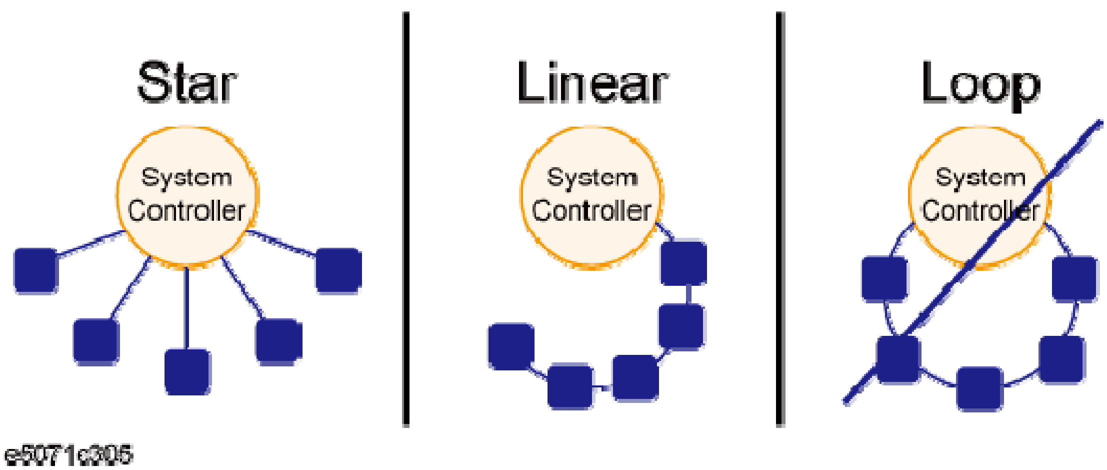
NOTE While the E5061B is turned OFF, the SRQ status of the E5061B is active. To prevent an incorrect operation on the SRQ of the GPIB remote control system, disconnect the E5061B from the system when the E5061B is turned OFF.

Required Equipment

- E5061B
- External controller (PC or workstation that can be connected to LAN and Agilent I/O Library is installed into)
- Other devices (other instruments and/or peripherals that serve your purpose)
- GPIB cables

Scale of system you can construct

- You can connect up to 15 devices in a single GPIB system.
- The length of cables to connect 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 the system in which the total cable length exceeds 20 m.
- The number of connectors connected to an individual device must be 4 or less. If you connect 5 or more connectors, excessive force is 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 set for each system. The GPIB address must be set to a unique value for each device, which 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 E5061B

To set the GPIB address for talker/listener mode, See Setting_talker_listener_GPIB_address_of_E5061B.

LAN remote control system

- Overview
- System Configuration
- Required Equipment
- Control over SICL-LAN Server
- Control using C or Visual Basic
- Control using Agilent VEE
- Control with Telnet Server
- About LXI

Other topics about Overview

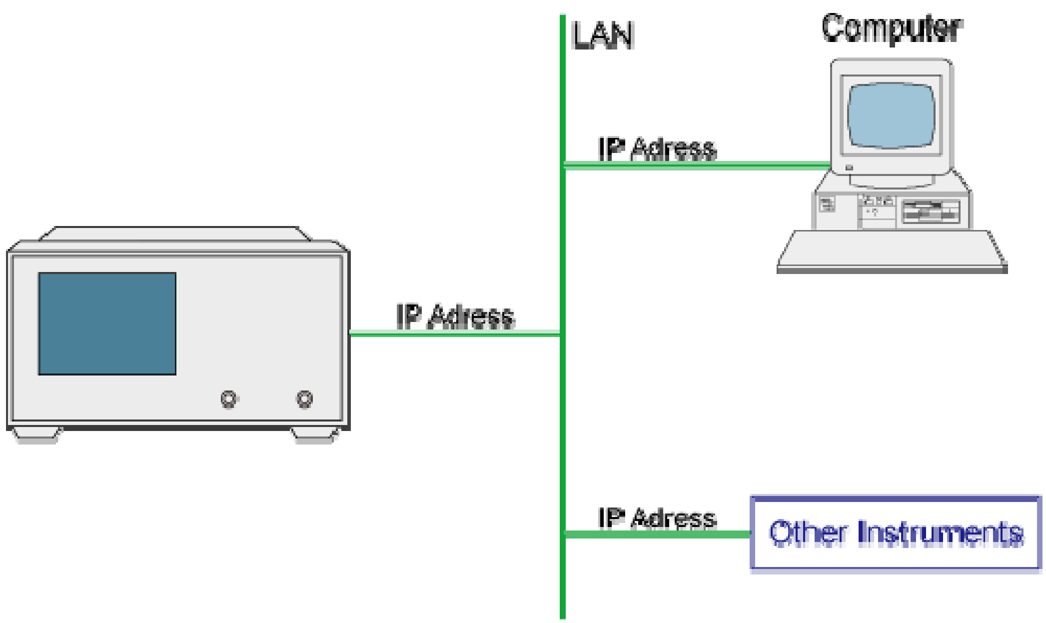
Overview

The LAN (Local Area Network) remote control system provides two methods: controlling the E5061B using the SICL-LAN server and controlling the E5061B using the telnet server.

System Configuration

Use a LAN cable to connect between the E5061B and the external controller (computer). The following figure shows the overview of the system configuration of the LAN remote control system.

Configuration of the LAN remote control system



e5061b065

Required Equipment

- E5061B
- External controller (PC or workstation that can be connected to LAN)
- Other devices (other instruments and/or peripherals that serve your purpose)
- LAN cables

Control over SICL-LAN Server

In the control system using the SICL-LAN server, communication between the external controller (client) and the E5061B (server) is performed using the SICL-LAN protocol. Communication is performed using SICL (Standard Instrument Control Library). You can control the E5061B by programming using SICL or VISA with the C language in the UNIX environment, or Visual C++, Visual Basic, or VEE in the Windows environment.

Preparing the E5061B

To communicate with the external controller, follow these steps to turn ON the SICL-LAN server of the E5061B in advance.

1. Turn ON the SICL-LAN server of the E5061B.

   System > **Misc Setup** > **Network Setup** > **SICL-LAN Server [ON]**

   NOTE When the SICL-LAN server is turned ON for the first time, the windows firewall setting dialog box appears. Select **Unblock** and click **OK**. If you select **Keep Blocking** on firewall setting, you need to unblock for the remote server in Windows firewall to use the SICL-LAN server.

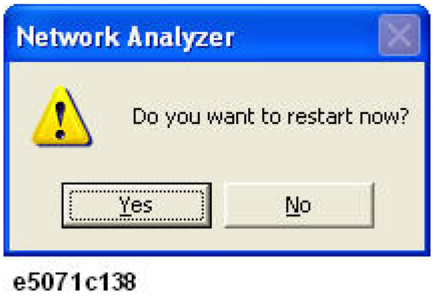
2. Set the GPIB address of the E5061B for control with the SICL-LAN server. "XX" represents an address number.

   System > **Misc Setup** > **Network Setup** > **SICL-LAN Address [XX]**

3. By default, the SICL-LAN Address does not changes until the firmware of E5061B is restarted.

e5071c137

4. On pressing any key, a message appears for restarting the firmware. Click **Yes** to restart the firmware.

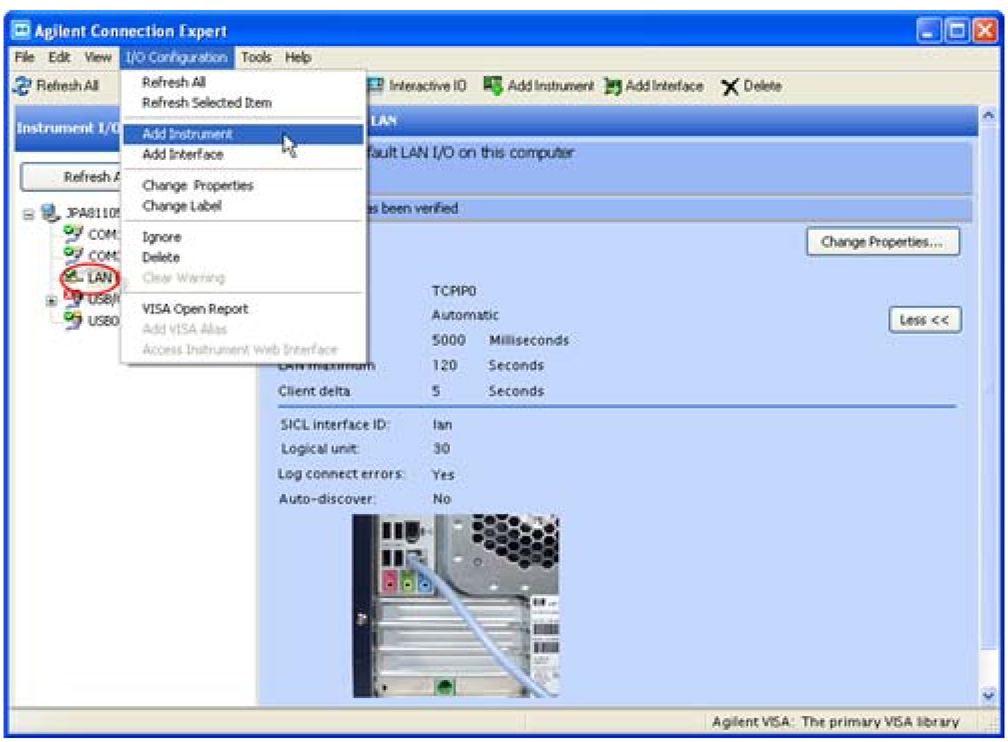


Preparing the external controller

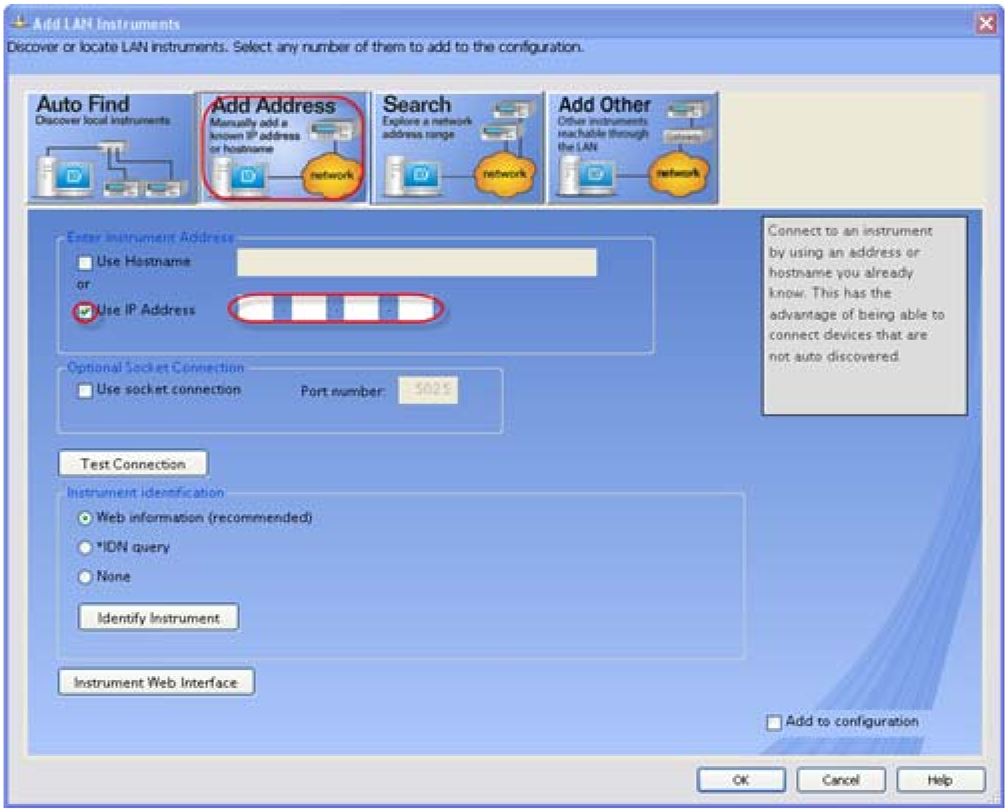
In order to establish communication with/ the E5061B using the TCP/IP protocol, you need to set the I/O interface of the external controller in advance. This section shows the setting procedure when using the external controller in the Windows environment.

NOTE You must install the Agilent I/O Libraries on your PC in advance. Use Agilent I/O Libraries Suite 14.2 or later.

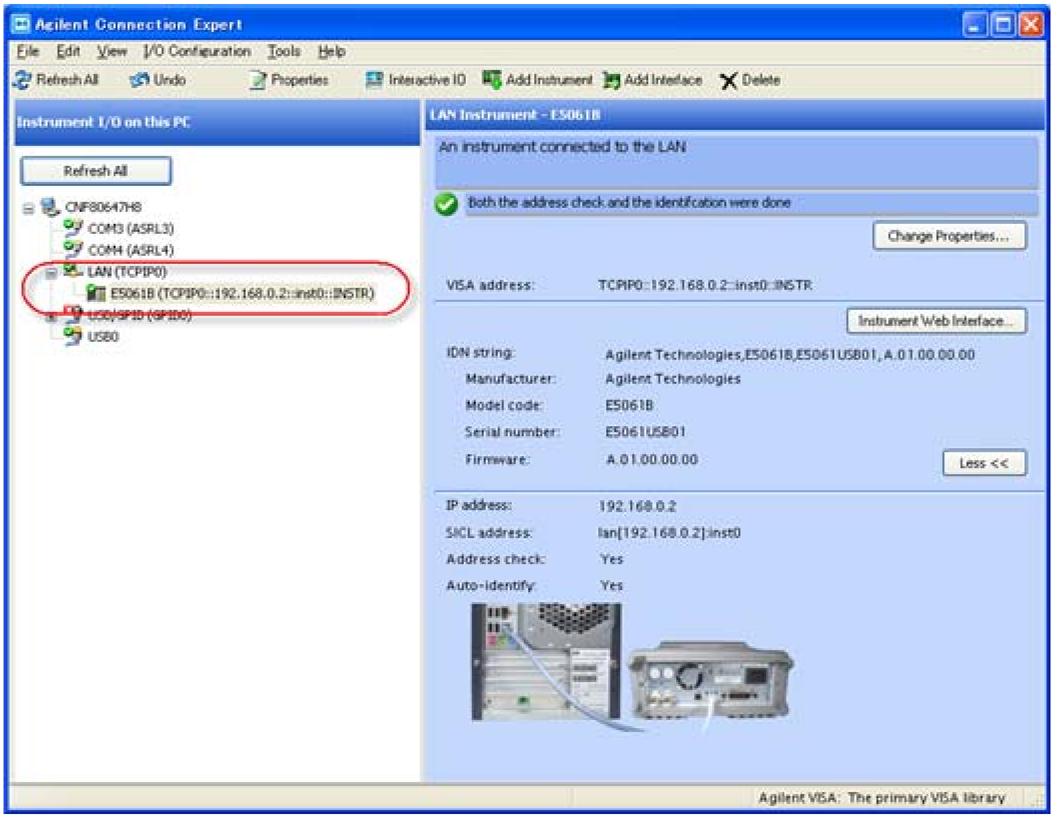
1. From your PC's Start menu, click **Program** > **Agilent I/O Libraries Suite** > **Agilent Connection Expert** to open the Agilent Connection Expert setting screen.
2. In the Agilent Connection Expert setting screen, select **LAN(TCPIP0)** in the **Instrument I/O on this PC** frame, and then click **I/O Configuration** > **Add Instrument.**



3. In the Add LAN Instrument Properties screen, set up the IP address of the E5061B and click **OK**. You can change settings as necessary. For details, refer to the Agilent I/O Libraries Suite documentation.



5. In the Agilent Connection Expert screen, check that the E5061B has been added under **LAN(TCPIP0)** in the **Instrument I/O on this PC** frame.



Control using C or Visual Basic

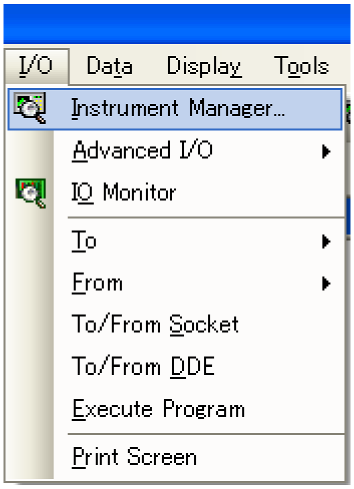
You can control the E5061B by programming using SICL with the C language in the UNIX environment, or Visual C++ or Visual Basic in the Windows environment.

Control using Agilent VEE

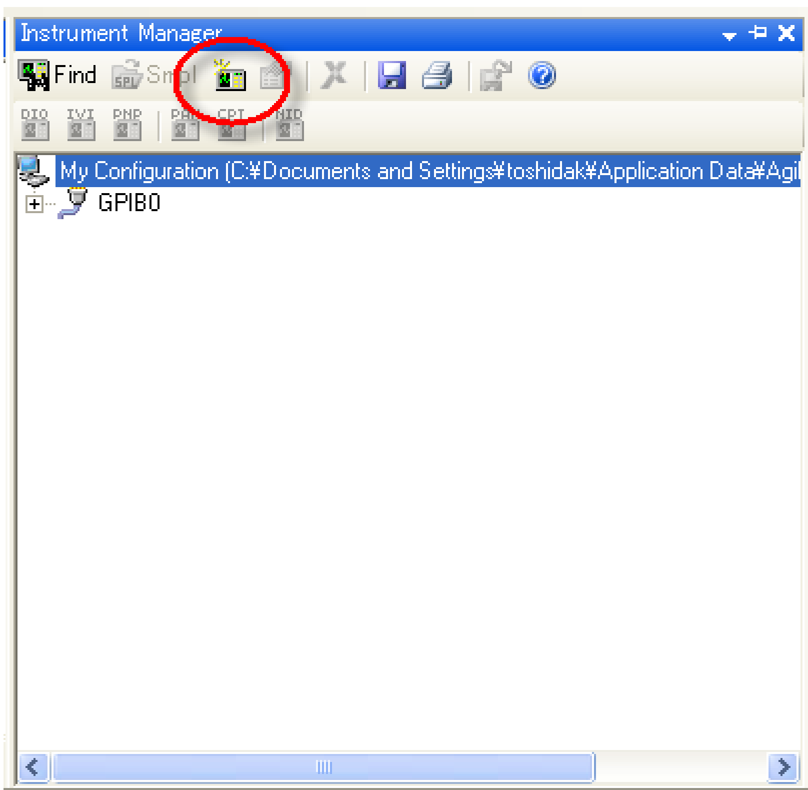
Agilent VEE allows you to control the E5061B via the I/O interface. The following example shows how to control the E5061B that is set as follows: the address of the SICL-LAN server is 17 and the IP address is 146.208.116.90.

NOTE When using Agilent VEE for PC, use Agilent VEE Pro 7.5 for Windows or later.

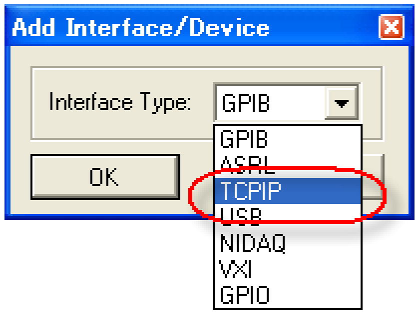
1. On the Agilent VEE's **I/O** menu, click **Instrument Manager**



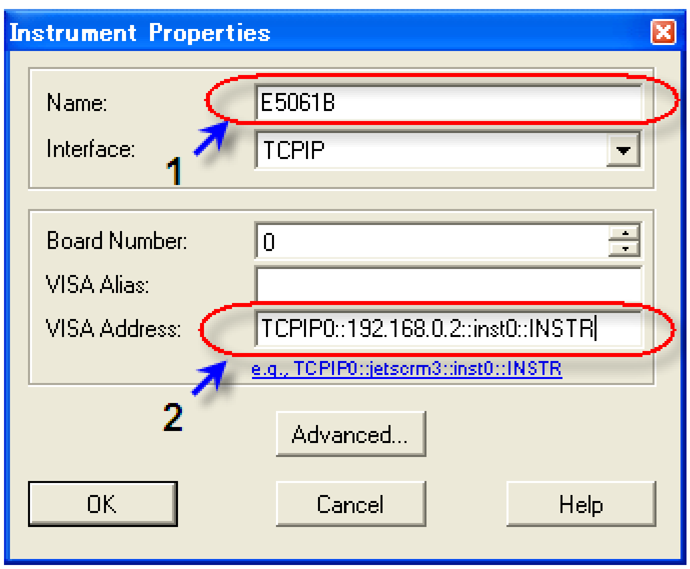
2. In **Instrument Manager**, click **Add Instrument** Icon.



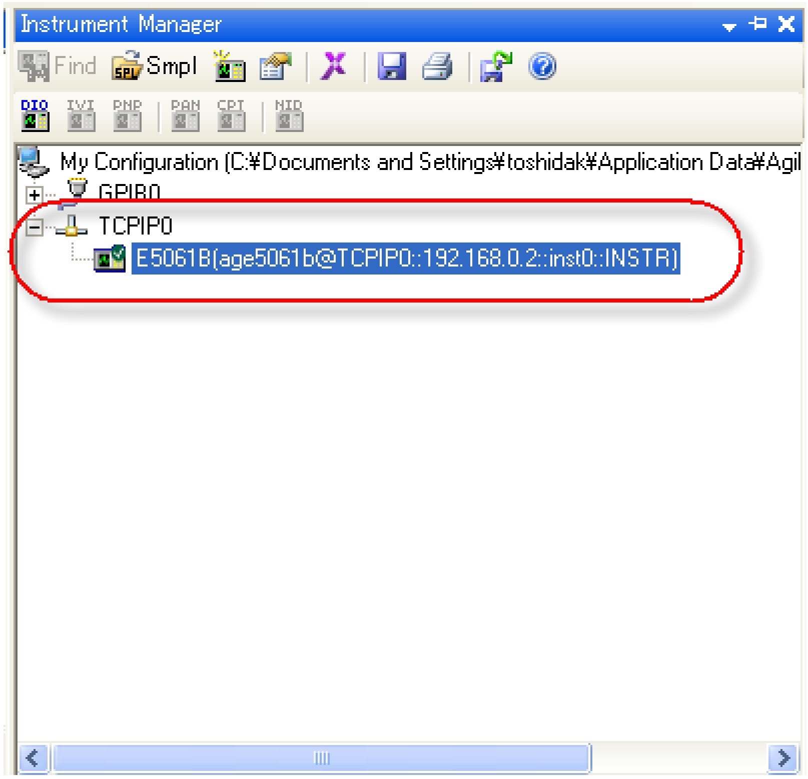
3. A new windows appears for the selection of Interface Type. Select **TCPIP** and click **OK**.



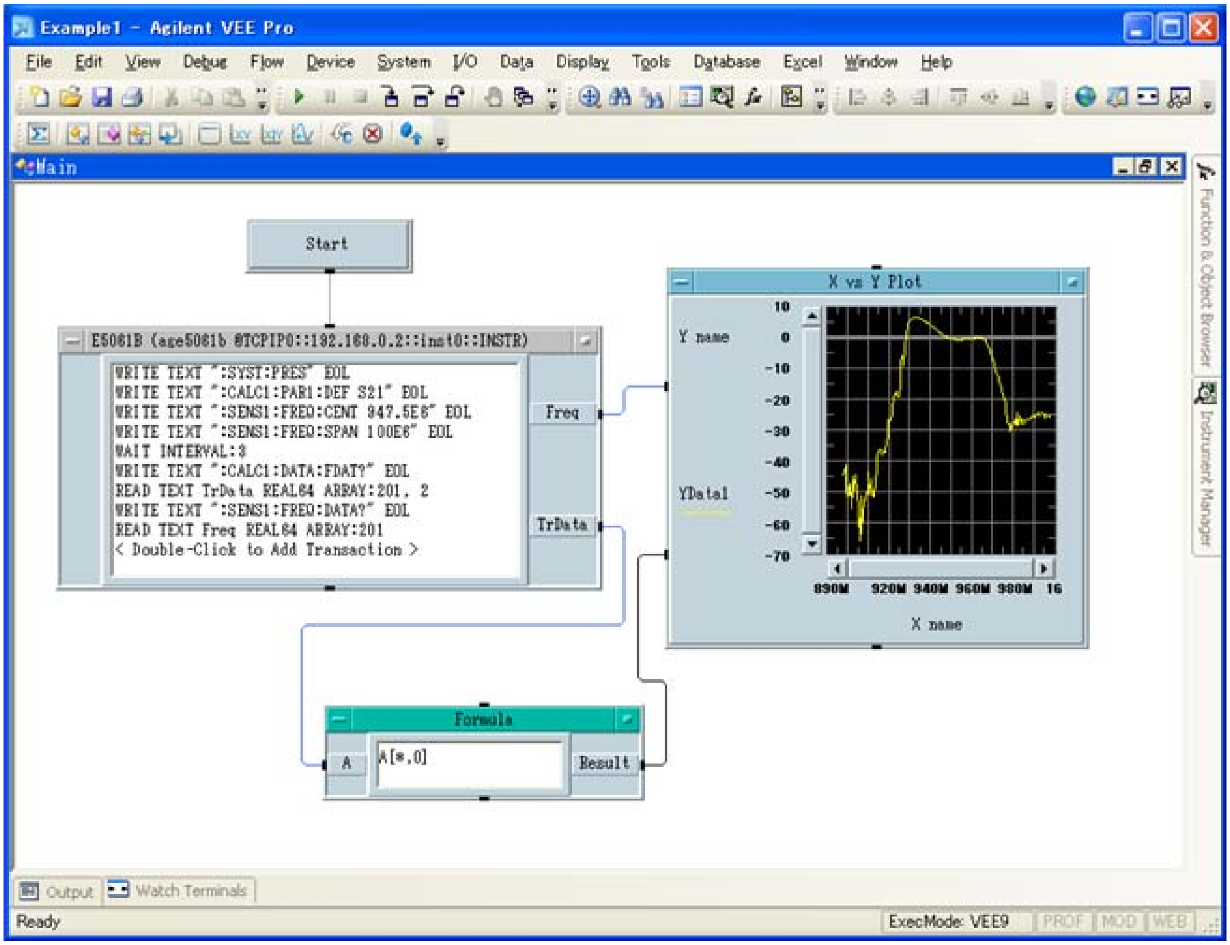
4. In **Instrument Properties** dialog box, type any name for the Instrument in Name (1 in the following figure, for example: ENA or E5061B), and add TCPIP0::<IP Address>::inst0::INSTR in the **TCIP Address**, where <IP Address> is the IP address for E5061B (2 in the following figure). For example, if the IP address for E5061B is 192.168.0.2, then the value for **TCPIP Address** would be TCPIP0::192.168.0.2::inst0::INSTR. Click **OK** after entering all the parameters.



5. The Instrument manager displays the connection with E5061B.



The following figure shows an example of control using the I/O interface that has been set in the above procedure.



Control with Telnet Server

In the control system over telnet server, communications are performed through connection between the sockets provided by the processes of the external controller and the E5061B 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 E5061B. Port 5024 is provided for conversational control using telnet (user interface program for the TELNET protocol) and port 5025 for control from a program.

NOTE To use telnet, port 5024 and 5025 should be opened through Windows firewall.

Preparing the E5061B

To communicate with the external controller, follow these steps to turn on the telnet server of the E5061B in advance.

System > **Misc Setup** > **Network Setup** > **Telnet Server [ON]**

NOTE When the telnet server is turned ON for the first time, the windows firewall setting dialog box appears. Select

Unblock and click **OK**. If you select **Keep Blocking** on firewall setting, you need to unblock for the remote server in Windows firewall to use the telnet server.

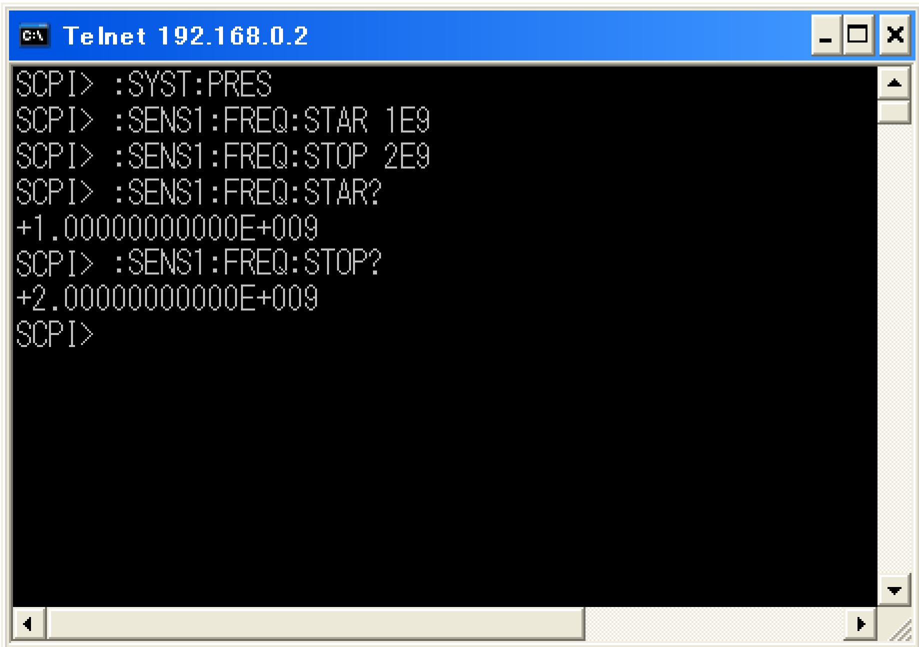
Conversational control using telnet (using port 5024)

You can use telnet to perform conversational control by sending SCPI commands to the E5061B on a message-by-message basis. For telnet, the socket of port 5024 is used for communications.

In this example, in order to show you the control procedure using telnet, you control the E5061B (IP address: 192.168.0.2 and host name: e5061b) from the external controller in the Windows environment.

1. Open the MS-DOS command prompt screen.
2. At the MS-DOS prompt, type **telnet 192.168.0.2 5024** and press the return key.
3. The telnet screen opens.
4. Type a command and press the return key; it is sent to the E5061B and executed. If you enter a command that queries some data, the query response is displayed below the line you have entered the command.
5. The following figure shows the screen after using the :SYST:PRES command to reset, the :SENS{1-4}:FREQ:STAR command and :SENS{1-4}:FREQ:STOP commands to set the sweep start value and stop value to 1 GHz and 2 GHz respectively, and checking the settings.

Example of control using telnet



6. Press] while holding down Ctl in the telnet screen to break the connection with the E5061B. The telnet prompt appears. At the telnet prompt, type quit and press the Enter key. The connection to the E5061B breaks and telnet ends.

Control from a program (using port 5025)

When controlling the E5061B from a program on the external controller, use the socket of port 5025 for connection.

NOTE Some functions such as service requests that are available in the GPIB remote control system are not available in control over telnet server.

Control using C or Visual Basic

You can control the E5061B by socket programming using the C language in the UNIX environment, or Visual C++ or Visual Basic in the Windows environment.

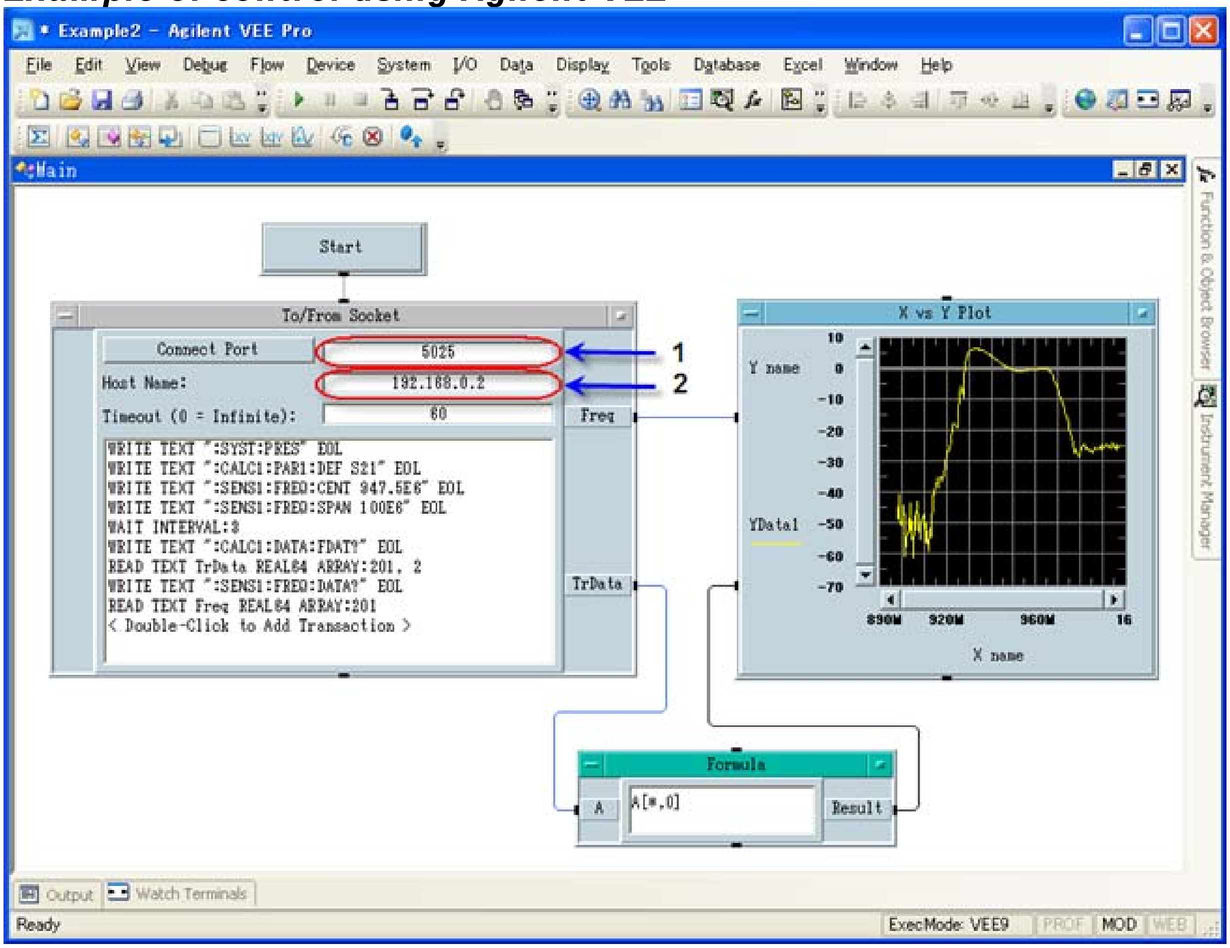
For socket programming, the library for network connection on the TCP/IP protocol is required. For the UNIX environment, BSD (Berkeley Software Distribution) Sockets API is available; for the Windows environment, WinSock (WinSock1.1 and WinSock2.0) is created by porting BSD Sockets to Windows and expanding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

For more information on the control method, see a sample program for control using WinSock described in "Controlling Using Telnet Server".

Control using Agilent VEE

Agilent VEE allows you to control the E5061B through the connection to the socket of port 5025 using To/From Socket. The following figure shows an example (when the IP address of the E5061B is 192.168.0.2). Enter 5025 in **Connect Port** to specify the port for connection (1 in the following figure) and enter the IP address or host name of the E5061B in the field to specify the **Host Name** (2 in the following figure).

Example of control using Agilent VEE



About LXI

LXI (LAN eXtensions 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.
- Synchronization of local and remote instruments through the IEEE 1588 precision time protocol.

NOTE For more information on LXI, refer to www.lxistandard.org

USB Remote Control System

- Overview
- System Configuration

Other topics about Overview

Overview

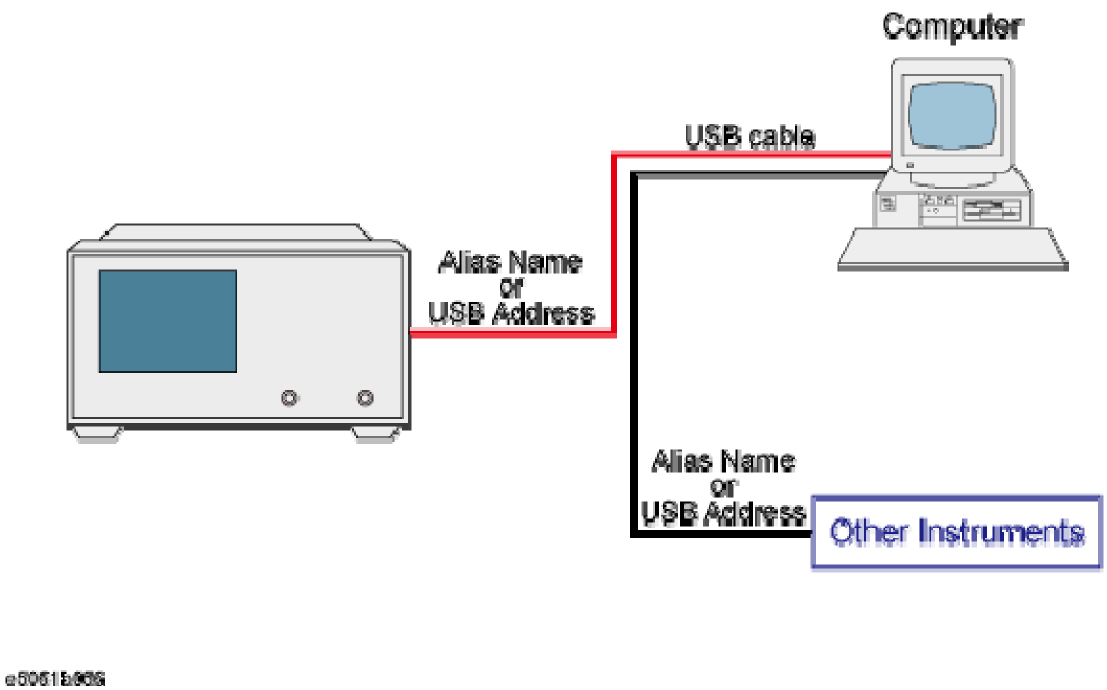
The USB (Universal Serial Bus) remote control system provides device control via USB, which is equivalent to control via GPIB. Connection is made through an interface in compliance with USBTMC-USB488 and USB 2.0.

System Configuration

The USB remote control system controls instrument with either the name "alias" or the USB address.

Use a USB cable to connect the E5061B to an external controller (personal computer). The following figure shows an overview of the system configuration for the USB remote control system.

USB Remote Control System Configuration



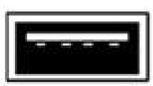
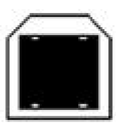
Required Equipment

- E5061B
- External controller (PC with USB host port (type A)).

- Other USB compatible devices (instruments and/or peripherals for specific purposes).
- USB cable connecting E5061B and external controller (with type A/4-prong male or type B/4-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 E5061B and other USB compatible devices must be connected via the USB interface port (type B).

Port Type	Description
	Type A: USB host port
	Type B: USB (USBTMC) interface port

Preparing E5061B

You do not have to configure any softkey or command of the E5061B in order to control the E5061B from an external controller. Simply connect a USB cable to the USB interface port.

Preparing External Controller

In order to establish communication with the E5061B via USB, you must set up the I/O interface of the external controller in advance. The USB can identify devices automatically, so once you connect a USB cable to a target device, a dialog box appears for USB device registration.

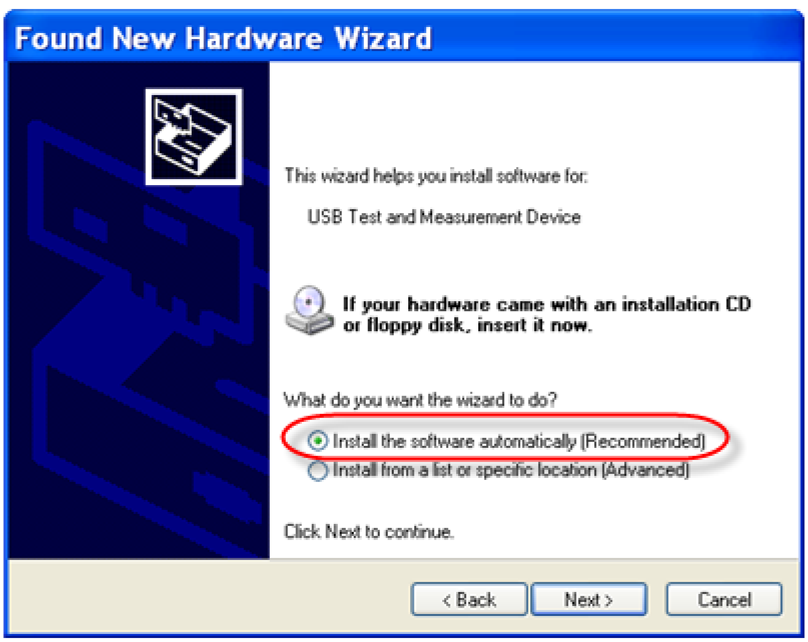
NOTE The E5061B is identified as new device if its serial number has been changed.

NOTE You must install the Agilent I/O Libraries on your PC in advance. Use Agilent I/O Libraries Suite 14.2 or later.

1. Setting E5061B when USB Cable Is Connected
    1. When new device is connected via USB cable, the following dialog box appears automatically. Select **No, not this time**, and then click **Next >**.

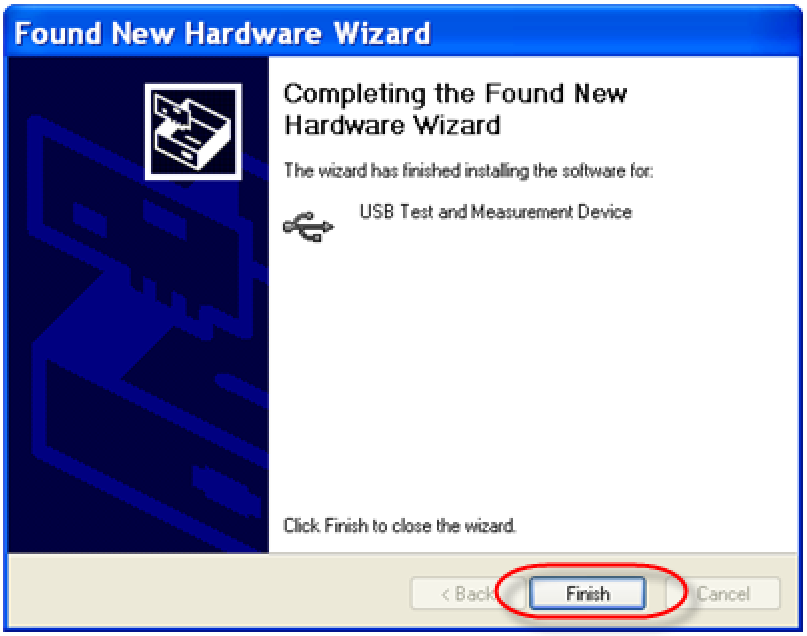
e5071c155

2. Select **Install the software automatically (Recommended)**, and then click **Next >**.



e5071c156

3. The drivers for E5061B are automatically installed and the completion screen appears. Click **Finish** to complete the process.



e5071c157

4. If you use Agilent I/O libraries 14.x and below, after finishing the setting, the dialog box named "Assign USB device alias" appears. Check "When a new USB device is plugged in.", press OK.

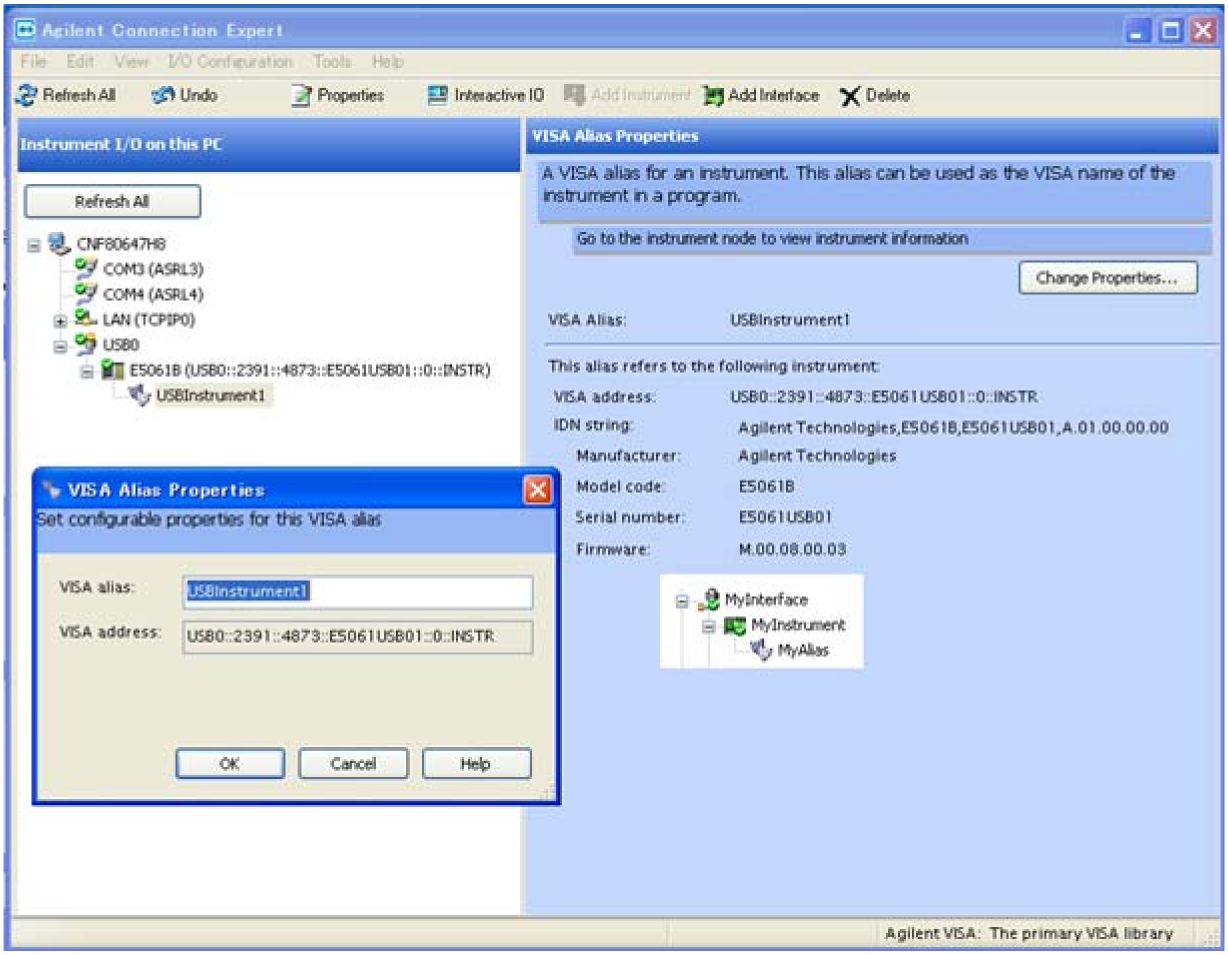
   NOTE When the device is detected at the different USB port, the "New Hardware Search Wizard" will start. Follow the instruction to implement the processing.

3. Changing Alias on Setting Screen

The following are steps using the Agilent I/O Libraries Suite 15.

1. From the Start menu of your PC, click **Programs** > **Agilent IO Libraries Suite** > **Agilent Connection Expert** to open the Config setting screen.
2. In the Config setting screen, select the alias names from **USB0** onward in the **Instrument I/O on this PC** frame, and then use the **Change Properties** from **I/O Configuration** on the menu bar.

Changing Alias



NOTE For alias, use the ASCII format less than 127 digits. Alias is upper/lower case insensitive.

Control using C or Visual Basic

You can control the E5061B by programming using Visual C++ or Visual Basic in the Windows environment as well as SICL/VISA.For further information on controlling the E5061B, see the manual of SICL or VISA. Use Agilent I/O Libraries Suite 14.2 or later.

You may use alias in the programming using SICL/VISA.

The following example shows an OPEN command to control the E5061B to which alias is given as ENA_USBIF.

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

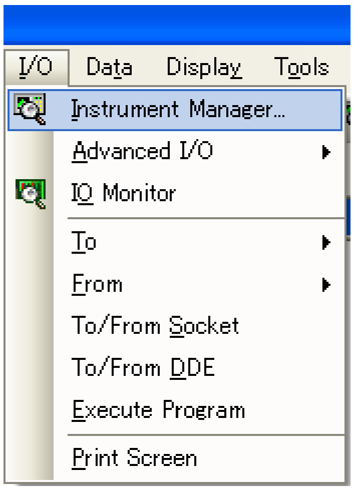
NOTE For further details of the programming using SICL/VISA, see the SICL Users Guide or the VISA Users Guide.

Control using Agilent VEE

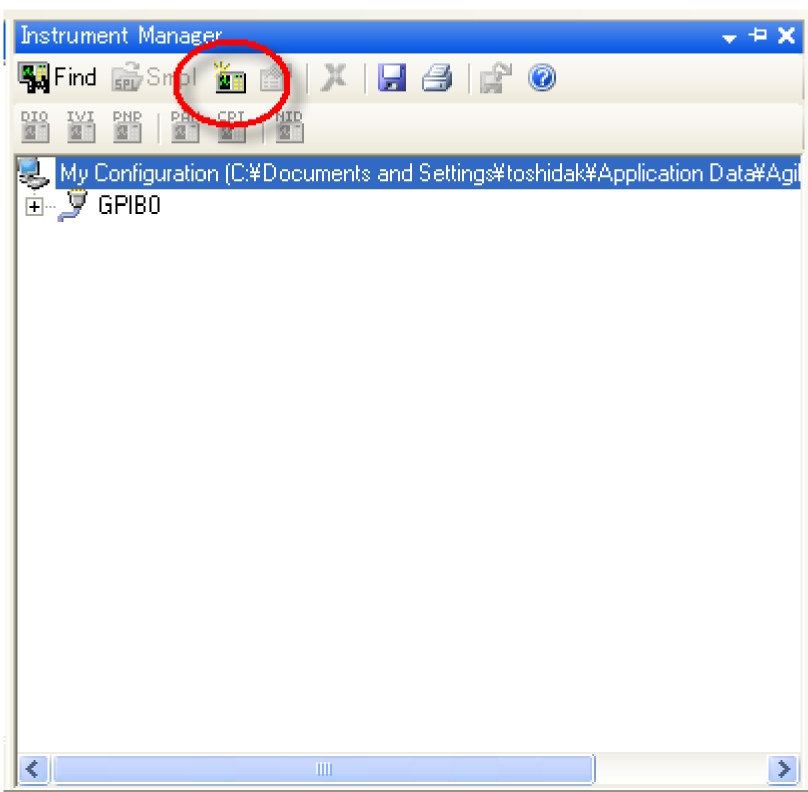
Agilent VEE allows you to control the E5061B via the direct I/O interface. The following example shows how to control the E5061B to which alias is given as ENA_USBIF.

NOTE When using Agilent VEE for PC, use Agilent VEE Pro 7 for Windows or later version.

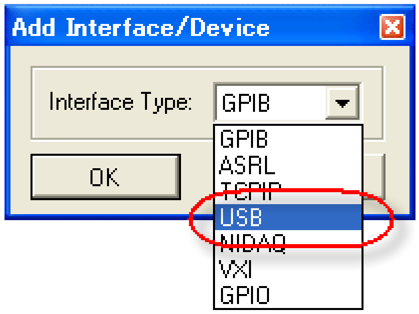
1. On the Agilent VEE's **I/O** menu, click **Instrument Manager**.



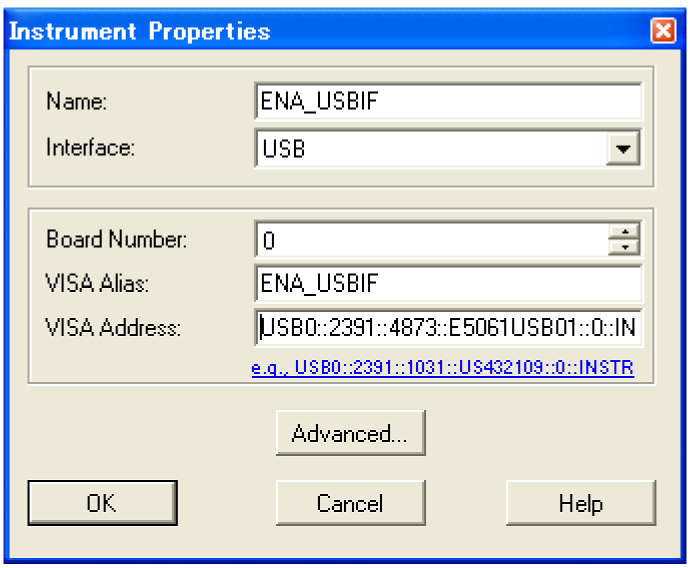
2. In **Instrument Manager**, click **Add Instrument** Icon.



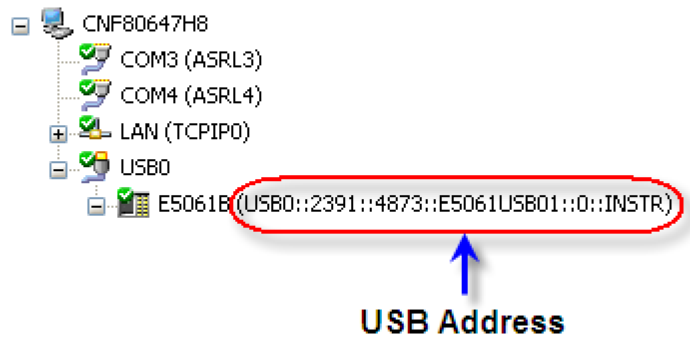
3. A new windows appears for the selection of Interface Type. Select **USB** and click **OK**.



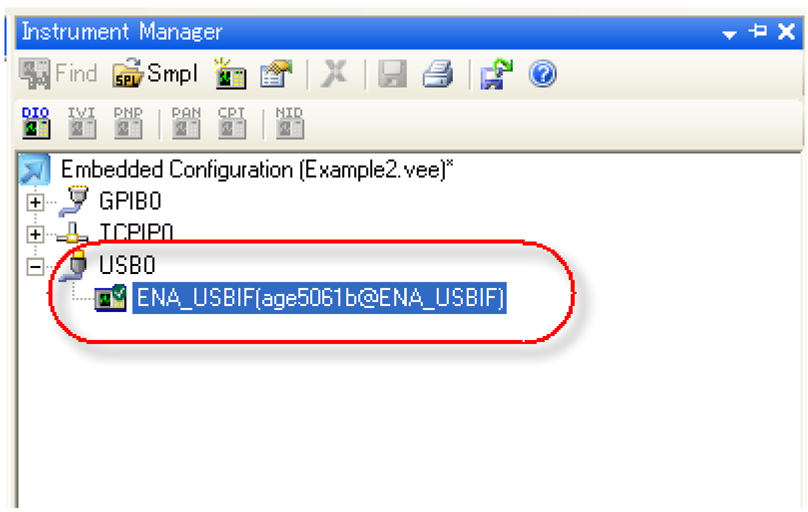
4. In **Instrument Properties**, type any name for the Instrument in Name (for example: ENA_USBIF or E5061B_USB), and add USB Address in the **VISA Address.** Click **OK** after entering all the parameters.



NOTE The USB address can be retrieved from Agilent Connection Expert.



5. The E5061B successfully appears in the **Instrument Manager**.



Sending SCPI command messages

- Type and Structure of Commands
- Grammar of Messages
- Remote Mode

Other topics about Overview

Type and Structure of Commands

The SCPI commands available for the E5061B are classified into 2 groups as follows.

E5061B commands

Commands specific to the E5061B. They cover all measurement functions that the E5061B has and some general-purpose functions. The commands in this group are arranged in a hierarchical structure called the command tree. 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 available commonly to instruments that support this standard. The commands in this group have an asterisk (*) at the beginning. For the commands in this group, there is no hierarchical structure.

Concepts of the command tree

The commands at the top of the command tree are called "root command" or simply "root." 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 power-on or reset, 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 (:)

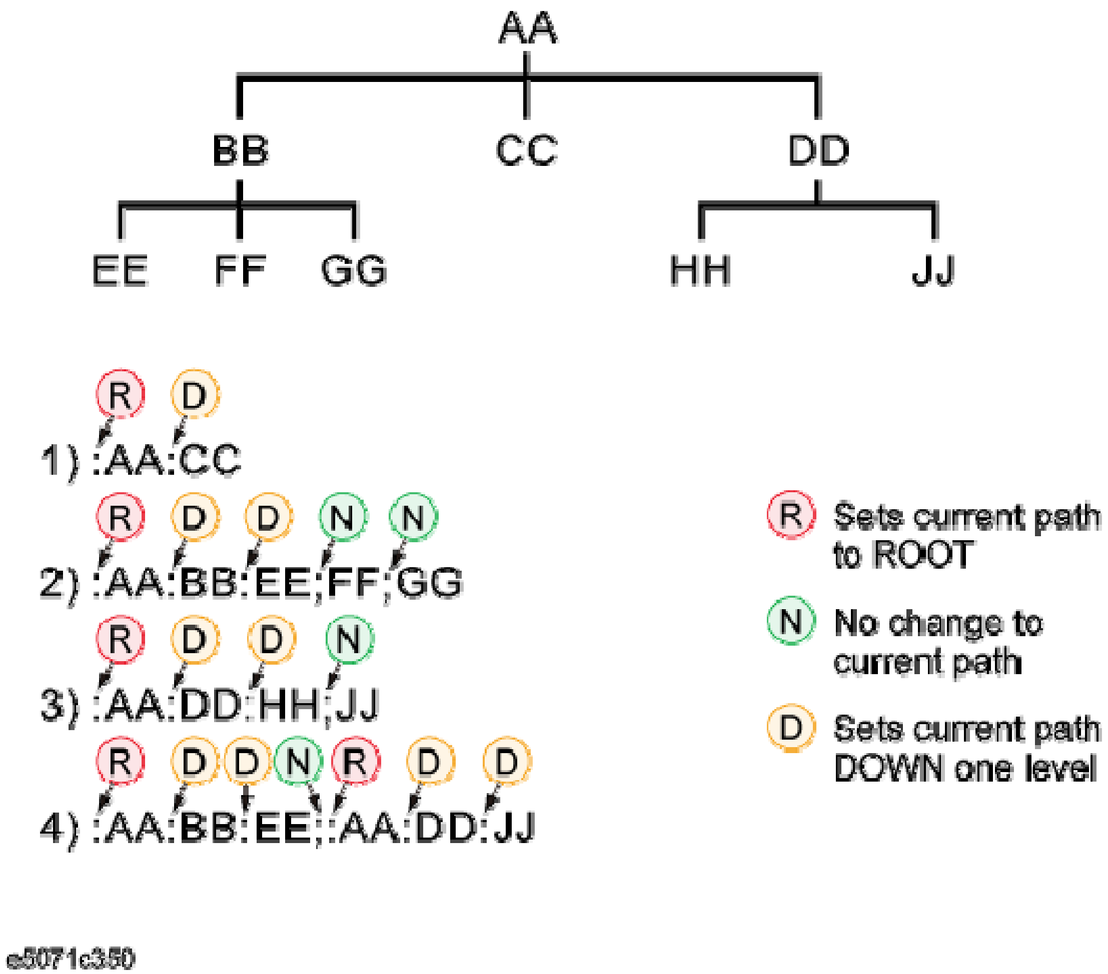
A colon between 2 command mnemonics lowers the level of the current path in the command tree. A colon used as the first character of a command specifies the command mnemonic that follows as the root-level command.

Semicolon (;)

A semicolon does not change the current path and separates 2 commands in the same message.

The following figure shows an example of how to use colons and semicolons to efficiently access commands in the command tree.

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 1 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 3 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 2 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 E5061B does not provide remote mode. Therefore, even if you send a GPIB command, it never enters into the remote mode automatically. There is no local key to release remote mode.

If you need to prevent misoperation during remote control due to entry from the front panel or mouse, lock the input devices using the following commands.

- :SYST:KLOC:KBD
- :SYST:KLOC:MOUS

Setting up Analyzer

Setting up Analyzer

- Selecting the Active Channel/Trace
- Configuring Measurement Conditions
- Configuring Display Settings
- Saving and Loading the Settings

Selecting the Active Channel / Trace

You can configure the E5061B by using various commands. Some commands require you to specify and work with a particular channel or trace, while other commands do not have this restriction.

Those commands that do not require you to specify a particular channel or trace apply to the currently active channels and traces. Before issuing such a command, therefore, you must make the appropriate channels and traces active.

To make a channel active, use the following command:

:DISP:WIND{1-4}:ACT

NOTE Only the currently displayed channels can be active channels. Therefore, you must display the desired channels by using the :DISP:SPL command before making them active.

To make a trace active, use the following command:

:CALC{1-4}:PAR{1-4}:SEL

NOTE Only the currently displayed traces can be active traces. Therefore, you must display the desired traces by using the :CALC{1-4}:PAR:COUN command before making them active.

Other topics about Setting up Analyzer

Configuring Measurement Conditions

- Setting the Number of Traces
- Selecting Measurement Parameters
- Setting Sweep Condition (Stimulus)
- Configuring Averaging Settings
- Setting the System Z0

Other topics about Setting up Analyzer

Setting the Number of Traces

When you set the number of traces, that setting determines the upper limit trace number; for example, if the setting is 3, traces 1 through 3 are displayed. To set the number of traces, use the following command:

:CALC{1-4}:PAR:COUN

NOTE Only the currently displayed traces can be active traces. Therefore, you must set the number of traces appropriately before making them active.

Selecting Measurement Parameters

To select the measurement parameter (S parameter) for each trace, use the following command:

:CALC{1-4}:PAR{1-4}:DEF

There is no equivalent command with **Meas** > **Measurement Port.** Using this command allows you to select the parameters for Gain-Phase and S-Parameter measurements in one channel. (For example, Trace 1: T/R, Trace 2: S11).

Setting Sweep Condition (Stimulus)

To select one of the above sweep types, use the following command:

:SENS{1-4}:SWE:TYPE

Turning ON/OFF stimulus signal output

To turn ON/OFF the stimulus signal output, use the following command:

:OUTP

Configuring linear/log sweep settings

To set the sweep range, use the following commands:

Type	Command
Start value	:SENS{1-4}:FREQ:STAR

Stop value	:SENS{1-4}:FREQ:STOP
Center value	:SENS{1-4}:FREQ:CENT
Span value	:SENS{1-4}:FREQ:SPAN

To set the number of measurement points, use the following command:

:SENS{1-4}:SWE:POIN

To set the sweep time, use the following commands:

Type	Command
Sweep time	:SENS{1-4}:SWE:TIME
Turning ON/OFF auto setting	:SENS{1-4}:SWE:TIME:AUTO

To set the sweep delay time, use the following command:

:SENS{1-4}:SWE:DEL

To set the IF bandwidth, use the one of the following commands (both provide the same function):

:SENS{1-4}:BAND

:SENS{1-4}:BWID

To set the AUTO IF bandwidth, use the following command:

:SENS{1-4}:BWA

Setting power level

To set the power level, use the following command:

:SOUR{1-4}:POW

To select whether to output the same power level (the set value for port 1) or a different power level for each port, use the following command:

:SOUR{1-4}:POW:PORT:COUP

:SOUR{1-4}:POW:PORT{1-2}

Configuring segment sweep settings

When you opt to use segment sweep, you can set all items (in the segment sweep table) by using a single command:

:SENS{1-4}:SEGM:DATA

Alternatively, you can configure the segment sweep settings based on the data contained in a CSV file by issuing the following command:

:MMEM:LOAD:SEGM

Also, you can save the contents of the current segment sweep table to a file by issuing the following command:

:MMEM:STOR:SEGM

For more information on how to save and load the segment sweep table, refer to Saving and recalling the segment sweep table.

Configuring power sweep settings

To set the sweep range, use the following commands:

Type	Command
Start value	:SOUR{1-4}:POW:STAR
Stop value	:SOUR{1-4}:POW:STOP
Center value	:SOUR{1-4}:POW:CENT
Span value	:SOUR{1-4}:POW:SPAN

To set the fixed frequency (CW frequency), use the following command:

:SENS{1-4}:FREQ

To set the number of points, the sweep time, the sweep delay time, and the IF bandwidth, use the same commands as for the linear/log sweep.

Configuring Averaging Settings

To configure the averaging settings, use the following commands:

Type	Command
On/off	:SENS{1-4}:AVER
Averaging factor	:SENS{1-4}:AVER:COUN
Clear (Restart)	:SENS{1-4}:AVER:CLE

For averaging, normally, the instrument must be triggered according to the number of averaging; however, when the averaging trigger is turned ON, sweeps for the number of averaging can be executed by a single trigger. For details on the averaging trigger, refer to Averaging Trigger Function.

Setting the System Z_0

To set the system characteristic impedance (Z_0), use the following command:

:SENS:CORR:IMP

Configuring Display Settings

- Setting the Layout of Windows and Graphs
- Configuring Trace Display Settings
- Setting Display Color

Other topics about Setting up Analyzer

Setting the Layout of Windows and Graphs

You can split the E5061B's LCD screen into multiple windows that display channel-specific result information, and the window layout can be selected from a number of variations. In addition, you can place on screen a segment sweep table or echo window, which you can use to display messages from your custom program.

Selecting the window layout (Channel Display Mode)

One window displays the results for a single channel. You cannot have a single window to display the results from more than one channel. This means that setting the window layout determines the number of channels displayed on screen.

To select one of the window layouts shown in the figure below, use the following command:

:DISP:SPL

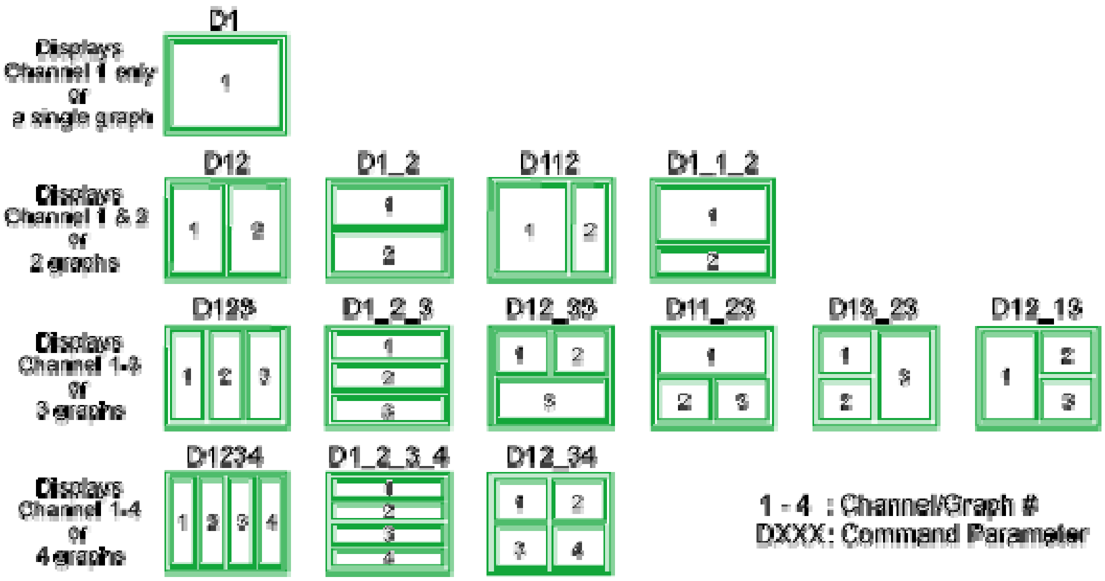
Selecting the graph layout (Trace Display Mode)

You can place a number of trace graphs in each window by selecting one of the pre-defined graph layouts. The number of graphs differs depending on your selected graph layout. If the number of graphs is equal to or larger than the number of traces (set by the :CALC{1-4}:PAR:COUN command), each graph always displays one trace. On the other hand, if the number of graphs is smaller than the number of traces, some of the graphs display two or more traces. Graph 1 is populated with trace 1, graph 2 with trace 2, and so on. Traces whose numbers exceed the last graph's number populates graph 1, graph 2, and so on.

To select one of the 14 different graph layouts shown in the figure below, use the following command:

:DISP:WIND{1-4}:SPL

Window/graph layouts and command parameters



e5061b027

Maximizing a window or a trace graph

When you have multiple windows displayed, you can maximize the active channel window so that it covers the entire screen area. To maximize a window, use the following command:

:DISP:MAX

Similarly, when you have multiple traces displayed, you can maximize the active trace so that it extends throughout the entire window. To maximize a trace, use the following command:

:DISP:WIND{1-4}:MAX

Showing/hiding a table or echo window

You can display the following items at the bottom of the LCD screen:

- Segment sweep table
- Limit table
- Marker list table
- Echo window (a window that displays messages from a custom program)
- Loss compensation table

To show or hide each of the above items, use the following command:

:DISP:TABL

You cannot have two or more of the above items displayed at a time. The screen displays only the selected item by using the following command:

:DISP:TABL:TYPE

Showing/hiding softkey labels

You can show or hide the softkey labels placed alongside the right-hand edge of the LCD screen. To show or hide the softkey labels, use the following command:

:DISP:SKEY

Configuring Trace Display Settings

Selecting which traces to display

Each trace has two different representations: data and memory traces. You can show or hide the data and memory traces independently of each other. To show or hide the data or memory traces, use the following commands:

Type	Command
Data trace	:DISP:WIND{1-4}:TRAC{1-4}:STAT
Memory trace	:DISP:WIND{1-4}:TRAC{1-4}:ANN:MARK:POS:X

To copy the data trace to the memory trace, use the following command:

:CALC{1-4}:MATH:MEM

Configuring cross-trace math operations

You can perform math operations between the data and memory traces and have the results displayed as the data trace. To perform cross-trace math operations, use the following command:

:CALC{1-4}:MATH:FUNC

Configuring smoothing settings

To turn ON/OFF smoothing, use the following command:

:CALC{1-4}:SMO

The smoothing aperture is expressed as a percentage with respect to the sweep range. To set the smoothing aperture, use the following command:

:CALC{1-4}:SMO:APER

Selecting the data format

To select the measurement parameter data format, use the following command:

:CALC{1-4}:FORM

Configuring the display scale

Depending on the measurement parameter data format, you can configure the display scale in one of the following two ways:

Rectangular display formats:

When you use one of rectangular display formats, you can configure the display scale by setting the following four items:

Type	Command
Number of divisions	:DISP:WIND{1-4}:Y:DIV
Scale per division	:DISP:WIND{1-4}:TRAC{1-4}:Y:PDIV
Reference graticule line	:DISP:WIND{1-4}:TRAC{1-4}:Y:RPOS
Reference graticule line value	:DISP:WIND{1-4}:TRAC{1-4}:Y:RLEV

NOTE The number of divisions is a channel-wide setting (shared among all traces), while the remaining three settings are trace-specific.

You can show or hide graticule label (the label on the left-hand side of the graticule lines) by issuing the following command:

:DISP:WIND{1-4}:LAB

Smith chart/Polar formats:

When you are using one of Smith chart/Polar formats, you can only set the full scale value (the outermost circle's value) using the following command:

:DISP:WIND{1-4}:TRAC{1-4}:Y:PDIV

Auto Scale

You can use Auto Scale to automatically set the display scale. This feature works by automatically adjusting the reference division line value and the scale value per division when you are using one of the rectangular display formats or the full scale value when you are using one of Smith chart/Polar formats.

To perform Auto Scale, use the following command:

:DISP:WIND{1-4}:TRAC{1-4}:Y:AUTO

Displaying a message in the echo window

You can display a message in the echo window by issuing the following command:

:DISP:ECHO

You can clear any message displayed in the echo window by issuing the following command:

:DISP:ECHO:CLE

Turning ON/OFF display update

To turn ON/OFF the update of the LCD screen, use the following command:

:DISP:ENAB

Showing/hiding frequencies

To show or hide frequencies on the LCD screen, use the following command:

:DISP:ANN:FREQ

Showing or hiding the title

To show or hide the title, use the following command:

:DISP:WIND{1-4}:TITL

To define the title string that appears in the title display area, use the following command:

:DISP:WIND{1-4}:TITL:DATA

Configuring date/time display

To show or hide the current date and time on the right-hand side of the instrument status bar, use the following command:

:DISP:CLOC

To set the date and time, use the following command:

:SYST:DATE

:SYST:UPR

Turning ON/OFF the LCD backlight

To turn ON/OFF the LCD backlight, use the following command (note that turning OFF the backlight makes the screen unreadable):

:SYST:BACK

Setting Display Color

Selecting Display Mode

You can select one of the two LCD display modes: normal display (black background) or inverted display (white background).

To select the display mode, use the following command:

:DISP:IMAG

Setting display color for each item

To set the display colors, use the following commands:

Data trace	:DISP:COL{1-2}:TRAC{1-4}:DATA
Memory trace	:DISP:COL{1-2}:TRAC{1-4}:MEM
Graph	:DISP:COL{1-2}:GRAT{1-2}
Limit test	:DISP:COL{1-2}:LIM{1-2}
Background	:DISP:COL{1-2}:BACK

Resetting display colors to factory state

You can reset the display colors in normal display and inverted display to the preset factory state.

To reset the display colors, use the following command:

:DISP:COL{1-2}:RES

Saving and Loading the Settings

You can save the settings for measurement conditions and screen display to a file along with other instrument settings, and these settings can later be loaded from the file.

Once you have saved the measurement condition and screen display settings to a file, you can later load them whenever necessary; therefore, you can quickly modify the settings loaded from a file to create new settings without having to issue many commands.

To save the current settings to a file, use the following command:

:MMEM:STOR

To load the settings from a file, use the following command:

:MMEM:LOAD

Other topics about Setting up Analyzer

Performing Calibration

Performing Calibration

- Calibration
- Partial overwrite

Calibration

- Overview
- Performing Calibration
- Defining Calibration Kits
- Standard Definitions
- Reading/Writing Calibration Coefficient Alone
- Clearing Calibration Data and Calibration Coefficients

Other topics about Performing Calibration

Overview

You need to execute calibration to eliminate error elements related to measurement, thus allowing you to perform accurate measurement.

Performing Calibration (Obtaining calibration coefficients)

Selecting a Calibration Kit

To select a calibration kit, use the following command:

:SENS{1-4}:CORR:COLL:CKIT

Selecting a Calibration Type

The calibration coefficients are calculated based on the selected calibration type. Therefore, before you can calculate the calibration coefficients, you must select the appropriate calibration type by using one of the following commands.

Calibration type		Command
Response	OPEN	:SENS{1-4}:CORR:COLL:METH:OPEN :SENS{1-4}:CORR:COLL:METH:GPR:OPEN
	SHORT	:SENS{1-4}:CORR:COLL:METH:SHOR :SENS{1-4}:CORR:COLL:METH:GPR:SHOR
	THRU	:SENS{1-4}:CORR:COLL:METH:THRU :SENS{1-4}:CORR:COLL:METH:GPR:THRU
Enhanced Response		:SENS{1-4}:CORR:COLL:METH:ERES
1-Port		:SENS{1-4}:CORR:COLL:METH:SOLT1 :SENS{1-4}:CORR:COLL:METH:GPS1

Full 2-Port	:SENS{1-4}:CORR:COLL:METH:SOLT2

To check the currently selected calibration type, use the following command:

:SENS{1-4}:CORR:COLL:METH:TYPE?

Measuring Calibration Data

To measure the calibration data, use one of the following commands:

Calibration data items	Command
OPEN	:SENS{1-4}:CORR:COLL:OPEN :SENS{1-4}:CORR:COLL:GPAC:OPEN
SHORT	:SENS{1-4}:CORR:COLL:SHOR :SENS{1-4}:CORR:COLL:GPAC:SHOR
LOAD	:SENS{1-4}:CORR:COLL:LOAD :SENS{1-4}:CORR:COLL:GPAC:LOAD
THRU	:SENS{1-4}:CORR:COLL:THRU :SENS{1-4}:CORR:COLL:GPAC:THRU
Isolation	:SENS{1-4}:CORR:COLL:ISOL :SENS{1-4}:CORR:COLL:GPAC:ISOL

NOTE You cannot run more than one of the commands listed above at a time; if you issue another command before the currently running command completes successfully, the current command is aborted. Therefore, when you write a program that issues multiple calibration commands in series, you should use the *OPC? command or some other means to ensure that no command is executed before the preceding command completes itself.

As shown in the table below, the data required to calculate the calibration coefficients differ depending on the selected calibration type.

Calibration type (Selected ports are enclosed in parentheses)		Data				
		OPEN	SHOR T	LOAD	THRU	Isolati on
Respon se	OPEN (a)	a	Not requir	[a]	Not requir	Not require

			ed		ed	d
	SHORT (a)	Not required	a	[a]	Not required	Not required
	THRU (a-b)	Not required	Not required	Not required	a-b	[a-b]
Enhanced Response (a-b)		b	b	b	a-b	[a-b]
1-Port (a)		a	a	a	Not required	Not required
Full 2-Port (a-b)		a, b	a, b	a, b	a-b, b-a	[a-b], [b-a]

In the data section in the table, the letter m (for example, 1, a) represents the measurement data at port m; m-n (for example, 1-2, a-b) represents the measurement data between response port m and stimulus port n. You can omit data enclosed in brackets.

Calculating Calibration Coefficients

To calculate the calibration coefficients, use one of the following commands:

Calibration type	Command
Response, 1/2 port	:SENS{1-4}:CORR:COLL:SAVE

Before issuing the above commands, you must measure all required calibration data items according to your selected calibration type. Calculating the calibration coefficients clears all calibration data regardless of whether they are used for the calculation. The calibration type selection is also cleared, which results in a state where no calibration type is selected.

Turning ON/OFF Error Correction

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

:SENS{1-4}:CORR:STAT

Also, once you have calculated the calibration coefficient using the :SENS{1-4}:CORR:COLL:SAVE command, error correction is automatically turned on.

Using ECal

An ECal (Electronic Calibration) module allows you to perform 1/2-port calibration and response (THRU) calibration without having to replace the standard device.

ECal works by using the calibration kit data contained in the ECal module instead of the calibration kit data selected for the E5061B. This means that you do not have to define or select a calibration kit when using ECal.

NOTE When two or more ECal modules are connected through the USB port, the system uses the calibration kit data of the first ECal module.

To perform ECal, use one of the following commands:

Calibration type	Command
1-Port Calibration	:SENS{1-4}:CORR:COLL:ECAL :SOLT1
Full 2-Port Calibration	:SENS{1-4}:CORR:COLL:ECAL :SOLT2
Enhanced Response Calibration	:SENS{1-4}:CORR:COLL:ECAL :ERES
Response Calibration (THRU)	:SENS{1-4}:CORR:COLL:ECAL :THRU

Simply issuing one of the above commands completes all of the tasks necessary for error correction, including measuring the calibration data, calculating the calibration coefficients, and running the error correction feature.

NOTE Once you have initiated ECal, you cannot cancel the operation.

NOTE No command entered following the initiation of ECal is processed until ECal completes successfully. Accordingly, if you issue a command that queries some data, the system does not respond to the query until ECal is complete.

The below command is intended to turn ON/OFF the isolation measurement for performing ECal. However, as the isolation performance of ENA is better than ECal, this command no longer works. ENA ignores this command.

:SENS{1-4}:CORR:COLL:ECAL:ISOL

NOTE This command takes no action and only exists to maintain backward compatibility.

To select the ECal characteristic for a user-characterized ECal, use the following command:

:SENS{1-4}:CORR:COLL:ECAL:UCH

ECal Auto-detect Function

The ECal module can automatically detect which port of the ECal module is connected to the E5061B test port. Turn OFF the auto-detect function to specify a port manually.

To turn OFF the auto-detect function, use the following command.

:SENS:CORR:COLL:ECAL:ORI

To turn OFF the auto-detect function and set a port manually, use the following command.

:SENS:CORR:COLL:ECAL:PATH

Checking the Applied Calibration Type

When you turn on the error correction, you can check the calibration type actually applied to each trace. To check the calibration type, use the following command:

:SENS{1-4}:CORR:TYPE{1-4}?

Defining Calibration kits

Selecting a Calibration Kit

To select a calibration kit, use the following command:

:SENS{1-4}:CORR:COLL:CKIT

Setting the Calibration Kit Name

To set the name of a calibration kit, use the following command:

:SENS{1-4}:CORR:COLL:CKIT:LAB

Standard Definitions

Selecting a Standard Type

To select a standard type, use the following command:

:SENS{1-4}:CORR:COLL:CKIT:STAN{1-21}:TYPE

Setting the Standard Name

To set the standard name, use the following command:

:SENS{1-4}:CORR:COLL:CKIT:STAN{1-21}:LAB

Setting the Standard Value

To set the standard value, use one of the following commands:

Item	Command
C0	:SENS{1-4}:CORR:COLL:CKIT:STAN{1-21} :C0
C1	:SENS{1-4}:CORR:COLL:CKIT:STAN{1-21} :C1
C2	:SENS{1-4}:CORR:COLL:CKIT:STAN{1-21} :C2
C3	:SENS{1-4}:CORR:COLL:CKIT:STAN{1-21} :C3
L0	:SENS{1-4}:CORR:COLL:CKIT:STAN{1-21} :L0
L1	:SENS{1-4}:CORR:COLL:CKIT:STAN{1-21} :L1
L2	:SENS{1-4}:CORR:COLL:CKIT:STAN{1-21} :L2
L3	:SENS{1-4}:CORR:COLL:CKIT:STAN{1-21} :L3
Offset Delay	:SENS{1-4}:CORR:COLL:CKIT:STAN{1-21} :DEL
Offset Loss	:SENS{1-4}:CORR:COLL:CKIT:STAN{1-21} :LOSS
Offset Z0	:SENS{1-4}:CORR:COLL:CKIT:STAN{1-21} :Z0
Arbitrary Impedance	:SENS{1-4}:CORR:COLL:CKIT:STAN{1-21}: ARB

As shown in the table below, you need to set different items depending on the standard type.

Standard Types	C0 to C3	L0 to L3	Offset Delay	Offset Loss	Offset Z0	Arbitrary Impedance	Min. Freq.	Max. Freq.	Connector Type
OPEN	*		*	*	*		*	*	*
SHORT		*	*	*	*		*	*	*
LOAD			*	*	*		*	*	*

THRU			*	*	*		*	*	*
Arbitrary Impedance			*	*	*	*	*	*	*

You need to set the items identified by * marks in the table above.

Defining a Standard Class Assignment

To select the standard to be applied to the OPEN measurement for each port, use the following command:

:SENS{1-4}:CORR:COLL:CKIT:ORD:OPEN

To select the standard to be applied to the OPEN measurement for Gain-Phase calibration, use the following command:

:SENS{1-4}:CORR:COLL:CKIT:GPOR:OPEN

To select the standard to be applied to the SHORT measurement for each port, use the following command:

:SENS{1-4}:CORR:COLL:CKIT:ORD:SHOR

To select the standard to be applied to the SHORT measurement for Gain-Phase calibration, use the following command:

:SENS{1-4}:CORR:COLL:CKIT:GPOR:SHOR

To select the standard to be applied to the LOAD measurement for each port, use the following command:

:SENS{1-4}:CORR:COLL:CKIT:ORD:LOAD

To select the standard to be applied to the LOAD measurement for Gain-Phase calibration, use the following command:

:SENS{1-4}:CORR:COLL:CKIT:GPOR:LOAD

To select the standard to be applied to the THRU measurement between each pair of ports, use the following command:

:SENS{1-4}:CORR:COLL:CKIT:ORD:THRU

To select the standard to be applied to the THRU measurement for Gain-Phase calibration, use the following command:

:SENS{1-4}:CORR:COLL:CKIT:GPOR:THRU

Saving and loading calibration coefficients

You can save calibration coefficients to a file along with other instrument settings and then later load them from the file.

By default, the system does not save calibration coefficients when it saves instrument settings. Therefore, to save calibration coefficients, you must explicitly configure the system to save them by issuing the following command:

:MMEM:STOR:STYP

To save calibration coefficients to a file, use the following command:

:MMEM:STOR

To load calibration coefficients from a file, use the following command:

:MMEM:LOAD

For more information on how to save and load calibration coefficients, refer to Saving and recalling instrument status

Reading/Writing Calibration Coefficient Alone

The calibration coefficient alone can be read from and written to the E5061B by using the following command:

:SENS{1-4}:CORR:COEF

To write a positive calibration coefficient, use one of the following commands to declare the calibration type:

:SENS{1-4}:CORR:COEF:METH:ERES

:SENS{1-4}:CORR:COEF:METH:GPR:OPEN

:SENS{1-4}:CORR:COEF:METH:GPR:SHOR

:SENS{1-4}:CORR:COEF:METH:GPR:THRU

:SENS{1-4}:CORR:COEF:METH:GPS1

:SENS{1-4}:CORR:COEF:METH:OPEN

:SENS{1-4}:CORR:COEF:METH:SHOR

:SENS{1-4}:CORR:COEF:METH:SOLT1

:SENS{1-4}:CORR:COEF:METH:SOLT2

:SENS{1-4}:CORR:COEF:METH:THRU

To validate the written calibration coefficient, use the following command:

:SENS{1-4}:CORR:COEF:SAVE

About Calibration Types and Coefficients

The following table shows the required calibration coefficients for each calibration type.

Calibration Type	Calibration Coefficient					
	ES	ER	ED	EL	ET	EX

Response calibration (OPEN)		*	*			
Response calibration (SHORT)		*	*			
Response calibration (THRU)					*	*
Enhanced response calibration	*	*	*		*	*
1-port calibration	*	*	*			
Full 2-port calibration	*	*	*	*	*	*

NOTE If either an invalid calibration coefficient is specified for the writing command or a nonexistent calibration coefficient is specified for its reading command, the following error occurs: 23, Specified error term does not exist.

Procedures for Writing Calibration Coefficient

You must follow the steps below to write the calibration coefficient.

1. Declare the calibration type to write.

Execute :SENS{1-4}:CORR:COEF:METH:xxxx command

2. Write any calibration coefficient.

Execute :SENS{1-4}:CORR:COEF command as needed for the written calibration coefficients

3. Validate the calibration coefficients.

Execute :SENS{1-4}:CORR:COEF:SAVE command

NOTE Do not execute any other command while writing the calibration coefficients. This may cause the system to function incorrectly.

Clearing Calibration Data and Calibration Coefficients

Clearing Calibration Data

You can use the following command to clear the measurement values of calibration data executed with :SENS{1-4}:CORR:COLL:OPEN command, etc.

:SENS{1-4}:CORR:COLL:CLE

These clear functions make the temporary settings during the calibration, such as trace number and measurement parameters, recover to the original state.

Clearing Calibration Coefficients

You can use the following command to clear the calibration coefficients used.

:SENS{1-4}:CORR:CLE

Partial Overwrite

- Overview
- Executing_calculation_of_calibration_coefficients_using_partial_overwrite

Other topics about Performing Calibration

Overview

The E5061B has the following calibration coefficients for full N-port calibration: Er, Es, Ed (reflection), Et (transmission), and Ex (isolation).The partial overwrite function is used to measure some of these calibration coefficients after completion of the initial calibration and then overwrite them.

The conditions under which the calibration coefficients can be calculated by the partial overwrite are as follows:

- Calibration is completed once and valid (status other than C? or C!)
- One or more measurements for re-calculation are performed.

NOTE The isolation calibration coefficient, Ex, cannot be returned to the initial value, 0, once it is calculated.

If calculation of the calibration coefficients is attempted without the measurements required to execute the partial overwrite, an error message (20: Additional Standard Needed) is displayed.

Executing calculation of calibration coefficients using partial overwrite

To calculate the calibration coefficients using partial overwrite, use the following command:

:SENS{1-4}:CORR:COLL:PART:SAVE

NOTE Before you can calculate the calibration coefficients with the partial overwrite, you must select the appropriate calibration type in the same way used for normal calibration. If calculation of the calibration coefficients is attempted without selecting the calibration type, an error message (28: Invalid Calibration Method) is displayed.

Making Measurement

Making Measurement

- Trigger System
- Starting a Measurement Cycle (triggering the instrument)
- Waiting for the End of Measurement
- Detecting Occurrence of an Error
- Improving Command Processing Speed

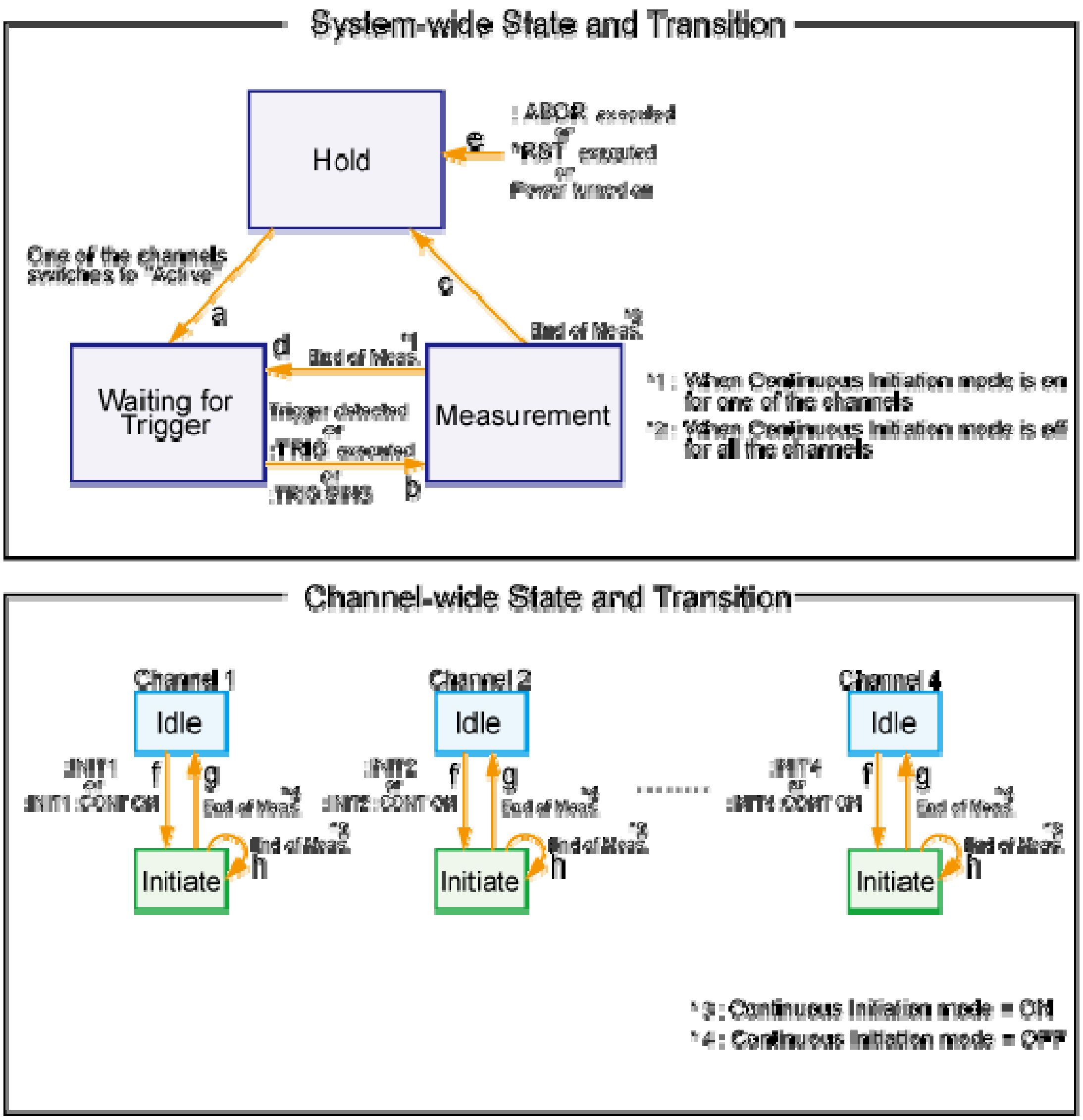
Trigger System

Other topics about Making Measurement

Overview

The trigger system is responsible for such tasks as detecting the start of a measurement cycle (triggering) and enabling/disabling measurement on each channel. As shown in the following figure, the trigger system has two types of states: system-wide and channel-wide. The system-wide state can be "Hold", "Waiting for Trigger", or "Measurement", while the channel-wide state can be "Idle" or "Initiate".

Trigger system



e5061b023

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

System-Wide States and Transitions

"Hold" State

The trigger system switches to "Hold" state when one of the following commands is executed (arrow "e" in Trigger system). Also, turning ON the power to the instrument puts the trigger system into "Hold" state. When the power is turned ON, the continuous initiation mode is ON for channel 1 and the trigger source is set to "Internal". Accordingly, the trigger system immediately switches to "Waiting for Trigger" state and subsequently repeats transitions between "Measurement" and "Waiting for Trigger" states.

- :ABOR
- *RST

When the trigger system is in "Hold" state and one of the channels switches to "Initiate" state (arrow "f" in Trigger system), the trigger system switches to "Waiting for Trigger" state (arrow "a" in Trigger system).

"Waiting for Trigger" State

When the trigger system is in "Waiting for Trigger" state and either the instrument is triggered (i.e., a trigger is detected) or one of the following commands is executed, the trigger system switches to "Measurement" state (arrow "b" in Trigger system)

- :TRIG
- :TRIG:SING

As shown in the table below, the instrument is triggered differently depending on which trigger source is specified. To specify the trigger source, use the :TRIG:SOUR command.

Trigger Source	How instrument is triggered
Internal trigger	The instrument is automatically triggered within itself.
External trigger	The instrument is triggered when a trigger signal is input through the Ext Trig terminal or the handler interface
Bus trigger	The instrument is triggered when the *TRG command is issued.
Manual trigger	The instrument is triggered when you press **Trigger** > **Trigger** on the front panel.

"Measurement" State

In "Measurement" state, the instrument waits for the elapse of the sweep delay time (set by the :SENS{1-4}:SWE:DEL) and then starts a measurement cycle. This process is performed sequentially on each of those channels that are in "Initiate" state immediately before the transition to this state, in ascending order of channel number.

When the instrument has finished measuring all the active channels, the trigger system behaves in one of the following ways depending on the setting of the continuous initiation mode.

If continuous initiation mode is OFF for all channels:

The trigger system switches to "Hold" state (arrow "c" in Trigger system).

If continuous initiation mode is ON for one of the channels:

The trigger system switches to "Waiting for Trigger" state (arrow "d" in Trigger system).

Channel-wide States and Transitions

"Idle" State

A channel switches to "Initiate" state when one of the following commands is executed (arrow "f" in Trigger system).

- :INIT{1-4}
- :INIT{1-4}:CONT("ON" specified)

"Initiate" State

A channel in this state is measured just before the entire system switches to "Measurement" state.

When the instrument has finished measuring a channel, the channel behaves in one of the following ways depending on the setting of the continuous initiation mode (set by the :INIT{1-4}:CONT).

- If continuous initiation mode is OFF: The channel switches to "Idle" state (arrow "g" in Trigger system).
- If continuous initiation mode is ON: The channel remains in "Initiate" state (arrow "h" in Trigger system).

Starting a Measurement Cycle (Triggering the Instrument)

- Configuring the Instrument
- Starting Measurement on Demand

Other topics about Making Measurement

Configuring the Instrument to Automatically Perform Continuous Measurement

1. Use the :INIT{1-4}:CONT command to turn ON the continuous initiation mode for the channels you want to measure and turn the mode OFF for any other channel.
2. Issue the :TRIG:SOUR command to set the trigger source to Internal trigger.

Starting Measurement on Demand

1. Use the :INIT{1-4}:CONT command to turn ON the continuous initiation mode for the channels you want to measure and turn the mode OFF for any other channel.
2. Issue the :TRIG:SOUR command to set the trigger source to "Bus Trigger".
3. Trigger the instrument whenever you want to perform the measurement. An external controller can trigger the instrument by using one of the following three commands:

<table>
<tr><th>Command</th><th>Can *OPC? command be used to wait for end of sweep?</th><th>Applicable trigger source</th></tr>
<tr><td>*TRG</td><td rowspan="2">No</td><td>Bus trigger only</td></tr>
<tr><td>:TRIG</td><td rowspan="2">External trigger
Bus trigger
Manual trigger</td></tr>
<tr><td>:TRIG:SING</td><td>Yes</td></tr>
</table>

4. Repeat step 3 to start the next measurement cycle.

Waiting for the End of Measurement

Other topics about Making Measurement

Using the Status Register

The status of the E5061B can be detected through the status registers. This section explains how to detect the end of measurement by using the status registers.

Measurement status is reported by the operation status condition register. An SRQ (service request) is useful when creating a program that uses the information reported by this register to detect the end of measurement.

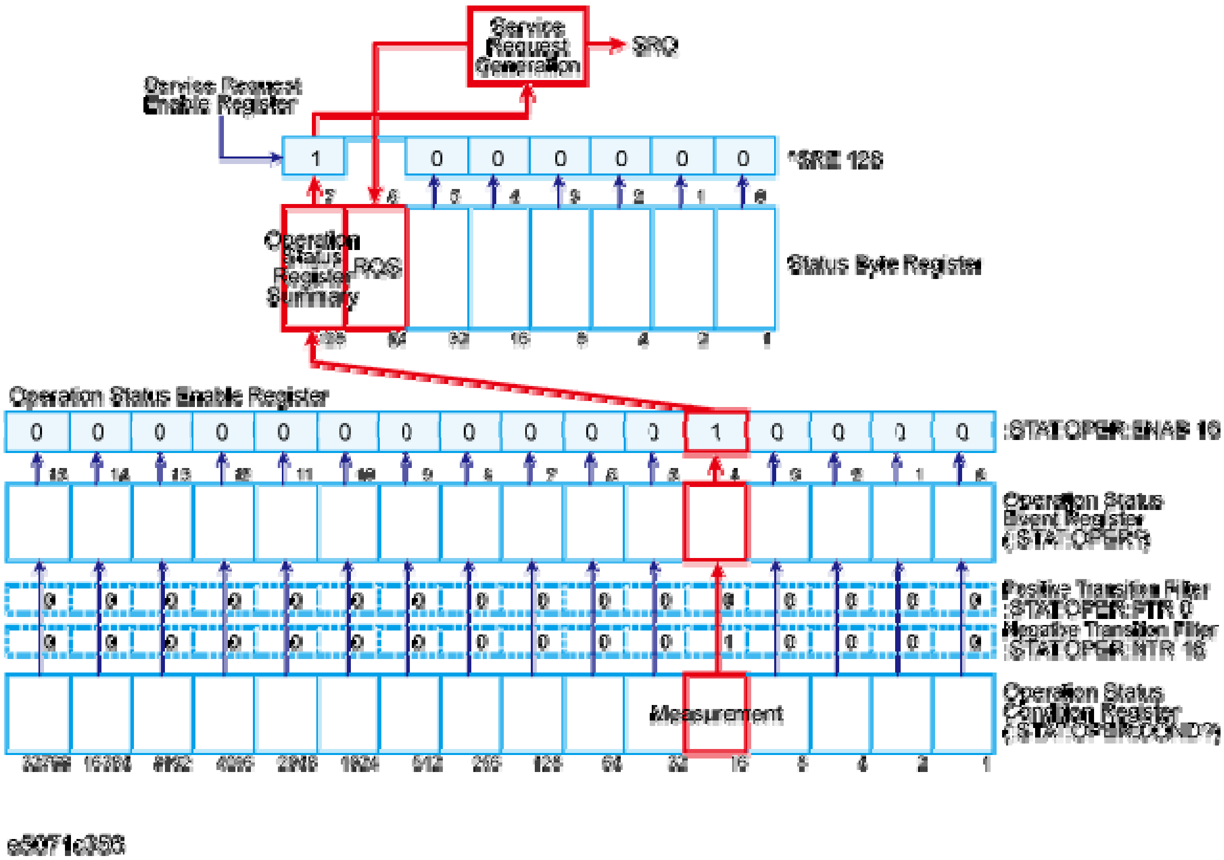
To detect the end of measurement via an SRQ, use one of the following commands:

- *SRE
- :STAT:OPER:ENAB
- :STAT:OPER:PTR
- :STAT:OPER:NTR

Follow these steps to utilize an SRQ:

1. Configure the E5061B so that it generates an SRQ when the operation status condition register's bit 4 (a bit that is set to 1 during measurement) is changed from 1 to 0.
2. Trigger the instrument to start a measurement cycle.
3. When an SRQ is generated, the program interrupts the measurement cycle.

SRQ generation sequence (at end of measurement)



Sample Program

See the Waiting for Trigger (SRQ).

Using :TRIG:SING Command

When you trigger the instrument by issuing the :TRIG:SING command, you can use the *OPC command to wait until the measurement cycle is completed.

Sample Program

See the Waiting for Trigger (OPC?).

Using Wait Time

Before creating your program, actually measure the time between the start and end of the measurement cycle. Then code your program so that the controller waits for the actual measured time by using the appropriate command (for example, the WAIT command for HTBasic). This is a straightforward method, but care must be taken: an incorrect wait time could result in an unexpected error.

Detecting Occurrence of an Error

- Using Status Reporting System
- Using Error Queue
- Sample Program

Other topics about Working with Automatic Test System

Using Status Reporting System

The status of the E5061B can be detected through the status registers. This section describes how to detect the end of measurement by using the status registers.

The occurrence of an error presents in the standard event status register. An SRQ (service request) is useful when you create a program that uses the information reported by this register to detect the occurrence of an error.

To detect the end of sweep via an SRQ, use one of the following commands:

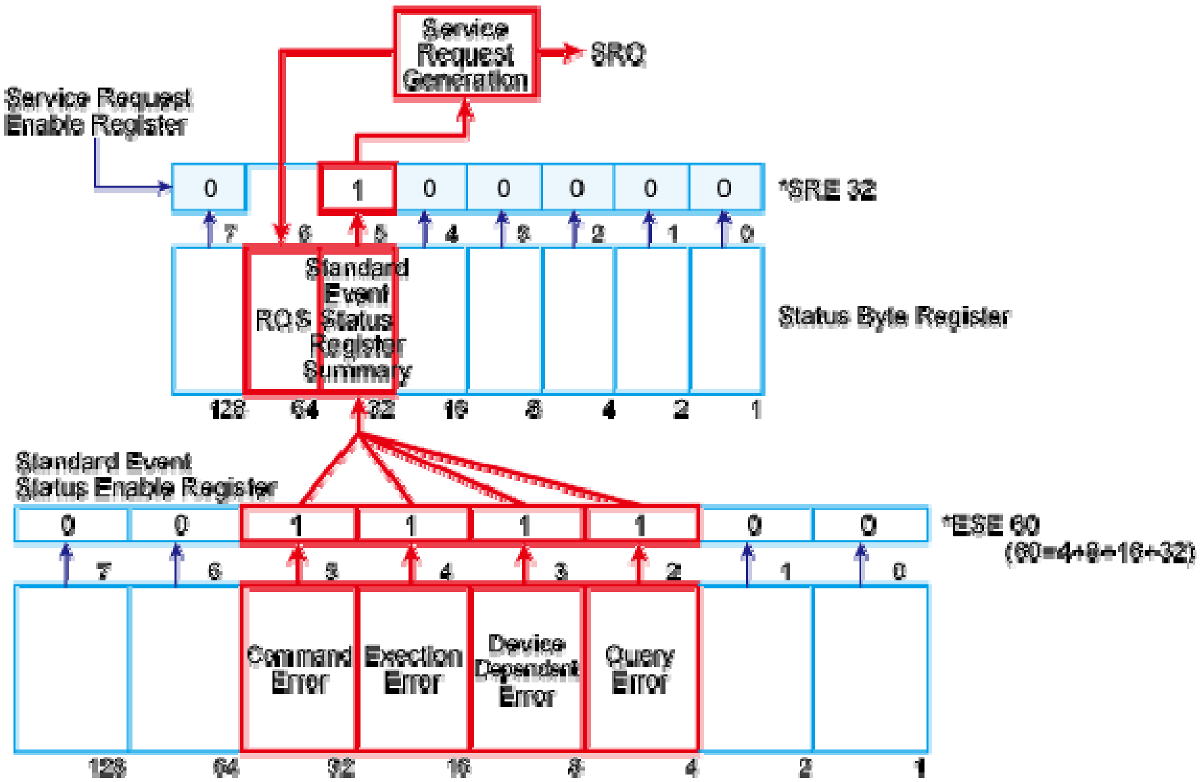
*SRE

*ESE

Follow these steps:

1. Set the E5061B so that it generates an SRQ when any of the error occurrence bits is set to 1 in the standard event status register.
2. When an SRQ is generated, the program interrupts the measurement cycle.

SRQ generation sequence (when an error occurs)



e5071c357

Using Error Queue

An error queue holds the number for the error and the error message. Reading the error queue allows the user to verify the error that has occurred. To retrieve the content of an error queue, use the following command:

:SYST:ERR?

The error queue can be used in the following ways:

1. It is used as a branch for error handling. When an error queue is retrieved, it returns 0 as the error number and "No error" as the error message if no error is detected. This can be used to detect an error and for branching the flow of a program. This is also useful when you wish to handle specific error(s). Note that this method does not allow the user to perform any processing during the occurrence of an error.
2. When an error is detected using SRQ, the error queue is used to examine the error. Refer to the sample program in this section.

Sample Program

See Error Detection (SRQ).

Improving Command Processing Speed

SCPI commands should be processed quickly to improve the throughput when such commands are frequently executed (for example, reading out traces for each measurement).

With the E5061B, the processing time for SCPI commands can be improved by decreasing the refresh rate of the LCD display.

Measurement results (trace) do not need to be updated

When the measurement trace does not need to be updated, turn OFF the updating feature of the LCD display. This improves the processing speed of SCPI commands and eliminates the time used for updating the screen.

To turn OFF the updating feature of the LCD display, use the following command:

:DISP:ENAB

Measurement results (trace) need to be updated

When the measurement trace needs to be updated, the processing speed of SCPI commands can still be improved by controlling the update timing of the LCD display:

1. Execute all SCPI commands that are required before measurement, including commands setting conditions.
2. Turn OFF the update of the LCD display.
3. Perform the measurement.
4. Execute the commands for reading out measurement result or analyzing the result. Note that reading out the result in binary format accelerates data transfer.
5. Execute the following command to update the LCD display once :DISP:UPD
6. Return to Step 3.

Sample program

See Control LCD Update Timing.

Reading-Writing Measurement Data

Reading-Writing Measurement Data

Data Transfer Format

Other topics about Reading-Writing Measurement Data

Overview

When you transfer data using the one of the following commands, you can choose among ASCII transfer format, IEEE 64-bit floating point binary transfer format and IEEE 32-bit floating point binary transfer format.

NOTE The instrument always uses the ASCII transfer format when you transfer data without using any of the following commands:

- :CALC{1-4}:BLIM:REP?
- :CALC{1-4}:DATA:FDAT
- :CALC{1-4}:DATA:FMEM
- :CALC{1-4}:DATA:SDAT
- :CALC{1-4}:DATA:SMEM
- :CALC{1-4}:FUNC:DATA?
- :CALC{1-4}:LIM:DATA
- :CALC{1-4}:LIM:REP?
- :CALC{1-4}:LIM:REP:ALL?
- :CALC{1-4}:RLIM:DATA
- :CALC{1-4}:RLIM:REP?
- :SENS{1-4}:CORR:COEF
- :SENS{1-4}:CORR:COEF:GPD
- :SENS{1-4}:FREQ:DATA?
- :SENS{1-4}:SEGM:DATA

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

:FORM:DATA

NOTE Executing the :SYST:PRES or *RST does not affect the current setting of the data transfer format.

ASCII Transfer Format

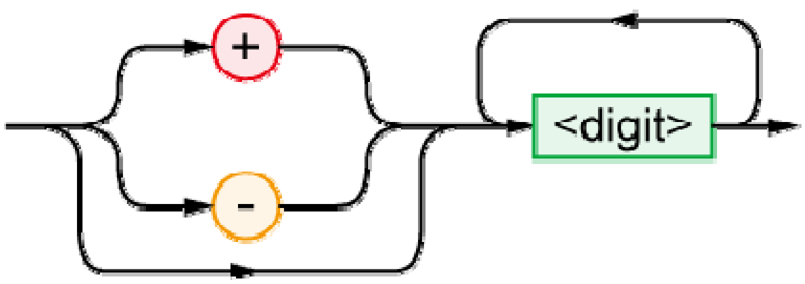
When you select the ASCII transfer format as the data transfer format, numbers are transferred as ASCII bytes, each of which corresponds to one of the formats shown below. Note that numbers are separated from one another with a comma (,) in accordance with the IEEE 488.2 specification.

NOTE Numeric data strings vary in length. Keep this in mind when you extract some data from retrieved numeric data strings in your program.

Integer Format

The figure below shows this format. Numbers are expressed as integers. For example, 201 is expressed as "+201" or "201."

Integer format

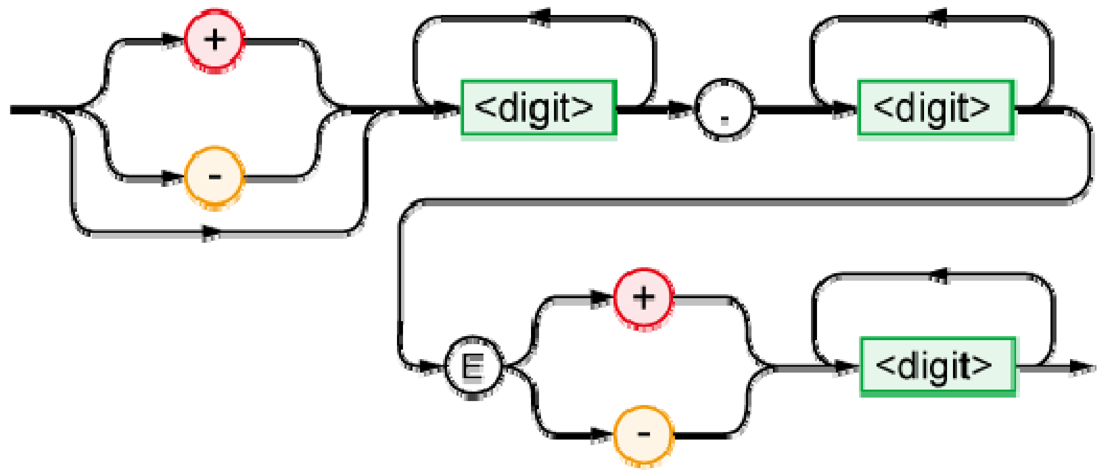


e5071c351

Floating-Point Number Format

The figure below shows this format. Numbers are expressed with floating points. The number of decimal is 12 at default. For example, 1000 is expressed as "+1.00000000000E+003." You can change the number of decimal to 14 by the SCPI.FORMat.REAL.ASCii.LENGth command.

Floating-point number format



e5071c352

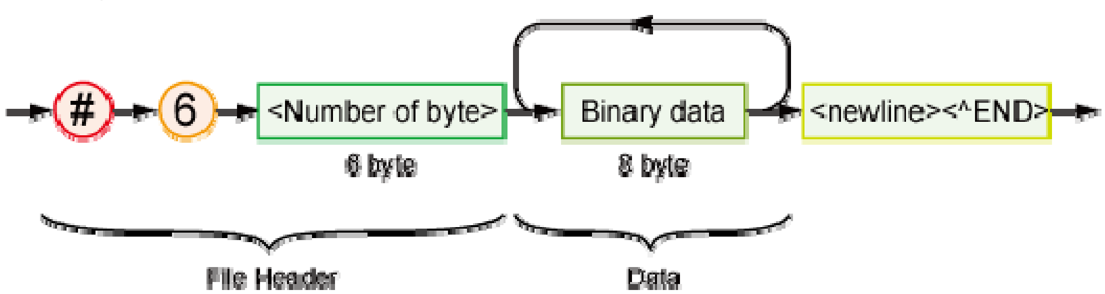
Binary Transfer Format

You can select the binary transfer format from the IEEE 64-bit floating point format or the IEEE 32-bit floating point format depending on the controller you use.

IEEE 64-bit floating point format

When you select the IEEE 64-bit floating point binary transfer format as the data transfer format, numbers are transferred in the format shown in the figure below.

Binary transfer format

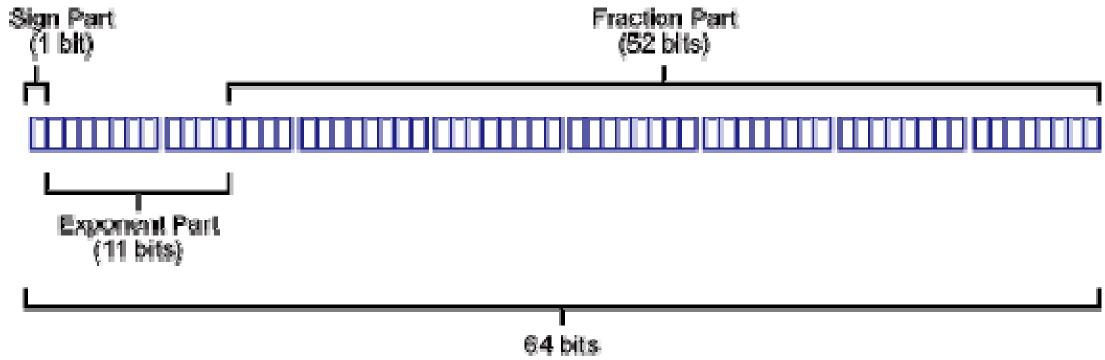


e5071c480

This data transfer format uses a header that consists of a sharp character (#), a number of 6 (which indicates the byte size of the <number of bytes transferred> part), and the <number of bytes transferred> part in this order. The header is followed by the binary data (each number consists of 8 bytes and the total is the byte size indicated by <number of bytes transferred>) and the message terminator <new line>^END.

The binary data is expressed in the IEEE 754 64-bit floating-point number format shown in the figure below.

64-bit floating point format

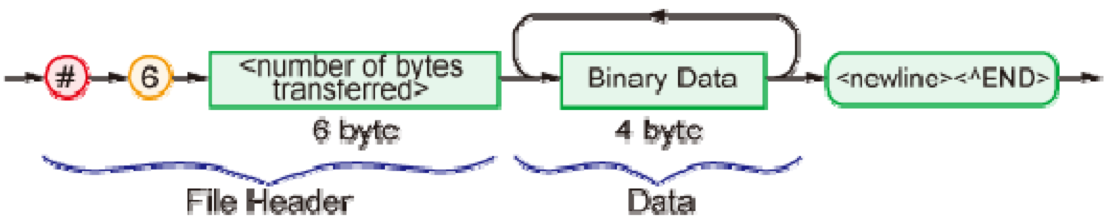


e5071c482

IEEE 32-bit floating point format

When you select the IEEE 32-bit floating point binary transfer format as the data transfer format, numbers are transferred in the format shown in the figure below.

IEEE 32-bit floating point binary transfer format

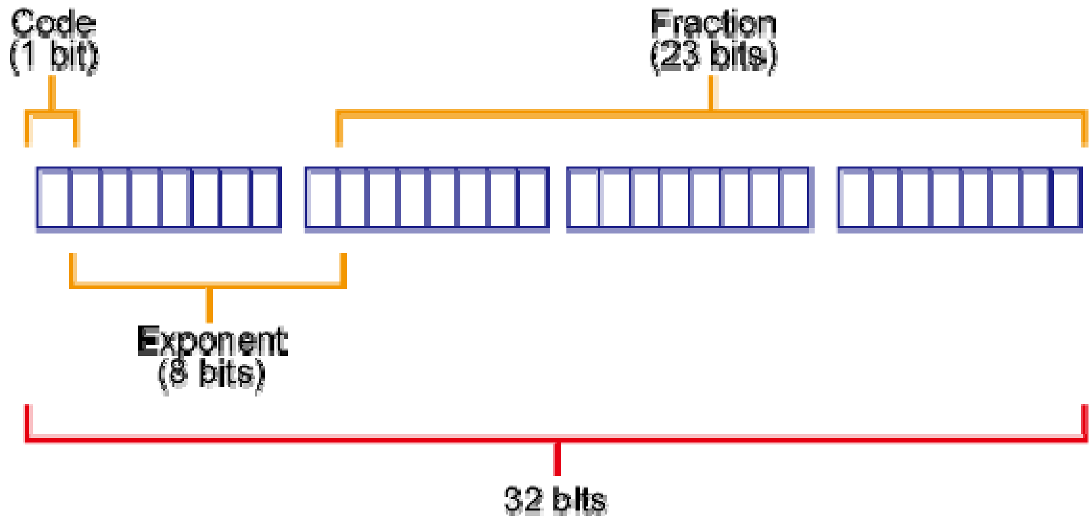


e5071c439

This data transfer format uses a header that consists of a sharp character (#), a number of 6 (which indicates the byte size of the <number of bytes transferred> part), and the <number of bytes transferred> part in this order. The header is followed by the binary data (each number consists of 4 bytes and the total is the byte size indicated by <number of bytes transferred>) and the message terminator <new line>^END.

The binary data is expressed in the IEEE 754 32-bit floating-point number format shown in the figure below.

32-bit floating point data



e5071c440

Byte order

When you opt to perform binary transfer, you can configure the instrument to transfer the bytes of the data in one of the following two byte orders:

NORMal

Transfer begins with the byte that contains the MSB (Most Significant Bit); that is, the leftmost byte in 64 bit floating point format and 32 bit floating point data.

SWAPped

Transfer begins with the byte that contains the LSB (Least Significant Bit); that is, the rightmost byte in 64 bit floating point format and 32 bit floating point data.

To set the byte order, use the following command:

:FORM:BORD

NOTE Executing the :SYST:PRES or *RST does not affect the current setting of the byte order.

Internal Data Processing

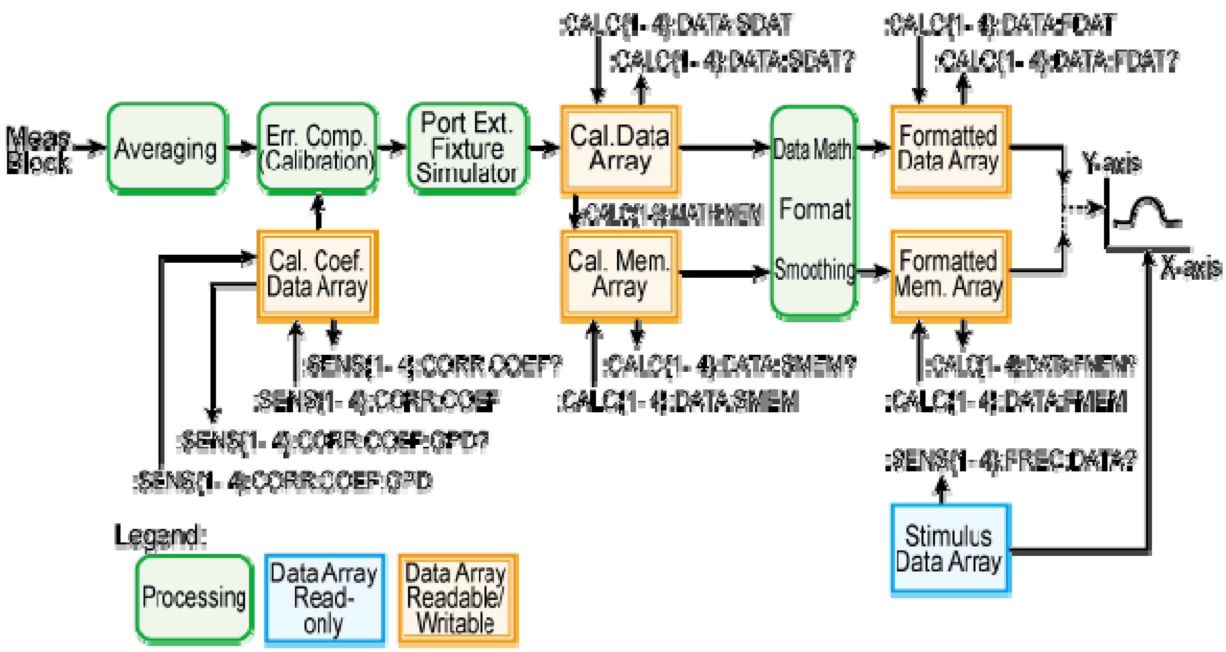
- [Data Flow]
- [Internal Data Arrays]

Other topics about Reading-Writing Measurement Data

Data Flow

The following figure provides an overview of the E5061B's internal data processing flow.

E5061B's data processing flow



e5061b028

Internal Data Arrays

Corrected data arrays

A corrected data array contains the corrected data obtained by performing error correction and port extension compensation (calibration) operations on the raw measured data of S parameter specified for each trace of each channel. Each data element is stored as a complex number (Re/Im).

The instrument retains 16 corrected data arrays at maximum, each of which is associated with one of the 4 traces contained in one of the 4 channels (4 × 4 = 16). To read/write one of the corrected data arrays, use the following command:

:CALC{1-4}:DATA:SDAT

Corrected memory arrays

When the :CALC{1-4}:MATH:MEM command is executed on a particular corrected data array, its copy is stored into the corrected memory array corresponding to that corrected data array.

The instrument retains 16 corrected memory arrays at maximum, each of which is associated with one of the 4 traces contained in one of the 4 channels (4 × 4 = 16). To read/write one of the corrected data arrays, use the following command:

:CALC{1-4}:DATA:SMEM

Formatted data array

A formatted data array contains the formatted data (values to be displayed) obtained by performing data math operations, measurement parameter conversion, and smoothing on a particular corrected data array. Regardless of the data format, it contains two data elements per measurement point as shown in the following table:

Data format	Data element (primary value)	Data element (secondary value)
log magnitude	log magnitude	Always 0
Phase	Phase	Always 0
Group delay	Group delay	Always 0
Smith chart (Lin)	Liner magnitude	Phase
Smith chart (Log)	log magnitude	Phase
Smith chart (Re/Im)	Real part of a complex number	Imaginary part of a complex number
Smith chart (R+jX)	Resistance	Reactance
Smith chart (G+jB)	Conductance	Susceptance
Polar (Lin)	Liner magnitude	Phase

Polar (Log)	log magnitude	Phase
Polar (Re/Im)	Real part of a complex number	Imaginary part of a complex number
Liner magnitude	Liner magnitude	Always 0
SWR	SWR	Always 0
Real number	Real part of a complex number	Always 0
Imaginary number	Imaginary part of a complex number	Always 0
Expanded phase	Expanded phase	Always 0
Positive phase	Positive phase	Always 0

The instrument retains 16 formatted data arrays at maximum, each of which is associated with one of the 4 traces contained in one of the 4 channels (4 × 4 = 16). To read/write one of the formatted data arrays, use the following command:

:CALC{1-4}:DATA:FDAT

Formatted memory arrays

A formatted memory array contains the formatted data (values to be displayed) obtained by performing data math operations, measurement parameter conversion, and smoothing on a particular corrected memory array.

The instrument retains 16 formatted memory arrays at maximum, each of which is associated with one of the 4 traces contained in one of the 4 channels (4 × 4 = 16). To read/write one of the formatted memory arrays, use the following command:

:CALC{1-4}:DATA:FMEM

Stimulus data arrays

A stimulus data array contains the stimulus values for all measurement points.

The instrument retains 4 stimulus data arrays at maximum, each of which is associated with one of the 4 channels. Stimulus data arrays are read-only. To retrieve one of the stimulus data arrays, use the following command:

:SENS{1-4}:FREQ:DATA?

Calibration Coefficient Data Arrays

A calibration coefficient data array contains the calibration coefficients calculated based on the results of measurement performed with standard devices.

The instrument retains 12 calibration coefficient data arrays at maximum, each of which is associated with each channel. Commands are available for reading or writing calibration coefficient data arrays. To read or write, first use the following command:

:SENS{1-4}:CORR:COEF

NOTE If any calibration coefficient is interpolated, the interpolated calibration coefficient data array is retrieved. Once a calibration coefficient data array has been written, execute the command SENS{1-4}:CORR:COEF:SAVE to validate it.

Retrieving Measurement Results

- Overview
- Retrieving Internal Data Arrays
- Sample Program

Other topics about Reading-Writing Measurement Data

Overview

Internal data arrays allows you to retrieve all measurement results throughout a particular trace. Alternatively, markers allow you to retrieve measurement results at your specified points. For information on how to retrieve marker values, refer to Retrieving measurement results at marker positions.

Retrieving Internal Data Arrays

You can chose between the ASCII and binary data transfer formats when you retrieve internal data arrays. For more information, please refer to Data Transfer Format.

Sample Program

See Reading Data in Ascii Format and Reading Data in Binary Format.

Entering Data into a Trace

- Overview
- Sample Program

Other topics about Reading-Writing Measurement Data

Overview

You can change the data/memory trace on the LCD by writing the new data into the Formatted data array/Formatted memory arrays.

When you write data into formatted data/memory array, you can choose either the ASCII or binary transfer format (see Data Transfer Format).

Using the ASCII Transfer Format to Write Formatted Data Arrays (write_a.htb) and Using the Binary Transfer Format to Write Formatted Data Arrays (write_b.htb) show sample programs that demonstrate how to write data into formatted data arrays. The sample program in Using the ASCII Transfer Format to Write Formatted Data Arrays (write_a.htb) uses the ASCII transfer format while the sample in Using the Binary Transfer Format to Write Formatted Data Arrays (write_b.htb) uses the binary transfer format. You can find the source files of these programs, named write_a.htb and write_b.htb, on the sample program disk.

Each of the sample programs holds the sweep on channel 1, retrieves the data from a specified file (a file saved measurement data using the :MMEM:STOR:FDAT command), and populates trace 1 for channel 1 with the retrieved data.

Sample Program

See Writing Data in Ascii Format and Writing Data in Binary Format.

Analyzing Data

Analyzing Data

- Retrieving Measurement Results at Specified Points
- Searching for Positions That Match Specified Criteria
- Obtaining Limit/Bandwidth Limit/Ripple Limit Test Results
- Analysis in Time Domain (time domain function)

Retrieving Measurement Results at Specified Points

- Overview
- Showing/Hiding Markers
- Turning ON/OFF Reference Marker Mode
- Setting and Retrieving Stimulus Value at Marker Positions
- Retrieving Measurement Results at Marker Positions

Other topics about Analyzing Data

Overview

Markers allow you to retrieve measurement results at specified points. You can use up to nine markers for each trace, and you can move them to any point on the trace. In addition to the regular markers, you can use a reference marker.

Showing/Hiding Markers

To show or hide markers, including the reference marker, use the following command:

:CALC{1-4}:MARK{1-10}

NOTE You can move markers or retrieve the data at a marker even when the markers are hidden.

NOTE The display of the reference marker is turned ON or OFF when you turn ON or OFF Reference Marker mode.

Turning ON/OFF Reference Marker Mode

Turning ON the Reference Marker mode provides relative marker values with respect to the reference marker (by subtracting the value at the reference marker from the value at a particular marker).

To turn ON or OFF Reference Marker mode, use the following command:

:CALC{1-4}:MARK:REF

Setting and Retrieving Stimulus Value at Marker Positions

To set (or change along the frequency axis) the stimulus value at a particular marker or the reference maker or to retrieve the current stimulus value, use the following command:

:CALC{1-4}:MARK{1-10}:X

When Reference Marker mode is ON, the stimulus value at a regular marker is a relative stimulus value obtained by subtracting the stimulus value at the reference marker from the actual stimulus value at that particular marker.

Retrieving Measurement Results at Marker Positions

To retrieve the measurement results (response values) at a particular marker or the reference marker, use the following command:

:CALC{1-4}:MARK{1-10}:Y?

When Reference Marker mode is ON, the response value at a regular marker is a relative value obtained by subtracting the response value at the reference marker from the actual response value at that particular marker.

Searching for Positions Matching Specified Criteria

- Overview
- Using Marker Search
- Analysis Commands
- Sample Program

Other topics about Analyzing Data

Overview

You can search for a position that matches specified criteria by using the Marker Search feature or analysis commands.

Using Marker Search

NOTE Marker Search is available whether the markers are shown or hidden.

Setting the Search Range

You can use either the entire sweep range or a user-defined range for the marker search range by using the following command:

:CALC{1-4}:MARK:FUNC:DOM

When you opt to use a user-defined range, use the following commands to set the range:

Description	Command
Start value (lower limit value)	:CALC{1-4}:MARK:FUNC:DOM :STAR
Stop value (upper limit value)	:CALC{1-4}:MARK:FUNC:DOM :STOP

You can also select whether to specify the marker search range independently for each trace by using the following command.

:CALC{1-4}:MARK:FUNC:DOM:COUP

Selecting a Search Type

Marker Search allows you to choose from the following eight search types:

- Maximum value
- Minimum value
- Peak (3 types)
  - Maximum peak (for a positive peak), minimum peak (for a negative peak)

  - Peak nearest to the marker position on its left-hand side
  - Peak nearest to the marker position on its right-hand side
- Target (3 types)
  - Peak nearest to the marker position
  - Target nearest to the marker position on its left-hand side
  - Target nearest to the marker position on its right-hand side

To select a search type, use the following command:

:CALC{1-4}:MARK{1-10}:FUNC:TYPE

Defining a Peak

You can define a peak by specifying the lower limit for the peak excursion value and polarity (positive or negative peak).

To define a peak, use the following commands:

Lower limit for the peak excursion value	:CALC{1-4}:MARK{1-10}:FUNC:PEXC
Polarity	:CALC{1-4}:MARK{1-10}:FUNC:PPOL

Defining a Target

You can define a target by specifying the target value (response value) and transitional direction (positive or negative value change).

To define a target, use the following commands:

Target value	:CALC{1-4}:MARK{1-10}:FUNC:TARG
Transitional direction	:CALC{1-4}:MARK{1-10}:FUNC:TTR

Performing Marker Search

To perform Marker Search, use the following command:

:CALC{1-4}:MARK{1-10}:FUNC:EXEC

To turn ON or OFF the Search Tracking feature, which performs Marker Search every time the trace is updated, use the following command:

:CALC{1-4}:MARK{1-10}:FUNC:TRAC

Retrieving Search Results

Performing Marker Search moves the marker to the points that matches the search criteria, so you can obtain the search results by retrieving the marker value. For information on how to retrieve marker values, refer to Setting (changing) and retrieving stimulus value at marker positions and Retrieving measurement results at marker positions.

Analysis Commands

You can use the analysis commands to perform search and analysis.

Setting the Search (Analysis) Range

You can use either the entire sweep range or a user-defined range as the search (analysis) range by using the following command:

:CALC{1-4}:FUNC:DOM

When you opt to use a user-defined range, use the following commands to set the range:

Start value (lower limit value)	:CALC{1-4}:FUNC:DOM:STAR
Stop value (upper limit value)	:CALC{1-4}:FUNC:DOM:STOP

You can also select whether to specify the marker search (analysis) range independently for each trace by using the following command:

:CALC{1-4}:FUNC:DOM:COUP

Selecting the Search (Analysis) Type

The analysis commands allow you to choose from the following five search types:

- Maximum value
- Minimum value
- Maximum peak (for a positive peak), minimum peak (for a negative peak)
- All peaks
- All targets

In addition, you can choose from the following three analysis types:

- Difference between the maximum and minimum values
- Standard deviation
- Average

To select the search (analysis) type, use the following command:

:CALC{1-4}:FUNC:TYPE

Defining a Peak

You can define a peak by specifying the lower limit for the peak excursion value and polarity (positive or negative peak).

To define a peak, use the following commands:

Lower limit for the peak excursion value	:CALC{1-4}:FUNC:PEXC
Polarity	:CALC{1-4}:FUNC:PPOL

Defining a Target

You can define a target by specifying the target value (response value) and transitional direction (positive or negative value change).

To define a target, use the following commands:

Target value	:CALC{1-4}:FUNC:TARG
Transitional direction	:CALC{1-4}:FUNC:TTR

Performing Search (Analysis)

To perform search (analysis), use the following command:

:CALC{1-4}:FUNC:EXEC

Retrieving Search (Analysis) Results

To retrieve search (analysis) results, use the following command:

:CALC{1-4}:FUNC:DATA?

The number of data items contained in search (analysis) results differ depending on the search (analysis) type and the number of points found by the search operation. To retrieve the number of data items, use the following command:

:CALC{1-4}:FUNC:POIN?

Sample Program

See Peak Search.

Notch Search

- Overview
- Setting the notch definition value
- Displaying the notch search result
- Reading out the notch search result

Other topics about Analyzing Data

Overview

The notch search function is used to obtain the bandwidth, center frequency, cutoff points (high-frequency side and low-frequency side), Q and insertion loss of a trace based on the active marker position. The notch search function starts at the left side of the active marker position, and ends when points that meet the condition are found.

- Bandwidth (BW = high – low)
- Center frequency (cent = (high +low)/2)
- Q value (Q = cent/BW)
- Loss (response value at marker position)

Where, high is the right-hand cutoff point frequency, and low is the left-hand cutoff point frequency.

NOTE For more information on notch search, see Determining the bandwidth of a trace (Notch Search)

Setting the notch definition value

The notch search function finds a point whose response value is different, by the amount defined as the notch definition value, than the response value at the marker position, and identifies that point as the cutoff point.

To set the notch definition value, use the following command:

:CALC{1-4}:MARK{1-10}:NOTC:THR

Displaying the notch search result

The following command is used to control whether to display the notch search result on the LCD:

:CALC{1-4}:MARK:NOTC

Reading out the notch search result

Once the marker is moved to an appropriate position using the marker search function or some other function, it is able to retrieve the notch search result using the following command:

:CALC{1-4}:MARK{1-10}:NOTC:DATA

NOTE It is able to retrieve the notch search result regardless of whether the marker display and the notch search result display is ON/OFF.

Analysis in Time Domain (time domain function)

- Overview
- Transforming Measurement Data to Time Domain
- Sample Program

Other topics about Analyzing Data

Overview

The time domain function provides the following functions:

- Transforming measurement data to data in the time domain (Transformation Function)
- Deleting unnecessary measurement data in the time domain (Gating Function)

Transforming Measurement Data to Time Domain

By using the Transformation Function, you can convert the results measured in the frequency domain to data in the time domain and analyze it.

ON/OFF

To turn ON or OFF the transformation function, use the following command:

:CALC{1-4}:TRAN:TIME:STAT

Selecting Transformation Type

To select the transformation type (band pass/low pass), use the following command:

:CALC{1-4}:TRAN:TIME

To select the stimulus type (impulse/step) when the transformation type is low pass, use the following command:

:CALC{1-4}:TRAN:TIME:STIM

When the transformation type is low pass, you need to execute the following command because each measurement point must be a multiple of the start frequency.

:CALC{1-4}:TRAN:TIME:LPFR

Setting Window Shape

To set the window shape, use one of the following items.

Item	Command

β	:CALC{1-4}:TRAN:TIME:KBES
Impulse width	:CALC{1-4}:TRAN:TIME:IMP :WIDT
Rise time of step signal	:CALC{1-4}:TRAN:TIME:STEP :RTIM

The above three items are dependent on each other. When the value of one of them is changed, the values of the other two are automatically changed to the corresponding values.

Unlike the manual operation, you cannot set the window shape by selecting the window type (maximum/normal/minimum). However, you can set the same shape as each window type by setting β as follows:

	Maximum	Normal	Minimum
Value of β	13	6	0

Setting Display Range

To set the display range after time domain transformation, use the following commands:

Description	Command
Start value	:CALC{1-4}:TRAN:TIME:STAR
Stop value	:CALC{1-4}:TRAN:TIME:STOP
Center value	:CALC{1-4}:TRAN:TIME:CENT
Span value	:CALC{1-4}:TRAN:TIME:SPAN

Deleting unnecessary measurement data in the time domain

You can use the Gating Function to delete unnecessary time domain data.

ON/OFF

To turn ON or OFF the gating function, use the following command:

:CALC{1-4}:FILT:TIME:STAT

Selecting Gate Type

To select the gate type, use the following command:

:CALC{1-4}:FILT:TIME

Setting Gate Shape

To select the gate shape, use the following command:
:CALC{1-4}:FILT:TIME:SHAP

Setting Gate Range

To set the gate range, use the following commands:

Description	Command
Start value	:CALC{1-4}:FILT:TIME:STAR
Stop value	:CALC{1-4}:FILT:TIME:STOP
Center value	:CALC{1-4}:FILT:TIME:CENT
Span value	:CALC{1-4}:FILT:TIME:SPAN

Sample Program

See the Time Domain.

Obtaining Limit / Bandwidth Limit / Ripple Limit Test Results

- Overview
- Test Results at each Measurement Point
- Test Results for each Trace
- Test Results for each Channel
- Overall Test Results

Other topics about Limit Test

Overview

You can obtain test results by issuing a result retrieval command or through the status register.

Test Results at each Measurement Point

Using commands that retrieve test results

You can obtain the test results at each measurement point by retrieving the stimulus value at failed measurement points. To retrieve failed measurement points, use the following command:

Stimulus value	:CALC{1-4}:LIM:REP?
Number of measurement points	:CALC{1-4}:LIM:REP:POIN?

Using the status register

You cannot use the status register to obtain the test results at each measurement point.

Test Results for each Trace

Using commands that retrieve test results

You can retrieve the test result for each trace (i.e., the trace-wide result that combines the results for all measurement points in a particular trace) by issuing the following command:

:CALC{1-4}:LIM:FAIL?

Using the status register

The condition register and event register under the questionable limit channel {1-4} status register provide 4 bits that correspond to traces 1 to 4 and contain the test results (0: Pass, 1: Fail) for the respective traces; for example, you can obtain the test result for trace 1 from bit 1 and that for trace 4 from bit 4.

Every bit of the condition register is set to 0 when a measurement cycle is started. Upon completion of measurement, those bits that correspond to failed traces are set to 1.

If the corresponding bit of the positive transition filter is set to 1 (preset value), each bit of the event register is set to 1 when the corresponding bit of the condition register changes from 0 to 1 (indicating that the corresponding trace failed the test).

To retrieve the registers, use the following commands:

Questionable limit channel {1-4} status register	
Condition register	:STAT:QUES:LIM:CHAN{1-4}:COND?
Event register	:STAT:QUES:LIM:CHAN{1-4}?
Questionable limit channel {1-4} extra status register	
Condition register	:STAT:QUES:LIM:CHAN{1-4}:ECH:COND?
Event register	:STAT:QUES:LIM:CHAN{1-4}:ECH?

Test Results for each Channel

Using commands that retrieve test results

No command is available that allows you to directly retrieve the test result for each channel (i.e., the channel-wide result that combines the results for all traces in a particular channel).

Using the status register

The questionable limit status event register provides 4 bits that correspond to channels 1 to 4 and contain the test results (0: Pass, 1: Fail) for the respective channels; for example, you can obtain the test result for channel 1 from bit 1 and that for channel 4 from bit 4.

Every bit of the condition register is set to 0 after the event registers are cleared by the *CLS. Upon completion of measurement, if the channel-wide test result that combines the results for all traces in a channel is fails, the corresponding bit of the condition register is set to 1.

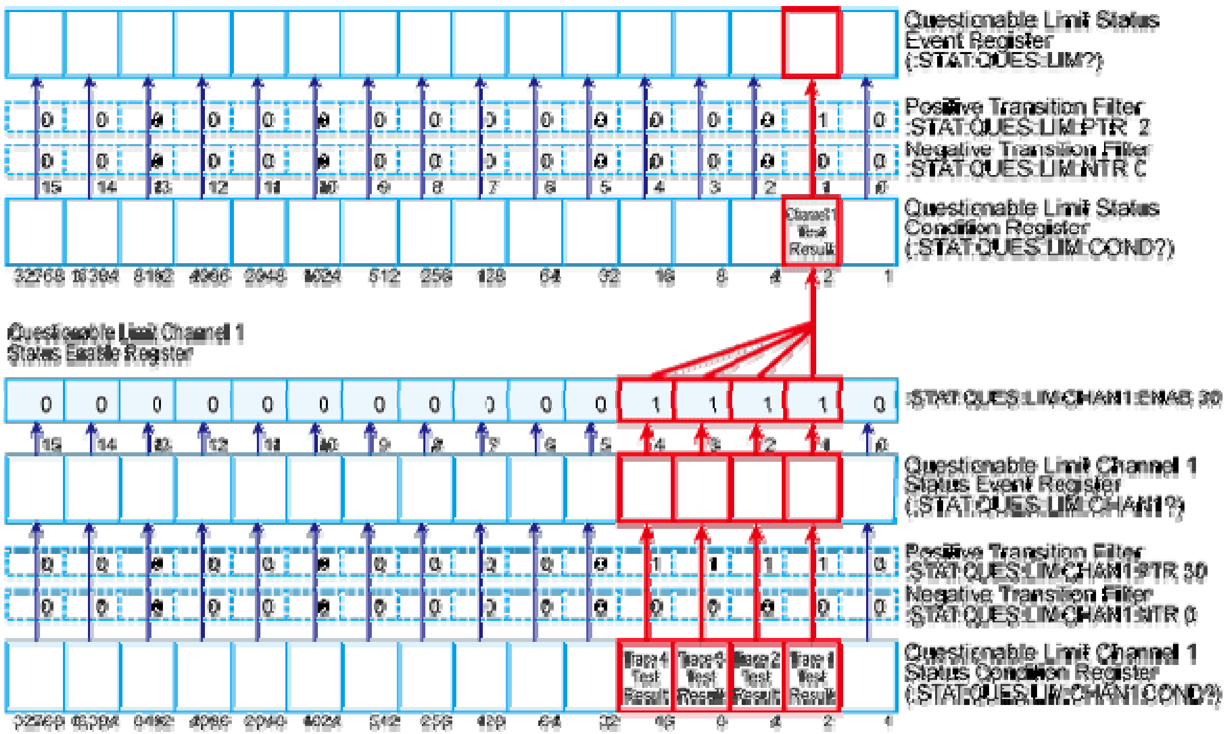
If the corresponding bit of the positive transition filter is set to 1 (preset value), every bit of the event register is set to 1 when the corresponding bit of the condition register changes from 0 to 1.

To retrieve the registers, use the following commands:

Questionable limit status register

Condition register	:STAT:QUES:LIM:COND?
Event register	:STAT:QUES:LIM?
Questionable limit extra status register	
Condition register	:STAT:QUES:LIM:ELIM:COND?
Event register	:STAT:QUES:LIM:ELIM?

Obtaining test results for a channel (channel 1 in this example) using the status register



e5061b021

Overall Test Results

Using commands that retrieve test results

No command is available that allows you to directly retrieve the overall test results that combine the test results for all channels.

Using the status register

The condition register and event register under the questionable status event register provides bit 10 each, from which you can obtain the overall test result (0: Pass, 1: Fail).

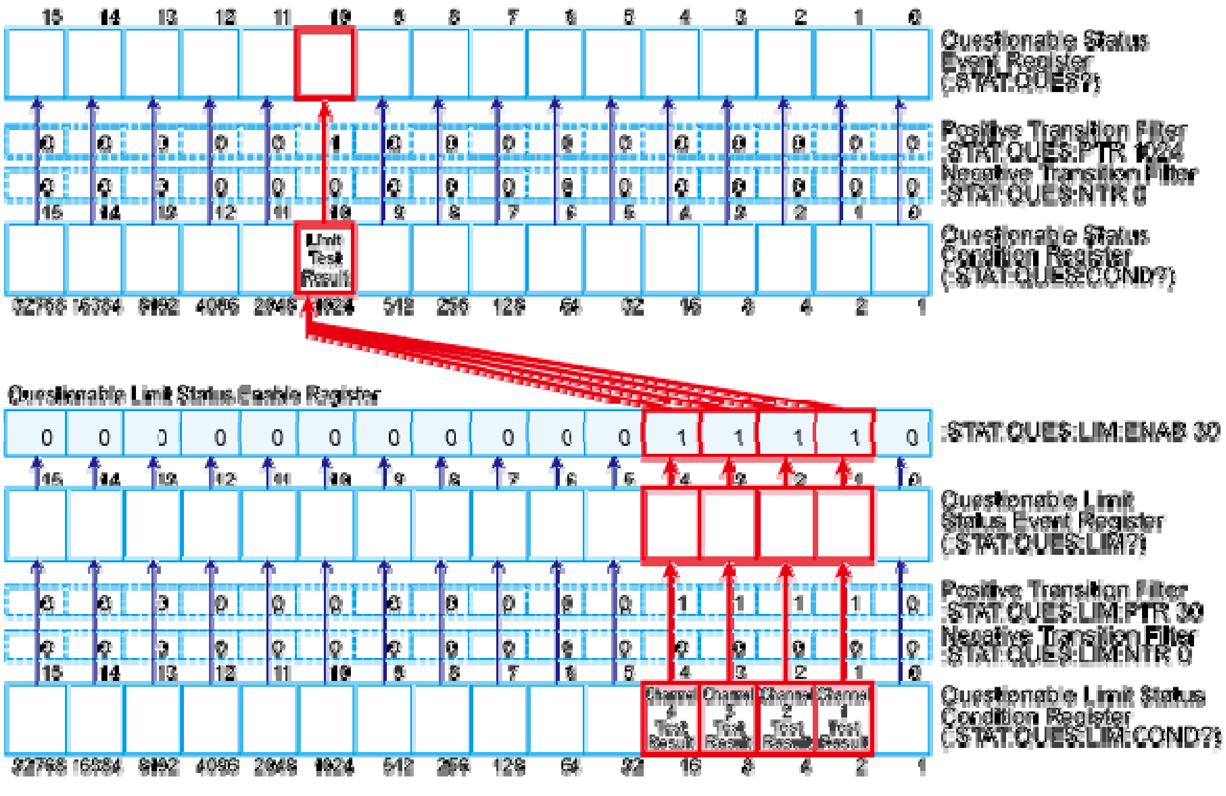
The condition register's bit 10 is set to 0 after the event registers are cleared by the *CLS. Upon completion of measurement, this bit is set to 1 if the overall test result that combines the results for all channels fails.

If the positive transition filter's bit 10 is set to 1 (preset value), the event register's bit 10 is set to 1 when the condition register's bit 10 changes from 0 to 1.

To retrieve the condition register and event register under the questionable status event register, use the following commands:

Condition register	:STAT:QUES:COND?
Event register	:STAT:QUES?

Obtaining overall test results using the status register



e5061b022

Saving and Recalling

Saving and Recalling

- Saving and Recalling File
- Managing Files

Saving and Recalling File

Other topics about Saving and Recalling

Specifying File

When running a command for saving, recalling, and managing files, use a filename with extension to specify a particular file. Specify "D:" at the beginning of the file name, when specifying a file at the user area of hard disk. Also, when specifying a file name with directory, use "/" (slash) or "\" (backslash) as a delimiter.

Saving and Recalling Instrument Status

You can save the instrument state using one of the following 2 methods:

- Saving the entire instrument state into a file
- Saving the state for each channel into registers A to D (volatile memory)

Selecting content to be saved

When saving the instrument status into a file or register, the content to be saved can be selected among the following 4 options:

- Instrument status only
- Instrument status and calibration coefficient array.
- Instrument status, corrected data/memory array (measurement data)
- Instrument status, calibration coefficient array, and corrected data/memory array (measurement data)

To select a content to be saved, use the following command:

:MMEM:STOR:STYP

Selecting Content to be Saved

To select whether to save the setting of all channels/traces or that of the displayed channels/traces, use the following command:

:MMEM:STOR:SALL

Saving and recalling entire instrument status

To save the entire instrument status into a file, use the following command:

:MMEM:STOR

Recalling a file saved with the above command can reproduce the status when it was saved. To recall the settings from a file, use the following command:

:MMEM:LOAD

Auto Recall

The file saved with the name autorec.sta will be automatically recalled when the E5061B is powered ON.

Saving the state for each channel into a register

For the active channel, when you want to save the instrument state specific to that channel into only one of the registers A to D, use the following command:

:MMEM:STOR:CHAN

Recalling an instrument state saved in a register can reproduce it as the state of the active channel. To recall a register, use the following command:

:MMEM:LOAD:CHAN

> **NOTE** It is possible to recall a file from a different channel where it was saved.

The contents in the registers are lost when you turn OFF the power. You can delete (clear) the contents of all registers using the following command.

:MMEM:STOR:CHAN:CLE

Saving Measurement Data

Measurement data (in a formatted data array) can be saved to a file in CSV (Comma Separated Value) format.

To save measurement data in a file, use the following command:

:MMEM:STOR:FDAT

Executing the above command saves the measurement data of the active trace. Note that the data saved using the above command cannot be recalled from the E5061B.

Saving Measurement Data in Touchstone Format

Measurement data for the active channel can be saved to a file in touchstone format.

To determine the file type in touchstone file format and specify a port, use one of the following commands according to the number of ports used:

- :MMEM:STOR:SNP:TYPE:S1P
- :MMEM:STOR:SNP:TYPE:S2P

To set the data type of the files saved in touchstone format, use the following command:

:MMEM:STOR:SNP:FORM

To save measurement data in touchstone format, use the following command:

:MMEM:STOR:SNP

NOTE Only trace data of frequency sweep can be saved in touchstone format files. Trace data of power sweep measurement cannot be saved.

Saving Images

Images displayed on the LCD screen can be saved to a file in the bitmap (.bmp) or portable network graphics (.png) format.

To save the screen image to a file, use the following command:

:MMEM:STOR:IMAG

Executing the above command saves the screen image when the command is invoked.

NOTE This gives different screen image results from those obtained by pressing the **Capture** key on the front panel.

Saving and Recalling Segment Sweep Table

Segment sweep table can be saved in the file with CSV (Comma Separated Value) format.

To save segment sweep table on a file, use the following command:

:MMEM:STOR:SEGM

Executing the above command saves the segment sweep table for the active channel.

Recalling the file saved using the above command can reproduce the segment sweep table on the active channel.

To recall the settings from a file, use the following command:

:MMEM:LOAD:SEGM

NOTE It is possible to recall a file from a different channel where it was saved. Note that recalling operation is not guaranteed for the file that might have been modified with editor.

Saving and Recalling Limit Table

Limit table can be saved in the file with CSV (Comma Separated Value) format. To save limit table on a file, use the following command:

:MMEM:STOR:LIM

Executing the above command saves the limit table for the active trace of the active channel.

Recalling the file saved using the above command can reproduce the limit table on the active trace of the active channel. To recall the settings from a file, use the following command:

:MMEM:LOAD:LIM

NOTE It is possible to recall a file from a different channel or trace where it was saved. Note that recalling operation is not guaranteed for the file that might have been modified with editor.

Saving/Loading (Importing) a VBA Program

Saving

Only the VBA project file can be saved using command.

To save the VBA project that is opened on the VBA editor on the file, use the following command.

:MMEM:STOR:PROG

Loading (importing)

To load the VBA project to the VBA editor, or to import the module/form file, use the following command.

:MMEM:LOAD:PROG

Executing the above command loads/imports the file according to its extension as follows:

Extension	File type
vba	VBA Project
bas	Standard module

frm	User Forms
cls	Class Modules

Sample program

See the Saving Files.

Managing Files

- Various Commands
- Sample Program

Other topics about Saving and Recalling

Various Commands

Creating directory (folder)

To create a directory (folder), use the following command:

:MMEM:MDIR

Deleting file (directory)

To delete a file or a directory, use the following command:

:MMEM:DEL

Copying file

To copy a file, use the following command:

:MMEM:COPY

Transferring files

File transfer from the external controller to the E5061B is possible by reading data from a file on the controller and then writing them to the file on the E5061B.

:MMEM:TRAN

Also, file transfer from the E5061B to the external controller can be possible by reading data from a file on the E5061B using the commands as query and then writing them to the file on the controller.

Retrieving data from storage

To retrieve information for the storage that is built in the E5061B (usage, property of file located in a specified directory), use the following command;

:MMEM:CAT?

Sample Program

See the Transferring Files.

Communication with External Devices using I/O Port

Communication with External Devices (24 Bit I/O Port)

- 24 Bit (Handler) I/O Port Overview
- I/O Signal Pin Layout and Description
- Inputting/Outputting Data
- Preset states at power-on
- Timing Chart
- Electrical Characteristics

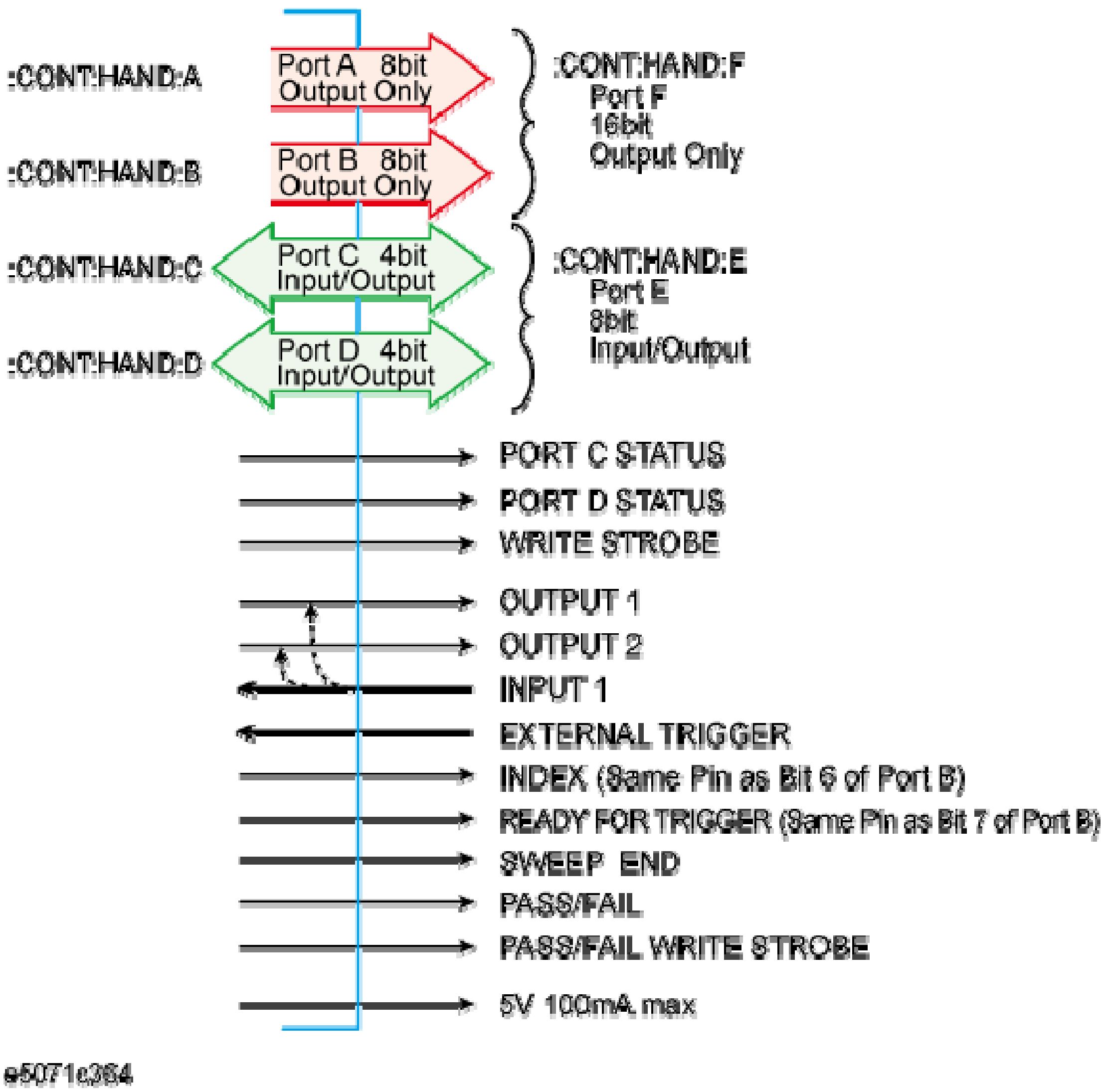
24 Bit (Handler) I/O Port Overview

The E5061B 24 Bit (handler) I/O port provides four independent parallel ports for data I/O associated with several control signal lines and the power line. All signals operate in TTL logic.

The data I/O ports are configured with 2 pairs of 8 bit output port and 2 pairs of 4 bit bi-directional port. Also those ports can cooperate to provide a maximum 16-bit-width output port or a maximum 8-bit-width input port.

The I/O signals operate on the negative logic basis, which can be altered. The control signal lines consist of various control output data, including completion of measurement or control signal for handshaking.

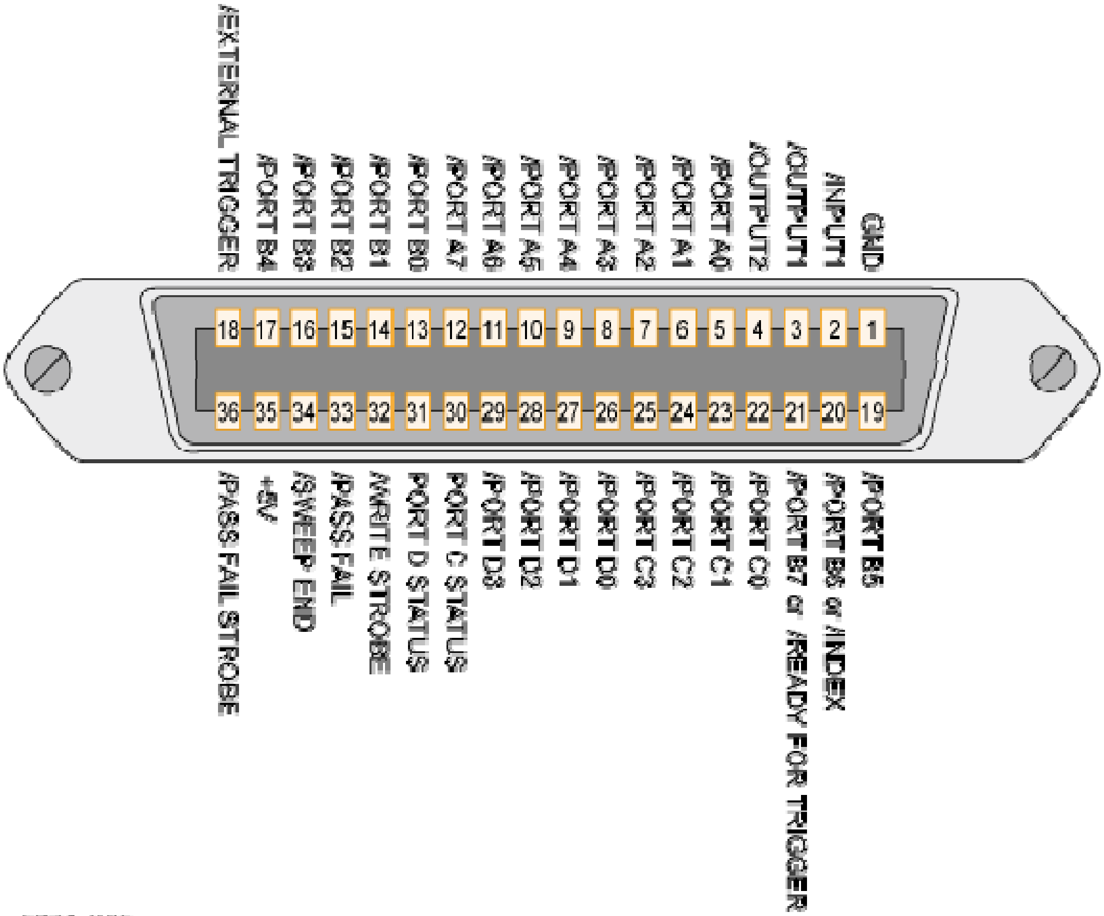
I/O ports and control signal lines



Other topics about Communication with External Devices

I/O Signal Pin Layout and Description

The layout of the I/O signal pins on the handler interface connector and its description are shown below.



A slash (/) symbol preceding signal names means that they are negative logic (active low).

Pin number	Signal name	Input/Output	Description
1	GND	N/A	Ground.
2	/INPUT1	Input	When this port receives a negative pulse, /OUTPUT1 and /OUTPUT2 are changed to the Low level.
3	/OUTPUT1	Output	Changes to the Low level when /INPUT1 receives a

			negative pulse. A command is available for altering the Low/High level logic.
4	/OUTPUT2	Output	Changes to the Low level when /INPUT1 receives a negative pulse. A command is available for altering the Low/High level logic.
5	/PORT A0	Output	Bit 0 of port A (8 bit parallel output port)
6	/PORT A1	Output	Bit 1 of port A.
7	/PORT A2	Output	Bit 2 of port A.
8	/PORT A3	Output	Bit 3 of port A.
9	/PORT A4	Output	Bit 4 of port A.
10	/PORT A5	Output	Bit 5 of port A.
11	/PORT A6	Output	Bit 6 of port A.
12	/PORT A7	Output	Bit 7 of port A.
13	/PORT B0	Output	Bit 0 of port B (8 bit parallel output port)
14	/PORT B1	Output	Bit 1 of port B.
15	/PORT B2	Output	Bit 2 of port B.
16	/PORT B3	Output	Bit 3 of port B.
17	/PORT B4	Output	Bit 4 of port B.
18	/EXTERNAL TRIGGER	Input	An external trigger signal. When the trigger source is set to the "External," this port generates a trigger in respond to the trailing edge of a negative pulse.

19	/PORT B5	Output	Bit 5 of port B.
20	/PORT B6	Output	Bit 6 of port B.
	/INDEX		Indicates that analog measurement is complete. The /INDEX signal changes to the Low level when analog measurement (all sweeps of all channels) is complete. When the handler receives the signal, it assumes that it is ready to connect the next DUT. However, no measurement data is available until data calculation is completed. When the point trigger function is ON, it goes to the High level before starting measurement of the first measurement point and returns to the Low level after completing measurement of all measurement points.
21	/PORT B7	Output	Bit 7 of port B.
	/READY FOR TRIGGER		Indicates that the instrument is ready for triggering. This signal is changed to the Low level when the instrument is ready to receive a trigger signal. The /READY FOR TRIGGER signal goes to the Low level when the instrument is ready to accept the trigger signal for the first

			point and goes to the High level when the trigger signal for the first point is received. **When the point trigger is OFF:** When measurement of all measurement points is completed and the instrument is ready to receive the trigger signal for the first point of the next sweep, this signal goes to the Low level again. **When the point trigger is ON:** When each measurement point is completed and the instrument is ready to receive the trigger signal for the next measurement point, this signal goes to the Low level again.
22	/PORT C0	Input/Output	Bit 0 of port C (4 bit parallel I/O port)
23	/PORT C1	Input/Output	Bit 1 of port C.
24	/PORT C2	Input/Output	Bit 2 of port C.
25	/PORT C3	Input/Output	Bit 3 of port C.
26	/PORT D0	Input/Output	Bit 0 of port D (4 bit parallel I/O port)
27	/PORT D1	Input/Output	Bit 1 of port D.
28	/PORT D2	Input/Output	Bit 2 of port D.
29	/PORT D3	Input/Output	Bit 3 of port D.
30	PORT C STATUS	Output	Port C status signal. This signal is changed to the High level when the port C

			is configured to output port. It is changed to the Low level when the port is configured to input port.
31	PORT D STATUS	Output	Port D status signal. This signal is changed to the High level when the port D is configured to output port. It is changed to the Low level when the port is configured to input port.
32	/WRITE STROBE	Output	A output port write strobe signal. When data is present (that is, output level changes) on any of the output ports, this signal provides a negative pulse.
33	/PASS FAIL	Output	Each limit test's results signal. This signal changes to the High level when limit test, bandwidth test, or ripple test results return FAIL. It changes to the Low level when all limit test results return PASS.
34	/SWEEP END	Output	A sweep completion signal. When measurement (all sweeps of all channels) and data calculation are completed, this signal provides a negative pulse.
35	+5V	Output	Provides +5V DC power supply for external instruments.
36	/PASS FAIL	Output	Each limit test's results write a strobe signal.

	STROBE		When limit test result is present on /PASS FAIL, this signal provides a negative pulse.

Other topics about Communication with External Devices

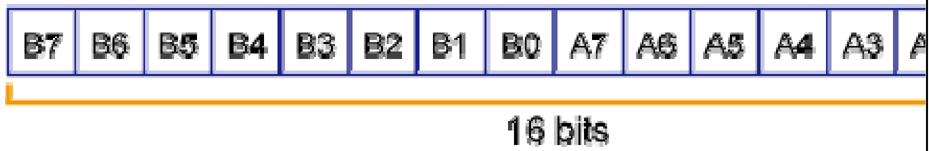
Inputting/Outputting Data

Other topics about Communication with External Devices

Overview

The E5061B 24 Bit (Handler) I/O port provides the ports for data I/O shown below.

Port Name	Usage	Data Structure
Port A	Output	A7 A6 A5 A4 A3 A2 A1 A0 8 bits
Port B	Output	B7 B6 B5 B4 B3 B2 B1 B0 8 bits
Port C	Input/Output	C3 C2 C1 C0 4 bits
Port D	Input/Output	D3 D2 D1 D0 4 bits

Port E	Input/Output	
Port F	Output	

Specifying Signal Direction of Port

Signal direction (input/output) can be changed for the ports C, D, and E as shown in I/O ports and control signal lines. Thus, before the ports are used, the directions should be determined according to their usage.

To specify the signal direction for the ports C and D, use the following command. Direction for the port E depends on the setting for the ports C and D.

Port Name	Command
Port C	:CONT:HAND:C:MODE
Port D	:CONT:HAND:D:MODE

Reading Data Input into Port

When the ports C, D, or E are configured to input ports, binary data represented by High(0)/Low(1) of each bit of the port is read as decimal data.

To retrieve the data, use the following command as query:

Port Name	Command
Port C	:CONT:HAND:C
Port D	:CONT:HAND:D
Port E	:CONT:HAND:E

Data Output to Port

To ports A through F (the ports C, D, and E should be configured to output ports), binary data (decimal data when output data is specified with a command) represented by High(0)/Low(1) of each bit of the port can be output.

To output data, use the following command:

Port Name	Command
Port A	:CONT:HAND:A
Port B	:CONT:HAND:B
Port C	:CONT:HAND:C
Port D	:CONT:HAND:D
Port E	:CONT:HAND:E
Port F	:CONT:HAND:F

NOTE The bit 6 of the data output by :CONT:HAND:B (the bit 14 of the data output by :CONT:HAND:F) is ignored when outputting the /INDEX signal is turned ON.

NOTE The bit 7 of the data output by :CONT:HAND:B (the bit 15 of the data output by :CONT:HAND:F command) is ignored when outputting the /READY FOR TRIGGER signal is turned ON.

Sample Program

See Handler Interface.

Preset states at power-on

The 24 bit (Handler) I/O port is set at power-on as follows (not affected at reset)

Description	Status
Port A	High (All Bits)
Port B	High (All Bits)
Port C	Input
Port D	Input
Port C STATUS	Low
Port D STATUS	Low
/OUTPUT1	High
/OUTPUT2	High
/SWEEP END	High
/PASS FAIL	High

Other topics about Communication with External Devices

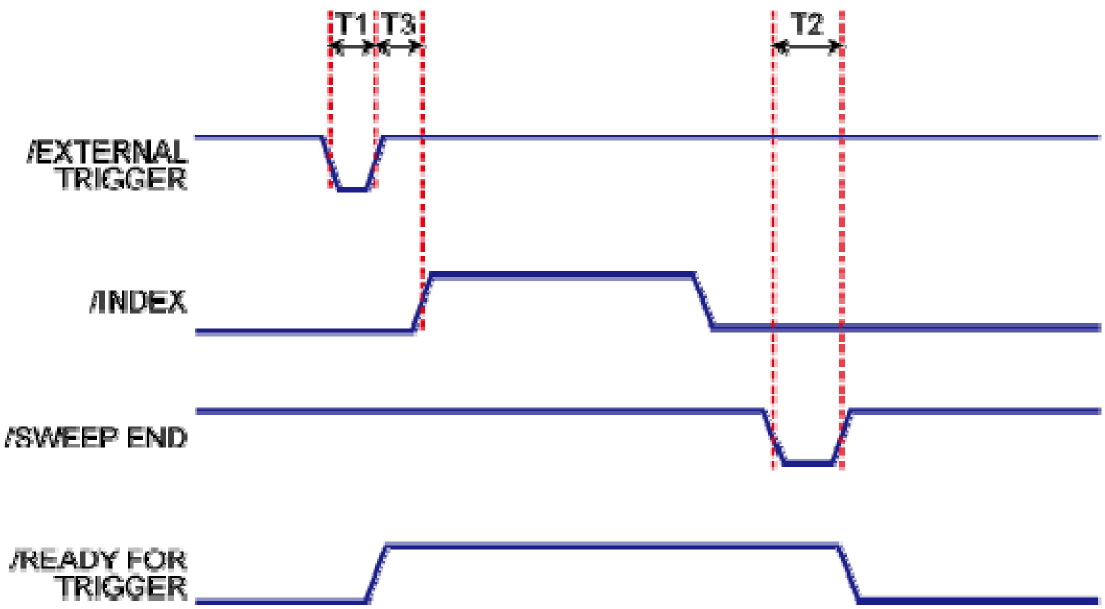
Timing Chart

Other topics about Communication with External Devices

Overview

This section shows the typical timing chart of I/O port Signal.

Timing chart of I/O Port Signal (Point Trigger: off)



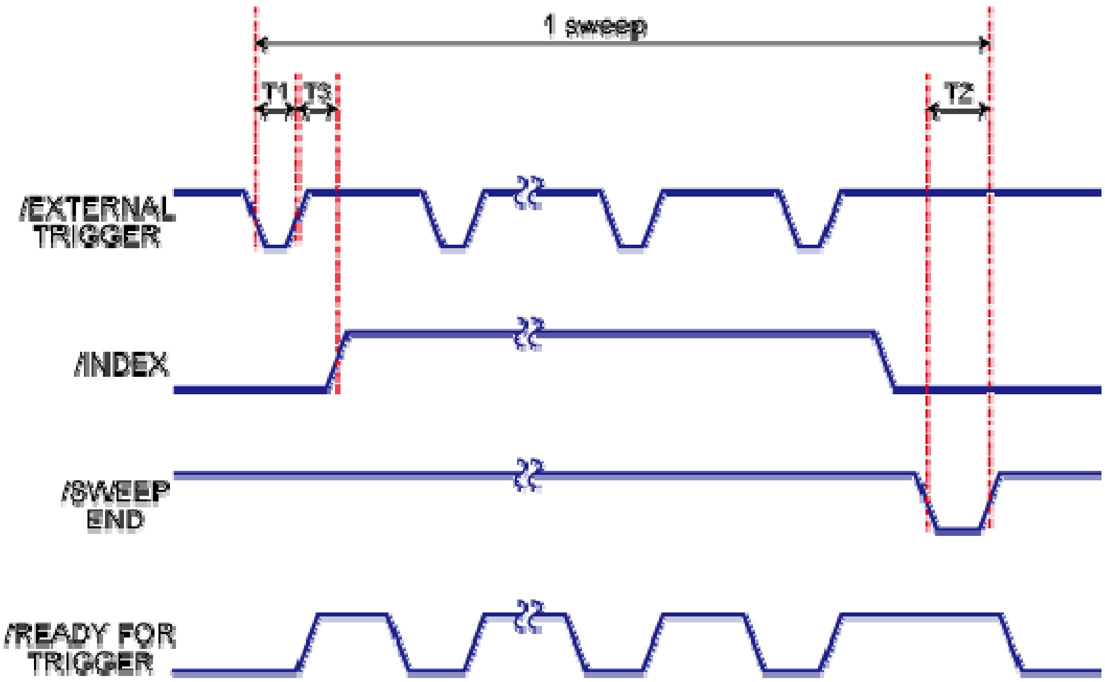
e5061b040

		Minimum value	Typical Value	Maximum value
T1	Pulse width of /EXTERNAL TRIGGER or External Trigger Input Port	1 μs	-	-
T2	Pulse width of /SWEEP END	10 μs	12 μs	-

T3	Time set as the trigger delay time	-	(See below note)	-

NOTE The Trigger Delay Time (T3) is not constant, because it is time that the user sets.

Timing Chart of I/O Port Signal (Point Trigger: on)



e5061b041

When the point trigger function is ON, the /EXTERNAL TRIGGER signal must be inputted for each measurement point during a single sweep. The /INDEX signal goes to the High level before starting measurement of the first measurement point and returns to the Low level after the completing measurement of all measurement points.

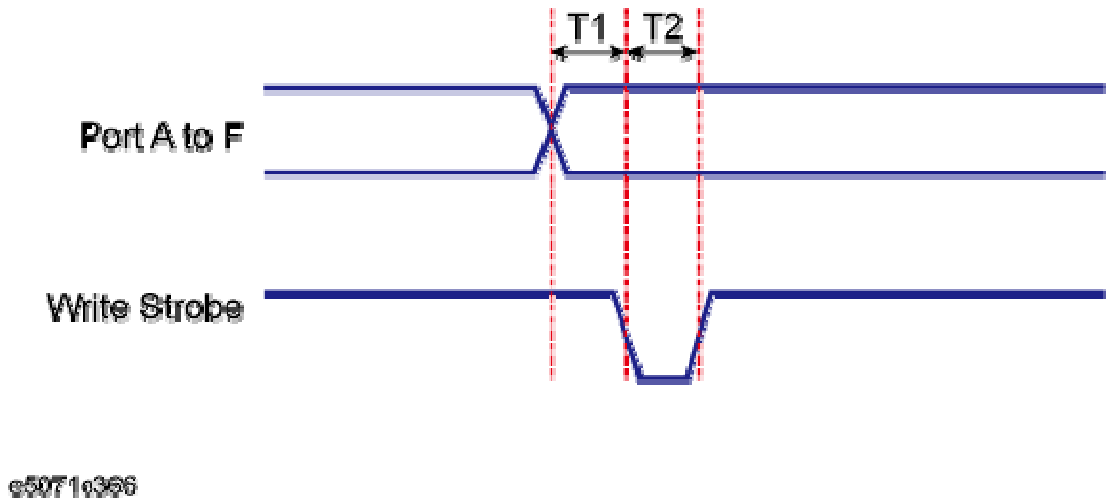
The /READY FOR TRIGGER signal goes to the Low level when the instrument is ready to accept the trigger signal for the first point and then goes to the High level when the trigger signal for the first point is received.

When measurement of all measurement points is completed and the instrument is ready to receive the trigger signal for the first point of the next sweep, this signal goes to the Low level again.

When the point trigger function is ON, the /READY FOR TRIGGER signal goes High each time a trigger signal is received and goes Low when measurement of each measurement point is completed and the instrument is ready to accept a trigger for the next measurement point.

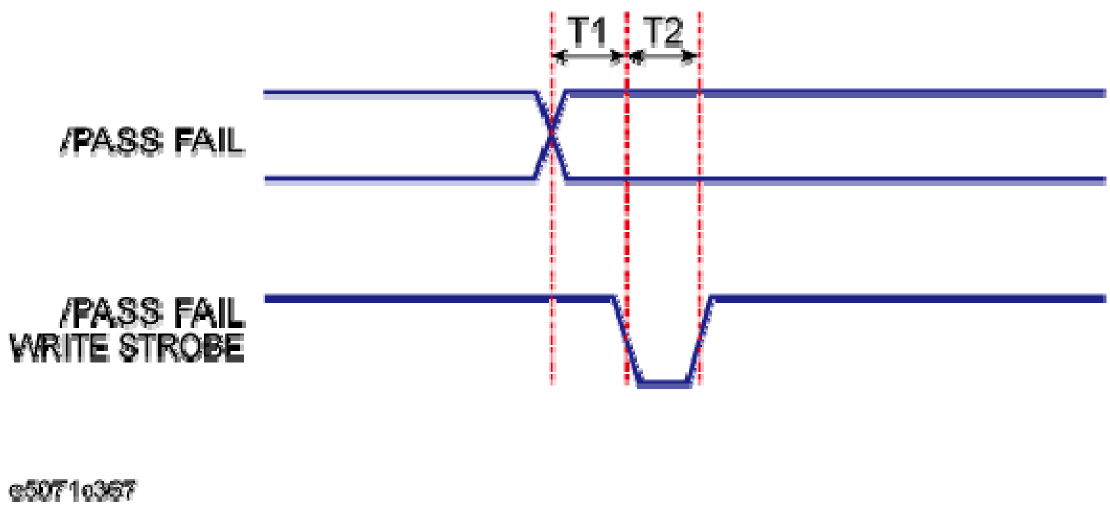
The times of T1 and T2 are the same as those when the point trigger function is OFF. For more information, see Timing chart of I/O Port Signal (Point trigger function:OFF).

Timing Chart of Data Output and Write Strobe Signal



T1	Response time of write strobe signal	1 μs
T2	Pulse width of write strobe signal	1 μs

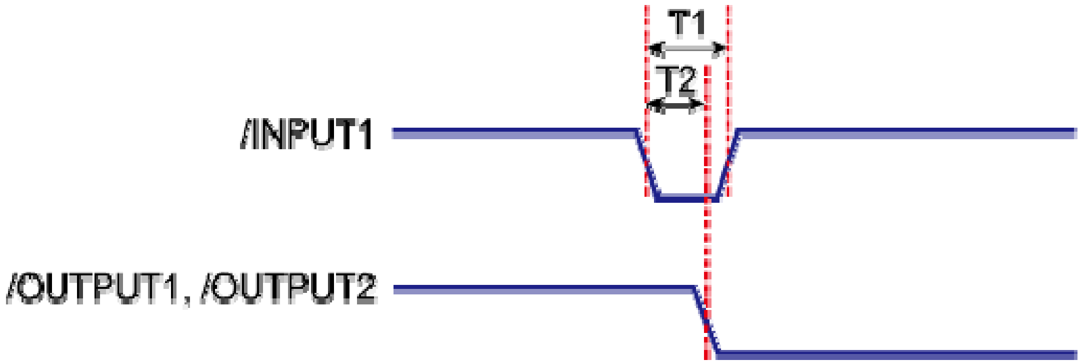
Timing Chart of Limit Test Result Output and Write Strobe Signal



T1	Response time of /PASS FAIL write strobe	1 μs
T2	Pulse width of /PASS FAIL write strobe	1 μs

NOTE When the average trigger function is activated, the fail and write strobe signals are output at the time that the average test result shows "failed" on a certain channel.

Timing Chart of /INPUT1 and /OUTPUT1, /OUTPUT2



e50f1c369

		Minimum value	Maximum value
T1	Pulse width of /INPUT1	1 μs	-
T2	Response time of /OUTPUT1, /OUTPUT2	0.2 μs	0.4 μs

Electrical Characteristics

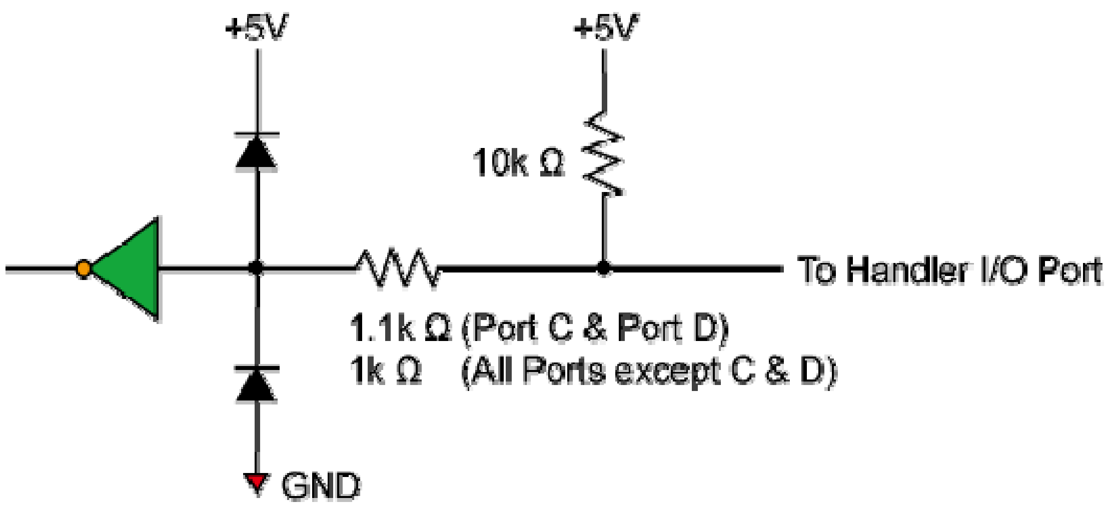
- Input Signal
- Output Signal
- Power Supply (+5 V)

Other topics about Communication with External Devices

Input Signal

All input signals are TTL compatible.

Maximum rate input voltage		-0.5 V to 5.5 V
Input voltage	**High level**	2.0 V to 5.0 V
	Low level	0 V to 0.5 V



e5071c370

Output Signal

All output signals are TTL compatible.

Maximum rate output current		-10 mA to 10 mA
Output current	**High level**	-5 mA

	Low level	3 mA
Output voltage	**High level**	2.0 V to 3.3 V (when output current is from -5 mA to 0 mA) 3.20 V (when output current is -1 mA) 2.75 V (when output current is -5 mA)
	Low level	0 V to 0.8 V (when output current is from 0 mA to 3 mA) 0.25 V (when output current is 1 mA) 0.55 V (when output current is 3 mA)

Power Supply (+5 V)

The following table shows electrical characteristics of +5 V power supply for external instruments.

Output voltage	4.5 V to 5.5 V
Maximum output current	100 mA

Status Reporting System

Status Reporting System

- General Status Register Model
- Using the Status Reporting System
- Status Register Structure

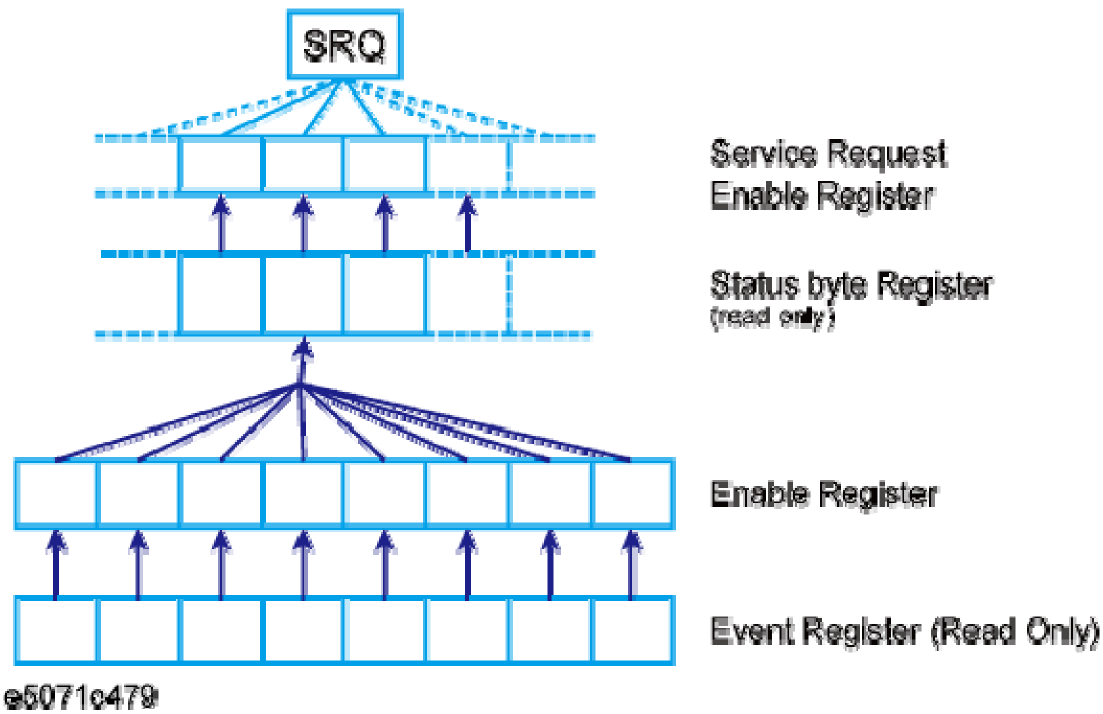
General Status Register Model

Other topics about Status Reporting System

Overview

The E5061B has a status reporting system to report the condition of the instrument.

General status register model



The status reporting system has a hierarchical structure as shown in the figure above. 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 General status register model) 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 SRQ, you can notify the controller that the E5061B is requesting service. In other words, interruption by SRQ can be programmed. For more information on using SRQ, see Using the status register or Using the status reporting system.

Event Register

Reflects the corresponding condition of the E5061B (e.g., occurrence of an event) as a bit status. These bits continuously monitor changes in the E5061B'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.

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 or serial poll (SPOLL statement in HTBasic) from the controller.

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 clears the RQS bit in the status byte register.

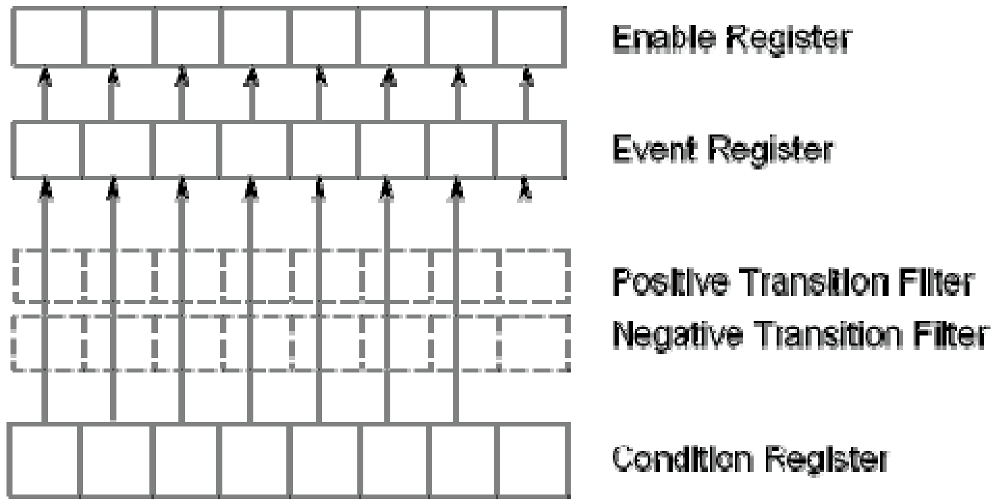
Also, setting the service request enable register using the *SRE command can generate a service request synchronously with the status byte register.

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, it changes from 1 to 0.

Transition filter and condition register



4294spe022

In the E5061B, the following registers provide a condition register and transition filter:

- Operation status register
- Questionable status register
- Questionable limit status register
- Questionable limit channel status register
- Questionable bandwidth limit status register
- Questionable bandwidth limit channel {1-4} status register
- Questionable ripple limit status register
- Questionable ripple limit channel {1-4} status register

Using the Status Reporting System

You can manage the status report system using the following commands in any combination:

- *CLS
- *SRE
- *STB?
- *ESE
- *ESR?
- :STAT:PRES
- :STAT:OPER:ENAB
- :STAT:OPER:COND?
- :STAT:OPER?
- :STAT:OPER:PTR
- :STAT:OPER:NTR
- :STAT:QUES:ENAB
- :STAT:QUES:COND?
- :STAT:QUES?
- :STAT:QUES:PTR
- :STAT:QUES:NTR
- :STAT:QUES:LIM:ENAB
- :STAT:QUES:LIM:COND?
- :STAT:QUES:LIM?
- :STAT:QUES:LIM:PTR
- :STAT:QUES:LIM:NTR
- :STAT:QUES:LIM:CHAN{1-4}:ENAB
- :STAT:QUES:LIM:CHAN{1-4}:COND?
- :STAT:QUES:LIM:CHAN{1-4}?
- :STAT:QUES:LIM:CHAN{1-4}:PTR
- :STAT:QUES:LIM:CHAN{1-4}:NTR
- :STAT:QUES:BLIM:ENAB
- :STAT:QUES:BLIM:COND?
- :STAT:QUES:BLIM?
- :STAT:QUES:BLIM:PTR
- :STAT:QUES:BLIM:NTR

- :STAT:QUES:BLIM:CHAN{1-4}:ENAB
- :STAT:QUES:BLIM:CHAN{1-4}:COND?
- :STAT:QUES:BLIM:CHAN{1-4}?
- :STAT:QUES:BLIM:CHAN{1-4}:PTR
- :STAT:QUES:BLIM:CHAN{1-4}:NTR
- :STAT:QUES:RLIM:ENAB
- :STAT:QUES:RLIM:COND?
- :STAT:QUES:RLIM?
- :STAT:QUES:RLIM:PTR
- :STAT:QUES:RLIM:NTR
- :STAT:QUES:RLIM:CHAN{1-4}:ENAB
- :STAT:QUES:RLIM:CHAN{1-4}:COND?
- :STAT:QUES:RLIM:CHAN{1-4}?
- :STAT:QUES:RLIM:CHAN{1-4}:PTR
- :STAT:QUES:RLIM:CHAN{1-4}:NTR

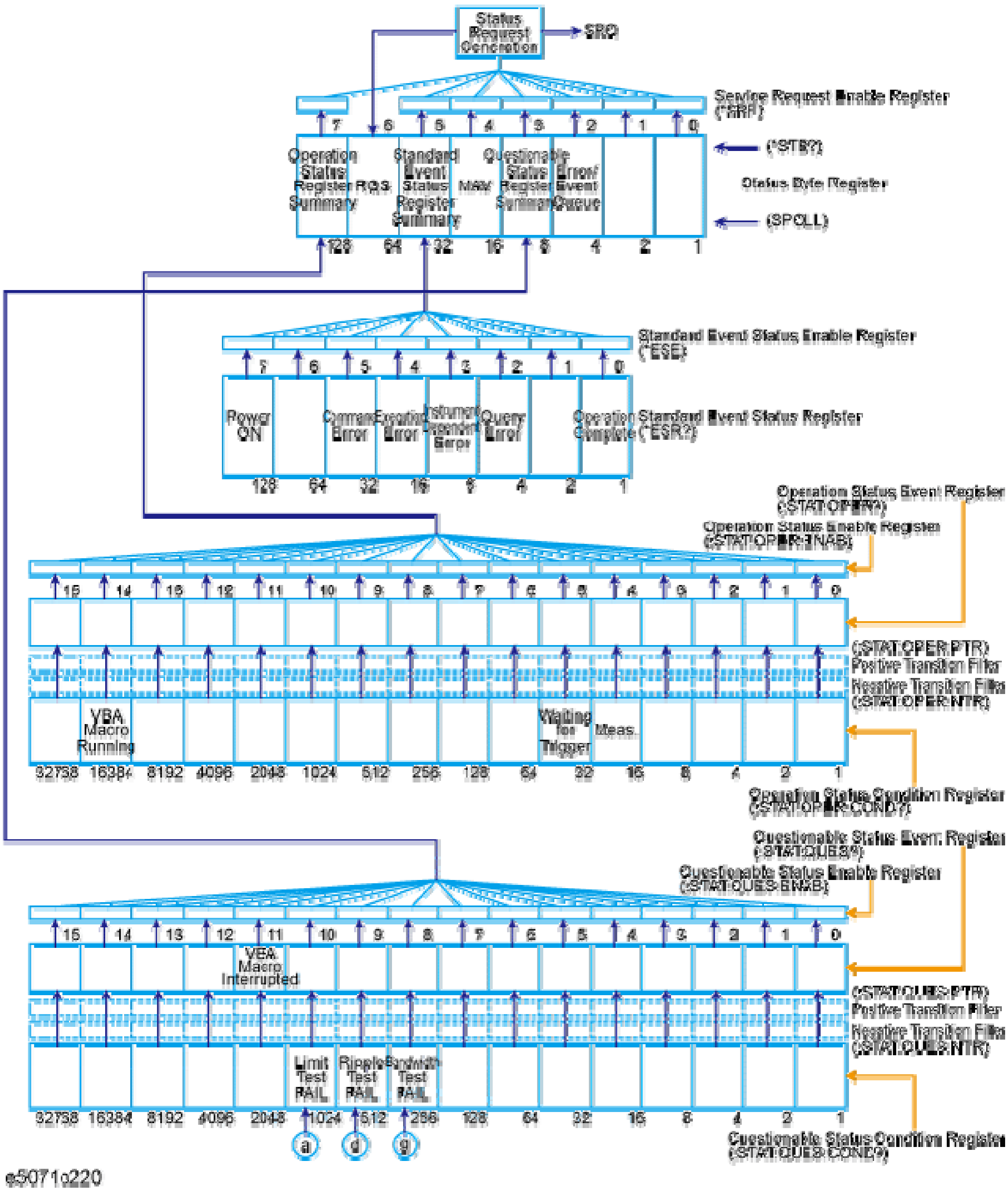
Other topics about Status Reporting System

Status Register Structure

Status Register Structure

- Status Register
- Status Register for Limit Test (Channel)
- Status Register for Limit Test (Trace)
- Status Register for Bandwidth Limit (Channel)
- Status Register for Bandwidth Limit (Trace)
- Status Register for Ripple Limit (Channel)
- Status Register for Ripple Limit (Trace)

Status Register



Status Bit Definitions of Status Byte Register

Bit Position	Name	Description

0, 1	Not used	Always 0
2	Error/Event Queue	Set to "1" if the error/event queue contains data; reset to "0" when all the data has been retrieved.
3	Questionable Status Register Summary	Set to "1" when one of the enabled bits in the questionable status register is set to "1."
4	MAV (Message Available)	Set to "1" when the output queue contains data; reset to "0" when all the data has been retrieved.
5	Standard Event Status Register Summary	Set to "1" when one of the enabled bits in the standard event status register is set to "1."
6	RQS	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 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 clears all bits from the status byte register.

Status Bit Definitions of Standard Event Status Register

Bit Position	Name	Description
0	Operation Complete	Set to "1" upon completion of all operations done by commands that precede the *OPC? command.
1	Not used	Always 0

2	Query Error	1. Set to "1" when the E5061B receives a data output request but there is no data to output. 2. Set to "1" when the data of the E5061B's output queue has been cleared because of a new message received before the completion of data output.
3	Instrument Dependent 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 an SCPI command exceeds its input range or is inconsistent with the E5061B's capabilities. 2. Set to "1" when an SCPI command cannot be properly executed due to some condition of the E5061B.
5	Command Error	1. Set to "1" when an IEEE 488.2 syntax error occurs (a command sent to the E5061B does not follow the IEEE 488.2 syntax). Possible violations include the command parameter violating the E5061B listening formats or being unacceptable. 2. Set to "1" when a semantic error occurs. Possible causes include a command containing misspellings being sent to the E5061B or an IEEE 488.2 command not supported by the E5061B being sent. 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 E5061B is powered ON, or when the firmware is restarted.

Issuing the *CLS command clears all bits from the standard event status register.

Status Bit Definitions of the Operation Status Condition Register

Bit Position	Name	Description
0 - 3	Not used	Always 0
4	Measurement	Set to "1" during measurement
5	Waiting for Trigger	Set to "1" while the instrument is waiting for a trigger.
6 - 13	Not used	Always 0
14	VBA Macro Running	Set to "1" while a VBA macro is running.
15	Not used	Always 0

Issuing the *CLS command clears all bits from the operation status event register.

Status Bit Definitions of the Questionable Status Condition Register

Bit Position	Name	Description
0 - 7	Not used	Always 0
8	Bandwidth Test Fail (Questionable bandwidth limit status register summary)	Set to "1" while one of the enabled bits in the questionable bandwidth limit status event register is set to "1."

9	Ripple Test Fail (Questionable ripple limit status register summary)	Set to "1" while one of the enabled bits in the questionable ripple limit status event register is set to "1."
10	Limit Test Fail (Questionable limit status register summary)	Set to "1" while one of the enabled bits in the questionable limit status event register is set to "1."
11 - 15	Not used	Always 0

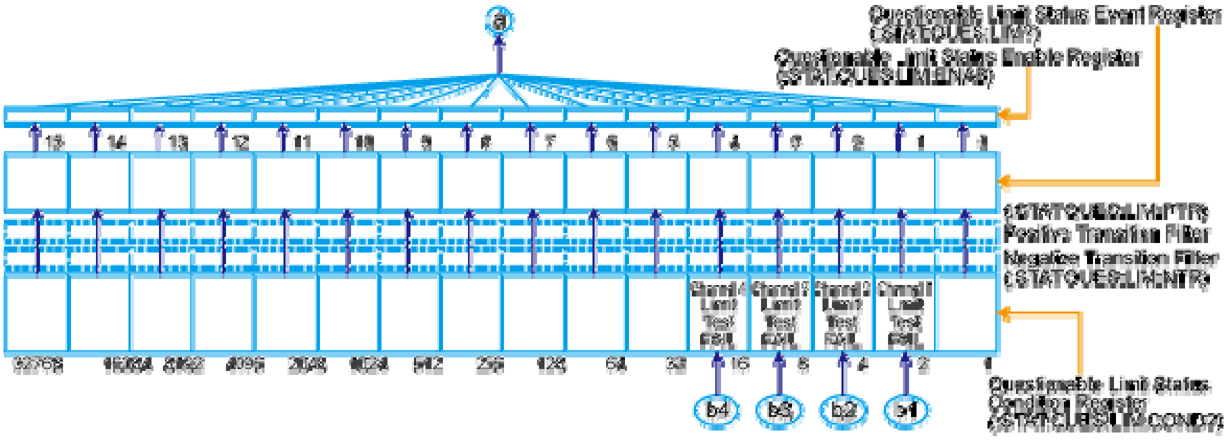
Status Bit Definitions of the Questionable Status Event Register

Bit Position	Name	Description
0 - 7	Not used	Always 0
8	Bandwidth Test Fail (Questionable bandwidth limit status register summary)	Set to "1" when a transition of the condition register occurs if the transition filters are set as valid values.
9	Ripple Test Fail (Questionable ripple limit status register summary)	Set to "1" when a transition of the condition register occurs if the transition filters are set as valid values.
10	Limit Test Fail (Questionable limit status register summary)	Set to "1" when a transition of the condition register occurs if the transition filters are set as valid values.
11	VBA Macro Interrupted	Set to "1" when a VBA macro is interrupted by one of the following

		reasons. Occurrence of an execution error Executing "End" statement in the VBA Macro Executing :PROG:STAT STOP Operating **CTRL** + **Break** using the keyboard Operating **Marcro Break** or **Macro Setup** > **Stop** using the front panel
12 - 15	Not used	Always 0

Issuing the *CLS command clears all bits from the questionable status event register.

Status Register for Limit Test (channel)



e5061b015

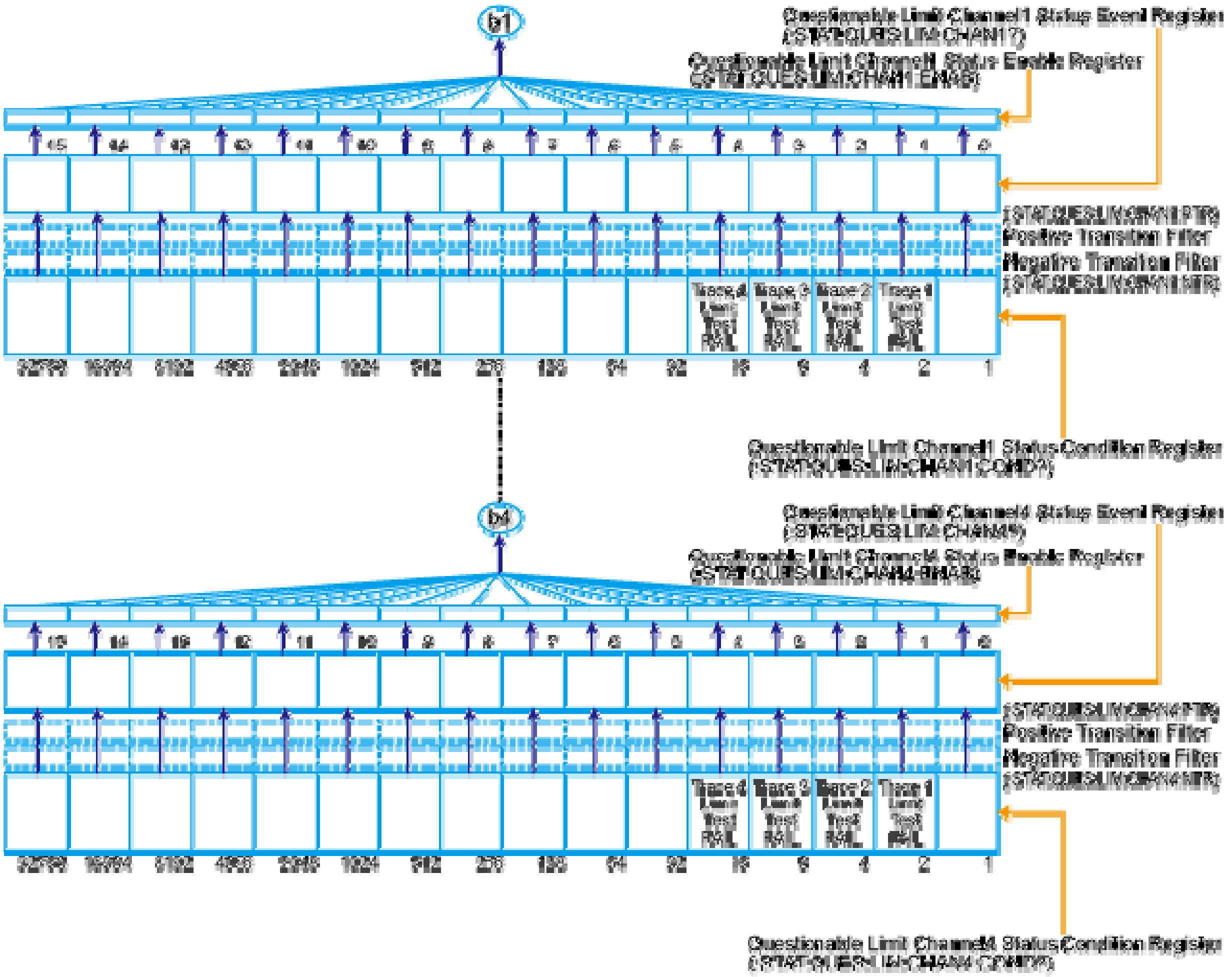
Status Bit Definitions of the Questionable Limit Status Condition Register

Bit Position	Name	Description
0	Not used	Always 0
1	Channel 1 Limit Test Fail (questionable limit channel 1 status register summary)	Set to "1" while one of the enabled bits in the questionable limit channel 1 status event register is set to "1."
2	Channel 2 Limit Test Fail (questionable limit channel 2 status register summary)	Set to "1" while one of the enabled bits in the questionable limit channel 2 status event register is set to "1."
3	Channel 3 Limit Test Fail (questionable limit channel 3 status register summary)	Set to "1" while one of the enabled bits in the questionable limit channel 3 status event register is set to "1."
4	Channel 4 Limit Test Fail (questionable limit channel 4 status register	Set to "1" while one of the enabled bits in the questionable limit channel 4

	summary)	status event register is set to "1."
5 to 15	Not used	Always 0

Issuing the *CLS command clears all bits from the questionable limit status event register.

Status Register for Limit Test (Trace)



e5061b016

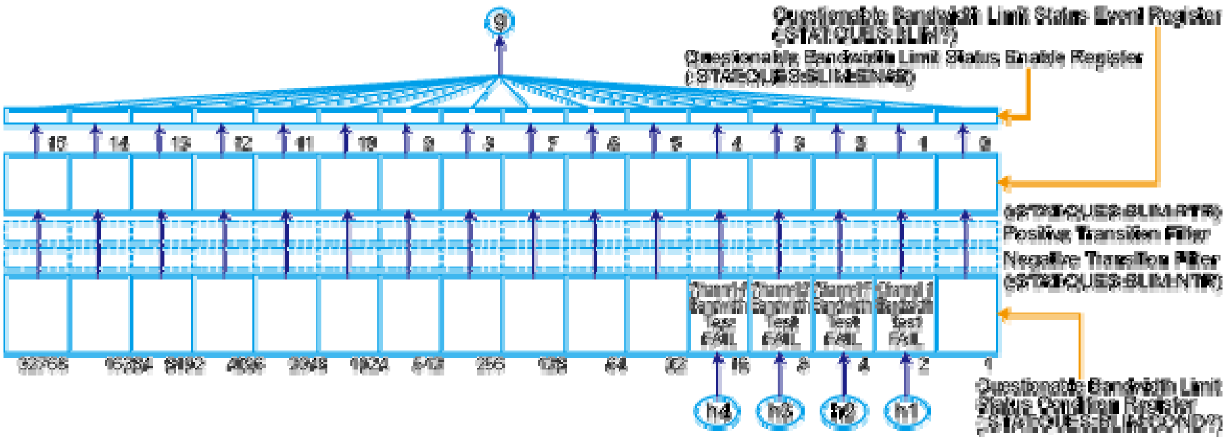
Status Bit Definitions of the Questionable Limit Channel Status Condition Register

Bit Position	Name	Description
0	Not used	Always 0
1	Trace 1 Limit Test Fail	Set to "0" when a measurement cycle begins; set to "1" when the measurement cycle finishes and returns "fail" as the limit test result for trace 1.

2	Trace 2 Limit Test Fail	Set to "0" when a measurement cycle begins; set to "1" when the measurement cycle finishes and returns "fail" as the limit test result for trace 2.
3	Trace 3 Limit Test Fail	Set to "0" when a measurement cycle begins; set to "1" when the measurement cycle finishes and returns "fail" as the limit test result for trace 3.
4	Trace 4 Limit Test Fail	Set to "0" when a measurement cycle begins; set to "1" when the measurement cycle finishes and returns "fail" as the limit test result for trace 4.
5 to 15	Not used	Always 0

Issuing the *CLS command clears all the bits in the questionable limit channel status event register.

Status Register for Bandwidth Limit (Channel)



e5061b017

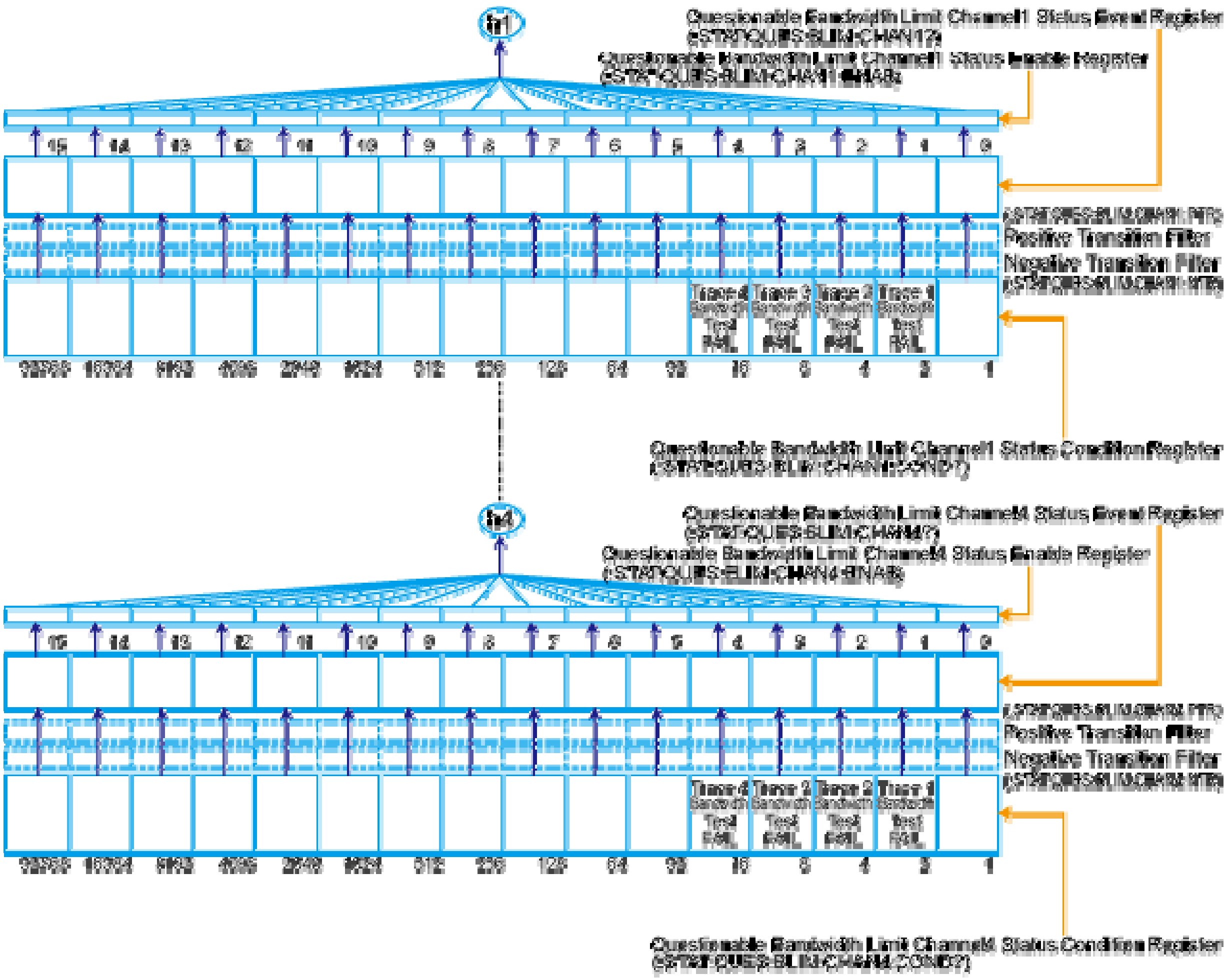
Status Bit Definitions of the Questionable Bandwidth Limit Channel Status Condition Register

Bit Position	Name	Description
0	Not Used	Always 0
1	Channel 1 Bandwidth Test Fail	Set to "0" when a measurement cycle begins; set to "1" when the measurement cycle finishes and returns "fail" as the bandwidth test result for channel 1.
2	Channel 2 Bandwidth Test Fail	Set to "0" when a measurement cycle begins; set to "1" when the measurement cycle finishes and returns "fail" as the bandwidth test result for channel 2.
3	Channel 3 Bandwidth Test Fail	Set to "0" when a measurement cycle begins; set to "1" when the measurement cycle finishes and returns "fail" as the bandwidth test result for channel 3.
4	Channel 4 Bandwidth Test	Set to "0" when a measurement cycle begins; set to "1" when the measurement cycle

	Fail	finishes and returns "fail" as the bandwidth test result for channel 4.
5 to 15	Not used	Always 0

Issuing the *CLS command clears all the bits in the questionable bandwidth limit channel {1-4} status event register.

Status Register for Bandwidth Limit (Trace)



e5071b018

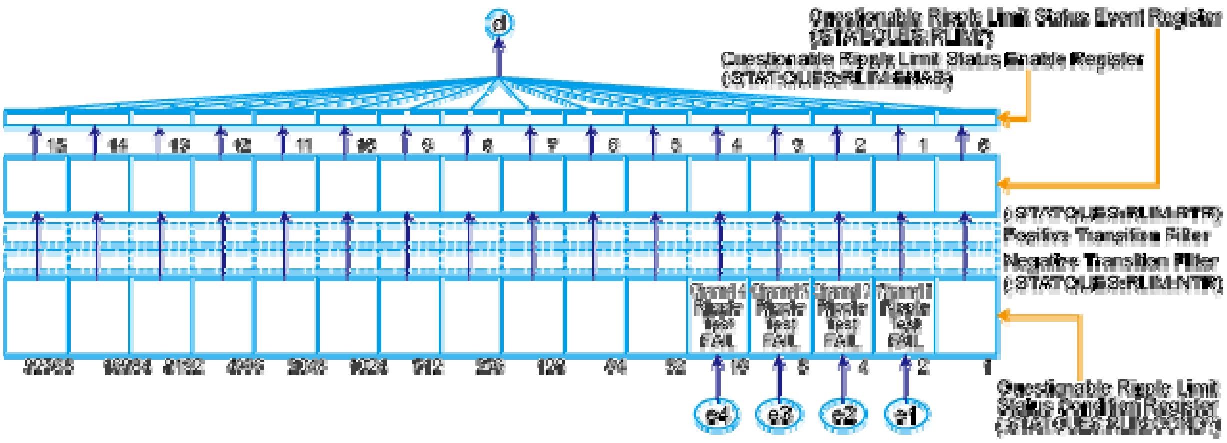
Status Bit Definitions of the Questionable Bandwidth Limit Status Condition Register

Bit Position	Name	Description
0	Not used	Always 0
1	Channel 1 Bandwidth Test Fail (questionable bandwidth limit channel 1 status register summary)	Set to "1" while one of the enabled bits in the questionable bandwidth limit channel 1 status event register is set to "1."

2	Channel 2 Bandwidth Test Fail (questionable bandwidth limit channel 2 status register summary)	Set to "1" while one of the enabled bits in the questionable bandwidth limit channel 2 status event register is set to "1."
3	Channel 3 Bandwidth Test Fail (questionable bandwidth limit channel 3 status register summary)	Set to "1" while one of the enabled bits in the questionable bandwidth limit channel 3 status event register is set to "1."
4	Channel 4 Bandwidth Test Fail (questionable bandwidth limit channel 4 status register summary)	Set to "1" while one of the enabled bits in the questionable bandwidth limit channel 4 status event register is set to "1."
5 to 15	Not used	Always 0

Issuing the *CLS command clears all bits from the questionable bandwidth limit status event register.

Status Register for Ripple Limit (Channel)



e5061b019

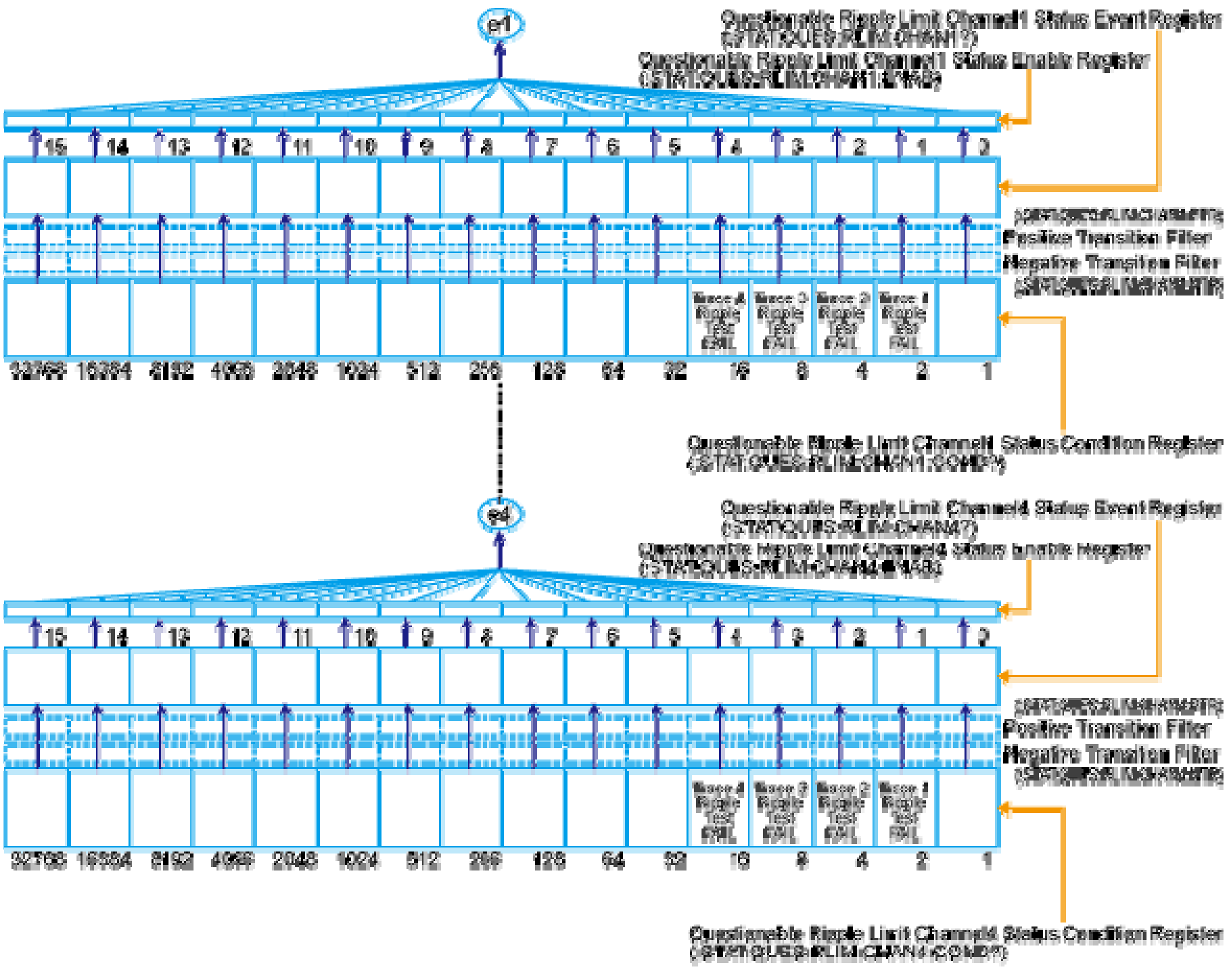
Status Bit Definitions of the Questionable Ripple Limit Status Condition Register

Bit Position	Name	Description
0	Not used	Always 0
1	Channel 1 Ripple Test Fail (questionable ripple limit channel 1 status register summary)	Set to "1" while one of the enabled bits in the questionable ripple limit channel 1 status event register is set to "1."
2	Channel 2 Ripple Test Fail (questionable ripple limit channel 2 status register summary)	Set to "1" while one of the enabled bits in the questionable ripple limit channel 2 status event register is set to "1."
3	Channel 3 Ripple Test Fail (questionable ripple limit channel 3 status register summary)	Set to "1" while one of the enabled bits in the questionable ripple limit channel 3 status event register is set to "1."
4	Channel 4 Ripple Test Fail (questionable ripple limit channel 4 status register	Set to "1" while one of the enabled bits in the questionable ripple limit

	summary)	channel 4 status event register is set to "1."
5 to 15	Not used	Always 0

Issuing the *CLS command clears all bits from the questionable ripple limit status event register.

Status Register for Ripple Limit (Trace)



e5061b020

Status Bit Definitions of the Questionable Ripple Limit Channel Status Condition Register

Bit Position	Name	**Description**
0	Not used	Always 0
1	Trace 1 Ripple Test Fail	Set to "0" when a measurement cycle begins; set to "1" when the measurement cycle finishes and returns "fail" as the ripple test result for trace 1.

2	Trace 2 Ripple Test Fail	Set to "0" when a measurement cycle begins; set to "1" when the measurement cycle finishes and returns "fail" as the ripple test result for trace 2.
3	Trace 3 Ripple Test Fail	Set to "0" when a measurement cycle begins; set to "1" when the measurement cycle finishes and returns "fail" as the ripple test result for trace 3.
4	Trace 4 Ripple Test Fail	Set to "0" when a measurement cycle begins; set to "1" when the measurement cycle finishes and returns "fail" as the ripple test result for trace 4.
5 to 15	Not used	Always 0

Issuing the *CLS command clears all the bits in the questionable ripple limit channel {1-4} status event register.

Sample Programs

Sample Programs

This section shows sample programs with the SCPI commands which can be executed from the external controller. See Application Programs under VBA Programming about the sample programs for built-in VBA.

- Analyzer Setup
- Calibration
- ECal
- Reading/Writing Error Coefficient
- Waiting for Trigger (OPC?)
- Waiting for Trigger (SRQ)
- Error Detection (SRQ)
- Reading Data in Ascii Format
- Reading Data in Binary Format
- Writing Data in Ascii Format
- Writing Data in Binary Format
- Peak Search
- Bandwidth Search
- Limit Test
- Saving Files
- Transferring Files
- Time Domain
- Control Using SICL-LAN Server
- Controlling Using Telnet Server
- Handler Interface

These sample program files can be downloaded from http://www.agilent.com/find/ena_support.

Analyzer Setup

- Overview
- Sample Program in Excel VBA
- Sample Program in HT Basic

Other topics about Sample Programs

Overview

The program listed in this section is a sample program that demonstrates how to configure measurement conditions.

The sample program puts the instrument into the preset state, configures it as shown in table below, and saves the settings to a file named **sample.sta**.

See Setting up Analyzer for this programming.

Target settings

Item		Setting
Window Layout		Channel 1 in the upper window (2/3 of the screen height) and channel 2 in the lower window (1/3 of the screen height)
Channel 1	Sweep type	Segment
	Sweep range	See Segment table.
	Number of measurement points	
	IF bandwidth	
	Power	
	Number of traces	4

	Graph Layout		Four graphs at upper left, upper right, lower left, and lower right.
	Trace 1	Measurement parameter	S11
		Data format	Smith chart (Lin)
		Full-scale value	2
	Trace 2	Measurement parameter	S21
		Data format	Log magnitude
		Reference division line number	9
		Reference division line value	2
		Scale per division	10 dBm
	Trace 3	Measurement parameter	S12
		Data format	Log magnitude
		Reference division line number	9
		Reference division line value	2
		Scale per division	10 dBm
	Trace 4	Measurement parameter	S22
		Data format	Smith chart (Lin)
		Full-scale value	2
Channel	Sweep type		Linear

2	Sweep range	Center value	1.9 GHz
		Span value	500 MHz
	Number of measurement points		101
	IF bandwidth		70 kHz
	Power		0 dBm
	Number of traces		2
	Graph Layout		Two graphs at left and right
	Trace 1	Measurement parameter	S21
		Data format	Log magnitude
		Reference division line number	9
		Reference division line value	2
		Scale per division	10 dBm
	Trace 2	Measurement parameter	S22
		Data format	Smith chart (Lin)
		Full-scale value	2

Segment table for channel 1

Segment Number	Start value	Stop value	Number of measurement points	IF bandwidth	Power
1	1.7 GHz	1.9 GHz	21	50 kHz	0 dBm
2	1.9 GHz	2 GHz	101	10 kHz	-10 dBm
3	2 GHz	2.2 GHz	21	50 kHz	0 dBm

Sample Program in Excel VBA

```
Sub Setup()
```

```
 '
Dim defrm As Long
Dim vi As Long
Const TimeOutTime = 20000

Dim Allocate1 As String, Allocate2 As String, File As String
Dim Para1(4) As String, Para2(2) As String
Dim Fmt1(4) As String, Fmt2(2) As String
Dim Star1(3) As String, Stop1(3) As String
Dim IfBw1(3) As Double, IfBw2 As Double
Dim Power1(3) As Double, Power2 As Double
Dim Cent2 As Double, Span2 As Double
Dim RefLev1(4) As Double, RefLev2(2) As Double, Scale1(4) As Double, Scale2(2) As Double
Dim Segm As Integer, Nop1(3) As Integer, Nop2 As Integer
Dim NumOfTr1 As Integer, NumOfTr2 As Integer
Dim RefPos1(4) As Integer, RefPos2(2) As Integer
Dim SendData As String
'
Segm = 3 ' Number of Segment Ch.1 : 3
Star1(1) = "1.7E9"      ' Start Frequency Ch.1 Segm.1: 1.7 GHz
Star1(2) = "1.9E9"      ' Segm.2: 1.9 GHz
Star1(3) = "2E9"        ' Segm.3: 2 GHz
Stop1(1) = "1.9E9"      ' Stop Frequency Ch.1 Segm.1: 1.9 GHz
Stop1(2) = "2E9"        ' Segm.2: 2 GHz
Stop1(3) = "2.2E9"      ' Segm.3: 2.2 GHz
Cent2 = 1900000000#     ' Center Frequency Ch.2 : 1.9 GHz
Span2 = 500000000#      ' Span Ch.2 : 500 MHz
Nop1(1) = 21            ' Number of points Segm.1: 21
Nop1(2) = 101           '               Segm.2: 101
Nop1(3) = 21            '               Segm.3: 21
Nop2 = 101              '       Ch.2 : 101
IfBw1(1) = 50000#       ' IFBW  Ch.1    Segm.1: 50 kHz
IfBw1(2) = 10000#       '               Segm.2: 10 kHz
IfBw1(3) = 50000#       '               Segm.3: 50 kHz
IfBw2 = 70000#          '       Ch.2 : 70 kHz
Power1(1) = 0           ' Power Ch.1    Segm.1: 0 dBm
Power1(2) = -10         '               Segm.2: -10 dBm
```

```
Power1(3) = 0           '               Segm.3: 0 dBm
Power2 = 0              '       Ch.2 : 0 dBm
NumOfTr1 = 4            ' Number of Trace Ch.1 : 4
NumOfTr2 = 2            ' Ch.2 : 2
Allocate1 = "D12_34"    ' Allocate Traces Ch.1 : D12_34
Allocate2 = "D12"       ' Ch.2 : D12
Para1(1) = "S11"        ' Measurement Ch.1 Trace1: S11
Para1(2) = "S21"        ' Parameter Trace2: S21
Para1(3) = "S12"        ' Trace3: S12
Para1(4) = "S22"        ' Trace4: S22
Para2(1) = "S21"        ' Ch.2 Trace1: S21
Para2(2) = "S22"        ' Trace2: S22
Fmt1(1) = "SLIN"        ' Data Format Ch.1 Trace1: Smith(Lin/Phase)
Fmt1(2) = "MLOG"        ' Trace2: Log Mag
Fmt1(3) = "MLOG"        ' Trace3: Log Mag
Fmt1(4) = "SLIN"        ' Trace4: Smith(Lin/Phase)
Fmt2(1) = "MLOG"        ' Ch.2 Trace1: Log Mag
Fmt2(2) = "SLIN"        ' Trace2: Smith(Lin/Phase)
RefPos1(1) = 9          ' Reference Ch.1 Trace2: 9
RefPos1(2) = 9          ' Position Trace3: 9
RefPos2(1) = 9          ' Ch.2 Trace1: 9
RefLev1(1) = 0          ' Reference Level Ch.1 Trace2: 0 dBm
RefLev1(2) = 0          ' Trace3: 0 dBm
RefLev2(1) = 0          ' Ch.2 Trace1: 0 dBm
Scale1(1) = 2           ' Scale Ch.1 Trace1: 2
Scale1(2) = 10          ' Trace2: 10 dBm
Scale1(3) = 10          ' Trace3: 10 dBm
Scale1(4) = 2           ' Trace4: 2
Scale2(1) = 10          ' Ch.2 Trace1: 10 dBm
Scale2(2) = 2           ' Trace2: 2
StaFileName = "sample.sta" ' Save File Name : sample.sta
'
' Assigns a GPIB address to the I/O pass.
Call viOpenDefaultRM(defrm)
Call viOpen(defrm, "GPIB0::17::INSTR", 0, 0, vi)
Call viSetAttribute(vi, VI_ATTR_TMO_VALUE, TimeOutTime) ' Set time out
'
```

```
    Call viVPrintf(vi, ":SYST:PRES" + vbLf, 0)
    Call viVPrintf(vi, ":DISP:SPL D1_1_2" + vbLf, 0) 'Allocate Channel
    Call viVPrintf(vi, ":INIT1:CONT ON" + vbLf, 0) 'Turn on Continuous Activation mode for channel 1
    Call viVPrintf(vi, ":INIT2:CONT ON" + vbLf, 0) 'Turn on Continuous Activation mode for channel 2
    ' ===============
    ' Setup Channel 1
    ' ===============
    Call viVPrintf(vi, ":SENS1:SWE:TYPE SEGM" + vbLf, 0) 'Sets channel 1 sweep type to segment
    '
    ' Create the data string for Segment Table
    SendData = "5,0,1,1,0,0," & Str(Segm)
    For i = 1 To Segm
        SendData = SendData + "," & Star1(i) & "," + Stop1(i) & "," & CStr(Nop1(i)) & "," & CStr(IfBw1(i)) &
"," & CStr(Power1(i))
    Next i
    Call viVPrintf(vi, ":SENS1:SEGM:DATA " + SendData + vbLf, 0)
    '
    Call viVPrintf(vi, ":CALC1:PAR:COUN " & CStr(NumOfTr1) & vbLf, 0) 'Set number of traces
    Call viVPrintf(vi, ":DISP:WIND1:SPL " & Allocate1 & vbLf, 0) 'Set graph layout


    For i = 1 To NumOfTr1
        Call viVPrintf(vi, ":CALC1:PAR" & CStr(i) & ":DEF " & Para1(i) & vbLf, 0) 'Set measurement
parameter
        Call viVPrintf(vi, ":CALC1:PAR" & CStr(i) & ":SEL" & vbLf, 0) ' Make trace active
        Call viVPrintf(vi, ":CALC1:FORM " & Fmt1(i) & vbLf, 0) ' Set data format
        '
        Select Case Fmt1(i)
        Case "SLIN", "SLOG", "SCOM", "SMIT", "SADM", "PLIN", "PLOG", "POL"
        ' If data format is neither Smith chart nor polar, sets reference division line number and scale per
division
            Call viVPrintf(vi, ":DISP:WIND1:TRAC" & CStr(i) & ":Y:PDIV " + CStr(Scale1(i)) + vbLf, 0)
        Case Else
        ' If data format is Smith chart or polar, set full-scale value
            Call viVPrintf(vi, ":DISP:WIND1:TRAC" & CStr(i) & ":Y:RPOS " & CStr(RefPos1(i)) & vbLf, 0)
            Call viVPrintf(vi, ":DISP:WIND1:TRAC" & CStr(i) & ":Y:RLEV " & CStr(RefLev1(i)) & vbLf, 0)
            Call viVPrintf(vi, ":DISP:WIND1:TRAC" & CStr(i) & ":Y:PDIV " & CStr(Scale1(i)) & vbLf, 0)
        End Select
    Next i
```

```
' ===============
' Setup Channel 2
' ===============
Call viVPrintf(vi, ":SENS2:SWE:TYPE LIN " + vbLf, 0) ' Set sweep type to linear
Call viVPrintf(vi, ":SENS2:FREQ:CENT " + CStr(Cent2) + vbLf, 0) ' Set center frequency
Call viVPrintf(vi, ":SENS2:FREQ:SPAN " + CStr(Span2) + vbLf, 0) ' Set span frequency
Call viVPrintf(vi, ":SENS2:SWE:POIN " + CStr(Nop2) + vbLf, 0) ' Set number of points
Call viVPrintf(vi, ":SENS2:BAND " + CStr(IfBw2) + vbLf, 0) ' Set IFBW
Call viVPrintf(vi, ":SOUR2:POW " + CStr(Power2) + vbLf, 0) ' Set power level
Call viVPrintf(vi, ":CALC2:PAR:COUN " & CStr(NumOfTr2) & vbLf, 0) ' Set number of traces
Call viVPrintf(vi, ":DISP:WIND2:SPL " & Allocate2 & vbLf, 0) 'Set graph layout

For i = 1 To NumOfTr2
    Call viVPrintf(vi, ":CALC2:PAR" & CStr(i) & ":DEF " & Para2(i) & vbLf, 0) 'Set measurement
parameter
    Call viVPrintf(vi, ":CALC2:PAR" & CStr(i) & ":SEL" & vbLf, 0) ' Make trace active
    Call viVPrintf(vi, ":CALC2:FORM " & Fmt2(i) & vbLf, 0) ' Set data format
    '
    Select Case Fmt2(i)
    Case "SLIN", "SLOG", "SCOM", "SMIT", "SADM", "PLIN", "PLOG", "POL"
    ' If data format is neither Smith chart nor polar, sets reference division line number and scale per
division
        Call viVPrintf(vi, ":DISP:WIND2:TRAC" & CStr(i) & ":Y:PDIV " + CStr(Scale2(i)) + vbLf, 0)
    Case Else
    ' If data format is Smith chart or polar, set full-scale value
        Call viVPrintf(vi, ":DISP:WIND2:TRAC" & CStr(i) & ":Y:RPOS " & CStr(RefPos2(i)) & vbLf, 0)
        Call viVPrintf(vi, ":DISP:WIND2:TRAC" & CStr(i) & ":Y:RLEV " & CStr(RefLev2(i)) & vbLf, 0)
        Call viVPrintf(vi, ":DISP:WIND2:TRAC" & CStr(i) & ":Y:PDIV " & CStr(Scale2(i)) & vbLf, 0)
    End Select
Next i
'
Call viVPrintf(vi, ":MMEM:STOR """ & StaFileName & """" & vbLf, 0) ' Save ENA settings to file
' Close IO
Call viClose(vi)
Call viClose(defrm)

End Sub
```

Sample Program in HT Basic (setup.htb)

```
10 DIM Allocate1$[9],Allocate2$[9],File$[20]
20 DIM Para1$(1:4)[9],Para2$(1:2)[9],Fmt1$(1:4)[9],Fmt2$(1:2)[9]
30 REAL Star1(1:3),Stop1(1:3),Pow1(1:3),Cent2,Span2,Pow2
40 REAL Ref_rev1(1:4),Ref_rev2(1:2),Scale1(1:4),Scale2(1:2)
50 INTEGER Segm,Nop1(1:3),Nop2,Num_of_tr1,Num_of_tr2
60 INTEGER Ref_pos1(1:4),Ref_pos2(1:2),I
70 ASSIGN @Agte506x TO 717
80 !
90 Segm=3 ! Number of Segment Ch.1 : 3
100 Star1(1)=1.7E+9 ! Start Frequency Ch.1 Segm.1: 1.7 GHz
110 Star1(2)=1.9E+9 ! Segm.2: 1.9 GHz
120 Star1(3)=2.E+9 ! Segm.3: 2 GHz
130 Stop1(1)=1.9E+9 ! Stop Frequency Ch.1 Segm.1: 1.9 GHz
140 Stop1(2)=2.E+9 ! Segm.2: 2 GHz
150 Stop1(3)=2.2E+9 ! Segm.3: 2.2 GHz
160 Cent2=1.9E+9 ! Center Frequency Ch.2 : 1.9 GHz
170 Span2=5.00E+8 ! Span Ch.2 : 500 MHz
180 Nop1(1)=21 ! Number Ch.1 Segm.1: 21
190 Nop1(2)=101 ! of Points Segm.2: 101
200 Nop1(3)=21 ! Segm.3: 21
210 Nop2=101 ! Ch.2 : 101
220 If_bw1(1)=5.0E+4 ! IF Bandwidth    Ch.1 Segm.1: 50 kHz
230 If_bw1(2)=1.0E+4 ! Segm.2: 10 kHz
240 If_bw1(3)=5.0E+4 ! Segm.3: 50 kHz
250 If_bw2=7.0E+4 ! Ch.2 : 70 kHz
260 Pow1(1)=0 ! Power Ch.1 Segm.1: 0 dBm
270 Pow1(2)=-10 ! Segm.2: -10 dBm
280 Pow1(3)=0 ! Segm.3: 0 dBm
290 Pow2=0 ! Ch.2 : 0 dBm
300 Num_of_tr1=4 ! Number Ch.1 : 4
310 Num_of_tr2=2 ! of Traces Ch.2 : 2
320 Allocate1$="D12_34" ! Allocate Traces Ch.1 : D12_34
330 Allocate2$="D12" ! Ch.2 : D12
340 Para1$(1)="S11" ! Measurement Ch.1 Trace1: S11
350 Para1$(2)="S21" ! Parameter Trace2: S21
360 Para1$(3)="S12" ! Trace3: S12
370 Para1$(4)="S22" ! Trace4: S22
```

```
380 Para2$(1)="S21" ! Ch.2 Trace1: S21
390 Para2$(2)="S22" ! Trace2: S22
400 Fmt1$(1)="SLIN" ! Data Format Ch.1 Trace1: Smith(Lin/Phase)
410 Fmt1$(2)="MLOG" ! Trace2: Log Mag
420 Fmt1$(3)="MLOG" ! Trace3: Log Mag
430 Fmt1$(4)="SLIN" ! Trace4: Smith(Lin/Phase)
440 Fmt2$(1)="MLOG" ! Ch.2 Trace1: Log Mag
450 Fmt2$(2)="SLIN" ! Trace2: Smith(Lin/Phase)
460 Ref_pos1(2)=9 ! Reference Ch.1 Trace2: 9
470 Ref_pos1(3)=9 ! Position Trace3: 9
480 Ref_pos2(1)=9 ! Ch.2 Trace1: 9
490 Ref_lev1(2)=0 ! Reference Level Ch.1 Trace2: 0 dBm
500 Ref_lev1(3)=0 ! Trace3: 0 dBm
510 Ref_lev2(1)=0 ! Ch.2 Trace1: 0 dBm
520 Scale1(1)=2 ! Scale Ch.1 Trace1: 2
530 Scale1(2)=10 ! Trace2: 10 dBm
540 Scale1(3)=10 ! Trace3: 10 dBm
550 Scale1(4)=2 ! Trace4: 2
560 Scale2(1)=10 ! Ch.2 Trace1: 10 dBm
570 Scale2(2)=2 ! Trace2: 2
580 File$="sample.sta" ! Save File Name : sample.sta
590 !
600 OUTPUT @Agte506x;":SYST:PRES"
610 !
620 OUTPUT @Agte506x;":DISP:SPL D1_1_2"
630 OUTPUT @Agte506x;":INIT1:CONT ON"
640 OUTPUT @Agte506x;":INIT2:CONT ON"
650 !
660 ! Channel 1
670 !
680 OUTPUT @Agte506x;":SENS1:SWE:TYPE SEGM"
690 OUTPUT @Agte506x;":SENS1:SEGM:DATA 5,0,1,1,0,0,";Segm;",";
700 FOR I=1 TO Segm-1
710 OUTPUT @Agte506x;Star1(I);",";Stop1(I);",";Nop1(I);",";If_bw1 (I);",";Pow1(I);",";
720 NEXT I
730 OUTPUT @Agte506x;Star1(Segm);",";Stop1(Segm);",";Nop1(Segm);","
;If_bw1(Segm);",";Pow(Segm)
```

```
740 !
750 OUTPUT @Agte506x;":CALC1:PAR:COUN ";Num_of_tr1
760 OUTPUT @Agte506x;":DISP:WIND1:SPL "&Allocate1$
770 FOR I=1 TO Num_of_tr1
780 OUTPUT @Agte506x;":CALC1:PAR"&VAL$(I)&":DEF "&Para1$(I)
790 OUTPUT @Agte506x;":CALC1:PAR"&VAL$(I)&":SEL"
800 OUTPUT @Agte506x;":CALC1:FORM "&Fmt1$(I)
810 SELECT Fmt1$(I)
820 CASE "SLIN","SLOG","SCOM","SMIT","SADM","PLIN","PLOG","POL"
830 OUTPUT @Agte506x;":DISP:WIND1:TRAC"&VAL$(I)&":Y:PDIV "; Scale1(I)
840 CASE ELSE
850 OUTPUT @Agte506x;":DISP:WIND1:TRAC"&VAL$(I)&":Y:RPOS "; Ref_pos1(I)
860 OUTPUT @Agte506x;":DISP:WIND1:TRAC"&VAL$(I)&":Y:RLEV "; Ref_rev1(I)
870 OUTPUT @Agte506x;":DISP:WIND1:TRAC"&VAL$(I)&":Y:PDIV "; Scale1(I)
880 END SELECT
890 NEXT I
900 !
910 ! Channel 2
920 !
930 OUTPUT @Agte506x;":SENS2:SWE:TYPE LIN"
940 OUTPUT @Agte506x;":SENS2:FREQ:CENT ";Cent2
950 OUTPUT @Agte506x;":SENS2:FREQ:SPAN ";Span2
960 OUTPUT @Agte506x;":SENS2:SWE:POIN ";Nop2
970 OUTPUT @Agte506x;":SENS2:BAND ";If_bw2
980 OUTPUT @Agte506x;":SOUR2:POW ";Pow2
990 !
1000 OUTPUT @Agte506x;":CALC2:PAR:COUN ";Num_of_tr2
1010 OUTPUT @Agte506x;":DISP:WIND2:SPL "&Allocate2$
1020 FOR I=1 TO Num_of_tr2
1030 OUTPUT @Agte506x;":CALC2:PAR"&VAL$(I)&":DEF "&Para2$(I)
1040 OUTPUT @Agte506x;":CALC2:PAR"&VAL$(I)&":SEL"
1050 OUTPUT @Agte506x;":CALC2:FORM "&Fmt2$(I)
1060 SELECT Fmt2$(I)
1070 CASE "SLIN","SLOG","SCOM","SMIT","SADM","PLIN","PLOG","POL"
1080 OUTPUT @Agte506x;":DISP:WIND2:TRAC"&VAL$(I)&":Y:PDIV "; Scale2(I)
1090 CASE ELSE
1100 OUTPUT @Agte506x;":DISP:WIND2:TRAC"&VAL$(I)&":Y:RPOS "; Ref_pos2(I)
```

```
1110 OUTPUT @Agte506x;":DISP:WIND2:TRAC"&VAL$(I)&":Y:RLEV "; Ref_rev2(I)
1120 OUTPUT @Agte506x;":DISP:WIND2:TRAC"&VAL$(I)&":Y:PDIV "; Scale2(I)
1130 END SELECT
1140 NEXT I
1150 !
1160 OUTPUT @Agte506x;":MMEM:STOR """&File$&""""
1170 END
```

Calibration

- Overview
- Sample Program in Excel VBA
- Sample Program in HT Basic

Other topics about Sample Programs

Overview

The sample program performs calibration with the specified calibration type.

See Calibration for this programming.

Sample Program in Excel VBA

```
Sub Cal_Click()
    Dim defrm As Long        'Session to Default Resource Manager
    Dim vi As Long           'Session to instrument
    Dim Ch As String
    Dim CalKit As Integer
    Dim Port(2) As String

    Const TimeOutTime = 40000   'timeout time.
    Const Cal85032F = 4         'cal kit number.

    Ch = Cells(5, 5)            'Select channel
    Port(1) = Cells(3, 6)       'Sets the select port 1.
    Port(2) = Cells(3, 7)       'Sets the select port 2.
    CalKit = Cal85032F          'Sets cal kit (85032F)

    Call viOpenDefaultRM(defrm)   'Initializes the VISA system.
    Call viOpen(defrm, "GPIB0::17::INSTR", 0, 0, vi)   'Opens the session to the specified instrument.
    Call viSetAttribute(vi, VI_ATTR_TMO_VALUE, TimeOutTime)   'The state of an attribute for the
specified session.

    Call viVPrintf(vi, "*RST" & vbLf, 0)   'Presets the setting state of the ENA.
    Call viVPrintf(vi, "*CLS" & vbLf, 0)   'Clears the all status register.

    Call viVPrintf(vi, ":SENS" & Ch & ":CORR:COLL:CKIT " & CalKit & vbLf, 0)  'Select the calibration kit

    Select Case Cells(3, 5)
        Case "Response (Open)"   'Perform response calibration (OPEN).
```

```
        Call Cal_Resp(vi, Ch, "OPEN", Port(1))
      Case "Response (Short)"   'Perform response calibration (SHORT).
        Call Cal_Resp(vi, Ch, "Short", Port(1))
      Case "Response (Thru)"    'Perform response calibration (Thru).
        Call Cal_RespThru(vi, Ch, "Thru", Port(1), Port(2))
      Case "Full 1 Port"    'Perform 1-port calibration.
        Call Cal_Slot(vi, Ch, 1, Port)
      Case "Full 2 Port"     'Perform full 2-port calibration.
        Call Cal_Slot(vi, Ch, 2, Port)
    End Select

    Call viClose(vi)  'Closes the resource manager session.
    Call viClose(defrm)  'Breaks the communication and terminates the VISA system.

    End                                                   'End
End Sub

Sub Cal_Resp(vi As Long, Ch As String, CalType As String, Port As String)
    Dim Dummy As Variant   'Variant to receive the result

    Call viVPrintf(vi, ":SENS" & Ch & ":CORR:COLL:METH:" & CalType & " " & Port & vbLf, 0)  'Sets the
calibration type.

    MsgBox ("Set " & CalType & " to Port " & Port & ". then click [OK] button")  'Display the message box.

    Call viVPrintf(vi, ":SENS" & Ch & ":CORR:COLL:" & CalType & " " & Port & vbLf, 0) 'Measurement the
calibration data.
    Call viVQueryf(vi, "*OPC?" & vbLf, "%t", Dummy)    'Reads the *OPC? result.
    Call viVPrintf(vi, ":SENS" & Ch & ":CORR:COLL:SAVE" & vbLf, 0)      'Calculating the calibration
coefficients.

    Call ErrorCheck(vi)      'Checking the error.

End Sub

Sub Cal_RespThru(vi As Long, Ch As String, CalType As String, Port1 As String, Port2 As String)
    Dim Dummy As Variant   'Variant to receive the result.
```

```
    If Port1 <> Port2 Then
        Call viVPrintf(vi, ":SENS" & Ch & ":CORR:COLL:METH:" & CalType & " " & Port1 & "," & Port2 &
vbLf, 0)   'Sets the calibration type

        MsgBox ("Set " & CalType & " to Port " & Port1 & "&" & Port2 & ". then click [OK] button")   'Display
the message box.

        Call viVPrintf(vi, ":SENS" & Ch & ":CORR:COLL:" & CalType & " " & Port1 & "," & Port2 & vbLf, 0)
 'Measurement the calibration data.
        Call viVQueryf(vi, "*OPC?" & vbLf, "%t", Dummy)      'Reads the *OPC? result.
        Call viVPrintf(vi, ":SENS" & Ch & ":CORR:COLL:SAVE" & vbLf, 0)     'Calculating the calibration
coefficients.

        Call ErrorCheck(vi)     'Checking the error.
    Else
        MsgBox ("Thru calibration select port error!")   'Displaying the error message when selected same
ports.
        Exit Sub
    End If

End Sub
Sub Cal_Slot(vi As Long, Ch As String, NumPort As String, Port() As String)
    Dim Dummy
    Dim i As Integer, j As Integer

    Select Case NumPort
        Case 1
            Call viVPrintf(vi, ":SENS" & Ch & ":CORR:COLL:METH:SOLT" & NumPort & " " & Port(1) & vbLf,
0)   'Set the 1-port calibration type.
        Case 2
            Call viVPrintf(vi, ":SENS" & Ch & ":CORR:COLL:METH:SOLT" & NumPort & " " & Port(1) & "," &
Port(2) & vbLf, 0)   'Set the full 2-port calibration type.
    End Select
    'Reflection
    For i = 1 To NumPort
        MsgBox ("Set Open to Port " & Port(i) & ". then click [OK] button")   'Display the message box.
        Call viVPrintf(vi, ":SENS" & Ch & ":CORR:COLL:OPEN " & Port(i) & vbLf, 0)    'Measurement the
OPEN calibration.
        Call viVQueryf(vi, "*OPC?" & vbLf, "%t", Dummy)    'Reads the *OPC? result.
```

```
        MsgBox ("Set Short to Port " & Port(i) & ". then click [OK] button")    'Display the message box.
        Call viVPrintf(vi, ":SENS" & Ch & ":CORR:COLL:SHORT " & Port(i) & vbLf, 0)  'Measurement the
SHORT calibration.
        Call viVQueryf(vi, "*OPC?" & vbLf, "%t", Dummy)    'Reads the *OPC? result.

        MsgBox ("Set Load to Port " & Port(i) & ". then click [OK] button")   'Display the message box.
        Call viVPrintf(vi, ":SENS" & Ch & ":CORR:COLL:LOAD " & Port(i) & vbLf, 0)    'Measurement the
LOAD calibration.
        Call viVQueryf(vi, "*OPC?" & vbLf, "%t", Dummy)     'Reads the *OPC? result.
    Next i
    'Transmission
    For i = 1 To NumPort - 1
        For j = i + 1 To NumPort
            MsgBox ("Set Thru to Port " & Port(i) & "&" & Port(j) & ". then click [OK] button")   'Display the
message box.
            Call viVPrintf(vi, ":SENS" & Ch & ":CORR:COLL:THRU " & Port(i) & "," & Port(j) & vbLf, 0)
   'Measurement the THRU calibration.
            Call viVQueryf(vi, "*OPC?" & vbLf, "%t", Dummy)    'Reads the *OPC result.
            Call viVPrintf(vi, ":SENS" & Ch & ":CORR:COLL:THRU " & Port(j) & "," & Port(i) & vbLf, 0)
   'Measurement the THRU calibration.
            Call viVQueryf(vi, "*OPC?" & vbLf, "%t", Dummy)    'Reads the *OPC result.
        Next j
    Next i
    Call viVPrintf(vi, ":SENS" & Ch & ":CORR:COLL:SAVE" & vbLf, 0)  'Calculating the calibration
coefficients.

    Call ErrorCheck(vi)    'Checking the error.

End Sub
Sub ErrorCheck(vi As Long)
    Dim err As String * 50, ErrNo As Variant, Response

    Call viVQueryf(vi, ":SYST:ERR?" & vbLf, "%t", err)    'Reads error message.
    ErrNo = Split(err, ",")    'Gets the error code.

    If Val(ErrNo(0)) <> 0 Then
        Response = MsgBox(CStr(ErrNo(1)), vbOKOnly)    'Display the message box.
    End If
```

```
End Sub
```

Sample Program in HT Basic (cal.htb)

```
10 DIM File$[20],Ch$[9],Inp_char$[9]
20 INTEGER Cal_kit,Cal_type,Port(1:2)
30 !
40 ASSIGN @Agte506x TO 717
50 File$="Ex_4_1.sta"
60 Ch$="1"
70 !
80 Select_cal_kit(@Agte506x,Ch$)
90 !
100 CLEAR SCREEN
110 ON ERROR GOTO Type_select
120 Type_select: !
130 PRINT "## Calibration Type Selection ##"
140 PRINT " 1: Response (Open)"
150 PRINT " 2: Response (Short)"
160 PRINT " 3: Response (Thru)"
170 PRINT " 4: Full 1 Port"
180 PRINT " 5: Full 2 Port"
210 PRINT ""
220 PRINT "Input 1 to 5"
230 INPUT "Input number? (1 to 5)",Inp_char$
240 Cal_type=IVAL(Inp_char$,10)
250 IF Cal_type<1 OR Cal_type>5 THEN Type_select
260 OFF ERROR
270 !
280 SELECT Cal_type
290 CASE 1
300 Select_port(1,Port(*))
310 Cal_resp(@Agte506x,Ch$,"OPEN",Port(1))
320 CASE 2
330 Select_port(1,Port(*))
340 Cal_resp(@Agte506x,Ch$,"SHOR",Port(1))
350 CASE 3
360 Select_port(2,Port(*))
370 Cal_resp_thru(@Agte506x,Ch$,Port(1),Port(2))
```

```
380 CASE 4
390 Select_port(1,Port(*))
400 Cal_solt(@Agte506x,Ch$,1,Port(*))
410 CASE 5
420 Select_port(2,Port(*))
430 Cal_solt(@Agte506x,Ch$,2,Port(*))
500 END SELECT
510 !
520 OUTPUT @Agte506x;":MMEM:STOR:STYP CST"
530 OUTPUT @Agte506x;":MMEM:STOR """&File$&""""
540 END
550 !==============================================
560 ! Calibration Kit Selection Function
570 !==============================================
580 SUB Select_cal_kit(@Agte506x,Ch$)
590 DIM Cal_kit_lbl$(1:10)[20],Inp_char$[9]
600 INTEGER Cal_kit,I
610 CLEAR SCREEN
620 !
630 FOR I=1 TO 10
640 OUTPUT @Agte506x;":SENS1:CORR:COLL:CKIT ";I
650 OUTPUT @Agte506x;":SENS1:CORR:COLL:CKIT:LAB?"
660 ENTER @Agte506x;Cal_kit_lbl$(I)
670 NEXT I
680 ON ERROR GOTO Kit_select
690 Kit_select: !
700 PRINT "## Calibration Kit Selection ##"
710 FOR I=1 TO 10
720 PRINT USING "X,2D,A,X,20A";I,":",Cal_kit_lbl$(I)
730 NEXT I
740 PRINT ""
750 PRINT "Input 1 to 10"
760 INPUT "Input number? (1 to 10)",Inp_char$
770 Cal_kit=IVAL(Inp_char$,10)
780 IF Cal_kit<1 OR Cal_kit>10 THEN Kit_select
790 OFF ERROR
800 !
```

```
810 OUTPUT @Agte506x;":SENS"&Ch$&":CORR:COLL:CKIT ";Cal_kit
820 SUBEND
830 !==============================================
840 ! Port Selection Function
850 !==============================================
860 SUB Select_port(INTEGER Num_of_ports,INTEGER Port(*))
870 DIM Inp_char$[9]
880 !
890 CLEAR SCREEN
900 IF Num_of_ports=2 THEN
910 Port(1)=1
920 Port(2)=2
950 ELSE
960 PRINT "## Test Ports Selection ##"
970 ON ERROR GOTO Port_select
990 PRINT "Port(1):";
1000 Port_select:!
1010 INPUT "Number?",Inp_char$
1020 Port(1)=IVAL(Inp_char$,10)
1070 PRINT Port(1)
1090 OFF ERROR
1100 END IF
1110 SUBEND
1120 !==============================================
1130 ! Response (Open/Short) Calibration Function
1140 !==============================================
1150 SUB Cal_resp(@Agte506x,Ch$,Type$,INTEGER Port)
1160 DIM Buff$[9]
1170 !
1180 PRINT "## Response ("&Type$&") Calibration ##"
1190 OUTPUT @Agte506x;":SENS"&Ch$&":CORR:COLL:METH:"&Type$&" ";Port
1200 PRINT "Set "&Type$&" to Port "&VAL$(Port)&". Then push [Enter] key."
1210 INPUT "",Buff$
1220 OUTPUT @Agte506x;":SENS"&Ch$&":CORR:COLL:"&Type$&" ";Port
1230 OUTPUT @Agte506x;"*OPC?"
1240 ENTER @Agte506x;Buff$
1250 OUTPUT @Agte506x;":SENS"&Ch$&":CORR:COLL:SAVE"
```

```
1260 PRINT "Done"
1270 SUBEND
1280 !=============================================
1290 ! Response (Thru) Calibration Function
1300 !=============================================
1310 SUB Cal_resp_thru(@Agte506x,Ch$,INTEGER Port1,Port2)
1320 DIM Buff$[9]
1330 !
1340 PRINT "## Response (Thru) Calibration ##"
1350 OUTPUT @Agte506x;":SENS"&Ch$&":CORR:COLL:METH:THRU ";Port1;",";Port2
1360 PRINT "Set THRU between Port "&VAL$(Port1)&" and Port "&VAL$(Port2 )&". Then push [Enter] key."
1370 INPUT "",Buff$
1380 OUTPUT @Agte506x;":SENS"&Ch$&":CORR:COLL:THRU ";Port1;",";Port2
1390 OUTPUT @Agte506x;"*OPC?"
1400 ENTER @Agte506x;Buff$
1410 OUTPUT @Agte506x;":SENS"&Ch$&":CORR:COLL:SAVE"
1420 PRINT "Done"
1430 SUBEND
1440 !=============================================
1450 ! Full n Port Calibration Function
1460 !=============================================
1470 SUB Cal_solt(@Agte506x,Ch$,INTEGER Num_of_ports,INTEGER Port(*))
1480 DIM Buff$[9]
1490 INTEGER I,J
1500 !
1510 PRINT "## Full "&VAL$(Num_of_ports)&" Port Calibration ##"
1520 !
1530 ! Calibration Type Selection
1540 !
1550 OUTPUT @Agte506x;":SENS"&Ch$&":CORR:COLL:METH:SOLT"&VAL$(Num_of_ ports)&" ";
1560 FOR I=1 TO Num_of_ports-1
1570 OUTPUT @Agte506x;Port(I);",";
1580 NEXT I
1590 OUTPUT @Agte506x;Port(Num_of_ports)
1600 !
1610 ! Reflection Measurement
```

```
1620 !
1630 FOR I=1 TO Num_of_ports
1640 PRINT "Set OPEN to Port "&VAL$(Port(I))&". Then push [Enter] key."
1650 INPUT "",Buff$
1660 OUTPUT @Agte506x;":SENS"&Ch$&":CORR:COLL:OPEN ";Port(I)
1670 OUTPUT @Agte506x;"*OPC?"
1680 ENTER @Agte506x;Buff$
1690 PRINT "Set SHORT to Port "&VAL$(Port(I))&". Then push [Enter] key."
1700 INPUT "",Buff$
1710 OUTPUT @Agte506x;":SENS"&Ch$&":CORR:COLL:SHOR ";Port(I)
1720 OUTPUT @Agte506x;"*OPC?"
1730 ENTER @Agte506x;Buff$
1740 PRINT "Set LOAD to Port "&VAL$(Port(I))&". Then push [Enter] key."
1750 INPUT "",Buff$
1760 OUTPUT @Agte506x;":SENS"&Ch$&":CORR:COLL:LOAD ";Port(I)
1770 OUTPUT @Agte506x;"*OPC?"
1780 ENTER @Agte506x;Buff$
1790 NEXT I
1800 !
1810 ! Transmission Measurement
1820 !
1830 FOR I=1 TO Num_of_ports-1
1840 FOR J=I+1 TO Num_of_ports
1850 PRINT "Set THRU between Port "&VAL$(Port(I))&" and Port "& VAL$(Port(J))&". Then push [Enter] key."
1860 INPUT "",Buff$
1870 OUTPUT @Agte506x;":SENS"&Ch$&":CORR:COLL:THRU ";Port(I);"," ;Port(J)
1880 OUTPUT @Agte506x;"*OPC?"
1890 ENTER @Agte506x;Buff$
1900 OUTPUT @Agte506x;":SENS"&Ch$&":CORR:COLL:THRU ";Port(J);"," ;Port(I)
1910 OUTPUT @Agte506x;"*OPC?"
1920 ENTER @Agte506x;Buff$
1930 NEXT J
1940 NEXT I
1950 !
1960 ! Done
1970 !
```

```
1980 OUTPUT @Agte506x;":SENS"&Ch$&":CORR:COLL:SAVE"
1990 PRINT "Done"
2000 SUBEND
```

ECal

- Overview
- Sample Program in Excel VBA
- Sample Program in HT Basic

Other topics about Sample Programs

Overview

The sample program performs 1-port or 2-port calibration using ECal.

See Calibration for this programming.

Sample Program in Excel VBA

```
Sub ECal_Click()
    Dim defrm As Long         'Session to Default Resource Manager
    Dim vi As Long            'Session to instrument
    Dim Ch As String
    Dim CalKit As Integer
    Dim Port(4) As String
    Const TimeOutTime = 40000   'timeout time.

    Ch = Cells(5, 5)          'Select channel
    Port(1) = Cells(3, 6)      'Sets the select port 1.
    Port(2) = Cells(3, 7)      'Sets the select port 2.

    Call viOpenDefaultRM(defrm)     'Initializes the VISA system.
    Call viOpen(defrm, "GPIB0::17::INSTR", 0, 0, vi)      'Opens the session to the specified instrument.
    Call viSetAttribute(vi, VI_ATTR_TMO_VALUE, TimeOutTime)   'The state of an attribute for the
specified session.

    Call viVPrintf(vi, "*RST" & vbLf, 0)   'Presets the setting state of the ENA-L.
    Call viVPrintf(vi, "*CLS" & vbLf, 0)   'Clears the all status register.

    Select Case Cells(3, 5)
        Case "1 Port"
            Call ECal(vi, Ch, 1, Port)   'Perform 1-port calibration.
        Case "2 Port"
            Call ECal(vi, Ch, 2, Port)   'Perform full 2-port calibration.
    End Select
```

```
    Call viClose(vi)    'Closes the resource manager session.
    Call viClose(defrm)    'Breaks the communication and terminates the VISA system.

    End
End Sub
Sub ECal(vi As Long, Ch As String, NumPort As String, Port() As String)
    Dim Dummy As Variant
    Dim i As Integer, j As Integer

    Select Case NumPort
        Case 1
            MsgBox ("Connect Port " & Port(1) & ". then click [OK] button")  'Display the message box.
            Call viVPrintf(vi, ":SENS" & Ch & ":CORR:COLL:ECAL:SOLT" & NumPort & " " & Port(1) & vbLf,
0) 'Execute the 1-port calibration.
        Case 2
            MsgBox ("Connect Port " & Port(1) & " and Port " & Port(2) & ". then click [OK] button")   'Display
the message box.
            Call viVPrintf(vi, ":SENS" & Ch & ":CORR:COLL:ECAL:SOLT" & NumPort & " " & Port(1) & "," &
Port(2) & vbLf, 0) 'Execute the full 2-port calibration.
    End Select

    Call ErrorCheck(vi)      'Checking the error.

End Sub
Sub ErrorCheck(vi As Long)
    Dim err As String * 50, ErrNo As Variant, Response

    Call viVQueryf(vi, ":SYST:ERR?" & vbLf, "%t", err)    'Reads error message.
    ErrNo = Split(err, ",")   'Gets the error code.

    If Val(ErrNo(0)) <> 0 Then
        Response = MsgBox(CStr(ErrNo(1)), vbOKOnly)    'Display the message box.
    End If
End Sub
```

Sample Program in HT Basic (ecal.htb)

```
10 DIM File$[20],Ch$[9],Inp_char$[9]
20 INTEGER Cal_kit,Cal_type,Port(1:2)
30 !
```

```
40 ASSIGN @Agte506x TO 717
50 File$="Ex_4_2.sta"
60 Ch$="1"
70 !
80 CLEAR SCREEN
90 ON ERROR GOTO Type_select
100 Type_select: !
110 PRINT "## Calibration Type Selection ##"
120 PRINT " 1: Full 1 Port"
130 PRINT " 2: Full 2 Port"
160 PRINT ""
170 PRINT "Input 1 to 2"
180 INPUT "Input number? (1 to 2)",Inp_char$
190 Cal_type=IVAL(Inp_char$,10)
200 IF Cal_type<1 OR Cal_type>2 THEN Type_select
210 OFF ERROR
220 !
230 Select_port(Cal_type,Port(*))
240 Ecal(@Agte506x,Ch$,Cal_type,Port(*))
250 !
260 OUTPUT @Agte506x;":MMEM:STOR:STYP CST"
270 OUTPUT @Agte506x;":MMEM:STOR """&File$&""""
280 END
290 !=============================================
300 ! Port Selection Function
310 !=============================================
320 SUB Select_port(INTEGER Num_of_ports,INTEGER Port(*))
330 DIM Inp_char$[9]
340 !
350 CLEAR SCREEN
360 IF Num_of_ports=2 THEN
370 Port(1)=1
380 Port(2)=2
410 ELSE
420 PRINT "## Test Ports Selection ##"
430 ON ERROR GOTO Port_select
450 PRINT "Port(1):";
```

```
460 Port_select: !
470 INPUT "Number?",Inp_char$
480 Port(1)=IVAL(Inp_char$,10)
490 IF Port(1)<1 OR Port(1)>4 THEN Port_select
530 PRINT Port(1)
550 OFF ERROR
560 END IF
570 SUBEND
580 !============================================
590 ! Electronic Calibration Function
600 !============================================
610 SUB Ecal(@Agte506x,Ch$,INTEGER Num_of_ports,INTEGER Port(*))
620 DIM Buff$[9],Err_msg$[100]
630 INTEGER Err_no,Port1
640 !
650 PRINT "## Full "&VAL$(Num_of_ports)&" Port ECal ##"
660 !
670 OUTPUT @Agte506x;"*CLS"
680 SELECT Num_of_ports
690 CASE 1
700 PRINT "Connect Port "&VAL$(Port(1))&" to ECal Module."
710 PRINT "Then push [Enter] key."
720 INPUT "",Buff$
730 OUTPUT @Agte506x;":SENS"&Ch$&":CORR:COLL:ECAL:SOLT1 ";Port(1)
740 CASE 2
750 PRINT "Connect Port "&VAL$(Port(1));
760 PRINT " and Port "&VAL$(Port(2))&" to ECal Module."
770 PRINT "Then push [Enter] key."
780 INPUT "",Buff$
790 OUTPUT @Agte506x;":SENS"&Ch$&":CORR:COLL:ECAL:SOLT2 ";Port(1); ",";Port(2)
920 END SELECT
930 PRINT "Executing ..."
940 OUTPUT @Agte506x;":SYST:ERR?"
950 ENTER @Agte506x;Err_no,Err_msg$
960 IF Err_no<>0 THEN
970 PRINT "Error occurred!!"
980 PRINT " No:";Err_no,"Description: "&Err_msg$
```

```
990 PRINT "ECAL INTERRUPT!!"
1000 ELSE
1010 PRINT "Done"
1020 END IF
1030 SUBEND
```

Reading / Writing Error Coefficient

- Overview
- Sample Program in Excel VBA
- Sample Program in HT Basic

Other topics about Sample Programs

Overview

This sample program reads/writes the error coefficient.

This program sets measurement conditions and perform full 2-port calibration, preset the E5061B with the read error coefficient to be written, and then again read the error coefficient.

NOTE The error coefficient read from the E5061B is displayed in a graph.

Sample Program in Excel VBA

```
Sub Err_Term_Click()
    Dim defrm As Long        'Session to Default Resource Manager
    Dim vi As Long           'Session to instrument
    Dim Ch As String
    Dim CalKit As Integer
    Dim Port(2) As String
    Dim Result As String * 10
    Dim tNop As Long
    Dim Respons As String
    Dim Stimulus As String
    Dim ErrTerm As String

    Const TimeOutTime = 40000   'timeout time.
    Const Cal85032F = 4         'cal kit number

    Ch = Cells(2, 6)            'Select channel
    Port(1) = Cells(4, 6)       'Sets the select port 1.
    Port(2) = Cells(5, 6)       'Sets the select port 2.
    Respons = Cells(6, 6)       'Sets the respons port.
    Stimulus = Cells(7, 6)      'Sets the stimulus port.
    ErrTerm = Cells(8, 6)       'Sets the error term.

    CalKit = Cal85032F          'Set cal kit (85032F)
```

```
    Call viOpenDefaultRM(defrm)    'Initializes the VISA system.
    Call viOpen(defrm, "GPIB0::17::INSTR", 0, 0, vi)    'Opens the session to the specified instrument.
    Call viSetAttribute(vi, VI_ATTR_TMO_VALUE, TimeOutTime)      'The state of an attribute for the
specified session.

    Call viVPrintf(vi, "*RST" & vbLf, 0)   'Presets the setting state of the ENA.
    Call viVPrintf(vi, "*CLS" & vbLf, 0)   'Clears the all status register.

    Call viVPrintf(vi, ":SENS" & Ch & ":CORR:COLL:CKIT " & CalKit & vbLf, 0)  'Select the calibration kit.

    Call Set_sgm_tbl(vi, Ch)     'Configures the segment table.

    Select Case Cells(3, 6)    'Sets the read/write.
       Case "Read"
          Call Cal_Slot(vi, Ch, 2, Port)  'Full 2-Port Calibration.
       Case "Write"
          Call viVPrintf(vi, ":SENS" & Ch & ":CORR:COEF:METH:SOLT2 1,2" & vbLf, 0)   'Sets the
calibration type to the full 2-port calibration.
    End Select

    Call viVPrintf(vi, ":SENS" & Ch & ":SEGM:SWE:POIN?" & vbLf, 0)  'Reads out the total number of the
measurement points of all segments.
    Call viVScanf(vi, "%t", Result)

    Call Exec_Error_Term(vi, Ch, Val(Result), ErrTerm, Respons, Stimulus)    'Reads the error coefficient.

    Call viClose(vi)      'Closes the resource manager session.
    Call viClose(defrm)      'Breaks the communication and terminates the VISA system.

    End
End Sub
Sub Exec_Error_Term(vi As Long, Ch As String, Nop As Long, ErrTerm As String, Respons As String,
Stimulus As String)

    Dim Error_Term_Data As Variant
    Dim Freq_Data As Variant
    Dim i As Integer, j As Integer
```

```
    Dim SelMode As String
    Dim Result As String * 10000
    Dim RealData As Double
    Dim ImagData As Double
    Dim FreqData As Double

    ReDim Error_Term_Data(Nop * 2) As String        'Defines the stock variables for the error coefficient
as needed for NOP.
    ReDim Freq_Data(Nop) As String               'Defines the stock variables for the frequency values.

    SelMode = Cells(3, 6)   'Reads the read/write mode.

    Select Case SelMode
        Case "Read"                              'Reads the error coefficient from the ena.
            Call viVPrintf(vi, ":SENS" & Ch & ":CORR:COEF? " & ErrTerm & "," & Respons & "," & Stimulus &
vbLf, 0)  'Read the calibration coefficient data.
            Call viVScanf(vi, "%t", Result)
            Error_Term_Data = Split(Result, ",")    'Splits the read data by comma.

            Freq_Data = Make_Freq(vi, Nop)       'Calculates the frequency values.

            For i = 0 To Nop - 1
                RealData = CDbl(Error_Term_Data(i * 2))       'Reads the real data from error coefficient items.
                ImagData = CDbl(Error_Term_Data(i * 2 + 1))  'Reads the imag data from error coefficient
items.
                FreqData = CDbl(Freq_Data(i + 1))              'Reads the frequency values.
                Cells(10 + i, 2) = RealData                   'Displays the real data to the excel sheet.
                Cells(10 + i, 3) = ImagData                   'Displays the imag data to the excel sheet.
                Cells(10 + i, 1) = FreqData                   'Displays the frequency values to the excel sheet.
            Next i

        Case "Write"     'Write the error coefficient to the ena.
            Error_Term_Data = ErrTerm & "," & Respons & "," & Stimulus     'Sets the command parameter.
            For i = 0 To Nop - 1
                RealData = Cells(10 + i, 2)                   'Retrieves the real data from the excel sheet.
                ImagData = Cells(10 + i, 3)                   'Retrieves the imag data from the excel sheet.
                Error_Term_Data = Error_Term_Data & "," & RealData & "," & ImagData        'Sets the
command parameter.
```

```
        Next i

        Call viVPrintf(vi, ":SENS" & Ch & ":CORR:COEF " & Error_Term_Data & vbLf, 0)    'Write the
calibration coefficient data.
        Call viVPrintf(vi, ":SENS" & Ch & ":CORR:COEF:SAVE" & vbLf, 0)         'Calculates the
calibration coefficients.

    End Select
End Sub
Function Make_Freq(vi As Long, tPoint As Long) As Variant
    Dim start_freq As Double
    Dim stop_freq As Double
    Dim Nop As Integer
    Dim fStep As Double
    Dim fPoint As Double
    Dim freq_arry() As Variant
    Dim MeasPoint As Integer

    Const SegmentCnt = 2        'number of segment table.

    ReDim freq_arry(tPoint) As Variant

    MeasPoint = 1

    For j = 1 To SegmentCnt
        start_freq = Cells(3 + j - 1, 9)                    'Sets the start frequency of segment table.
        stop_freq = Cells(3 + j - 1, 10)                    'Sets the stop frequency of segment table.
        Nop = Cells(3 + j - 1, 13)                          'Sets the nop of segment table.
        fStep = (stop_freq - start_freq) / (Nop - 1)        'Calculate the frequency step.

        fPoint = start_freq                                 'Sets the frequency start point.
        For i = 1 To Nop
            freq_arry(MeasPoint) = fPoint                   'Sets the frequency value.
            fPoint = fPoint + fStep                         'Calculate the frequency points.
            MeasPoint = MeasPoint + 1                       'Add to measurement points.
        Next i
    Next j
```

```
    Make_Freq = freq_arry                                   'Sets the frequency data array.

End Function

Sub Set_sgm_tbl(vi As Long, Ch As String)
    Dim Star1(2) As Double, Stop1(2) As Double, Pow1(2) As Double, If_bw1(2) As Double
    Dim Segm As Integer, Nop1(2) As Integer, Num_of_tr1 As Integer
    Dim i As Integer

    Segm = 2
    Star1(1) = Cells(3, 9)          'Sets the start frequency of segment 1 table.
    Stop1(1) = Cells(3, 10)          'Sets the stop frequency of segment 1 table.
    Pow1(1) = Cells(3, 11)           'Sets the power of segment 1 table.
    If_bw1(1) = Cells(3, 12)         'Sets the ifbw of segment 1 table.
    Nop1(1) = Cells(3, 13)           'Sets the nop of segment 1 table.
    Star1(2) = Cells(4, 9)          'Sets the start frequency of segment 2 table.
    Stop1(2) = Cells(4, 10)          'Sets the stop frequency of segment 2 table.
    Pow1(2) = Cells(4, 11)           'Sets the power of segment 2 table.
    If_bw1(2) = Cells(4, 12)         'Sets the ifbw of segment 2 table.
    Nop1(2) = Cells(4, 13)           'Sets the nop of segment 2 table.

    Call viVPrintf(vi, ":SENS" & Ch & ":SWE:TYPE SEGM" & vbLf, 0)   'Sets sweep type to segment.
    Call viVPrintf(vi, ":SENS" & Ch & ":SEGM:DATA 5,0,1,1,0,0," & Segm & ",", 0)      'Sets the header of
segment table.
    Call viVPrintf(vi, Star1(1) & "," & Stop1(1) & "," & Nop1(1) & "," & If_bw1(1) & "," & Pow1(1) & ",", 0)
 'Sets the 1st parameter.
    Call viVPrintf(vi, Star1(2) & "," & Stop1(2) & "," & Nop1(2) & "," & If_bw1(2) & "," & Pow1(2) & vbLf, 0)
 'Sets the 2nd parameter.

    Call ErrorCheck(vi)   'Checking the error.

End Sub
Sub Cal_Slot(vi As Long, Ch As String, NumPort As String, Port() As String)
    Dim Dummy
    Dim i As Integer, j As Integer

    Select Case NumPort
        Case 1
```

```
        Call viVPrintf(vi, ":SENS" & Ch & ":CORR:COLL:METH:SOLT" & NumPort & " " & Port(1) & vbLf,
0)   'Set the 1-port calibration type.
      Case 2
        Call viVPrintf(vi, ":SENS" & Ch & ":CORR:COLL:METH:SOLT" & NumPort & " " & Port(1) & "," &
Port(2) & vbLf, 0)  'Set the full 2-port calibration type.
    End Select
    'Reflection
    For i = 1 To NumPort
      MsgBox ("Set Open to Port " & Port(i) & ". then click [OK] button")    'Display the message box.
      Call viVPrintf(vi, ":SENS" & Ch & ":CORR:COLL:OPEN " & Port(i) & vbLf, 0)      'Measurement the
OPEN calibration.
      Call viVQueryf(vi, "*OPC?" & vbLf, "%t", Dummy)     'Reads the *OPC? result.

      MsgBox ("Set Short to Port " & Port(i) & ". then click [OK] button")  'Display the message box.
      Call viVPrintf(vi, ":SENS" & Ch & ":CORR:COLL:SHORT " & Port(i) & vbLf, 0)  'Measurement the
SHORT calibration.
      Call viVQueryf(vi, "*OPC?" & vbLf, "%t", Dummy)  'Reads the *OPC? result.

      MsgBox ("Set Load to Port " & Port(i) & ". then click [OK] button")   'Display the message box.
      Call viVPrintf(vi, ":SENS" & Ch & ":CORR:COLL:LOAD " & Port(i) & vbLf, 0)   'Measurement the
LOAD calibration.
      Call viVQueryf(vi, "*OPC?" & vbLf, "%t", Dummy)   'Reads the *OPC? result.
    Next i
    'Transmission
    For i = 1 To NumPort - 1
      For j = i + 1 To NumPort
        MsgBox ("Set Thru to Port " & Port(i) & "&" & Port(j) & ". then click [OK] button")  'Display the
message box.
        Call viVPrintf(vi, ":SENS" & Ch & ":CORR:COLL:THRU " & Port(i) & "," & Port(j) & vbLf, 0)
  'Measurement the THRU calibration.
        Call viVQueryf(vi, "*OPC?" & vbLf, "%t", Dummy)   'Reads the *OPC? result.
        Call viVPrintf(vi, ":SENS" & Ch & ":CORR:COLL:THRU " & Port(j) & "," & Port(i) & vbLf, 0)
  'Measurement the THRU calibration.
        Call viVQueryf(vi, "*OPC?" & vbLf, "%t", Dummy)      'Reads the *OPC? result.
      Next j
    Next i

    Call viVPrintf(vi, ":SENS" & Ch & ":CORR:COLL:SAVE" & vbLf, 0)    'Calculating the calibration
coefficients.
```

```
    Call ErrorCheck(vi)   'Checking the error.

End Sub
Sub ErrorCheck(vi As Long)
    Dim err As String * 50, ErrNo As Variant, Response

    Call viVQueryf(vi, ":SYST:ERR?" & vbLf, "%t", err)   'Reads error message.
    ErrNo = Split(err, ",")      'Gets the error code.

    If Val(ErrNo(0)) <> 0 Then
        Response = MsgBox(CStr(ErrNo(1)), vbOKOnly)    'Display the message box.
    End If
End Sub
```

Sample Program in HT Basic (ErrTerm.htb)

```
2000 Main:!
2010 INTEGER Agte506x,Ii,Nop
2020 INTEGER Respons,Stimulas
2030 INTEGER Port(1:2)
2040 REAL Stok(12,1:5000)
2050 REAL Stok2(12,1:5000)
2060 REAL Stok3(12,1:5000)
2070 DIM Ch$[10],Wk$[128]
2080 !
2090 ! PC's Monitor Clear
2100 CLEAR SCREEN
2110 GINIT
2130 !
2140 ! Set ENA++'s Addr
2150 Agte506x=717
2160 !
2170 Ch$="1"
2180 !
2190 ! Set ENA++'s I/O Path
2200 ASSIGN @Agte506x TO Agte506x
2210 !
2220 ON TIMEOUT SC(@Agte506x),15 RECOVER Tout
2230 !
```

```
2240 ! Set Start Port and End Port
2250 Port(1)=1
2260 Port(2)=2
2270 !
2280 ! Setup Segment Table
2290 CALL Set_sgm_tbl(@Agte506x)
2300 !
2310 ! Select Cal Kit
2320 CALL Select_cal_kit(@Agte506x,Ch$)
2330 !
2340 ! Execute Full-2Port Calibration
2350 CALL Cal_solt(@Agte506x,Ch$,2,Port(*))
2360 !
2370 ! Get All Segment's Points
2380 CALL Get_nop(@Agte506x,Nop,Ch$)
2390 !
2400 REDIM Stok(12,1:Nop*2)
2410 REDIM Stok2(12,1:Nop*2)
2420 REDIM Stok3(12,1:Nop*2)
2430 !
2440 CALL Exec_error_term(@Agte506x,"READ","ES",Ch$,1,Nop,1,1,Stok(*))
2450 CALL Exec_error_term(@Agte506x,"READ","ES",Ch$,2,Nop,2,2,Stok(*))
2460 CALL Exec_error_term(@Agte506x,"READ","ER",Ch$,3,Nop,1,1,Stok(*))
2470 CALL Exec_error_term(@Agte506x,"READ","ER",Ch$,4,Nop,2,2,Stok(*))
2480 CALL Exec_error_term(@Agte506x,"READ","ED",Ch$,5,Nop,1,1,Stok(*))
2490 CALL Exec_error_term(@Agte506x,"READ","ED",Ch$,6,Nop,2,2,Stok(*))
2500 !
2510 CALL Exec_error_term(@Agte506x,"READ","EL",Ch$,7,Nop,1,2,Stok(*))
2520 CALL Exec_error_term(@Agte506x,"READ","EL",Ch$,8,Nop,2,1,Stok(*))
2530 CALL Exec_error_term(@Agte506x,"READ","ET",Ch$,9,Nop,1,2,Stok(*))
2540 CALL Exec_error_term(@Agte506x,"READ","ET",Ch$,10,Nop,2,1,Stok(*))
2550 !
2560 CLEAR SCREEN
2570 PRINT "Push [Preset] - OK of ENA. Then push [Enter] key."
2580 INPUT "",Wk$
2590 !
2600 CALL Set_sgm_tbl(@Agte506x)
```

```
2610 !
2620 OUTPUT @Agte506x;":SENS"&Ch$&":CORR:COEF:METH:SOLT2 ";Port(1);",";Port(2)
2630 !
2640 CALL Exec_error_term(@Agte506x,"WRITE","ES",Ch$,1,Nop,1,1,Stok(*))
2650 CALL Exec_error_term(@Agte506x,"WRITE","ES",Ch$,2,Nop,2,2,Stok(*))
2660 CALL Exec_error_term(@Agte506x,"WRITE","ER",Ch$,3,Nop,1,1,Stok(*))
2670 CALL Exec_error_term(@Agte506x,"WRITE","ER",Ch$,4,Nop,2,2,Stok(*))
2680 CALL Exec_error_term(@Agte506x,"WRITE","ED",Ch$,5,Nop,1,1,Stok(*))
2690 CALL Exec_error_term(@Agte506x,"WRITE","ED",Ch$,6,Nop,2,2,Stok(*))
2700 !
2710 CALL Exec_error_term(@Agte506x,"WRITE","EL",Ch$,7,Nop,1,2,Stok(*))
2720 CALL Exec_error_term(@Agte506x,"WRITE","EL",Ch$,8,Nop,2,1,Stok(*))
2730 CALL Exec_error_term(@Agte506x,"WRITE","ET",Ch$,9,Nop,1,2,Stok(*))
2740 CALL Exec_error_term(@Agte506x,"WRITE","ET",Ch$,10,Nop,2,1,Stok(*))
2750 !
2760 OUTPUT @Agte506x;":SENS"&Ch$&":CORR:COEF:SAVE"
2770 !
2780 CALL Exec_error_term(@Agte506x,"READ","ES",Ch$,1,Nop,1,1,Stok2(*))
2790 CALL Exec_error_term(@Agte506x,"READ","ES",Ch$,2,Nop,2,2,Stok2(*))
2800 CALL Exec_error_term(@Agte506x,"READ","ER",Ch$,3,Nop,1,1,Stok2(*))
2810 CALL Exec_error_term(@Agte506x,"READ","ER",Ch$,4,Nop,2,2,Stok2(*))
2820 CALL Exec_error_term(@Agte506x,"READ","ED",Ch$,5,Nop,1,1,Stok2(*))
2830 CALL Exec_error_term(@Agte506x,"READ","ED",Ch$,6,Nop,2,2,Stok2(*))
2840 !
2850 CALL Exec_error_term(@Agte506x,"READ","EL",Ch$,7,Nop,1,2,Stok2(*))
2860 CALL Exec_error_term(@Agte506x,"READ","EL",Ch$,8,Nop,2,1,Stok2(*))
2870 CALL Exec_error_term(@Agte506x,"READ","ET",Ch$,9,Nop,1,2,Stok2(*))
2880 CALL Exec_error_term(@Agte506x,"READ","ET",Ch$,10,Nop,2,1,Stok2(*))
2890 !
2900 ASSIGN @Agte506x TO *
2910 !
2920 DISP CHR$(139)&" Done ..."&CHR$(136)
2930 STOP
2940 !
2950 Tout: OFF TIMEOUT SC(@Agte506x)
2960 !
2970 ASSIGN @Agte506x TO *
```

```
2980 !
2990 PRINT CHR$(137)&" ENA Timeout ..."&CHR$(136)
3000 END
3010!
3020 Set_sgm_tbl: SUB Set_sgm_tbl(@Agte506x)
3030 REAL Star1(1:2),Stop1(1:2),Pow1(1:2)
3040 INTEGER Segm,Nop1(1:2),Num_of_tr1
3050 INTEGER I
3060 !
3070 CLEAR SCREEN
3080 DISP CHR$(138)&" Wait ..."&CHR$(136)
3090 !
3100 Segm=2 ! Number of Segment Ch.1 : 2
3110 Star1(1)=3.E+6 ! Start Frequency Ch.1 Segm.1: 3.0 MHz
3120 Star1(2)=5.0E+7 ! Segm.2: 50.0 MHz
3130 Stop1(1)=1.0E+7 ! Stop Frequency Ch.1 Segm.1: 10.0 MHz
3140 Stop1(2)=8.E+9 ! Segm.2: 8.0 GHz
3150 Nop1(1)=2 ! Number Ch.1 Segm.1: 2
3160 Nop1(2)=170 ! of Points Segm.2: 170
3170 If_bw1(1)=7.0E+4 ! IF Bandwidth Ch.1 Segm.1: 70 kHz
3180 If_bw1(2)=7.0E+4 ! Segm.2: 70 kHz
3190 Pow1(1)=0 ! Power Ch.1 Segm.1: 0 dBm
3200 Pow1(2)=0 ! Segm.2: 0 dBm
3210 !
3220 OUTPUT @Agte506x;":SYST:PRES"
3230 !
3240 WAIT 5
3250 !
3260 ! Channel 1
3270 !
3280 OUTPUT @Agte506x;":SENS1:SWE:TYPE SEGM"
3290 OUTPUT @Agte506x;":SENS1:SEGM:DATA 5,0,1,1,0,0,";Segm;",";
3300 FOR I=1 TO Segm-1
3310 OUTPUT @Agte506x;Star1(I);",";Stop1(I);",";Nop1(I);",";If_bw1(I);",";Pow1(I);",";
3320 NEXT I
3330 OUTPUT
@Agte506x;Star1(Segm);",";Stop1(Segm);",";Nop1(Segm);",";If_bw1(Segm);",";Pow1(Segm)
```

```
3340 !
3350 OUTPUT @Agte506x;":CALC1:PAR:COUN ";Num_of_tr1
3360 FOR I=1 TO Num_of_tr1
3370 OUTPUT @Agte506x;":CALC1:PAR"&VAL$(I)&":SEL"
3380 NEXT I
3390 SUBEND
3400!
3410 Select_cal_kit: SUB Select_cal_kit(@Agte506x,Ch$)
3420 !=============================================
3430 ! Calibration Kit Selection Function
3440 !=============================================
3450 !
3460 DIM Cal_kit_lbl$(1:10)[20],Inp_char$[9]
3470 DIM Msg$[80],Wk$[10]
3480 INTEGER Cal_kit,I,Noc
3490 !
3500 ! PC's Monitor Clear
3510 CLEAR SCREEN
3520 !
3530 ! Number of Cal Kid
3540 Noc=10
3550 !
3560 FOR I=1 TO Noc
3570 OUTPUT @Agte506x;":SENS1:CORR:COLL:CKIT ";I
3580 OUTPUT @Agte506x;":SENS1:CORR:COLL:CKIT:LAB?"
3590 ENTER @Agte506x;Cal_kit_lbl$(I)
3600 NEXT I
3610 ON ERROR GOTO Kit_select
3620 !
3630 PRINT "## Calibration Kit Selection ##"
3640 FOR I=1 TO Noc
3650 PRINT USING "X,2D,A,X,20A";I,":",Cal_kit_lbl$(I)
3660 NEXT I
3670 PRINT ""
3680 PRINT "Input 1 to "&VAL$(Noc)
3690 !
3700 Msg$="Input number? (1 to "&VAL$(Noc)&") "
```

```
3710 LOOP
3720 LOOP
3730 DISP Msg$;
3740 INPUT Inp_char$
3750 Cal_kit=IVAL(Inp_char$,10)
3760 EXIT IF 1<=Cal_kit AND Cal_kit<=Noc
3770 Kit_select:!
3780 BEEP
3790 END LOOP
3800 !
3810 Wk$=""
3820 PRINT TABXY(1,Cal_kit+1);
3830 PRINT USING "X,B,2D,A,X,20A,B";139,Cal_kit,":",Cal_kit_lbl$(Cal_kit),136
3840 INPUT "Sure ? [Y/N]",Wk$
3850 EXIT IF (UPC$(Wk$)="Y")
3860 PRINT TABXY(1,Cal_kit+1);
3870 PRINT USING "X,2D,A,X,20A";Cal_kit,":",Cal_kit_lbl$(Cal_kit)
3880 BEEP
3890 BEEP
3900 END LOOP
3910 OFF ERROR
3920 !
3930 OUTPUT @Agte506x;":SENS"&Ch$&":CORR:COLL:CKIT ";Cal_kit
3940 SUBEND
3950!
3960 Cal_solt: SUB Cal_solt(@Agte506x,Ch$,INTEGER Num_of_ports,INTEGER Port(*))
3970 !============================================
3980 ! Full n Port Calibration Function
3990 !============================================
4000 !
4010 DIM Buff$[9]
4020 INTEGER I,J
4030 !
4040 ! PC's Monitor Clear
4050 CLEAR SCREEN
4060 !
4070 PRINT "## Full "&VAL$(Num_of_ports)&" Port Calibration ##"
```

```
4080 !
4090 ! Calibration Type Selection
4100 !
4110 OUTPUT @Agte506x;":SENS"&Ch$&":CORR:COLL:METH:SOLT"&VAL$(Num_of_ports)&" ";
4120 FOR I=1 TO Num_of_ports-1
4130 OUTPUT @Agte506x;Port(I);",";
4140 NEXT I
4150 OUTPUT @Agte506x;Port(Num_of_ports)
4160 !
4170 ! Reflection Measurement
4180 !
4190 FOR I=1 TO Num_of_ports
4200 PRINT "Set OPEN to Port "&VAL$(Port(I))&". Then push [Enter] key."
4210 INPUT "",Buff$
4220 OUTPUT @Agte506x;":SENS"&Ch$&":CORR:COLL:OPEN ";Port(I)
4230 OUTPUT @Agte506x;"*OPC?"
4240 ENTER @Agte506x;Buff$
4250 PRINT "Set SHORT to Port "&VAL$(Port(I))&". Then push [Enter] key."
4260 INPUT "",Buff$
4270 OUTPUT @Agte506x;":SENS"&Ch$&":CORR:COLL:SHOR ";Port(I)
4280 OUTPUT @Agte506x;"*OPC?"
4290 ENTER @Agte506x;Buff$
4300 PRINT "Set LOAD to Port "&VAL$(Port(I))&". Then push [Enter] key."
4310 INPUT "",Buff$
4320 OUTPUT @Agte506x;":SENS"&Ch$&":CORR:COLL:LOAD ";Port(I)
4330 OUTPUT @Agte506x;"*OPC?"
4340 ENTER @Agte506x;Buff$
4350 NEXT I
4360 !
4370 ! Transmission Measurement
4380 !
4390 FOR I=1 TO Num_of_ports-1
4400 FOR J=I+1 TO Num_of_ports
4410 PRINT "Set THRU between Port "&VAL$(Port(I))&" and Port "&VAL$(Port(J))&". Then push [Enter] key."
4420 INPUT "",Buff$
4430 OUTPUT @Agte506x;":SENS"&Ch$&":CORR:COLL:THRU ";Port(I);",";Port(J)
```

```
4440 OUTPUT @Agte506x;"*OPC?"
4450 ENTER @Agte506x;Buff$
4460 OUTPUT @Agte506x;":SENS"&Ch$&":CORR:COLL:THRU ";Port(J);",";Port(I)
4470 OUTPUT @Agte506x;"*OPC?"
4480 ENTER @Agte506x;Buff$
4490 NEXT J
4500 NEXT I
4510 !
4520 ! Done
4530 !
4540 OUTPUT @Agte506x;":SENS"&Ch$&":CORR:COLL:SAVE"
4550 PRINT "Done"
4560 SUBEND
4570!
4580 Get_nop: SUB Get_nop(@Agte506x,INTEGER Nop,Ch$)
4590 ! Get All Segment's Points
4600 OUTPUT @Agte506x;":SENS"&Ch$&":SEGM:SWE:POIN?"
4610 ENTER @Agte506x;Nop
4620 SUBEND
4630 Exec_error_term: SUB Exec_error_term(@Agte506x,Rw$,Id$,Ch$,INTEGER
Idx,Nop,Respons,Stimulas,REAL Stok(*))
4640 INTEGER Ii
4650 REAL Error_term_data(1:5000)
4660 !
4670 DISP CHR$(138)&" Wait ..."&CHR$(136)
4680 !
4690 REDIM Error_term_data(1:Nop*2)
4700 !
4710 SELECT Rw$
4720 CASE "WRITE"
4730 FOR Ii=1 TO Nop
4740 Error_term_data(2*Ii-1)=Stok(Idx,2*Ii-1)
4750 Error_term_data(2*Ii)=Stok(Idx,2*Ii)
4760 NEXT Ii
4770 !
4780 OUTPUT @Agte506x;":SENS"&Ch$&":CORR:COEF
"&Id$&",";Respons;",";Stimulas;",";Error_term_data(*)
4790 !
```

```
4800 CASE "READ"
4810 FOR Ii=1 TO Nop
4820 Error_term_data(2*Ii-1)=-999
4830 Error_term_data(2*Ii)=-999
4840 NEXT Ii
4850 !
4860 OUTPUT @Agte506x;":SENS"&Ch$&":CORR:COEF? "&Id$&",";Respons;",";Stimulas
4870 ENTER @Agte506x;Error_term_data(*)
4880 !
4890 CALL Data_plot(Id$,Respons,Stimulas,Nop,Error_term_data(*))
4900 !
4910 FOR Ii=1 TO Nop
4920 Stok(Idx,2*Ii-1)=Error_term_data(2*Ii-1)
4930 Stok(Idx,2*Ii)=Error_term_data(2*Ii)
4940 NEXT Ii
4950 !
4960 END SELECT
4970 SUBEND
4980!
4990 Data_plot: SUB Data_plot(Error_term$,INTEGER Respons,Stimulas,Nop,REAL
Error_term_data(*))
5000 INTEGER Ii,Pen(1:2)
5010 REAL Y_minmax(1:2)
5020 DIM Wk$[20]
5030 !
5040 CLEAR SCREEN
5050 GINIT
5060 GCLEAR
5070 !
5080 Pen(1)=3
5090 Pen(2)=4
5100 !
5110 ! Get Min Value and Max Value from all data
5120 Y_minmax(1)=MIN(Error_term_data(*))
5130 Y_minmax(2)=MAX(Error_term_data(*))
5150 !
5160 IF (Y_minmax(1)=Y_minmax(2)) AND (Y_minmax(1)=0) THEN
```

```
5170 Y_minmax(1)=1
5180 Y_minmax(2)=-1
5190 ELSE
5200 IF (Y_minmax(1)=Y_minmax(2)) THEN
5210 Y_minmax(1)=Y_minmax(1)*.5
5220 Y_minmax(2)=Y_minmax(2)*1.5
5230 END IF
5240 END IF
5250 !
5260 VIEWPORT 25*RATIO,80*RATIO,40,90
5270 WINDOW 1,Nop,Y_minmax(1),Y_minmax(2)
5280 FRAME
5290 !
5300 VIEWPORT 80*RATIO,100*RATIO,40,90
5310 WINDOW 0,2,0,2
5320 PEN Pen(1)
5330 CSIZE 2.5
5340 LORG 2
5350 MOVE .2,1.5
5360 DRAW .4,1.5
5370 MOVE .5,1.5
5380 PEN 1
5390 LABEL ":Real Value"
5400 !
5410 PEN Pen(2)
5420 MOVE .2,1
5430 DRAW .4,1
5440 MOVE .5,1
5450 PEN 1
5460 LABEL ":Image Value"
5470 !
5480 VIEWPORT 25*RATIO,80*RATIO,90,100
5490 WINDOW 0,2,0,2
5500 CSIZE 3
5510 LORG 5
5520 MOVE 1,1.2
5530 LABEL "Error Term:"&Error_term$
```

```
5540 !
5550 MOVE 1,.5
5560 LABEL "Respons Port:"&VAL$(Respons)&" Stimulas Port:"&VAL$(Stimulas)
5570 !
5580 VIEWPORT 0,25*RATIO,40,90
5590 WINDOW 0,2,0,2
5600 CLIP -10,10,-10,10
5610 LORG 8
5620 CSIZE 3
5630 !
5640 MOVE 1.9,0
5650 LABEL VAL$(Y_minmax(1))
5660 MOVE 1.9,2
5670 LABEL VAL$(Y_minmax(2))
5680 !
5690 VIEWPORT 25*RATIO,80*RATIO,30,40
5700 WINDOW 0,2,0,2
5710 CLIP -10,10,-10,10
5720 LORG 5
5730 MOVE 0,1.5
5740 LABEL VAL$(1)
5750 MOVE 2,1.5
5760 LABEL VAL$(Nop)
5770 !
5780 VIEWPORT 25*RATIO,80*RATIO,40,90
5790 WINDOW 1,Nop,Y_minmax(1),Y_minmax(2)
5800 FOR Ii=2 TO Nop
5820 PEN Pen(1)
5830 MOVE Ii-1,Error_term_data(2*(Ii-1)-1)
5840 DRAW Ii,Error_term_data(2*Ii-1)
5860 !
5870 PEN Pen(2)
5880 MOVE Ii-1,Error_term_data(2*(Ii-1))
5890 DRAW Ii,Error_term_data(2*Ii)
5900 NEXT Ii
5910 !
5920 PEN 1
```

```
5930 BEEP
5940 INPUT "Cont:push [Enter] key",Wk$
5950 SUBEND
5960!
```

Waiting for Trigger (OPC?)

- Overview
- Sample Program in Excel VBA
- Sample Program in HT Basic

Other topics about Sample Programs

Overview

This sample program demonstrates how to use the :TRIG:SING command to wait until the measurement cycle is completed.

NOTE This sample program correctly runs when the maximum number of channels/traces is set to 4 channels/4 traces.

The sample program uses the :TRIG:SING command to start a sweep (measurement) cycle, uses the *OPC command to wait until the measurement cycle is completed, then prints a message and exits.

See Waiting for the End of Measurement for this programming.

Sample Program in Excel VBA

```
Sub trg_sing_Click()
    Dim defrm As Long
    Dim vi As Long
    Dim ContMode(4) As String
    Dim Result As String * 10
    Dim i As Integer
    Const TimeOutTime = 100000 ' TimeOut time should be greater than the measurement time.
    '
    ' Assign a GPIB address to the I/O pass.
    Call viOpenDefaultRM(defrm)
    Call viOpen(defrm, "GPIB0::17::INSTR", 0, 0, vi)
    Call viSetAttribute(vi, VI_ATTR_TMO_VALUE, TimeOutTime)
    '
    ' Store the settings of continuous initiation mode for each channel
    ' (on for channels 1 and 2; off for channels 3 and 4)
    ' into the array variable ContMode().
    ContMode(1) = "ON"
    ContMode(2) = "ON"
    ContMode(3) = "OFF"
    ContMode(4) = "OFF"
```

```
    '
    ' Turn on or off continuous initiation mode for each channel
    ' depending on the value of ContMode(*).
    Call viVPrintf(vi, ":DISP:SPL D12_34" & vbLf, 0)
    Call viVPrintf(vi, ":SENS1:SWE:TIME:AUTO OFF" & vbLf, 0)
    Call viVPrintf(vi, ":SENS1:SWE:TIME 5" & vbLf, 0)
    Call viVPrintf(vi, ":SENS2:SWE:TIME:AUTO OFF" & vbLf, 0)
    Call viVPrintf(vi, ":SENS2:SWE:TIME 3" & vbLf, 0)
    Call viVPrintf(vi, ":SENS3:SWE:TIME:AUTO OFF" & vbLf, 0)
    Call viVPrintf(vi, ":SENS3:SWE:TIME 1" & vbLf, 0)
    Call viVPrintf(vi, ":SENS4:SWE:TIME:AUTO OFF" & vbLf, 0)
    Call viVPrintf(vi, ":SENS4:SWE:TIME 3" & vbLf, 0)
    For i = 1 To 4
      Call viVPrintf(vi, ":INIT" & CStr(i) & ":CONT " & ContMode(i) & vbLf, 0)
    Next i
    ' Set the trigger source to Bus Trigger.
    Call viVPrintf(vi, ":TRIG:SOUR BUS" & vbLf, 0)
    '
    ' Trigger the instrument to start a sweep cycle.
    Call viVPrintf(vi, ":TRIG:SING" & vbLf, 0)
    '
    ' Execute the *OPC? command and wait until the command
    ' returns 1 (i.e., the measurement cycle is completed).
    Call viVPrintf(vi, "*OPC?" & vbLf, 0)
    Call viVScanf(vi, "%t", Result)
    '
    ' Display a measurement completion message.
    Stat = MsgBox("Measurement complete", vbOKOnly)
    Call viClose(vi)
    Call viClose(defrm)
  End Sub
```

Sample Program in HT Basic (trg_sing.htb)

```
10 DIM Cont_mode$(1:4)[9],Buff$[9]
20 INTEGER I
30 !
40 ASSIGN @Agte506x TO 717
50 !
```

```
60 Cont_mode$(1)="ON"
70 Cont_mode$(2)="ON"
80 Cont_mode$(3)="OFF"
90 Cont_mode$(4)="OFF"
150 !
160 FOR I=1 TO 4
170 OUTPUT @Agte506x;":INIT"&VAL$(I)&":CONT "&Cont_mode$(I)
180 NEXT I
190 OUTPUT @Agte506x;":TRIG:SOUR BUS"
200 !
210 OUTPUT @Agte506x;":TRIG:SING"
220 OUTPUT @Agte506x;"*OPC?"
230 ENTER @Agte506x;Buff$
240 !
250 PRINT "Measurement complete"
260 END
```

Waiting for Trigger (SRQ)

- Overview
- Sample Program in Excel VBA
- Sample Program in HT Basic

Other topics about Sample Programs

Overview

This sample program demonstrates how to use an SRQ to detect the end of measurement.

NOTE This sample program correctly runs when the maximum number of channels/traces is set to 4 channels/4 traces.

The sample program sets up the trigger system, configures the instrument to properly generate an SRQ, and then triggers the instrument. When the instrument has generated an SRQ that indicates the end of measurement, the program exits after printing a measurement completion message.

See Waiting for the End of Measurement for this programming.

Sample Program in Excel VBA

```
Sub srq_meas_Click()
  Dim defrm As Long
  Dim vi As Long
  Dim ContMode(9) As String
  Dim Result As String * 10
  Dim i As Integer, StbStatus As Integer
  Const TimeOutTime = 100000 ' TimeOut time should be greater than the measurement time.
  '
  ' Assign a GPIB address to the I/O pass.
  Call viOpenDefaultRM(defrm)
  Call viOpen(defrm, "GPIB0::17::INSTR", 0, 0, vi)
  Call viSetAttribute(vi, VI_ATTR_TMO_VALUE, TimeOutTime)
  '
  ' Store the settings of continuous initiation mode for eachchannel
  ' (on for channels 1 and 2; off for channels 3 and 4)
  ' into the array variable ContMode().
  ContMode(1) = "ON"
  ContMode(2) = "ON"
  ContMode(3) = "OFF"
```

```
ContMode(4) = "OFF"
'
' Turn on or off continuous initiation mode for each channel
' depending on the value of ContMode(*).
Call viVPrintf(vi, ":DISP:SPL D12_34" & vbLf, 0)
Call viVPrintf(vi, ":SENS1:SWE:TIME:AUTO OFF" & vbLf, 0)
Call viVPrintf(vi, ":SENS1:SWE:TIME 5" & vbLf, 0)
Call viVPrintf(vi, ":SENS2:SWE:TIME:AUTO OFF" & vbLf, 0)
Call viVPrintf(vi, ":SENS2:SWE:TIME 3" & vbLf, 0)
Call viVPrintf(vi, ":SENS3:SWE:TIME:AUTO OFF" & vbLf, 0)
Call viVPrintf(vi, ":SENS3:SWE:TIME 1" & vbLf, 0)
Call viVPrintf(vi, ":SENS4:SWE:TIME:AUTO OFF" & vbLf, 0)
Call viVPrintf(vi, ":SENS4:SWE:TIME 3" & vbLf, 0)
For i = 1 To 4
  Call viVPrintf(vi, ":INIT" & CStr(i) & ":CONT " & ContMode(i) & vbLf, 0)
Next i
'
' Set the trigger source to Bus Trigger.
Call viVPrintf(vi, ":TRIG:SOUR BUS" & vbLf, 0)
'
' .
Call viVPrintf(vi, ":STAT:OPER:PTR 0" & vbLf, 0) 'Set 0 at all bits of Position Transition Filter
Call viVPrintf(vi, ":STAT:OPER:NTR 16" & vbLf, 0) 'Set 1 at bit 4 of Negative Transition Filter
Call viVPrintf(vi, ":STAT:OPER:ENAB 16" & vbLf, 0) 'Set 1 at bit 4 of Operation status enable
Call viVPrintf(vi, "*SRE 128" & vbLf, 0) 'Set 1 at bit 7 of Service Request Enable Register
Call viVPrintf(vi, "*CLS" & vbLf, 0) ' Clear Register.
'
Call viVPrintf(vi, "*TRG" & vbLf, 0) 'Make a trigger
' Wait until Status Byte Register became 192
Do
  Call viReadSTB(vi, StbStatus) ' Read Status Byte Register
  Range("B5").Value = StbStatus
Loop Until StbStatus = 192
'
' Display a measurement completion message.
Stat = MsgBox("Measurement complete", vbOKOnly)
' Close IO
```

```
    Call viClose(vi)
    Call viClose(defrm)
  End Sub
```

Sample Program in HT Basic (srq_meas.htb)

```
10 DIM Cont_mode$(1:4)[9],Buff$[9]
20 INTEGER I
30 !
40 ASSIGN @Agte506x TO 717
50 !
60 Cont_mode$(1)="ON"
70 Cont_mode$(2)="ON"
80 Cont_mode$(3)="OFF"
90 Cont_mode$(4)="OFF"
150 !
160 FOR I=1 TO 4
170 OUTPUT @Agte506x;":INIT"&VAL$(I)&":CONT "&Cont_mode$(I)
180 NEXT I
190 OUTPUT @Agte506x;":TRIG:SOUR BUS"
200 !
210 OUTPUT @Agte506x;":STAT:OPER:PTR 0"
220 OUTPUT @Agte506x;":STAT:OPER:NTR 16"
230 OUTPUT @Agte506x;":STAT:OPER:ENAB 16"
240 OUTPUT @Agte506x;"*SRE 128"
250 OUTPUT @Agte506x;"*CLS"
260 OUTPUT @Agte506x;"*OPC?"
270 ENTER @Agte506x;Buff$
280 !
290 ON INTR 7 GOTO Meas_end
300 ENABLE INTR 7;2
310 OUTPUT @Agte506x;"*TRG"
320 PRINT "Waiting..."
330 Meas_wait: GOTO Meas_wait
340 Meas_end: OFF INTR 7
350 PRINT "Measurement Complete"
360 END
```

Description

Line 40

Assigns a GPIB address to the I/O pass.

Lines 60 to 90

These lines store the settings of continuous initiation mode for each channel (on for channels 1 and 2; off for channels 3 and 4) into the array variable Cont_mode$(*).

Lines 160 to 180

These lines turn ON or OFF continuous initiation mode for each channel depending on the value of Cont_mode$(*).

Line 190

Sets the trigger source to "Bus Trigger".

Lines 210 to 220

These lines configure the instrument so that operation status event register's bit 4 is set to 1 only when the operation status condition register's bit 4 is changed from 1 to 0 (negative transition).

Lines 230 to 240

These lines enable the operation status event register's bit 4 and status byte register's bit 7.

Lines 250 to 270

These lines clear the status byte register and operation status event register.

Lines 290 to 300

These lines set the branch target for an SRQ interrupt to enable SRQ interruptions.

Lines 310 to 320

These lines trigger the instrument and wait until the measurement cycle finishes.

Line 350

Displays a measurement completion message.

Error Detection (SRQ)

- Overview
- Sample Program in Excel VBA using VISA-COM
- Sample Program in HT Basic

Other topics about Sample Programs

Overview

This sample program demonstrates how to use an SRQ to detect the occurrence of an error.

This program sets SRQs and then intentionally sends an invalid parameter to generate an error to be handled by this program. In the error handling part, this program examines the error, displays the error number and error message, and then displays the message indicating the suspension of the program. See Detecting Occurrence of an Error for this programming.

NOTE The sequence interception by an error can not be performed on Excel VBA using VISA.

Sample Program in Excel VBA using VISA-COM

```
Dim Age506x As VisaComLib.FormattedIO488
Option Explicit
Implements VisaComLib.IEventHandler

Private Sub IEventHandler_HandleEvent(ByVal vi As VisaComLib.IEventManager, ByVal SRQevent As VisaComLib.IEvent, ByVal userHandle As Long)
  On Error Resume Next
  Dim readErr As Variant
  Dim i As Integer
  Age506x.WriteString "SYST:ERR?"
  readErr = Age506x.ReadList
  MsgBox "Error : " & readErr(0) & " , " & readErr(1), vbOKOnly, "Error occured."
  End

End Sub

Private Sub Err_Detect_Click()
  Dim gmgr As VisaComLib.ResourceManager
  Dim SRQ As VisaComLib.IEventManager

  Dim strdmy As String
```

```
On Error GoTo CommandButton1_Error

Set gmgr = New VisaComLib.ResourceManager

Set Age506x = New VisaComLib.FormattedIO488
Set Age506x.IO = gmgr.Open("GPIB0::17::INSTR")

Set SRQ = Age506x.IO
SRQ.InstallHandler EVENT_SERVICE_REQ, Me, 1, 0
SRQ.EnableEvent EVENT_SERVICE_REQ, EVENT_HNDLR

With Age506x
  .WriteString "*RST"
  .WriteString "*ESE 60"
  .WriteString "*SRE 32"
  .WriteString "*CLS"
  .WriteString "*OPC?"
  strdmy = .ReadString
End With

With Age506x
  .WriteString "CALC1:PAR:COUN 2"
  .WriteString "CALC1:PAR1:DEF S21"
  .WriteString "CALC1:PAR1:SEL"
  .WriteString "CALC1:FORM MLOG"

  .WriteString "CALC1:PAR2:DEF S11"
  .WriteString "CALC1:PAR2:SEL"
  .WriteString "CALC1:FORM LOG"
  .WriteString "*OPC?"
  strdmy = .ReadString
End With

SRQ.DisableEvent ALL_ENABLED_EVENTS, EVENT_ALL_MECH, 0

Exit Sub
```

```
CommandButton1_Error:
  MsgBox "Error Occured. Error=" & vbTab & Err.Number & " , " & Err.Description
End Sub

Private Sub EndBtn_Click()
  End
End Sub
```

Sample Program in HT Basic (srq_err.bas)

```
10 DIM Buff$[9],Err_mes$[50]
20 INTEGER Err_no
30 !
40 ASSIGN @Agte506x TO 717
50 !
60 OUTPUT @Agte506x;"*ESE 60"
70 OUTPUT @Agte506x;"*SRE 32"
80 OUTPUT @Agte506x;"*CLS"
90 OUTPUT @Agte506x;"*OPC?"
100 ENTER @Agte506x;Buff$
110 !
120 ON INTR 7 GOTO Err_proc
130 ENABLE INTR 7;2
140 OUTPUT @Agte506x;":CALC1:PAR:COUN 2"
150 PRINT "Trace 1 Meas.Para: S21"
160 PRINT "Trace 1 Format : Log Mag"
170 OUTPUT @Agte506x;":CALC1:PAR1:DEF S21"
180 OUTPUT @Agte506x;":CALC1:PAR1:SEL"
190 OUTPUT @Agte506x;":CALC1:FORM MLOG"
200 PRINT "Trace 2 Meas.Para: S11"
210 PRINT "Trace 2 Format : Log Mag"
220 OUTPUT @Agte506x;":CALC1:PAR2:DEF S11"
230 OUTPUT @Agte506x;":CALC1:PAR2:SEL"
240 OUTPUT @Agte506x;":CALC1:FORM LOG"
250 OUTPUT @Agte506x;"*OPC?"
260 ENTER @Agte506x;Buff$
270 GOTO Skip_err_proc
```

```
280 Err_proc: OFF INTR 7
290 OUTPUT @Agte506x;":SYST:ERR?"
300 ENTER @Agte506x;Err_no,Err_mes$
310 PRINT "Error occurred!!"
320 PRINT " No:";Err_no,"Description: "&Err_mes$
330 PRINT "PROGRAM INTERRUPT!!"
340 GOTO Prog_end
350 Skip_err_proc: PRINT "PROGRAM DONE."
360 Prog_end: END
```

Description

Line 40

Assigns a GPIB address to the I/O pass.

Lines 60 to 70

These lines enable bits 2, 3, 4 and 5 in the standard event status register and set bit 5 to 1 in the service request enable register.

Lines 80 to 100

These lines clear the status byte register, the standard event status register, and the error queue.

Lines 120 to 130

These lines set the branch target for an SRQ interrupt to enable SRQ interruptions.

Lines 140 to 260

These lines set the measurement parameters and their data formats for traces 1 and 2. An invalid parameter is given to the data format setting for trace 2, causing an error.

Lines 280 to 330

These lines define an error handler in the following way.

Lines 290 to 300: These lines retrieve the error number and error messages for the error from the error queue.

Lines 310 to 330 These lines display the message indicating the occurrence of the error, the error number, the error message, and the message showing that the program is suspended.

Line 350

Displays a closing message. Note that this message will not display unless this program is re-executed after setting a corrected parameter to the data format setting for trace 2.

Reading Data in ASCII Format

- Overview
- Sample Program in Excel VBA using VISA
- Sample Program in Excel VBA using VISA-COM
- Sample Program in HT Basic

Other topics about Sample Programs

Overview

This sample program demonstrates how to retrieve formatted data arrays in the ASCII transfer format.

This program holds the sweep on channel 1, then retrieves and displays the stimulus array for channel 1 and the formatted data array for trace 1.

See Retrieving Measurement Results for this programming.

Sample Program in Excel VBA using VISA

```
Sub read_asc_Click()
  '
  Dim defrm As Long
  Dim vi As Long
  Dim Result As String * 10000
  Dim Res As Variant
  Dim Res2 As Variant
  Dim Nop As Long
  Const TimeOutTime = 10000
  '
  ' Open the Analyzer
  Call viOpenDefaultRM(defrm)
  Call viOpen(defrm, "GPIB0::17::INSTR", 0, 0, vi)
  Call viSetAttribute(vi, VI_ATTR_TMO_VALUE, TimeOutTime)
  '
  ' Select Parameter 1
  Call viVPrintf(vi, ":CALC1:PAR1:SEL" + vbLf, 0)
  Call viVPrintf(vi, ":INIT1:CONT OFF" + vbLf, 0)
  Call viVPrintf(vi, ":ABOR" + vbLf, 0)
  '
  ' Read out NOP Data in ASCII transfer format
  Call viVPrintf(vi, ":SENS1:SWE:POIN?" + vbLf, 0)
  Call viVScanf(vi, "%t", Result)
```

```
    Nop = Val(Result)
    ReDim FMTData(Nop, 2)
    ReDim Freq(Nop)
    '
    ' Read out Measurement Data in ASCII transfer format
    Call viVPrintf(vi, ":FORM:DATA ASC" + vbLf, 0)
    Result = ""
    Call viVPrintf(vi, ":CALC1:DATA:FDAT?" + vbLf, 0)
    Call viVScanf(vi, "%t", Result)
    Res = Split(Result, ",")
    '
    Range("A6:D1607").Clear 'Clear cells of Excel
    '
    ' Write data in cells of Excel
    j = 0
    For i = 1 To Nop
        Cells(i + 5, 1) = i
        Cells(i + 5, 3) = Val(Res(j))
        Cells(i + 5, 4) = Val(Res(j + 1))
        j = j + 2
    Next i
    '
    ' Read out Measurement Frequency Data in ASCII transfer format
    Result = ""
    Call viVPrintf(vi, ":SENS1:FREQ:DATA?" + vbLf, 0)
    Call viVScanf(vi, "%t", Result)
    Res2 = Split(Result, ",")
    '
    '  Write data in cells of Excel
    For i = 1 To Nop
        Cells(i + 5, 2) = Val(Res2(i - 1))
    Next i
    '
    ' Close the Analyzer
    Call viClose(vi)
    Call viClose(defrm)
End Sub
```

Sample Program in Excel VBA using VISA-COM

```
Private Sub Read_ASC_Click()
  Dim ReadData() As Double
  Dim Poin As Integer
  Dim FreqData() As Double

  '*** The variables of the resource manager and the instrument I/O are declared.
  Dim ioMgr As VisaComLib.ResourceManager
  Dim Age506x As VisaComLib.FormattedIO488

  '*** The memory area of the resource manager and the instrument I/O are acquired.
  Set ioMgr = New VisaComLib.ResourceManager
  Set Age506x = New VisaComLib.FormattedIO488

  '*** Open the instrument.
  Set Age506x.IO = ioMgr.Open("GPIB0::17::INSTR")
  Age506x.IO.timeout = 10000

  '*** Abort sweeping of channel1/trace1.
  Age506x.WriteString ":CALC1:PAR1:SEL", True
  Age506x.WriteString ":INIT1:CONT OFF", True
  Age506x.WriteString ":ABOR", True

  '*** Set the ascii format.
  Age506x.WriteString ":FORM:DATA ASC", True

  '*** Get num of point and the stimulus data.
  Age506x.WriteString ":SENS1:SWE:POIN?", True
  Poin = Age506x.ReadNumber
  ReDim FreqData(Poin - 1)
  Age506x.WriteString ":SENS1:FREQ:DATA?", True
  FreqData() = Age506x.ReadList(ASCIIType_R8, ",")

  '*** Get the measurement data.
  ReDim ReadData(Poin * 2 - 1)
  Age506x.WriteString ":CALC1:DATA:FDAT?", True
  ReadData() = Age506x.ReadList(ASCIIType_R8, ",")
```

```
  '*** set data for new sheet
  Sheets.Add
  ActiveSheet.Name = Format(Year(Date), "0000") & Format(Month(Date), "00") & Format(Day(Date),
"00") _
            & Format(Hour(Time), "00") & Format(Minute(Time), "00") & Format(Second(Time), "00")
  ActiveSheet.Cells(5, 3) = "Frequency"
  ActiveSheet.Cells(5, 4) = "Data 1"
  ActiveSheet.Cells(5, 5) = "Data 2"
  For i = 1 To Poin
    ActiveSheet.Cells(i + 5, 3) = FreqData(i - 1)

    ActiveSheet.Cells(i + 5, 4).Value = ReadData(i * 2 - 2)
    ActiveSheet.Cells(i + 5, 5).Value = ReadData(i * 2 - 1)
  Next i

  '*** end procedure
  Age506x.IO.Close
End Sub
```

Sample Program in HT Basic (read_asc.htb)

```
10 REAL Fdata(1:1601,1:2),Freq(1:1601)
20 DIM Img$[30]
30 INTEGER Nop,I
40 !
50 ASSIGN @Agte506x TO 717
60 !
70 OUTPUT @Agte506x;":CALC1:PAR1:SEL"
80 OUTPUT @Agte506x;":INIT1:CONT OFF"
90 OUTPUT @Agte506x;":ABOR"
100 OUTPUT @Agte506x;":SENS1:SWE:POIN?"
110 ENTER @Agte506x;Nop
120 REDIM Fdata(1:Nop,1:2),Freq(1:Nop)
130 !
140 ! Reading out in ASCII transfer format
150 !
160 OUTPUT @Agte506x;":FORM:DATA ASC"
170 !
```

```
180 OUTPUT @Agte506x;":CALC1:DATA:FDAT?"
190 ENTER @Agte506x;Fdata(*)
200 OUTPUT @Agte506x;":SENS1:FREQ:DATA?"
210 ENTER @Agte506x;Freq(*)
220 !
230 ! Displaying
240 !
250 OUTPUT @Agte506x;":CALC1:FORM?"
260 ENTER @Agte506x;Fmt$
270 SELECT Fmt$
280 CASE "MLOG","PHAS","GDEL","MLIN","SWR","REAL","IMAG","UPH"
290 Img$="MD.4DE,2X,MD.6DE"
300 PRINT " Frequency Data"
310 FOR I=1 TO Nop
320 PRINT USING Img$;Freq(I),Fdata(I,1)
330 NEXT I
340 CASE ELSE
350 Img$="MD.4DE,2X,MD.6DE,2X,MD.6DE"
360 PRINT " Frequency Data1 Data2"
370 FOR I=1 TO Nop
380 PRINT USING Img$;Freq(I),Fdata(I,1),Fdata(I,2)
390 NEXT I
400 END SELECT
410 !
420 END
```

Reading Data in Binary Format

- Overview
- Sample Program in Excel VBA using VISA
- Sample Program in Excel VBA using VISA-COM
- Sample Program in HT Basic

Other topics about Sample Programs

Overview

This sample program demonstrates how to retrieve formatted data arrays in the Binary transfer format.

This program holds the sweep on channel 1, then retrieves and displays the stimulus array.

See Retrieving Measurement Results for this programming.

Sample Program in Excel VBA using VISA

```
Sub read_bin_Click()
  Dim defrm As Long
  Dim vi As Long
  Dim Result As String * 10000
  Dim Res() As Double
  Dim Nop As Long
  Const TimeOutTime = 10000
  ' Open Analyzer
  Call viOpenDefaultRM(defrm)
  Call viOpen(defrm, "GPIB0::17::INSTR", 0, 0, vi)
  Call viSetAttribute(vi, VI_ATTR_TMO_VALUE, TimeOutTime)
  '
  Call viVPrintf(vi, ":CALC1:PAR1:SEL" + vbLf, 0)
  Call viVPrintf(vi, ":INIT1:CONT OFF" + vbLf, 0)
  Call viVPrintf(vi, ":ABOR" + vbLf, 0)
  '
  ' Reading out Measurement Frequency Data in Binary transfer format
  Call viVPrintf(vi, ":FORM:DATA REAL" + vbLf, 0)
  Call viVPrintf(vi, ":CALC1:DATA:FDAT?" + vbLf, 0)
  Call Scpi_read_binary_double_array(vi, Res, Nop)
  '
  ' Write data in cells of Excel
  Range("A6:D1607").Clear
```

```
    For i = 0 To Nop - 1
        j = i Mod 2
        k = i \ 2
        Cells(k + 6, j + 3).Value = Res(i)
    Next i
    '
    ' Read out Measurement Frequency Data in Binary transfer format
    Call viVPrintf(vi, ":SENS1:FREQ:DATA?" + vbLf, 0)
    Call Scpi_read_binary_double_array(vi, Res, Nop)
    '
    ' Write data in cells of Excel
    For i = 0 To Nop - 1
        Cells(i + 6, 1) = i + 1
        Cells(i + 6, 2).Value = Res(i)
    Next i
    ' Close
    Call viClose(vi)
    Call viClose(defrm)
End Sub
    '===================================
    ' BinaryAry Read Subroutine
    '===================================
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
```

Sample Program in Excel VBA using VISA-COM

```
Private Sub Read_BINARY_Click()
 Dim ReadData() As Double
 Dim Poin As Integer
 Dim FreqData() As Double
 '*** The variables of the resource manager and the instrument I/O are declared.
 Dim ioMgr As VisaComLib.ResourceManager
 Dim Age506x As VisaComLib.FormattedIO488
 '*** The memory area of the resource manager and the instrument I/O are acquired.
 Set ioMgr = New VisaComLib.ResourceManager
 Set Age506x = New VisaComLib.FormattedIO488

 '*** Open the instrument.
 Set Age506x.IO = ioMgr.Open("GPIB0::17::INSTR")
 Age506x.IO.Timeout = 10000

 '*** Abort sweeping of channel1/trace1.
 Age506x.WriteString ":CALC1:PAR1:SEL", True
 Age506x.WriteString ":INIT1:CONT OFF", True
 Age506x.WriteString ":ABOR", True
 '*** Set the binary format.
 Age506x.WriteString ":FORM:DATA REAL", True

 '*** Get num of point and the stimulus data.
 Age506x.WriteString ":SENS1:SWE:POIN?", True
 Poin = Age506x.ReadNumber
 Age506x.WriteString ":SENS1:FREQ:DATA?", True
 FreqData = Age506x.ReadIEEEBlock(BinaryType_R8, False, True)
 '*** Get the measurement data.
 Age506x.WriteString ":CALC1:DATA:FDAT?", True
 ReadData = Age506x.ReadIEEEBlock(BinaryType_R8, False, True)

 '*** set data for new sheet
 Sheets.Add
```

```
  ActiveSheet.Name = Format(Year(Date), "0000") & Format(Month(Date), "00") & Format(Day(Date),
"00") _
                & Format(Hour(Time), "00") & Format(Minute(Time), "00") & Format(Second(Time), "00")
  ActiveSheet.Cells(5, 3) = "Frequency"
  ActiveSheet.Cells(5, 4) = "Data 1"
  ActiveSheet.Cells(5, 5) = "Data 2"
  For i = 1 To Poin
    ActiveSheet.Cells(i + 5, 3) = FreqData(i - 1)

    ActiveSheet.Cells(i + 5, 4).Value = ReadData(i * 2 - 2)
    ActiveSheet.Cells(i + 5, 5).Value = ReadData(i * 2 - 1)
  Next i

  '*** end procedure
  Age506x.WriteString ":FORM:DATA ASC", True
  Age506x.IO.Close
End Sub
```

Sample Program in HT Basic (read_bin.htb)

```
10 REAL Fdata(1:1601,1:2),Freq(1:1601)
20 DIM Buff$[9],Img$[30]
30 INTEGER Nop,I
40 !
50 ASSIGN @Agte506x TO 717
60 ASSIGN @Binary TO 717;FORMAT OFF
70 !
80 OUTPUT @Agte506x;":CALC1:PAR1:SEL"
90 OUTPUT @Agte506x;":INIT1:CONT OFF"
100 OUTPUT @Agte506x;":ABOR"
110 OUTPUT @Agte506x;":SENS1:SWE:POIN?"
120 ENTER @Agte506x;Nop
130 REDIM Fdata(1:Nop,1:2),Freq(1:Nop)
140 !
150 ! Reading out in binary transfer format
160 !
170 OUTPUT @Agte506x;":FORM:DATA REAL"
180 !
190 OUTPUT @Agte506x;":CALC1:DATA:FDAT?"
```

```
200 ENTER @Agte506x USING "#,8A";Buff$
210 ENTER @Binary;Fdata(*)
220 ENTER @Agte506x USING "#,1A";Buff$
230 OUTPUT @Agte506x;":SENS1:FREQ:DATA?"
240 ENTER @Agte506x USING "#,8A";Buff$
250 ENTER @Binary;Freq(*)
260 ENTER @Agte506x USING "#,1A";Buff$
270 !
280 ! Displaying
290 !
300 OUTPUT @Agte506x;":CALC1:FORM?"
310 ENTER @Agte506x;Fmt$
320 SELECT Fmt$
330 CASE "MLOG","PHAS","GDEL","MLIN","SWR","REAL","IMAG","UPH"
340 Img$="MD.4DE,2X,MD.6DE"
350 PRINT " Frequency Data"
360 FOR I=1 TO Nop
370 PRINT USING Img$;Freq(I),Fdata(I,1)
380 NEXT I
390 CASE ELSE
400 Img$="MD.4DE,2X,MD.6DE,2X,MD.6DE"
410 PRINT " Frequency Data1 Data2"
420 FOR I=1 TO Nop
430 PRINT USING Img$;Freq(I),Fdata(I,1),Fdata(I,2)
440 NEXT I
450 END SELECT
460 !
470 END
```

Writing Data in Ascii Format

- Overview
- Sample Program in Excel VBA using VISA
- Sample Program in Excel VBA using VISA-COM
- Sample Program in HT Basic

Other topics about Sample Programs

Overview

The sample program demonstrates to Write Formatted Data Arrays in Using the ASCII Transfer Format

See Entering Data into a Trace for this programming.

NOTE The sample programs of Excel VBA (using VISA/VISA-COM) does not create the Ch2, and the measurement point is not set. It is necessary to create Ch2 and to set the measurement point to a prescribed measurement point before the program is executed.

Sample Program in Excel VBA using VISA

```
Private Sub Write_ASC_Click()
 '*** Before this program is executed, create the channel 2,
 '*** and match the number of points and data format
 '*** of channel 2 to the transmitted data.
 Dim defrm As Long
 Dim Age506x As Long
 Dim Poin As Integer
 Dim Buf As String * 16
 Dim WriteData() As Double
 Dim AryPtr As Long
 Dim WrtFmt As String

 Call viOpenDefaultRM(defrm)
 '*** Open the instrument, and set times 10sec for timeout.
 Call viOpen(defrm, "GPIB0::17::INSTR", 0, 0, Age506x)
 Call viSetAttribute(Age506x, VI_ATTR_TMO_VALUE, 10000)

 '*** Abort sweeping of channel2/trace1.
 Call viVPrintf(Age506x, ":CALC2:PAR1:SEL" + vbLf, 0)
 Call viVPrintf(Age506x, ":INIT2:CONT OFF" + vbLf, 0)
```

```
 Call viVPrintf(Age506x, ":ABOR" + vbLf, 0)
 '*** Set the ascii format.
 Call viVPrintf(Age506x, ":FORM:DATA ASC" + vbLf, 0)

 '*** Get num of point.
 Call viVPrintf(Age506x, ":SENS1:SWE:POIN?" + vbLf, 0)
 Call viVScanf(Age506x, "%t", Buf)
 Poin = CInt(Buf)
 ReDim WriteData(Poin * 2 - 1)

 '*** Set data for array variable, and send data for E506x.
 For i = 1 To Poin
   WriteData(i * 2 - 2) = ActiveSheet.Cells(i + 5, 4).Value
   WriteData(i * 2 - 1) = ActiveSheet.Cells(i + 5, 5).Value
 Next i
 AryPtr = VarPtr(WriteData(0))
 Call viVPrintf(Age506x, ":CALC2:DATA:FDAT ", 0)

 WrtFmt = "%," & Poin * 2 & "lf"
 Call viVPrintf(Age506x, WrtFmt, AryPtr)
 Call viVPrintf(Age506x, vbLf, 0)

 '*** end procedure
 Call viClose(Age506x)
 Call viClose(defrm)
End Sub
```

Sample Program in Excel VBA using VISA-COM

```
Private Sub Write_ASC_Click()
 '*** Before this program executes, create the channel 2,
 '*** and match the number of points and data format
 '*** of channel 2 to the transmitted data.
 Dim WriteData() As Double
 Dim Poin As Integer
 '*** The variables of the resource manager and the instrument I/O are declared.
 Dim ioMgr As VisaComLib.ResourceManager
 Dim Age506x As VisaComLib.FormattedIO488
```

```
'*** The memory area of the resource manager and the instrument I/O are acquired.
Set ioMgr = New VisaComLib.ResourceManager
Set Age506x = New VisaComLib.FormattedIO488

'*** Open the instrument.
Set Age506x.IO = ioMgr.Open("GPIB0::17::INSTR")
Age506x.IO.Timeout = 10000

'*** Abort sweeping of channel2/trace1.
Age506x.WriteString ":CALC2:PAR1:SEL", True
Age506x.WriteString ":INIT2:CONT OFF", True
Age506x.WriteString ":ABOR", True
'*** Set the ascii format.
Age506x.WriteString ":FORM:DATA ASC", True

'*** Get num of point.
Age506x.WriteString ":SENS1:SWE:POIN?", True
Poin = Age506x.ReadNumber

ReDim WriteData(Poin * 2 - 1) As Double
'*** Set data for array variable, and send data for E506x.
For i = 1 To Poin
  WriteData(i * 2 - 2) = ActiveSheet.Cells(i + 5, 4).Value
  WriteData(i * 2 - 1) = ActiveSheet.Cells(i + 5, 5).Value
Next i
Age506x.WriteString "CALC2:DATA:FDAT ", False
Age506x.WriteList WriteData, ASCIIType_R8, ",", True

'*** end procedure
Age506x.IO.Close
End Sub
```

Sample Program in HT Basic (write_a.htb)

```
10 REAL Freq,Fdata(1:1601,1:2)
20 DIM File$[300]
30 INTEGER Nop
40 !
```

```
50 ASSIGN @Agte506x TO 717
60 !
70 CALL Inp_file_name(File$)
80 !
90 OUTPUT @Agte506x;":CALC1:PAR1:SEL"
100 OUTPUT @Agte506x;":INIT1:CONT OFF"
110 OUTPUT @Agte506x;":ABOR"
120 !
130 OUTPUT @Agte506x;":SENS1:SWE:POIN?"
140 ENTER @Agte506x;Nop
150 REDIM Fdata(1:Nop,1:2)
160 !
170 ON ERROR GOTO File_error
180 ASSIGN @File TO File$
190 ENTER @File USING "K";Buff$
200 ENTER @File USING "K";Buff$
210 ENTER @File USING "K";Buff$
220 FOR I=1 TO Nop
230 ENTER @File USING "19D,2X,19D,2X,19D";Freq,Fdata(I,1),Fdata (I,2)
240 NEXT I
250 ASSIGN @File TO *
260 OFF ERROR
270 !
280 OUTPUT @Agte506x;":FORM:DATA ASC"
290 !
300 OUTPUT @Agte506x;":CALC1:DATA:FDAT ";Fdata(*)
310 !
320 GOTO Prog_end
330 !
340 File_error: OFF ERROR
350 PRINT "############ ERROR ############"
360 PRINT File$&" is NOT exist."
370 PRINT " or"
380 PRINT File$&" has UNSUITABLE data."
390 !
400 Prog_end: END
410 !=============================================
```

```
420 ! File Name Input Function
430 !============================================
440 SUB Inp_file_name(Inp_name$)
450 DIM Inp_char$[9]
460 ON ERROR GOTO Inp_start
470 Inp_start: !
480 PRINT "Input File Name!"
490 INPUT "Name?",Inp_name$
500 PRINT "Input Name: "&Inp_name$
510 INPUT "OK? [Y/N]",Inp_char$
520 IF UPC$(Inp_char$)<>"Y" THEN Inp_start
530 OFF ERROR
540 SUBEND
```

Description

Line 50

Assigns a GPIB address to the I/O pass.

Line 70

Passes control to a subprogram named Inp_file_name, which lets the user input a file name, and then stores the returned file name into the File$ variable. For more information on the Inp_file_name subprogram, refer to the description in Using the Binary Transfer Format to write Formatted Data Arrays.

Lines 90 to 110

These lines set channel 1's active trace to trace 1 and hold the sweep.

Lines 130 to 140

These lines retrieve the number of points in channel 1 and stores that number into the Nop variable.

Line 150

Resizes the Fdata array based on the value of the Nop variable (the number of points).

Line 170

This line points to the statement block to be executed if an error occurs in retrieving data from the file (for example, if no file matches File$).

Lines 180 to 260

These lines retrieve the formatted data from the file identified by File$, and store the data into the Fdata array.

Line 280

Sets the data transfer format to ASCII.

Line 300

Writes Fdata into the formatted data array for the active trace (trace 1) in channel 1.

Lines 340 to 380

This statement block is executed if an error occurs in retrieving data from the file.

Writing Data in Binary Format

- Overview
- Sample Program in Excel VBA using VISA
- Sample Program in Excel VBA using VISA-COM
- Sample Program in HT Basic

Other topics about Sample Programs

Overview

The sample program demonstrates to Write Formatted Data Arrays in Using the Binary Transfer Format.

See Entering Data into a Trace for this programming.

NOTE The sample programs of Excel VBA (using VISA/VISA-COM) does not create the Ch2, and the measurement point is not set. It is necessary to create Ch2 and to set the measurement point to a prescribed measurement point before the program is executed.

Sample Program in Excel VBA using VISA

```
Private Sub Write_BINARY_Click()
 '*** Before this program is executed, create the channel 2,
 '*** and match the number of points and data format
 '*** of channel 2 to the transmitted data.
 Dim defrm As Long
 Dim Age506x As Long
 Dim Poin As Integer
 Dim Buf As String * 16
 Dim WriteData() As Double
 Dim WrtPtr As Long
 Dim WrtFmt As String

 Call viOpenDefaultRM(defrm)
 '*** Open the instrument.
 Call viOpen(defrm, "GPIB0::17::INSTR", 0, 0, Age506x)
 Call viSetAttribute(Age506x, VI_ATTR_TMO_VALUE, 10000)

 '*** Abort sweeping of channel2/trace1.
 Call viVPrintf(Age506x, ":CALC2:PAR1:SEL" + vbLf, 0)
 Call viVPrintf(Age506x, ":INIT2:CONT OFF" + vbLf, 0)
```

```
  Call viVPrintf(Age506x, ":ABOR" + vbLf, 0)
  '*** Set the binary format.
  Call viVPrintf(Age506x, ":FORM:DATA REAL" + vbLf, 0)
  '*** Get num of point, and re-define array size of data.
  Call viVPrintf(Age506x, ":SENS1:SWE:POIN?" + vbLf, 0)
  Call viVScanf(Age506x, "%t", Buf)
  Poin = CInt(Buf)
  ReDim WriteData(Poin * 2 - 1)
  WrtPtr = VarPtr(WriteData(0))

  '*** Set data for array variable.
  For i = 1 To Poin
    WriteData(i * 2 - 2) = ActiveSheet.Cells(i + 5, 4).Value
    WriteData(i * 2 - 1) = ActiveSheet.Cells(i + 5, 5).Value
  Next i
  '*** set Format code & Modifier.
  WrtFmt = "%#" & Trim(CStr(Poin * 2)) & "Zb"
  '*** send data for E506x.
  Call viVPrintf(Age506x, ":CALC2:DATA:FDAT ", 0)
  Call viVPrintf(Age506x, WrtFmt + vbLf, WrtPtr)

  '*** end procedure
  Call viVPrintf(Age506x, ":FORM:DATA ASC" + vbLf, 0)
  Call viClose(Age506x)
  Call viClose(defrm)

End Sub
```

Sample Program in Excel VBA using VISA-COM

```
Private Sub Write_BINARY_Click()
  '*** Before this program executes, create the channel 2,
  '*** and match the number of points and data format
  '*** of channel 2 to the transmitted data.
  Dim WriteData() As Double
  Dim Poin As Integer
  '*** The variables of the resource manager and the instrument I/O are declared.
  Dim ioMgr As VisaComLib.ResourceManager
```

```
  Dim Age506x As VisaComLib.FormattedIO488
  '*** The memory area of the resource manager and the instrument I/O are acquired.
  Set ioMgr = New VisaComLib.ResourceManager
  Set Age506x = New VisaComLib.FormattedIO488

  '*** Open the instrument.
  Set Age506x.IO = ioMgr.Open("GPIB0::17::INSTR")
  Age506x.IO.Timeout = 10000

  '*** Abort sweeping of channel2/trace1.
  Age506x.WriteString ":CALC2:PAR1:SEL", True
  Age506x.WriteString ":INIT2:CONT OFF", True
  Age506x.WriteString ":ABOR", True
  '*** Set the binary format.
  Age506x.WriteString ":FORM:DATA REAL", True

  '*** Get num of point.
  Age506x.WriteString ":SENS1:SWE:POIN?", True
  Poin = Age506x.ReadNumber
  ReDim WriteData(Poin * 2 - 1) As Double
  '*** Set data for array variable, and send data for E506x.
  For i = 1 To Poin
    WriteData(i * 2 - 2) = ActiveSheet.Cells(i + 5, 4).Value
    WriteData(i * 2 - 1) = ActiveSheet.Cells(i + 5, 5).Value
  Next i
  Age506x.WriteIEEEBlock ":CALC2:DATA:FDAT ", WriteData, True

  '*** end procedure
  Age506x.WriteString ":FORM:DATA ASC", True
  Age506x.IO.Close

End Sub
```

Sample Program in HT Basic (write_b.htb)

```
10 REAL Freq,Fdata(1:1601,1:2)
20 DIM File$[300],Header$[10]
30 INTEGER Nop
```

```
40 !
50 ASSIGN @Agte506x TO 717
60 ASSIGN @Binary TO 717;FORMAT OFF
70 CALL Inp_file_name(File$)
80 !
90 OUTPUT @Agte506x;":CALC1:PAR1:SEL"
100 OUTPUT @Agte506x;":INIT1:CONT OFF"
110 OUTPUT @Agte506x;":ABOR"
120 !
130 OUTPUT @Agte506x;":SENS1:SWE:POIN?"
140 ENTER @Agte506x;Nop
150 REDIM Fdata(1:Nop,1:2)
160 !
170 ON ERROR GOTO File_error
180 ASSIGN @File TO File$
190 ENTER @File USING "K";Buff$
200 ENTER @File USING "K";Buff$
210 ENTER @File USING "K";Buff$
220 FOR I=1 TO Nop
230 ENTER @File USING "19D,2X,19D,2X,19D";Freq,Fdata(I,1),Fdata (I,2)
240 NEXT I
250 ASSIGN @File TO *
260 OFF ERROR
270 !
280 OUTPUT @Agte506x;":FORM:DATA REAL"
290 Header$="#6"&IVAL$(8*2*Nop,10)
300 OUTPUT @Agte506x;":CALC1:DATA:FDAT ";Header$;
310 OUTPUT @Binary;Fdata(*),END
320 GOTO Prog_end
330 !
340 File_error: OFF ERROR
350 PRINT "############ ERROR ############"
360 PRINT File$&" is NOT exist."
370 PRINT " or"
380 PRINT File$&" has UNSUITABLE data."
390 !
400 Prog_end: END
```

```
410 !==============================================
420 ! File Name Input Function
430 !==============================================
440 SUB Inp_file_name(Inp_name$)
450 DIM Inp_char$[9]
460 ON ERROR GOTO Inp_start
470 Inp_start: !
480 PRINT "Input File Name!"
490 INPUT "Name?",Inp_name$
500 PRINT "Input Name: "&Inp_name$
510 INPUT "OK? [Y/N]",Inp_char$
520 IF UPC$(Inp_char$)<>"Y" THEN Inp_start
530 OFF ERROR
540 SUBEND
```

Description

Lines 50 to 60

Assigns a GPIB address to the I/O pass.

Line 70

Passes control to a subprogram named Inp_file_name, which lets the user input a file name, and then stores the returned file name into the File$ variable.

Lines 90 to 110

These lines set channel 1's active trace to trace 1 and hold the sweep.

Lines 130 to 140

These lines retrieve the number of points in channel 1 and stores that number into the Nop variable.

Line 150

Resizes the Fdata array based on the value of the Nop variable (the number of points).

Line 170

This line points to the statement block to be executed if an error occurs in retrieving data from the file (for example, if no file matches File$).

Lines 180 to 260

These lines retrieve the formatted data from the file identified by File$, and store the data into the Fdata array.

Line 280

Sets the data transfer format to binary.

Line 290

Creates the data header and stores it into the Header$ variable.

Line 300

Sends the command that writes data into the formatted data array for the active trace (trace 1) in channel 1, following it with the data header (Header$).

Line 310

Sends the data itself (Fdata), following it with a message terminator.

NOTE Because binary data must be written without being formatted, the program uses an I/O path (@Binary) that is configured to support writing unformatted data.

Lines 340 to 380

This statement block is executed if an error occurs in retrieving data from the file.

The Inp_file_name subprogram in lines 440 to 540, which is used to enter a save filename, is described below.

Line 460

Allows the user to return to the entry start line and re-enter the data if an error (such as an invalid entry) occurs while entering the target file name.

Lines 480 to 490

These lines prompt the user to enter the target file name. The program does not continue till the user actually enters the file name.

Lines 500 to 510

These lines display the entered file name and waits for a confirmation entry (y/n key).

Line 520

Returns to the entry start line if the key the user pressed in line 510 is not the y key.

Peak Search

- Overview
- Sample Program in Excel VBA
- Sample Program in HT Basic

Other topics about Sample Programs

Overview

This sample program demonstrates how to search for peaks using the Marker Search feature and analysis commands.

This program works in two steps: first, it uses Marker Search to search for the maximum positive peak and displays the results; second, it uses analysis commands to search for all positive peaks and displays the results.

See Searching for Positions That Match Specified Criteria for this programming.

Sample Program in Excel VBA

```
Private Sub PeakSearch_Click()
  Dim defrm As Long
  Dim vi As Long
  Const TimeOutTime = 20000
  '
  Dim Buff As String, Img As String, Err_msg As String
  Dim Excursion As String, Freq As String * 20, Resp As Variant, PeakPoint As Variant
  Dim Poin As String * 5, Result As String * 1000, errmsg As String * 20

  Excursion = "0.5"
  ' Open Analyzer
  Call viOpenDefaultRM(defrm)
  Call viOpen(defrm, "GPIB0::17::INSTR", 0, 0, vi)
  Call viSetAttribute(vi, VI_ATTR_TMO_VALUE, TimeOutTime)
  Call viVPrintf(vi, "*CLS" & vbLf, 0)
  '
  ' Setup Analyzer
  Call viVPrintf(vi, ":SENS1:FREQ:CENT 947.5E6" & vbLf, 0)
  Call viVPrintf(vi, ":SENS1:FREQ:SPAN 200E6" & vbLf, 0)
  Call viVPrintf(vi, ":CALC1:PAR1:DEF S11" & vbLf, 0)
  Call viVPrintf(vi, ":DISP:WIND1:TRAC1:Y:AUTO" & vbLf, 0)
  '
```

```
Call viVPrintf(vi, ":CALC1:PAR1:SEL" & vbLf, 0)
Call viVPrintf(vi, ":CALC1:MARK1:FUNC:TYPE PEAK" & vbLf, 0)
Call viVPrintf(vi, ":CALC1:MARK1:FUNC:PEXC " & Excursion & vbLf, 0)
Call viVPrintf(vi, ":CALC1:MARK1:FUNC:PPOL POS" & vbLf, 0)
Call viVPrintf(vi, ":CALC1:MARK1:FUNC:EXEC" & vbLf, 0)
Call ErrorCheck(vi)
Call viVPrintf(vi, ":CALC1:MARK1:X?" & vbLf, 0)
Call viVScanf(vi, "%t", Freq)
'
Call viVPrintf(vi, ":CALC1:MARK1:Y?" & vbLf, 0)
Call viVScanf(vi, "%t", Result)
'
Resp = Split(Result, ",")
Cells(5, 5).Value = Val(Freq)
Cells(5, 6).Value = Val(Resp(0))
'
Call viVPrintf(vi, ":CALC1:FUNC:DOM OFF" & vbLf, 0)
Call viVPrintf(vi, ":CALC1:FUNC:TYPE APE" & vbLf, 0)
Call viVPrintf(vi, ":CALC1:FUNC:PEXC " & Excursion & vbLf, 0)
Call viVPrintf(vi, ":CALC1:FUNC:PPOL NEG" & vbLf, 0)
Call viVPrintf(vi, ":CALC1:FUNC:EXEC" & vbLf, 0)
Call ErrorCheck(vi)
'
Call viVPrintf(vi, ":CALC1:FUNC:POIN?" & vbLf, 0)
Call viVScanf(vi, "%t", Poin)
Call viVPrintf(vi, ":CALC1:FUNC:DATA?" & vbLf, 0)
Call viVScanf(vi, "%t", Result)
PeakPoint = Split(Result, ",")
'
j = 0
For i = 1 To Val(Poin)
   Cells(6 + i, 5).Value = Val(PeakPoint(j))
   Cells(6 + i, 6).Value = Val(PeakPoint(j + 1))
   j = j + 2
Next i
'
Call viClose(vi)
```

```
    Call viClose(defrm)
    '
End Sub
Sub ErrorCheck(vi)
    Dim err As String * 50, ErrNo As Variant, Response As VbMsgBoxResult
    Call viVQueryf(vi, ":SYST:ERR?" & vbLf, "%t", err)
    ErrNo = Split(err, ",")
    If Val(ErrNo(0)) <> 0 Then
        Response = MsgBox(CStr(ErrNo(1)), vbOKOnly)
    End If
End Sub
```

Sample Program in HT Basic (search.htb)

```
10 DIM Buff$[9],Img$[50],Err_msg$[100]
20 REAL Excursion,Freq,Resp,Result(1:100,1:2)
30 INTEGER Poin,Err_no
40 !
50 ASSIGN @Agte506x TO 717
60 Excursion=.5
70 !
80 OUTPUT @Agte506x;"*ESE 60"
90 OUTPUT @Agte506x;"*SRE 32"
100 OUTPUT @Agte506x;"*CLS"
110 OUTPUT @Agte506x;"*OPC?"
120 ENTER @Agte506x;Buff$
130 ON INTR 7 GOTO Err
140 ENABLE INTR 7;2
150 !
160 PRINT "Maximum Peak Search using Marker 1"
170 !
180 OUTPUT @Agte506x;":CALC1:PAR1:SEL"
190 OUTPUT @Agte506x;":CALC1:MARK1:FUNC:TYPE PEAK"
200 OUTPUT @Agte506x;":CALC1:MARK1:FUNC:PEXC ";Excursion
210 OUTPUT @Agte506x;":CALC1:MARK1:FUNC:PPOL POS"
220 OUTPUT @Agte506x;":CALC1:MARK1:FUNC:EXEC"
230 OUTPUT @Agte506x;":CALC1:MARK1:X?"
240 ENTER @Agte506x;Freq
```

```
250 OUTPUT @Agte506x;":CALC1:MARK1:Y?"
260 ENTER @Agte506x;Resp
270 Img$="8A,MD.4DE,2X,MD.6DE"
280 PRINT " Frequency Response"
290 PRINT USING Img$;"Peak: ",Freq,Resp
300 !
310 PRINT "All Peaks Search using Command"
320 !
330 OUTPUT @Agte506x;":CALC1:FUNC:DOM OFF"
340 OUTPUT @Agte506x;":CALC1:FUNC:TYPE APE"
350 OUTPUT @Agte506x;":CALC1:FUNC:PEXC ";Excursion
360 OUTPUT @Agte506x;":CALC1:FUNC:PPOL POS"
370 OUTPUT @Agte506x;":CALC1:FUNC:EXEC"
380 OUTPUT @Agte506x;":CALC1:FUNC:POIN?"
390 ENTER @Agte506x;Poin
400 REDIM Result(1:Poin,1:2)
410 OUTPUT @Agte506x;":CALC1:FUNC:DATA?"
420 ENTER @Agte506x;Result(*)
430 Img$="4A,2D,2A,MD.4DE,2X,MD.6DE"
440 PRINT " Frequency Response"
450 FOR I=1 TO Poin
460 PRINT USING Img$;"Peak",I,": ",Result(I,2),Result(I,1)
470 NEXT I
480 GOTO No_err
490 Err: OFF INTR 7
500 OUTPUT @Agte506x;";:SYST:ERR?"
510 ENTER @Agte506x;Err_no,Err_msg$
520 PRINT "Error occurred!!"
530 PRINT " No:";Err_no,"Description: "&Err_msg$
540 No_err: OFF INTR 7
550 END
```

Limit Test

- Overview
- Sample Program in Excel VBA using VISA
- Sample Program in HT Basic

Other topics about Sample Programs

Overview

This sample program demonstrates how to perform limit tests.

The sample program creates a limit table as shown in the following two tables, turns on the Limit Test feature, performs one cycle of measurement, and then displays the test results.

Limit table of trace 1

No.	Type	Begin Stimulus	End Stimulus	Begin Response	End Response
1	MAX	847.5 MHz	905.0 MHz	-55.0 dBm	-55.0 dBm
2	MIN	935.0 MHz	960.0 MHz	-3.5 dBm	-3.5 dBm
3	MAX	935.0 MHz	960.0 MHz	0 dBm	0 dBm
4	MAX	980.0 MHz	1047.5 MHz	-25.0 dBm	-25.0 dBm

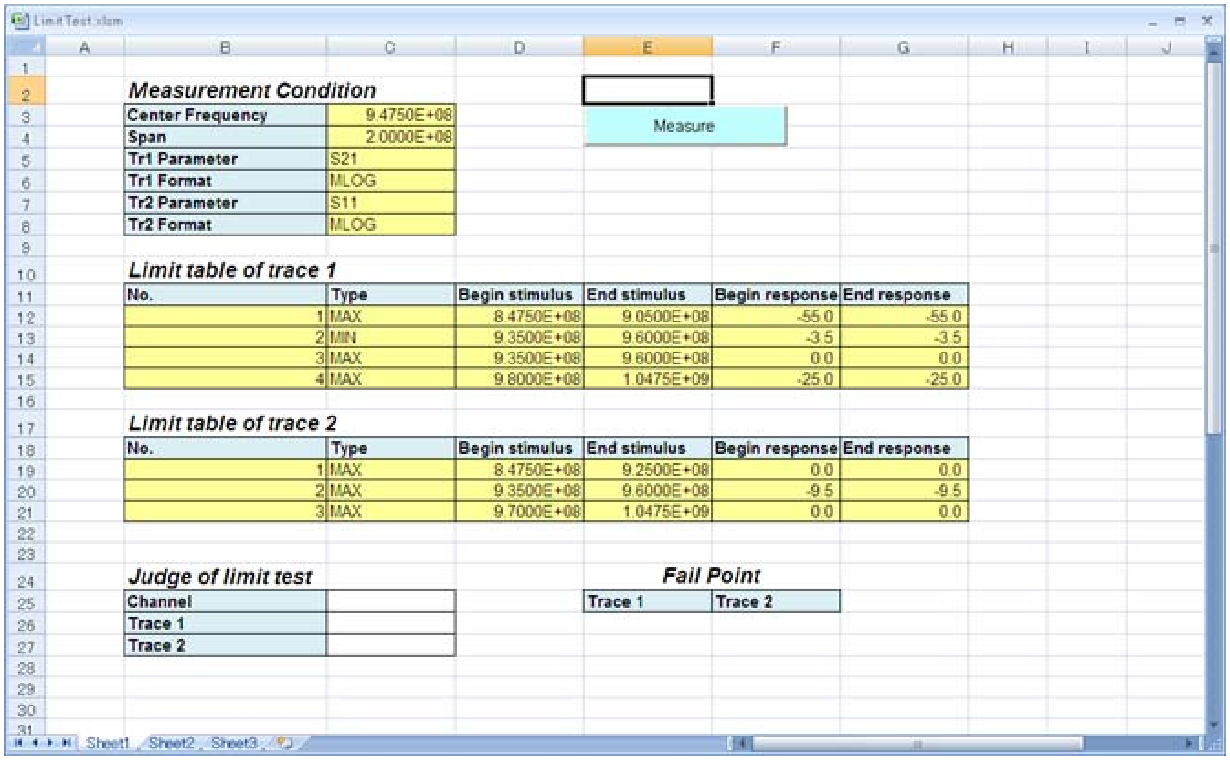
Limit table of trace 2

No.	Type	Begin Stimulus	End Stimulus	Begin Response	End Response
1	MAX	847.5 MHz	925.0 MHz	0 dBm	0 dBm
2	MAX	935.0 MHz	960.0 MHz	-9.5 dBm	-9.5 dBm
3	MAX	970.0 MHz	1047.5 MHz	0 dBm	0 dBm

See Limit Test for this programming.

Sample Program in Excel VBA using VISA

Example of excel sheet with limit test program




```
Private Sub Measure_Click()
    Dim defrm As Long
    Dim Age506x As Long

    Dim Cent As Double, Span As Double
    Dim Param(1) As String, Fmt(1) As String
    Dim NumofSeg(1) As Integer

    Dim LimTbl1 As LimitTbl1
    Dim LimTbl2 As LimitTbl2

    Dim Dummy As String * 20
    Dim ret As Integer

    Dim Lim_Judge As Integer
    Dim Tr1_Judge As Integer
    Dim Tr2_Judge As Integer
    Dim Fail_Point As Integer
    Dim Fail_Data() As Double
```

```
Dim ptr As Long
Dim Fail_Point2 As Integer
Dim Fail_Data2() As Double
Dim ptr2 As Long

'***
'*** Open session.
'***

Call viOpenDefaultRM(defrm)
Call viOpen(defrm, "GPIB0::17::INSTR", 0, 0, Age506x)
Call viSetAttribute(Age506x, VI_ATTR_TMO_VALUE, 10000)

'***
'*** Set variable of measurement condition.
'***

Cent = CDbl(Cells(3, 3).Value)
Span = CDbl(Cells(4, 3).Value)
Param(0) = Trim(Cells(5, 3).Value)
Fmt(0) = Trim(Cells(6, 3).Value)
Param(1) = Trim(Cells(7, 3).Value)
Fmt(1) = Trim(Cells(8, 3).Value)

'***
'*** Set variable of limit tables.
'***

NumofSeg(0) = 4
NumofSeg(1) = 3

For i = 0 To NumofSeg(0) - 1
    With LimTbl1
        If Trim(Cells(12 + i, 3).Value) = "MAX" Then
            .Typ(i) = 1
        Else
            .Typ(i) = 2
```

```
        End If
        .BeginStim(i) = CDbl(Cells(12 + i, 4).Value)
        .EndStim(i) = CDbl(Cells(12 + i, 5).Value)
        .BeginResp(i) = CDbl(Cells(12 + i, 6).Value)
        .EndResp(i) = CDbl(Cells(12 + i, 7).Value)
    End With
Next i

For i = 0 To NumofSeg(1) - 1
    With LimTbl2
        If Trim(Cells(19 + i, 3).Value) = "MAX" Then
            .Typ(i) = 1
        Else
            .Typ(i) = 2
        End If
        .BeginStim(i) = CDbl(Cells(19 + i, 4).Value)
        .EndStim(i) = CDbl(Cells(19 + i, 5).Value)
        .BeginResp(i) = CDbl(Cells(19 + i, 6).Value)
        .EndResp(i) = CDbl(Cells(19 + i, 7).Value)
    End With
Next i

'***
'*** Send measurement condition to E5061B.
'***

Call viVPrintf(Age506x, ":SENS1:FREQ:CENT " + CStr(Cent) + vbLf, 0)
Call viVPrintf(Age506x, ":SENS1:FREQ:SPAN " + CStr(Span) + vbLf, 0)
Call viVPrintf(Age506x, ":CALC1:PAR1:COUN 2" + vbLf, 0)
Call viVPrintf(Age506x, ":DISP:WIND1:SPL D1_2" + vbLf, 0)

Call viVPrintf(Age506x, ":TRIG:SOUR BUS" + vbLf, 0)
Call viVPrintf(Age506x, ":INIT1:CONT ON" + vbLf, 0)
Call viVPrintf(Age506x, "*OPC?" & vbLf, 0)
Call viVScanf(Age506x, "%t", Dummy)

'***
```

```
'*** Send measurement parameter and format of trace 1 to E5061B.
'***

Call viVPrintf(Age506x, ":CALC1:PAR1:SEL" + vbLf, 0)
Call viVPrintf(Age506x, ":CALC1:PAR1:DEF " + Param(0) + vbLf, 0)
Call viVPrintf(Age506x, ":CALC1:FORM " + Fmt(0) + vbLf, 0)

'***
'*** Send limit table of trace 1 to E5061B.
'***

Call viVPrintf(Age506x, ":CALC1:LIM:DATA " + CStr(NumofSeg(0)), 0)
For i = 0 To NumofSeg(0) - 1
   With LimTbl1
      Call viVPrintf(Age506x, "," + CStr(.Typ(i)), 0)
      Call viVPrintf(Age506x, "," + CStr(.BeginStim(i)), 0)
      Call viVPrintf(Age506x, "," + CStr(.EndStim(i)), 0)
      Call viVPrintf(Age506x, "," + CStr(.BeginResp(i)), 0)
      If i = NumofSeg(0) - 1 Then
         Call viVPrintf(Age506x, "," + CStr(.EndResp(i)) + vbLf, 0)
      Else
         Call viVPrintf(Age506x, "," + CStr(.EndResp(i)), 0)
      End If
   End With
Next i
Call viVPrintf(Age506x, ":CALC1:LIM:DISP ON" + vbLf, 0)
Call viVPrintf(Age506x, ":CALC1:LIM ON" + vbLf, 0)
Call viVPrintf(Age506x, "*OPC?" & vbLf, 0)
Call viVScanf(Age506x, "%t", Dummy)
'***
'*** Send measurement parameter and format of trace 2 to E5061B.
'***

Call viVPrintf(Age506x, ":CALC1:PAR2:SEL" + vbLf, 0)
Call viVPrintf(Age506x, ":CALC1:PAR2:DEF " + Param(1) + vbLf, 0)
Call viVPrintf(Age506x, ":CALC1:FORM " + Fmt(1) + vbLf, 0)
```

```
'***
'*** Send limit table of trace 2 to E5061B.
'***
Call viVPrintf(Age506x, ":CALC1:LIM:DATA " + CStr(NumofSeg(1)), 0)
For i = 0 To NumofSeg(1) - 1
   With LimTbl2
      Call viVPrintf(Age506x, "," + CStr(.Typ(i)), 0)
      Call viVPrintf(Age506x, "," + CStr(.BeginStim(i)), 0)
      Call viVPrintf(Age506x, "," + CStr(.EndStim(i)), 0)
      Call viVPrintf(Age506x, "," + CStr(.BeginResp(i)), 0)
      If i = NumofSeg(1) - 1 Then
         Call viVPrintf(Age506x, "," + CStr(.EndResp(i)) + vbLf, 0)
      Else
         Call viVPrintf(Age506x, "," + CStr(.EndResp(i)), 0)
      End If
   End With
Next i
Call viVPrintf(Age506x, ":CALC1:LIM:DISP ON" + vbLf, 0)
Call viVPrintf(Age506x, ":CALC1:LIM ON" + vbLf, 0)
Call viVPrintf(Age506x, "*OPC?" & vbLf, 0)
Call viVScanf(Age506x, "%t", Dummy)

'***
'*** Setting status resister.
'***

Call viVPrintf(Age506x, ":STAT:QUES:LIM:CHAN1:ENAB 6" + vbLf, 0)
Call viVPrintf(Age506x, ":STAT:QUES:LIM:CHAN1:PTR 6" + vbLf, 0)
Call viVPrintf(Age506x, ":STAT:QUES:LIM:CHAN1:NTR 0" + vbLf, 0)
Call viVPrintf(Age506x, ":STAT:QUES:LIM:PTR 2" + vbLf, 0)
Call viVPrintf(Age506x, ":STAT:QUES:LIM:NTR 0" + vbLf, 0)
Call viVPrintf(Age506x, "*CLS" + vbLf, 0)
Call viVPrintf(Age506x, "*OPC?" & vbLf, 0)
Call viVScanf(Age506x, "%t", Dummy)

'***
'*** Trigger.
```

```
'***

Call viVPrintf(Age506x, ":TRIG:SING" + vbLf, 0)
Call viVPrintf(Age506x, "*OPC?" & vbLf, 0)
Call viVScanf(Age506x, "%t", Dummy)

'***
'*** Checking test results.
'***

Call viVPrintf(Age506x, ":STAT:QUES:LIM?" + vbLf, 0)
Call viVScanf(Age506x, "%t", Dummy)
ret = CInt(Dummy)
Lim_Judge = ret And 2
Call viVPrintf(Age506x, ":STAT:QUES:LIM:CHAN1?" + vbLf, 0)
Call viVScanf(Age506x, "%t", Dummy)
ret = CInt(Dummy)
Tr1_Judge = ret And 2
Tr2_Judge = ret And 4

'***
'*** Displaying test results.
'***

If Lim_Judge = 0 Then
   Cells(25, 3).Value = "PASS"
Else
   Cells(25, 3).Value = "FAIL"
   If Tr1_Judge = 0 Then
      Cells(26, 3).Value = "PASS"
   Else
      Cells(26, 3).Value = "FAIL"
      Call viVPrintf(Age506x, ":CALC1:PAR1:SEL" + vbLf, 0)
      Call viVPrintf(Age506x, ":CALC1:LIM:REP:POIN?" + vbLf, 0)
      Call viVScanf(Age506x, "%t", Dummy)
      Fail_Point = CInt(Dummy)
      ReDim Fail_Data(Fail_Point - 1)
```

```
            ptr = VarPtr(Fail_Data(0))
            Call viVPrintf(Age506x, ":CALC1:LIM:REP?" + vbLf, 0)
            Call viVScanf(Age506x, "%," + CStr(Fail_Point) + "lf", ptr)
            For i = 0 To Fail_Point - 1
                Cells(26 + i, 5).Value = Fail_Data(i)
            Next i
        End If
        If Tr2_Judge = 0 Then
            Cells(27, 3).Value = "PASS"
        Else
            Cells(27, 3).Value = "FAIL"
            Call viVPrintf(Age506x, ":CALC1:PAR2:SEL" + vbLf, 0)
            Call viVPrintf(Age506x, ":CALC1:LIM:REP:POIN?" + vbLf, 0)
            Call viVScanf(Age506x, "%t", Dummy)
            Fail_Point2 = CInt(Dummy)
            ReDim Fail_Data2(Fail_Point2 - 1)
            ptr2 = VarPtr(Fail_Data2(0))
            Call viVPrintf(Age506x, ":CALC1:LIM:REP?" + vbLf, 0)
            Call viVScanf(Age506x, "%," + CStr(Fail_Point2) + "lf", ptr2)
            For i = 0 To Fail_Point2 - 1
                Cells(26 + i, 6).Value = Fail_Data2(i)
            Next i
        End If
    End If
    Call viClose(Age506x)
    Call viClose(defrm)

End Sub

'************************
'*** The public user definition type variable can not be defined in the object module.
'*** It is necessary to write the following codes in standard module, etc.
'************************
Type LimitTbl1
    Typ(3) As Integer       '1:MAX 2:MIN
    BeginStim(3) As Double
    EndStim(3) As Double
```

```
    BeginResp(3) As Double
    EndResp(3) As Double
End Type
Type LimitTbl2
    Typ(2) As Integer       '1:MAX 2:MIN
    BeginStim(2) As Double
    EndStim(2) As Double
    BeginResp(2) As Double
    EndResp(2) As Double
End Type
'************************
```

Sample Program in HT Basic (lim_test.htb)

```
10 DIM Param1$[9],Param2$[9],Fmt1$[9],Fmt2$[9],Buff$[9]
20 REAL Cent,Span,Lim1(1:4,1:5),Lim2(1:3,1:5),Fail_data(1:1601)
30 INTEGER Num_of_seg1,Num_of_seg2,Segment,Column,Fail_point
40 !
50 ASSIGN @Agte506x TO 717
60 !
70 Cent=9.475E+8
80 Span=2.00E+8
90 Param1$="S21"
100 Param2$="S11"
110 Fmt1$="MLOG"
120 Fmt2$="MLOG"
130 !
140 ! == Trace 1 Limit Line ==
150 Num_of_seg1=4 ! Number of segments: 4
160 ! -- Segment 1 --
170 Lim1(1,1)=1 ! Type : Maximum
180 Lim1(1,2)=8.475E+8 ! Frequency Start: 847.5 MHz
190 Lim1(1,3)=9.050E+8 ! Stop : 905.0 MHz
200 Lim1(1,4)=-55 ! Response Start: -55 dBm
210 Lim1(1,5)=-55 ! Stop : -55 dBm
220 ! -- Segment 2 --
230 Lim1(2,1)=2 ! Type : Minimum
240 Lim1(2,2)=9.350E+8 ! Frequency Start: 935.0 MHz
250 Lim1(2,3)=9.600E+8 ! Stop : 960.0 MHz
```

```
260 Lim1(2,4)=-3.5 ! Response Start: -3.5 dBm
270 Lim1(2,5)=-3.5 ! Stop : -3.5 dBm
280 ! -- Segment 3 --
290 Lim1(3,1)=1 ! Type : Maximum
300 Lim1(3,2)=9.350E+8 ! Frequency Start: 935.0 MHz
310 Lim1(3,3)=9.600E+8 ! Stop : 960.0 MHz
320 Lim1(3,4)=0 ! Response Start: 0 dBm
330 Lim1(3,5)=0 ! Stop : 0 dBm
340 ! -- Segment 4 --
350 Lim1(4,1)=1 ! Type : Maximum
360 Lim1(4,2)=9.800E+8 ! Frequency Start: 980.0 MHz
370 Lim1(4,3)=1.0475E+9 ! Stop : 1047.5 MHz
380 Lim1(4,4)=-25 ! Response Start: -25 dBm
390 Lim1(4,5)=-25 ! Stop : -25 dBm
400 ! == Trace 2 Limit Line ==
410 Num_of_seg2=3 ! Number of segments: 3
420 ! -- Segment 1 --
430 Lim2(1,1)=1 ! Type : Maximum
440 Lim2(1,2)=8.475E+8 ! Frequency Start: 847.5 MHz
450 Lim2(1,3)=9.250E+8 ! Stop : 925.0 MHz
460 Lim2(1,4)=0 ! Response Start: 0 dBm
470 Lim2(1,5)=0 ! Stop : 0 dBm
480 ! -- Segment 2 --
490 Lim2(2,1)=1 ! Type : Maximum
500 Lim2(2,2)=9.350E+8 ! Frequency Start: 935.0 MHz
510 Lim2(2,3)=9.600E+8 ! Stop : 960.0 MHz
520 Lim2(2,4)=-9.5 ! Response Start: -9.5 dBm
530 Lim2(2,5)=-9.5 ! Stop : -9.5 dBm
540 ! -- Segment 3 --
550 Lim2(3,1)=1 ! Type : Maximum
560 Lim2(3,2)=9.700E+8 ! Frequency Start: 970.0 MHz
570 Lim2(3,3)=1.0475E+9 ! Stop : 1047.5 MHz
580 Lim2(3,4)=0 ! Response Start: 0 dBm
590 Lim2(3,5)=0 ! Stop : 0 dBm
600 !
610 OUTPUT @Agte506x;":SENS1:FREQ:CENT ";Cent
620 OUTPUT @Agte506x;":SENS1:FREQ:SPAN ";Span
```

```
630 OUTPUT @Agte506x;":CALC1:PAR1:COUN 2"
640 OUTPUT @Agte506x;":DISP:WIND1:SPL D1_2"
650 OUTPUT @Agte506x;":TRIG:SOUR BUS"
660 OUTPUT @Agte506x;":INIT1:CONT ON"
670 !
680 ! Trace 1
690 !
700 OUTPUT @Agte506x;":CALC1:PAR1:SEL"
710 !
720 OUTPUT @Agte506x;":CALC1:PAR1:DEF "&Param1$
730 OUTPUT @Agte506x;":CALC1:FORM "&Fmt1$
740 !
750 OUTPUT @Agte506x;":CALC1:LIM:DATA ";Num_of_seg1;
760 FOR Segment=1 TO Num_of_seg1
770 FOR Column=1 TO 5
780 OUTPUT @Agte506x;",";Lim1(Segment,Column);
790 NEXT Column
800 NEXT Segment
810 OUTPUT @Agte506x;""
820 OUTPUT @Agte506x;":CALC1:LIM:DISP ON"
830 OUTPUT @Agte506x;":CALC1:LIM ON"
840 !
850 ! Trace 2
860 !
870 OUTPUT @Agte506x;":CALC1:PAR2:SEL"
880 !
890 OUTPUT @Agte506x;":CALC1:PAR2:DEF "&Param2$
900 OUTPUT @Agte506x;":CALC1:FORM "&Fmt2$
910 !
920 OUTPUT @Agte506x;":CALC1:LIM:DATA ";Num_of_seg2;
930 FOR Segment=1 TO Num_of_seg2
940 FOR Column=1 TO 5
950 OUTPUT @Agte506x;",";Lim2(Segment,Column);
960 NEXT Column
970 NEXT Segment
980 OUTPUT @Agte506x;""
990 OUTPUT @Agte506x;":CALC1:LIM:DISP ON"
```

```
1000 OUTPUT @Agte506x;":CALC1:LIM ON"
1010 !
1020 ! Setting status registers
1030 !
1040 OUTPUT @Agte506x;":STAT:QUES:LIM:CHAN1:ENAB 6"
1050 OUTPUT @Agte506x;":STAT:QUES:LIM:CHAN1:PTR 6"
1060 OUTPUT @Agte506x;":STAT:QUES:LIM:CHAN1:NTR 0"
1070 OUTPUT @Agte506x;":STAT:QUES:LIM:PTR 2"
1080 OUTPUT @Agte506x;":STAT:QUES:LIM:NTR 0"
1090 OUTPUT @Agte506x;"*CLS"
1100 !
1110 OUTPUT @Agte506x;":TRIG:SING"
1120 OUTPUT @Agte506x;"*OPC?"
1130 ENTER @Agte506x;Buff$
1140 !
1150 ! Checking test results
1160 !
1170 OUTPUT @Agte506x;":STAT:QUES:LIM?"
1180 ENTER @Agte506x;Reg_val
1190 Ch1_judge=BIT(Reg_val,1)
1200 OUTPUT @Agte506x;":STAT:QUES:LIM:CHAN1?"
1210 ENTER @Agte506x;Reg_val
1220 Tr1_judge=BIT(Reg_val,1)
1230 Tr2_judge=BIT(Reg_val,2)
1240 !
1250 ! Displaying test results
1260 !
1270 IF Ch1_judge=0 THEN
1280 PRINT "## PASS! ##"
1290 ELSE
1300 PRINT "## FAIL! ##"
1310 IF Tr1_judge=0 THEN
1320 PRINT " Trace1(S21): PASS"
1330 ELSE
1340 PRINT " Trace1(S21): FAIL"
1350 !
1360 ! Reading and displaying frequency at failed points
```

```
1370 !
1380 OUTPUT @Agte506x;":CALC1:PAR1:SEL"
1390 OUTPUT @Agte506x;":CALC1:LIM:REP:POIN?"
1400 ENTER @Agte506x;Fail_point
1410 REDIM Fail_data(1:Fail_point)
1420 OUTPUT @Agte506x;":CALC1:LIM:REP?"
1430 ENTER @Agte506x;Fail_data(*)
1440 PRINT " Frequency:"
1450 FOR I=1 TO Fail_point
1460 PRINT USING "3X,MD.4DE";Fail_data(I)
1470 NEXT I
1480 END IF
1490 IF Tr2_judge=0 THEN
1500 PRINT " Trace2(S11): PASS"
1510 ELSE
1520 PRINT " Trace2(S11): FAIL"
1530 !
1540 ! Reading and displaying frequency at failed points
1550 !
1560 OUTPUT @Agte506x;":CALC1:PAR2:SEL"
1570 OUTPUT @Agte506x;":CALC1:LIM:REP:POIN?"
1580 ENTER @Agte506x;Fail_point
1590 REDIM Fail_data(1:Fail_point)
1600 OUTPUT @Agte506x;":CALC1:LIM:REP?"
1610 ENTER @Agte506x;Fail_data(*)
1620 PRINT " Frequency:"
1630 FOR I=1 TO Fail_point
1640 PRINT USING "3X,MD.4DE";Fail_data(I)
1650 NEXT I
1660 END IF
1670 END IF
1680 END
```

Description

Line 50

Assigns a GPIB address to the I/O pass.

Lines 70 to 120

These lines store the sweep center value, sweep span value, trace 1 measurement parameter, trace 2 measurement parameter, trace 1 data format, and trace 2 data format into the variables Cent, Span, Param1$, Param2$, Fmt1$, and Fmt2$, respectively.

Line 150

Stores the number of segments in trace 1 limit table into the Num_of_seg1 variable.

Lines 160 to 390

These lines store the settings in trace 1 limit table into the Lim1(*) variable.

Line 410

Stores the number of segments in trace 2 limit table into the Num_of_seg2 variable.

Lines 420 to 590

These lines store the settings in trace 2 limit table into the Lim2(*) variable.

Lines 610 to 620

These lines configure the sweep range for channel 1's sweep range using the center and span values contained in the Cent and Span values.

Lines 630 to 660

These lines configure channel 1 so that it contains 2 traces, displays graphs in two windows tiled horizontally (i.e., with the screen split into upper and lower halves), uses a bus trigger source, and works in continuous activation mode.

Line 700

Sets channel 1's active trace to trace 1.

Lines 720 to 730

These lines store trace 1's measurement parameter and data format into the variables Param1$ and Fmt1$, respectively.

Lines 750 to 810

These lines set up the limit table for trace 1.

Line 750: Sends the command that sets up a limit table along with the Num_of_seg1 variable that contains the number of segments.

Lines 770 to 790: Sends five data items (type, start point stimulus value, end point stimulus value, start point response value, and end point response value) for each segment.

Lines 820 to 830

These lines turn ON the display of limit lines and the Limit Test feature for trace 1.

Line 870

Sets channel 1's active trace to trace 2.

Lines 890 to 900

These lines set trace 2's measurement parameter and data format to Param2$ and Fmt2$, respectively.

Lines 920 to 980

These lines set up the limit table for trace 2.

Lines 990 to 1000

These lines turn ON the display of limit lines and the Limit Test feature for trace 2.

Lines 1040 to 1060

These lines set, under the questionable limit channel 1 status register, the enable register and positive transition filter to 6 (0000000000000110 in binary notation) while setting the negative transition filter to 0 so that the questionable limit status condition register's bit 1 is set to 1 when the test result that combines the results for trace 1 and trace 2 is "fail."

The sample program provides an example of explicitly configuring the register bits so that they reflect the test results that only cover trace 1 and trace 2. However, because the results for traces 3 to 9 will never "fail" as long as the Limit Test feature is disabled for those traces, the register bits would reflect the test result that is limited to traces 1 and 2, even if the default setting is not changed.

Lines 1070 to 1080

These lines set transition filters so that the questionable limit status event register's bit 1 is set to 1 when the questionable limit status condition register's bit 1 changes from 0 to 1.

Line 1090

Clears the questionable limit status event register and questionable limit channel 1 status event register.

Lines 1110 to 1130

These lines trigger the instrument and wait until the sweep cycle is completed.

Lines 1170 to 1190

These lines retrieve the value of the questionable limit status event register and store the setting of bit 1 of the value into Ch1_judge.

Lines 1200 to 1230

These lines retrieve the value of the questionable limit channel 1 status event register and store the settings of bit 1 and bit 2 of the value into Tr1_judge and Tr2_judge, respectively.

Line 1280

Displays a message indicating that the DUT has passed the limit test if the test result for channel 1 is "Pass" (i.e., if Ch1_judge returns 0).

Lines 1300 to 1660

These lines are executed if the test result for channel 1 is "Fail" (i.e., if Ch1_judge returns 1).

Line 1300: Notifies the user that the limit test result is "Fail".

Line 1320: Displays a message indicating that trace 1 has passed the limit test if the test result for trace 1 is "Pass" (i.e., if Tr1_judge returns 0).

Lines 1340 to 1470: These lines are executed if the test result for trace 1 is "Fail" (i.e., if Tr1_judge returns 1). The lines notify the user that the test result for trace 1 is "Fail" and then retrieve and display the frequencies at the failed measurement points on trace 1.

Line 1340: Notifies the user that the limit test result for trace 1 is "Pass."

Line 1380: Sets channel 1's active trace to trace 2.

Lines 1390 to 1410: These lines retrieve the number of failed measurement points on trace 1 and, based on that number, resize the array that contains retrieved frequencies.

Lines 1420 to 1470: These lines retrieve and display the frequencies at the failed measurement points on trace 1.

Line 1500: Displays a message indicating that trace 2 has passed the limit test if the test result for trace 2 is "Pass" (i.e., if Tr2_judge returns 0).

Lines 1520 to 1650: If the test result for trace 2 is "Fail" (i.e., if Tr2_judge returns 1), these lines notify the user that trace 2 has failed to pass the limit test and then retrieve and display the frequencies at the failed measurement points on trace 2.

Bandwidth Search

- Overview
- Sample Program in Excel VBA
- Sample Program in HT Basic

Other topics about Sample Programs

Overview

The sample program demonstrates how to perform Bandwidth Search.

The sample program moves the marker to the maximum value position and then retrieves and displays the results of Bandwidth Search.

Sample Program in Excel VBA

```
Sub Bandwid_Click()
    Dim vi As Long
    Dim defrm As Long
    Dim Threshhold As Long
    Dim Result As String * 1000
    Dim Bdata As Variant

    Dim Dummy As String * 20

    Const TimeOutTime = 10000

    Call viOpenDefaultRM(defrm)
    Call viOpen(defrm, "GPIB0::17::INSTR", 0, 0, vi)
    Call viSetAttribute(vi, VI_ATTR_TMO_VALUE, TimeOutTime)

    Threshhold = -3
    Call viVPrintf(vi, ":SENS1:FREQ:CENT 947.5E6" & vbLf, 0)
    Call viVPrintf(vi, ":SENS1:FREQ:SPAN 200E6" & vbLf, 0)
    Call viVPrintf(vi, ":CALC1:PAR1:DEF S21" & vbLf, 0)
    Call viVPrintf(vi, ":DISP:WIND1:TRAC1:Y:AUTO" & vbLf, 0)
    Call viVPrintf(vi, ":CALC1:PAR1:SEL" & vbLf, 0)
    Call viVPrintf(vi, ":INIT1:CONT ON" & vbLf, 0)
    Call viVPrintf(vi, ":TRIG:SOUR BUS" & vbLf, 0)
    Call viVPrintf(vi, "*OPC?" & vbLf, 0)
    Call viVScanf(vi, "%t", Dummy)

    Call viVPrintf(vi, ":TRIG:SING" & vbLf, 0)
```

```
    Call viVPrintf(vi, "*OPC?" & vbLf, 0)
    Call viVScanf(vi, "%t", Dummy)

    Call viVPrintf(vi, ":CALC1:MARK1 ON" & vbLf, 0)

    Call viVPrintf(vi, ":CALC1:MARK1:FUNC:TYPE MAX" & vbLf, 0)
    Call viVPrintf(vi, ":CALC1:MARK1:FUNC:EXEC" & vbLf, 0)
    Call viVPrintf(vi, ":CALC1:MARK1:BWID:THR " + CStr(Threshhold) & vbLf, 0)
    Call viVPrintf(vi, ":CALC1:MARK:BWID ON" & vbLf, 0)
    Call viVPrintf(vi, "*OPC?" & vbLf, 0)
    Call viVScanf(vi, "%t", Dummy)

    Call viVPrintf(vi, ":CALC1:MARK1:BWID:DATA?" & vbLf, 0)
    Call viVScanf(vi, "%t", Result)
    Bdata = Split(Result, ",")
    Call ErrorCheck(vi)
    Cells(5, 2).Value = Val(Bdata(0))
    Cells(6, 2).Value = Val(Bdata(1))
    Cells(7, 2).Value = Val(Bdata(2))
    Cells(8, 2).Value = Val(Trim(Bdata(3)))

    Call viClose(vi)
    Call viClose(defrm)
End Sub
Sub ErrorCheck(vi)
    Dim err As String * 50, ErrNo As Variant, Response
    Call viVQueryf(vi, ":SYST:ERR?" & vbLf, "%t", err)
    ErrNo = Split(err, ",")
    If Val(ErrNo(0)) <> 0 Then
        Response = MsgBox(CStr(ErrNo(1)), vbOKOnly)
    End If
End Sub
```

Sample Program in HT Basic (bandwid.htb)

```
10 DIM Buff$[9],Err_msg$[100]
20 REAL Threshold,Bwid,Cent,Q,Loss
30 INTEGER Err_no
```

```
40 !
50 ASSIGN @Agte506x TO 717
60 Threshold=-3
70 !
80 OUTPUT @Agte506x;"*ESE 60"
90 OUTPUT @Agte506x;"*SRE 32"
100 OUTPUT @Agte506x;"*CLS"
110 OUTPUT @Agte506x;"*OPC?"
120 ENTER @Agte506x;Buff$
130 ON INTR 7 GOTO Err
140 ENABLE INTR 7;2
150 !
160 OUTPUT @Agte506x;":CALC1:PAR1:SEL"
170 OUTPUT @Agte506x;":CALC1:MARK1:FUNC:TYPE MAX"
180 OUTPUT @Agte506x;":CALC1:MARK1:FUNC:EXEC"
190 OUTPUT @Agte506x;":CALC1:MARK1:BWID:THR ";Threshold
200 OUTPUT @Agte506x;":CALC1:MARK1:BWID:DATA?"
210 WAIT .5
220 ENTER @Agte506x;Bwid,Cent,Q,Loss
230 !
240 PRINT "## Bandwidth Search ##"
250 PRINT "Bandwidth : ",Bwid
260 PRINT "Center Frequency: ",Cent
270 PRINT "Q : ",Q
280 PRINT "Loss : ",Loss
290 !
300 GOTO No_err
310 Err: OFF INTR 7
320 OUTPUT @Agte506x;";:SYST:ERR?"
330 ENTER @Agte506x;Err_no,Err_msg$
340 PRINT "Error occurred!!"
350 PRINT " No:";Err_no,"Description: "&Err_msg$
360 No_err: OFF INTR 7
370 END
```

Saving Files

- Overview
- Sample Program in Excel VBA
- Sample Program in HT Basic

Other topics about Sample Programs

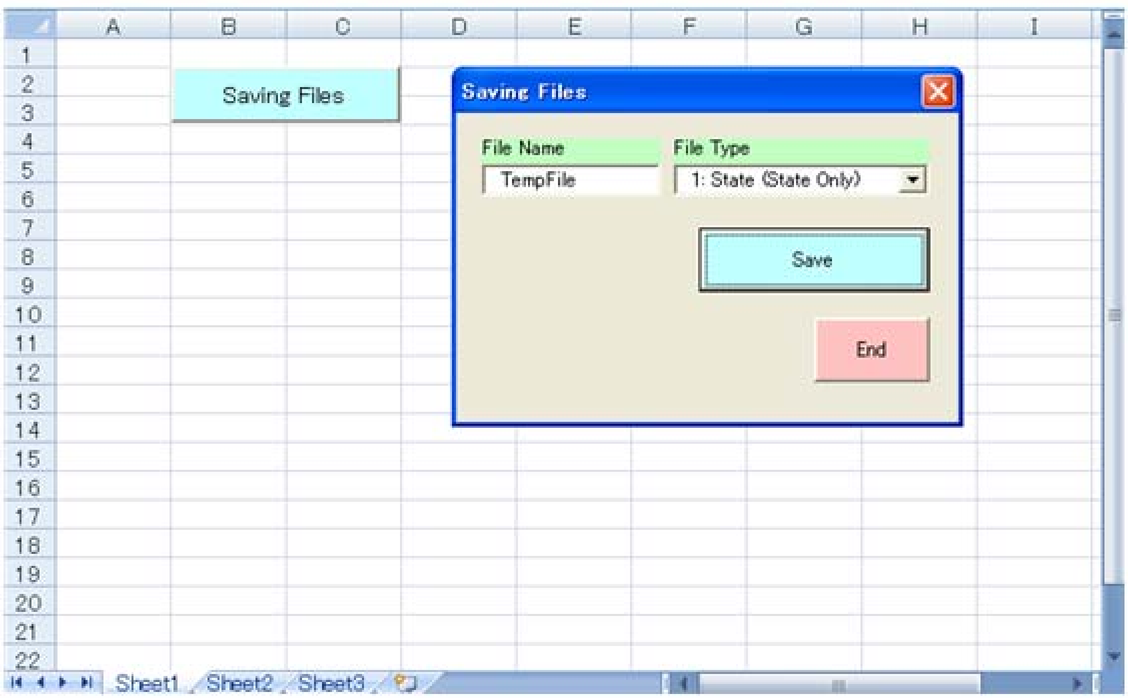
Overview

The sample program demonstrates how to save a file. This program saves selected content on a file with a specified name.

See Saving and Recalling File for this programming.

Sample Program in Excel VBA

Example of excel sheet and window form with saving files program




```
Private Sub File_Save_Click()
   ' Declare two string variables for file name and file type
   Dim File_Name As String
   Dim File_Type As String
   Dim defrm As Long
   Dim vi As Long
   Const TimeOutTime = 10000
   ' Check whether file name textbox is empty or not
   If TextBox1.Text <> "" Then
```

```
        File_Name = Trim(TextBox1.Text)
        File_Type = Trim(frmFileSave.ComboBox1.Value)
    ' Open connection to the E5061B
        Call viOpenDefaultRM(defrm)
        Call viOpen(defrm, "GPIB0::17::INSTR", 0, 0, vi)
        Call viSetAttribute(vi, VI_ATTR_TMO_VALUE, TimeOutTime)
        Select Case File_Type
            Case "1: State (State Only)"
                Call viVPrintf(vi, ":MMEM:STOR:STYP STAT" + vbLf, 0)
                Call viVPrintf(vi, ":MMEM:STOR """ & File_Name & ".sta""" + vbLf, 0)
            Case "2: State (State & Cal)"
                Call viVPrintf(vi, ":MMEM:STOR:STYP CST" + vbLf, 0)
                Call viVPrintf(vi, ":MMEM:STOR """ & File_Name & ".sta""" + vbLf, 0)
            Case "3: State (State & Trace)"
                Call viVPrintf(vi, ":MMEM:STOR:STYP DST" + vbLf, 0)
                Call viVPrintf(vi, ":MMEM:STOR """ & File_Name & ".sta""" + vbLf, 0)
            Case "4: State (All)"
                Call viVPrintf(vi, ":MMEM:STOR:STYP CDST" + vbLf, 0)
                Call viVPrintf(vi, ":MMEM:STOR """ & File_Name & ".sta""" + vbLf, 0)
            Case "5: Trace Data (CSV)"
                Call viVPrintf(vi, ":MMEM:STOR:FDAT """ & File_Name & ".csv""" + vbLf, 0)
            Case "6: Screen Image (BMP)"
                Call viVPrintf(vi, ":MMEM:STOR:IMAG """ & File_Name & ".bmp""" + vbLf, 0)
            Case Else
                MsgBox "Error in code"
        End Select
        Call viClose(defrm)
    Else
        MsgBox "Please enter a filename"
    End If
End Sub

Private Sub UserForm_Initialize()
  ComboBox1.AddItem "1: State (State Only)"
  ComboBox1.AddItem "2: State (State & Cal)"
  ComboBox1.AddItem "3: State (State & Trace)"
  ComboBox1.AddItem "4: State (All)"
```

```
  ComboBox1.AddItem "5: Trace Data (CSV)"
  ComboBox1.AddItem "6: Screen Image (BMP)"

  ComboBox1.ListIndex = 0

  TextBox1.Text = "TempFile"
End Sub

Private Sub EndBtn_Click()
  End
End Sub
```

Sample Program in HT Basic (file_sav.htb)

```
10 DIM File$[300],Inp_char$[30]
20 INTEGER Content
30 CLEAR SCREEN
40 ASSIGN @Agte506x TO 717
50 !
60 ON ERROR GOTO Content_select
70 Content_select: !
80 PRINT "## Save Content Selection ##"
90 PRINT "Select Content"
100 PRINT " 1: State (State only)"
110 PRINT " 2: State (State & Cal)"
120 PRINT " 3: State (State & Trace)"
130 PRINT " 4: State (State & Cal & Trace)"
140 PRINT " 5: Trace Data (CSV)"
150 PRINT " 6: Screen"
160 PRINT ""
170 PRINT "Input 1 to 6"
180 INPUT "Number?",Inp_char$
190 Content=IVAL(Inp_char$,10)
200 IF Content<1 OR Content>6 THEN Content_select
210 OFF ERROR
220 !
230 CALL Inp_file_name(File$)
240 !
```

```
250 SELECT Content
260 CASE 1
270 OUTPUT @Agte506x;":MMEM:STOR:STYP STAT"
280 OUTPUT @Agte506x;":MMEM:STOR """&File$&".sta"""
290 CASE 2
300 OUTPUT @Agte506x;":MMEM:STOR:STYP CST"
310 OUTPUT @Agte506x;":MMEM:STOR """&File$&".sta"""
320 CASE 3
330 OUTPUT @Agte506x;":MMEM:STOR:STYP DST"
340 OUTPUT @Agte506x;":MMEM:STOR """&File$&".sta"""
350 CASE 4
360 OUTPUT @Agte506x;":MMEM:STOR:STYP CDST"
370 OUTPUT @Agte506x;":MMEM:STOR """&File$&".sta"""
380 CASE 5
390 OUTPUT @Agte506x;":MMEM:STOR:FDAT """&File$&".csv"""
400 CASE 6
410 OUTPUT @Agte506x;":MMEM:STOR:IMAG """&File$&".bmp"""
420 END SELECT
430 !
440 END
450 !============================================
460 ! File Name Input Function
470 !============================================
480 SUB Inp_file_name(Inp_name$)
490 DIM Inp_char$[9]
500 ON ERROR GOTO Inp_start
510 Inp_start: !
520 PRINT "## File Name Input ##"
530 PRINT "Input Save File Name (without Extension)"
540 INPUT "Name?",Inp_name$
550 PRINT "Input Name: "&Inp_name$
560 INPUT "OK? [Y/N]",Inp_char$
570 IF UPC$(Inp_char$)<>"Y" THEN Inp_start
580 OFF ERROR
590 SUBEND
```

Transferring Files

- Overview
- Sample Program in Excel VBA using VISA
- Sample Program in Excel VBA using VISA-COM
- Sample Program in HT Basic

Other topics about Sample Programs

Overview

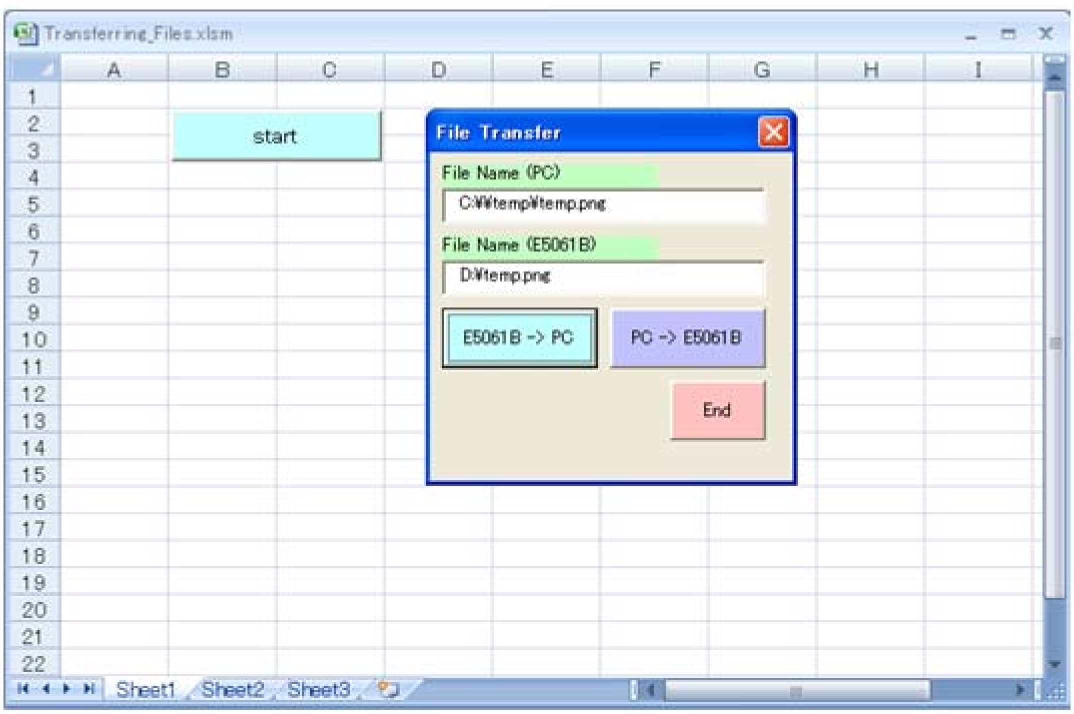
This sample program demonstrates hw to transfer files between the external controller and the E5061B.

This program reads out data from a specified file on the external controller (or the E5061B), then writes them to a specified file on the E5061B (or the external controller).

See Managing Files for this programming.

Sample Program in Excel VBA using VISA

Example of excel sheet and window form with transferring files program




```
Private Sub fromE5061B_toPC_Click()
    '*** This sequence is a sample code in which the file is transferred
```

```
'*** from the E5061B to the external controller.
Dim defrm As Long
Dim Age506x As Long
Dim Res As Variant

'*** The maximum size of transferring file is 5000000 byte.
Dim byteData(5000000) As Byte
Dim PtrData(1) As Long
Dim RetCnt As Long
Dim workData() As Byte
Dim hFile As Long
Dim isOpen As Boolean

Dim E5061B_File As String
Dim PC_File As String

PtrData(0) = VarPtr(RetCnt)
PtrData(1) = VarPtr(byteData(0))
RetCnt = UBound(byteData) - LBound(byteData) + 1

E5061B_File = """" & Trim(TextBox2.Text) & """"
PC_File = TextBox1.Text

Call viOpenDefaultRM(defrm)
Call viOpen(defrm, "GPIB0::17::INSTR", 0, 0, Age506x)
Call viSetAttribute(Age506x, VI_ATTR_TMO_VALUE, 10000)
Call viVPrintf(Age506x, ":MMEM:TRAN? " + E5061B_File + vbLf, 0)

Call viVScanf(Age506x, "%#b", PtrData(0))

ReDim workData(RetCnt - 1)

For i = 0 To RetCnt - 2
  workData(i) = byteData(i)
Next i

hFile = FreeFile()
```

```
    Open PC_File For Binary Access Write Shared As hFile
    isOpen = True

    Put #hFile, , workData
    If isOpen Then Close #hFile

    Call viClose(Age506x)
End Sub

Private Sub fromPC_toE5061B_Click()
    '*** This sequence is a sample code in which the file is transferred
    '*** from the external controller to the E5061B.
    Dim hFile As Long
    Dim isOpen As Boolean
    Dim byteData() As Byte
    Dim strBuf As String
    Dim fileSize As Long
    Dim PtrData As Long

    Dim defrm As Long
    Dim Age506x As Long

    Dim E5061B_File As String
    Dim PC_File As String

    E5061B_File = """" & Trim(TextBox2.Text) & """"
    PC_File = TextBox1.Text

    fileSize = FileLen(PC_File)
    ReDim byteData(fileSize - 1)
    PtrData = VarPtr(byteData(0))
    hFile = FreeFile()
    Open PC_File For Binary Access Read Shared As hFile
    isOpen = True

    Get #hFile, , byteData
    If isOpen Then Close #hFile
```

```
    Call viOpenDefaultRM(defrm)
    Call viOpen(defrm, "GPIB0::17::INSTR", 0, 0, Age506x)
    Call viSetAttribute(Age506x, VI_ATTR_TMO_VALUE, 10000)
    strBuf = ":MMEM:TRAN " & E5061B_File & "," & "%#" & CLng(fileSize - 1) & "b" & vbLf
    Call viVPrintf(Age506x, strBuf, PtrData)

    Call viClose(Age506x)
End Sub

Private Sub UserForm_Initialize()
  TextBox1.Text = "C:\\temp\temp.png"
  TextBox2.Text = "D:\temp.png"
End Sub

Private Sub EndBtn_Click()
    End
End Sub
```

Sample Program in Excel VBA using VISA-COM

```
Private Sub fromE5061B_toPC_Click()
    '*** This sequence is a sample code in which the file is transferred
    '*** from the E5061B to the external controller.
    Dim hFile As Long
    Dim isOpen As Boolean
    Dim ioMgr As VisaComLib.ResourceManager
    Set ioMgr = New VisaComLib.ResourceManager

    Dim Age506x As VisaComLib.FormattedIO488
    Set Age506x = New VisaComLib.FormattedIO488

    Set Age506x.IO = ioMgr.Open("GPIB0::17::INSTR")
    Age506x.IO.Timeout = 10000

    Dim byteData() As Byte
    Dim E5061B_File As String
    Dim PC_File As String
```

```
    E5061B_File = """" & Trim(TextBox2.Text) & """"
    PC_File = Trim(TextBox1.Text)

    Age506x.WriteString ":MMEM:TRAN? " & E5061B_File
    byteData = Age506x.ReadIEEEBlock(BinaryType_UI1)

    hFile = FreeFile()
    Open PC_File For Binary Access Write Shared As hFile
    isOpen = True

    Put #hFile, , byteData
    If isOpen Then Close #hFile
    Age506x.IO.Close
End Sub

Private Sub fromPC_toE5061B_Click()
    '*** This sequence is a sample code in which the file is transferred
    '*** from the external controller to the E5061B.
    Dim hFile As Long
    Dim isOpen As Boolean
    Dim ioMgr As VisaComLib.ResourceManager
    Set ioMgr = New VisaComLib.ResourceManager

    Dim Age506x As VisaComLib.FormattedIO488
    Set Age506x = New VisaComLib.FormattedIO488

    Set Age506x.IO = ioMgr.Open("GPIB0::17::INSTR")
    Age506x.IO.Timeout = 10000

    Dim byteData() As Byte
    Dim strBuf As String
    Dim fileSize As Long
    Dim E5061B_File As String
    Dim PC_File As String

    E5061B_File = """" & Trim(TextBox2.Text) & """"
    PC_File = Trim(TextBox1.Text)
```

```
    fileSize = FileLen(PC_File)
    ReDim byteData(fileSize - 1)
    hFile = FreeFile()
    Open PC_File For Binary Access Read Shared As hFile
    isOpen = True

    Get #hFile, , byteData
    If isOpen Then Close #hFile
    strBuf = ":MMEM:TRAN " & E5061B_File & ","
    Age506x.WriteIEEEBlock strBuf, byteData, True

    Age506x.IO.Close
  End Sub

  Private Sub UserForm_Initialize()
   TextBox1.Text = "C:\\temp\temp.png"
   TextBox2.Text = "D:\temp.png"
  End Sub

  Private Sub EndBtn_Click()
   End
  End Sub
```

Sample Program in HT Basic (file_xfr.htb)

```
10 DIM Src_file$[50],Dst_file$[50],Src_size_char$[50],Inp_char$[30]
20 INTEGER Direction
30 ASSIGN @Agte506x TO 717
40 !
50 CLEAR SCREEN
60 ON ERROR GOTO Direct_select
70 Direct_select: !
80 PRINT "#### File Transfer ####"
90 PRINT " 1: E506x -> Controller"
100 PRINT " 2: Controller -> E506x"
110 PRINT ""
120 PRINT "Input 1 or 2"
```

```
130 INPUT "Number?",Inp_char$
140 Direction=IVAL(Inp_char$,10)
150 IF Direction<1 OR Direction>2 THEN Direct_select
160 OFF ERROR
170 !
180 PRINT ""
190 PRINT " Input source file name. ";
200 INPUT "Name?",Src_file$
210 PRINT ": "&Src_file$
220 !
230 IF Direction=2 THEN
240 PRINT " Input source file size. ";
250 INPUT "Size[Byte]?",Src_size_char$
260 PRINT ": "&Src_size_char$&"[Byte]"
270 END IF
280 !
290 PRINT " Input destination file name. ";
300 INPUT "Name?",Dst_file$
310 PRINT ": "&Dst_file$
320 PRINT ""
330 !
340 IF Direction=1 THEN
350 Copy_to_contr(@Agte506x,Src_file$,Dst_file$)
360 ELSE
370 Copy_to_e506x(@Agte506x,Src_file$,Src_size_char$,Dst_file$)
380 END IF
390 !
400 END
410 !=============================================
420 ! File Transfer Function (E506x -> Controller)
430 !=============================================
440 SUB Copy_to_contr(@Agte506x,Src_file$,Dst_file$)
450 DIM Img$[32],Src_size_char$[10],Buff$[9],Err_msg$[100]
460 INTEGER Max_bsize,Block_size,Err_no
470 REAL Src_size
480 !
490 ON ERROR GOTO Skip_purge
```

```
500 PURGE Dst_file$
510 Skip_purge: OFF ERROR
520 CREATE Dst_file$,1
530 ASSIGN @Dst_file TO Dst_file$
540 Max_bsize=24576 ! 24KByte
550 !
560 OUTPUT @Agte506x;"*ESE 60"
570 OUTPUT @Agte506x;"*SRE 32"
580 OUTPUT @Agte506x;"*CLS"
590 OUTPUT @Agte506x;"*OPC?"
600 ENTER @Agte506x;Buff$
610 !
620 ON INTR 7 GOTO Err
630 ENABLE INTR 7;2
640 PRINT "Now Copying: "&Src_file$&"(@E506x) -> "&Dst_file$&"(@Contro ller)"
650 OUTPUT @Agte506x;":MMEM:TRAN? """&Src_file$&""""
660 WAIT .1
670 ENTER @Agte506x USING "#,A";Buff$
680 ENTER @Agte506x USING "#,A";Digit$
690 Img$="#,"&Digit$&"A"
700 ENTER @Agte506x USING Img$;Src_size_char$
710 !
720 Src_size=VAL(Src_size_char$)
730 WHILE Src_size>0
740 IF Src_size>Max_bsize THEN
750 Block_size=Max_bsize
760 ELSE
770 Block_size=Src_size
780 END IF
790 !
800 ALLOCATE Dat$[Block_size]
810 Img$="#,"&VAL$(Block_size)&"A"
820 ENTER @Agte506x USING Img$;Dat$
830 OUTPUT @Dst_file USING Img$;Dat$
840 DEALLOCATE Dat$
850 !
860 Src_size=Src_size-Block_size
```

```
870 END WHILE
880 !
890 PRINT "Done"
900 ENTER @Agte506x USING "#,A";Buff$
910 ASSIGN @Dst_file TO *
920 !
930 GOTO Skip_error
940 Err: OFF INTR 7
950 OUTPUT @Agte506x;";:SYST:ERR?"
960 ENTER @Agte506x;Err_no,Err_msg$
970 PRINT "Error occurred!!"
980 PRINT " No:";Err_no,"Description: "&Err_msg$
990 Skip_error: OFF INTR 7
1000 SUBEND
1010 !============================================
1020 ! File Transfer Function (Controller -> E506x)
1030 !============================================
1040 SUB Copy_to_e506x(@Agte506x,Src_file$,Src_size_char$,Dst_file$)
1050 DIM Img$[32],Header$[10],Buff$[9],Err_msg$[100]
1060 INTEGER Max_bsize,Block_size,Err_no
1070 REAL Src_size
1080 !
1090 ON ERROR GOTO File_error
1100 ASSIGN @Src_file TO Src_file$
1110 OFF ERROR
1120 Max_bsize=24576 ! 24KByte
1130 !
1140 OUTPUT @Agte506x;"*CLS"
1150 OUTPUT @Agte506x;"*OPC?"
1160 ENTER @Agte506x;Buff$
1170 !
1180 PRINT "Now Copying: "&Src_file$&"(@Controller) -> "&Dst_file$&"(@ E506x)"
1190 Header$="#"&VAL$(LEN(Src_size_char$))&Src_size_char$
1200 OUTPUT @Agte506x;":MMEM:TRAN """&Dst_file$&""","&Header$;
1210 !
1220 Src_size=VAL(Src_size_char$)
1230 WHILE Src_size>0
```

```
1240 IF Src_size>Max_bsize THEN
1250 Block_size=Max_bsize
1260 ELSE
1270 Block_size=Src_size
1280 END IF
1290 !
1300 ALLOCATE Dat$[Block_size]
1310 Img$="#,"&VAL$(Block_size)&"A"
1320 ENTER @Src_file USING Img$;Dat$
1330 OUTPUT @Agte506x USING Img$;Dat$
1340 DEALLOCATE Dat$
1350 !
1360 Src_size=Src_size-Block_size
1370 END WHILE
1380 !
1390 OUTPUT @Agte506x;"",END
1400 ASSIGN @Src_file TO *
1410 !
1420 OUTPUT @Agte506x;";:SYST:ERR?"
1430 ENTER @Agte506x;Err_no,Err_msg$
1440 IF Err_no=0 THEN
1450 PRINT "Done"
1460 ELSE
1470 PRINT "Error occurred!!"
1480 PRINT " No:";Err_no,"Description: "&Err_msg$
1490 END IF
1500 GOTO Skip_error
1510 File_error:OFF ERROR
1520 PRINT "File name NOT found!"
1530 Skip_error:!
1540 SUBEND
```

Description

Line 40

Assigns a GPIB address to the I/O pass.

Lines 60 to 130

These line allow the user to return to the entry start line and re-enter the data if an error (such as an invalid entry) occurs while entering the number that indicates the transfer direction. Then, these line display the list of transfer directions and prompt the user to input a selected number.

Lines 80 to 130

These lines display the list of transfer directions, and prompt the user to choose one of the items by typing in the appropriate number.

Lines 140 to 150

Converts the entered value into an integer and stores it into the Direction variable. Returns to the entry start line if an invalid value is contained in Direction.

Lines 180 to 210

These lines obtain the name of the source file for copying from the user input, store it into the Src_file$ variable, and display the value of Src_file$.

Lines 180 to 210

These lines obtain the name of the source file for copying from the user input, store it into the Src_file$ variable, and display the value of Src_file$.

Lines 230 to 270

If Direction is equal to 2 (from the external controller to the E5061B), these lines obtain the size of the source file for copying, store it into the Src_size_char$, and display the value of Src_size_char$.

Lines 290 to 320

These lines obtain the name of the destination file for copying from the user input, store it into the Dst_file$ variable, and display the value of Dst_file$.

Line 350

If Direction is equal to 1 (from the E5061B to the external controller), these lines use the subprogram Copy_to_contr to transfer (copy) a file with the name Src_file$ on the E5061B to a file with the name Dst_file$ on the external controller.

Line 370

If Direction is equal to 2, these lines use the subprogram Copy_to_e506x to transfer (copy) a file with the name Src_file$ on the external controller to a file with the name Dst_file$ on the E5061B.

Copy_to_contr, a subprogram for transferring files from the E5061B to the external controller that appears in lines 440 to 1000, is described below.

Lines 490 to 520

If any file with the name File$ already exists, these lines delete the file and create a new file with the name File$.

Line 530

Assigns a destination file for copying to the I/O pass.

Line 540

This line stores a maximum number of transferred data (in bytes) per one transfer, that is 24 KByte to meet the size limitation of string arrays in the HTBasic, into Max_bsize variable.

Lines 560 to 600

These lines configure the system to generate an SRQ when it cannot find a source file for copying due to an error.

Lines 620 to 630

These lines set the branch target for an SRQ interrupt to enable SRQ interrupts.

Lines 640 to 650

These lines display a message showing that the transfer has started, and execute commands for reading data from a file on the E5061B.

Lines 670 to 680

These lines read the header symbol (#) in a block data, read number of digits (characters) indicating the size of data in bytes, then store it into Digit$ variable.

Line 690

This line creates a format for reading characters in Digit$.

Line 700

This line reads the data size in byte and stores it into Src_size_char$ variable.

Line 720

This line converts Src_size_char$ to a real number and stores it into Src_size variable.

Lines 730 to 870

These lines repeat the procedures below until Src_size reaches 0.

Lines 740 to 780: If Src_size is greater than Max_bsize, these lines assign the value of the Max_bsize to Block_size variable (transferred data in bytes). If Src_size is equal or less than Max_bsize, assign the value of Src_size to Block_size.

Line 800 This line defines Dat$ string variable with the size as large as Block_size and reserves memory area.

Line 810 This line creates a format for reading characters as many as Block_size characters.

Line 820 This line reads data from the file on the E5061B, then stores them into Dat$.

Line 830 This line writes the contents of Dat$ to the file on the external controller.

Lines 840 to 860 These lines free the memory area for Dat$ and subtract Block_size from Src_size.

Lines 890 to 900

These lines display a message showing the completion of transfer, then read a message terminator at the end of the data.

Lines 940 to 980

These lines define an error handler that retrieves and displays the number and message of an error that has occurred.

Copy_to_e506x, a subprogram for transferring files from the external controller to the E5061B that appears in lines 1040 to 1540, is described below.

Lines 1090 to 1110

Assigns a destination file for copying to the I/O pass.

Line 1120

This line stores a maximum number of transferred data (in bytes) per one transfer, that is 24 KByte, into Max_bsize variable.

Lines 1140 to 1160

Clears the error queue.

Line 1180

Displays a measurement start message.

Lines 1190 to 1200

These lines create the header part indicating that data will be sent as many as Src_size_char$ bytes, then send the header part of the command and its parameters for writing the data to the file on the E5061B.

Line 1220

This line converts Src_size_char$ to a real number and stores it into Src_size variable.

Lines 1230 to 1370

These lines repeat the procedures below until Src_size reaches 0.

Lines 1240 to 1280: If Src_size is greater than Max_bsize, these lines assign the value of the Max_bsiize to Block_size variable (transferred data in bytes). If Src_size is equal or less than Max_bsize, assign the value of Src_size to Block_size.

Line 1300 This line defines Dat$ string variable with the size as large as Block_size and reserves memory area.

Line 1310 This line creates a format for reading characters as many as Block_size characters.

Line 1320 This line reads data from the file on the external controller, then stores them into Dat$.

Line 1330 This line writes the contents of Dat$ to the file on the E5061B.

Lines 1340 to 1360 These lines free the memory area for Dat$ and subtract Block_size from Src_size.

Line 1390

This line sends a message terminator at the end of data.

Lines 1420 to 1430

These lines retrieve the error number and error message from the error queue, and then store them into the variables Err_no and Err_msg$, respectively.

Lines 1440 to 1490

If Err_no is equal to 0 (no error occurred), these lines display the message indicating completion of transfer, and if Err_no is not equal to 0 (an error occurred), display Err_no along with Err_msg$.

Lines 1510 to 1520

These lines handle the case with no source file for copying is found.

Time Domain

- Overview
- Sample Program in Excel VBA
- Sample Program in HT Basic

Other topics about Sample Programs

Overview

This sample program demonstrates how to use the transformation function of the time domain function.

This program executes calibration (ECal), performs measurement once, converts the results to data in the time domain, and displays this data.

See Analysis in Time Domain (time domain function) for this programming.

Sample Program in Excel VBA

```
Private Sub Time_Domain_Click()
    Dim defrm As Long                   'Session to Default Resource Manager
    Dim vi As Long                      'Session to instrument
    Dim Para As String
    Dim Tran_Type As String
    Dim Stim_Type As String
    Dim stop_freq As Double
    Dim Win_Beta As Double
    Dim Star_Time As Double
    Dim Stop_Time As Double
    Dim Result As String

    Const TimeOutTime = 40000           'timeout time.

    stop_freq = 3000000000#             'Stop Frequendy  : 3 GHz
    Nop = 201                           'Nop             : 201
    Para = "S11"                        'Meas. Parameter : S11
    Tran_Type = "LPAS"                  'Transform Type  : Lowpass
    Stim_Type = "IMP"                   'Stimulus Type   : Impulse
    Win_Beta = 13                       'Window Beta     : 13 (Maximum Type)
    Star_Time = 0                       'Start time      : 0 sec
    Stop_Time = 0.00000001              'Stop time       : 10 nsec

    Call viOpenDefaultRM(defrm)                         'Initializes the VISA system.
```

```
    Call viOpen(defrm, "GPIB0::17::INSTR", 0, 0, vi)                'Opens the session to the
specified instrument.
    Call viSetAttribute(vi, VI_ATTR_TMO_VALUE, TimeOutTime)           'The state of an attribute
for the specified session.

    Call viVPrintf(vi, ":SYST:PRES" & vbLf, 0)                  'Presets the setting state of the ENA.
    Call viVPrintf(vi, "*CLS" & vbLf, 0)                     'Clears the all status register.

    Call viVPrintf(vi, ":SENS1:FREQ:STOP " & stop_freq & vbLf, 0)        'Sets the sweep stop
frequency.
    Call viVPrintf(vi, ":SENS1:SWE:POIN " & Nop & vbLf, 0)          'Sets the number of points.

    Call viVPrintf(vi, ":CALC1:TRAN:TIME:LPFR" & vbLf, 0)           'Sets a measurement point.

    Call viVPrintf(vi, ":CALC1:PAR1:DEF " & Para & vbLf, 0)          'Sets the measurement
parameter.
    Call viVPrintf(vi, ":TRIG:SOUR BUS" & vbLf, 0)               'Sets the trigger source to BUS.
    Call ErrorCheck(vi)                               'Checking the error.

    MsgBox "Connect Port1 to Ecal Module. Then click OK button.", vbOKOnly      'Display the message
box.

    Call viVPrintf(vi, ":SENS1:CORR:COLL:ECAL:SOLT1 1" & vbLf, 0)        'Execute the 1-port
calibration.
    Call viVPrintf(vi, "*OPC?" & vbLf, 0)                    'Sets the *OPC? command.
    Call viVScanf(vi, "%t", Result)                       'Reads the *OPC? result.
    Call ErrorCheck(vi)                               'Checking the error.
    MsgBox "Set DUT. Then click OK button.", vbOKOnly              'Display the message box.

    Call viVPrintf(vi, ":TRIG:SING" & vbLf, 0)                  'Execute the trigger.
    Call viVPrintf(vi, "*OPC?" & vbLf, 0)                    'Sets the *OPC? command.
    Call viVScanf(vi, "%t", Result)                       'Reads the *OPC? result.
    Call viVPrintf(vi, ":DISP:WIND1:TRAC1:Y:AUTO" & vbLf, 0)          'Execute auto scale.

    MsgBox "Click OK button. < Time Domain Transform >", vbOKOnly         'Display the message
box.

    Call viVPrintf(vi, ":CALC1:TRAN:TIME " & Tran_Type & vbLf, 0)        'Sets the transformation
type.
```

```
   Call viVPrintf(vi, ":CALC1:TRAN:TIME:STIM " & Stim_Type & vbLf, 0)           'Sets the stimulus type.
   Call viVPrintf(vi, ":CALC1:TRAN:TIME:KBES " & Win_Beta & vbLf, 0)            'Sets the beta value of
the window.
   Call viVPrintf(vi, ":CALC1:TRAN:TIME:STAR " & Star_Time & vbLf, 0)           'Sets the start value of
the display range.
   Call viVPrintf(vi, ":CALC1:TRAN:TIME:STOP " & Stop_Time & vbLf, 0)           'Sets the end value of
the display range.
   Call viVPrintf(vi, ":CALC1:TRAN:TIME:STAT ON" & vbLf, 0)                  'Turns on the time domain
function.
   Call ErrorCheck(vi)                                          'Checking the error.

   Call viVPrintf(vi, ":CALC1:PAR1:SEL" & vbLf, 0)                        'Sets the active trace.
   Call viVPrintf(vi, ":CALC1:FORM REAL" & vbLf, 0)                        'Sets the real data format.
   Call viVPrintf(vi, ":DISP:WIND1:TRAC1:Y:AUTO" & vbLf, 0)                 'Execute auto scale.
   Call ErrorCheck(vi)                                          'Checking the error.

   Call viClose(vi)                                           'Closes the resource manager session.
   Call viClose(defrm)                                          'Breaks the communication and terminates
the VISA system.

   End                                                      'End
End Sub

Private Sub EndBtn_Click()
   End
End Sub

Sub ErrorCheck(vi)
   Dim err As String * 50, ErrNo As Variant, Response As VbMsgBoxResult
   Call viVQueryf(vi, ":SYST:ERR?" & vbLf, "%t", err)
   ErrNo = Split(err, ",")
   If Val(ErrNo(0)) <> 0 Then
      Response = MsgBox(CStr(ErrNo(1)), vbOKOnly)
   End If
End Sub
```

Sample Program in HT Basic

```
10 DIM Para$[9],Tran_type$[9],Stim_type$[9],Buff$[9],Inp_ch ar$[9]
20 REAL Stop_freq,Win_beta,Star_time,Stop_time
```

```
30 INTEGER Nop
40 !
50 ASSIGN @Agte506x TO 717
60 !
70 Stop_freq=3.E+9 ! Stop Frequency : 3 GHz
80 Nop=201 ! Nop : 201
90 Para$="S11" ! Meas. Parameter : S11
100 !
110 Tran_type$="LPAS" ! Transform Type : Lowpass
120 Stim_type$="IMP" ! Stimulus Type : Impulse
130 Win_beta=13 ! Window Beta : 13 (Maximum Type)
140 Star_time=0 ! Start time : 0 s
150 Stop_time=1.E-8 ! Stop time : 10 ns
160 !
170 OUTPUT @Agte506x;":SYST:PRES"
180 OUTPUT @Agte506x;":SENS1:FREQ:STOP ";Stop_freq
190 OUTPUT @Agte506x;":SENS1:SWE:POIN ";Nop
200 !
210 OUTPUT @Agte506x;":CALC1:TRAN:TIME:LPFR"
220 !
230 OUTPUT @Agte506x;":CALC1:PAR1:DEF "&Para$
240 OUTPUT @Agte506x;":TRIG:SOUR BUS"
250 !
260 ! 1 Port Full Calibration (ECal)
270 !
280 PRINT "Connect Port 1 to ECal Module. Then push [Enter] key."
290 INPUT "",Buff$
300 OUTPUT @Agte506x;":SENS1:CORR:COLL:ECAL:SOLT1 1"
310 OUTPUT @Agte506x;":SYST:ERR?"
320 ENTER @Agte506x;Buff$
330 !
340 ! Measurement
350 !
360 PRINT "Set DUT. Then Push [Enter] key."
370 INPUT "",Inp_char$
380 !
390 OUTPUT @Agte506x;":TRIG:SING"
```

```
400 OUTPUT @Agte506x;"*OPC?"
410 ENTER @Agte506x;Buff$
420 !
430 OUTPUT @Agte506x;":DISP:WIND1:TRAC1:Y:AUTO"
440 PRINT "Push [Enter] key. -> [Time Domain Transform]"
450 INPUT "",Inp_char$
460 !
470 ! Time Domain Transform
480 !
490 OUTPUT @Agte506x;":CALC1:TRAN:TIME "&Tran_type$
500 OUTPUT @Agte506x;":CALC1:TRAN:TIME:STIM "&Stim_type$
510 OUTPUT @Agte506x;":CALC1:TRAN:TIME:KBES ";Win_beta
520 OUTPUT @Agte506x;":CALC1:TRAN:TIME:STAR ";Star_time
530 OUTPUT @Agte506x;":CALC1:TRAN:TIME:STOP ";Stop_time
540 OUTPUT @Agte506x;":CALC1:TRAN:TIME:STAT ON"
550 !
560 OUTPUT @Agte506x;":CALC1:PAR1:SEL"
570 OUTPUT @Agte506x;":CALC1:FORM REAL"
580 OUTPUT @Agte506x;":DISP:WIND1:TRAC1:Y:AUTO"
590 END
```

Control Using SICL-LAN Server

- Overview
- Sample Program in Excel VBA

Other topics about Sample Application Program

Overview

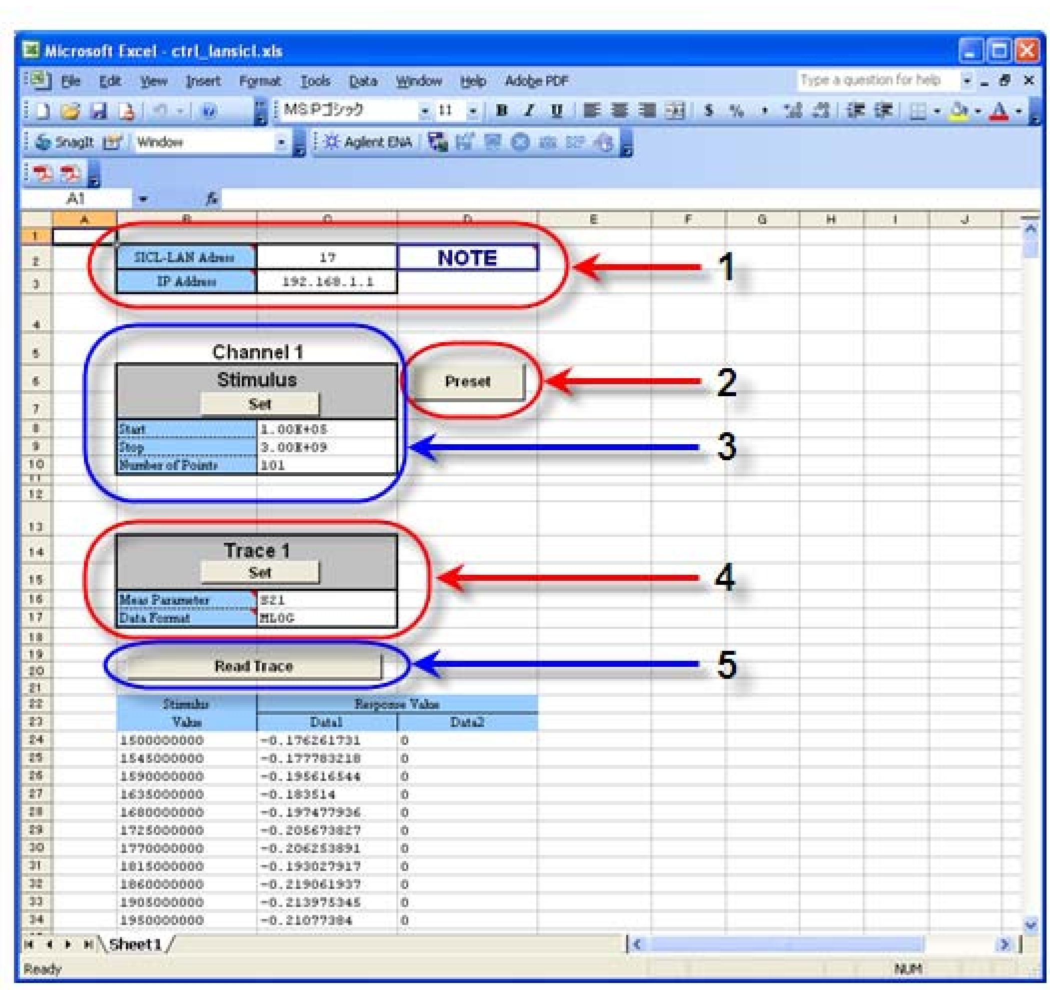
This section explains how to control the E5061B by using SICL in the Windows environment.

NOTE To control the E5061B using the SICL-LAN server, you need to make the preparations described in Control over SICL-LAN server.

Sample Program in Excel VBA

Opening **ctrl_lansicl.xls** in Microsoft Excel displays a screen as shown in the figure below:

ctrl_lansicl.xls



For how to use each element in ctrl_lansicl.xls, refer to the following description.

1. In part 1, in the cell to the right of the SICL-LAN Address, enter the address of the E5061B for control with the SICL-LAN server. This address is **XX**, which has been set with the command **System** > **Misc Setup** > **Network Setup** > **SICL-LAN Address** [**XX**]. Enter the IP address of the E5061B in the cell to the right of the IP Address. This VBA macro will not work properly without the correct values in these two cells.
2. Click Preset in part 2 to execute the presetting operation.
3. In part 3, the sweep range (start and stop points) and the number of measurement points for channel 1 are set. Click **Set** to execute the setting as shown in the setting table.
4. Part 4 sets the measurement parameters and data format for trace 1 in channel 1. Click **Set** to execute the setting as shown in the setting table.
5. Click **Read Trace** in part 5 retrieves the formatted data array of trace 1 in channel 1. The data is displayed in tabular.

Description of Operation in VBA macro

This section describes the operation of the VBA macro, focusing on the part related to control with SICL.

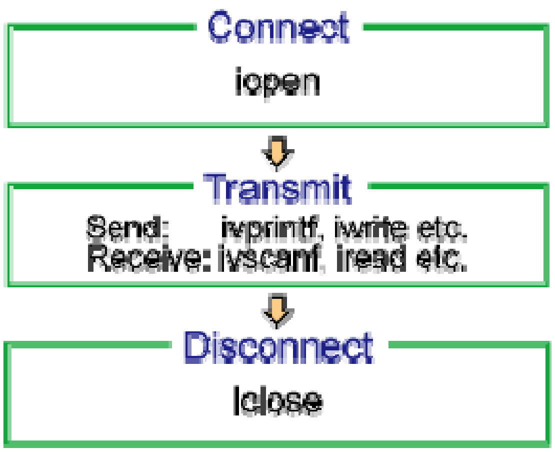
NOTE In order to use SICL in your VBA macro, you must declare functions and define variables with a SICL definition file (for VB).

NOTE In the VBA macro, ctrl_sicllan.xls, the standard module whose object name is "SICL," is the definition file**.**

The basic control flow with SICL is shown in Flow of control using SICL.

NOTE In this sample program, the **ivprintf** function, the **ivscanf** function, and the **iread** function are used in its communication part; you can use other SICL functions as well. For details, refer to sicl.hlp (the online help of SICL).

Flow of control using SICL



e5071c441

NOTE For more information on how to use each function of SICL, refer to the SICL manual.

The procedures of each step in Flow of control using SICL are described below.

Connection

The procedure corresponding to connection is OpenSession (OpenSession). OpenSession establishes a connection to the E5061B with the **iopen** function of SICL, using the SICL-LAN Address and IP Address entered in part 1 in ctrl_lansicl.xls. The **iopen** function takes the address information of the E5061B you specify as its parameters.

Syntax

addr = iopen(*dev*)

Variable	**addr**
Description	Session information (output)
Data type	Integer type
	dev
Description	Address information of the instrument you specify (input)
Data type	Character string type

Grammar	*sicl-name [ip-address]:interface, sicl-lan-address*

For example, if the parameter (*dev*) is "lan[192.168.0.1]:hpib9,17," connection is made to the address of **17** of the interface of **hpib9** with the E5061B whose IP address is **192.168.0.1** by using the external controller whose SICL interface name is **lan**.

OpenSession

```
Function OpenSession() As Integer
Dim ServAddr As String
Dim IpAddr As String
On Error GoTo ErrHandler
'''Get Sicl-Lan Address
Sheets("Sheet1").Select
Range("C2").Select
ServAddr = ActiveCell.FormulaR1C1
'''Get Ip Address
Sheets("Sheet1").Select
Range("C3").Select
IpAddr = ActiveCell.FormulaR1C1

OpenSession = iopen("lan[" & IpAddr & "]:hpib9," & ServAddr)
Call itimeout(OpenSession, 10000)
Exit Function
ErrHandler:
MsgBox "*** Error : " & Error$
Call siclcleanup
End
End Function
```

Sending

The procedure corresponding to sending in communication is OutputSiclLan. OutputSiclLan uses the **ivprintf** function of SICL to send messages (SCPI commands). The **ivprintf** function takes the session information output from the **iopen** function and a program message as its parameters.

Syntax

Status = ivprintf(*addr*,*mes*)

Variable	**Status**
Description	Return value of the function (output)
Data type	Integer type
	addr
Description	Session information (input)
Data type	Integer type
	mes
Description	Program message (input)
Data type	Character string type

OutputSiclLan

```
Sub OutputSiclLan(addr As Integer, message As String)

Dim Status As Integer
Dim actualcnt As Long
Dim length As Long

On Error GoTo ErrHandler
length = Len(message)
Status = ivprintf(addr, message & Chr$(10))
Exit Sub

ErrHandler:
MsgBox "*** Error : " & Error$
Call siclcleanup
```

```
End
End Sub
```

Receiving

The procedure corresponding to receiving ASCII format messages in communication is EnterSiclLan. EnterSiclLan uses the **ivscanf** function of SICL to receive a message in ASCII format and store it into the output variable. The **ivscanf** function takes the session information output from the **iopen** function, the format for output, and the data to be output as its parameters.

Syntax

Status = ivscanf(*addr,fmt,ap*)

Variable	**fmt**
Description	Format for output (input)
Data type	Character string type
	ap
Description	Data to be output (output)
Data type	Character string type

For information on the variable (*Status*) and the variable (*addr*), refer to Variable.

In Visual Basic, variables must be declared as a fixed-length string when receiving string data using the **ivscanf** function.

EnterSiclLan

```
Sub EnterSiclLan(addr As Integer, Query As String)

Dim Status As Integer
Dim actualcnt As Long
```

```
Dim res As String * 256

On Error GoTo ErrHandler
Status = ivscanf(addr, "%t", res)
Query = Trim(res)
Exit Sub

ErrHandler:
MsgBox "*** Error : " & Error$
Call siclcleanup
End

End Sub
```

The procedure corresponding to receiving array data in communication is EnterSiclLanArrayReal64, which uses the **iread** function of SICL to receive array data in the IEEE 64-bit floating point binary transfer format and store it into the output variable. The **iread** function takes the session information output from the **iopen** function, the data to be output, the number of data bytes, the condition to finish reading data, and the number of data bytes actually read out as its parameters.

Syntax

Status = iread(*addr,buf,bufsize,reason,actual*)

Variable	**buf**
Description	Data to be output (output)
Data type	Character string type
	bufsize
Description	The number of data bytes (input)
Data type	Long integer type
	reason
Description	The condition to finish reading out data (input)

Data type	Integer type
	actual
Description	The number of data bytes actually read out (output)
Data type	Long integer type

For information on the variable (*Status*) and the variable (*addr*), refer to Variable.

Each functional of EnterSiclLanArrayReal64 is described below.

(1) Retrieves the data header.

(2) Stores the number of data bytes into the size variable in the header part.

(3) Retrieves the formatted data array for trace 1 in channel 1 and stores it into the databuf variable.

(4) Retrieves the message terminator at the end of the data.

EnterSiclLanArrayReal64

Function EnterSiclLanArrayReal64(addr As Integer, databuf() As Double) As Long

```
Dim Status As Integer
Dim actualcnt As Long
Dim buf As String * 8
Dim size As Long
On Error GoTo ErrHandler
'''Read header info of "#6NNNNNN"
Status = iread(addr, buf, 8, I_TERM_MAXCNT, actualcnt) '.................(1)
size = Val(Mid$(buf, 3, 6)) '.................(2)
'''Read data
Status = iread(addr, databuf, size, I_TERM_MAXCNT, actualcnt) '.................(3)
'''Read ending LF
Status = iread(addr, buf, 1, I_TERM_MAXCNT, actualcnt) '.................(4)
EnterSiclLanArrayReal64 = size / 8
```

```
Exit Function

ErrHandler:
MsgBox "*** Error : " & Error$
Call siclcleanup
End
End Function
```

Disconnection

The **iclose** function of SICL is used to disconnect communication. The **iclose** function takes the session information output from the **iopen** function as its parameter.

Syntax

Status = iclose(*addr*)

For information on the variable (*Status*) and the variable (*addr*), refer to Variable

Sample control

The E5061B can be controlled by executing the above procedures in order, following the control flow in Flow of control using SICL. This is demonstrated by the Preset procedure (a procedure that is executed when the Preset button is clicked) as described in Preset.

Preset

```
Sub Preset()

''' Open Session
E506x = OpenSession

'''Presetting the analyzer
Call OutputSiclLan(E506x, ":SYST:PRES")
'''Close Session
Call iclose(E506x)

End Sub
```

Control Using Telnet Server

- Overview
- Sample Program in Excel VBA

Other topics about Sample Programs

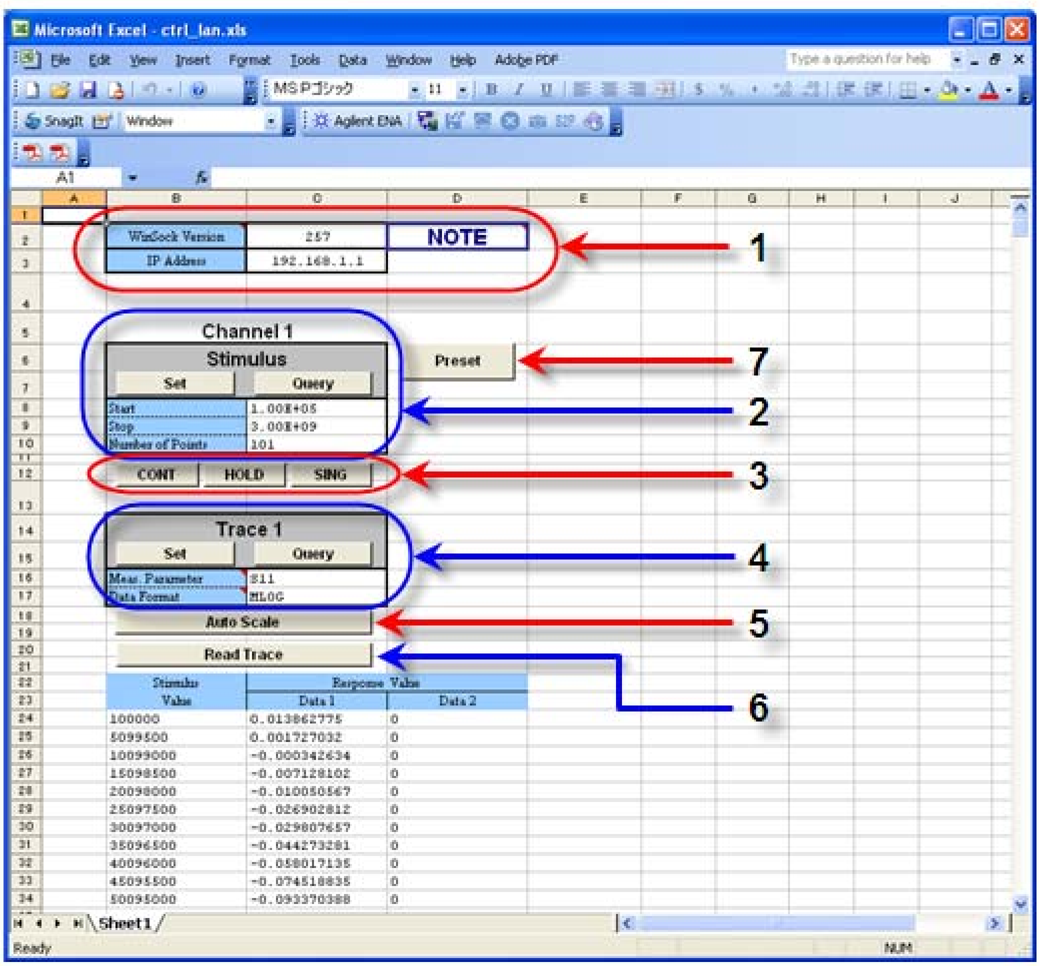
Overview

This section explains how to control the E5061B by using WinSock API in the Windows environment.

Sample Program in Excel VBA

Opening **ctrl_lan.xls** in Microsoft Excel displays the screen shown in the figure below.

ctrl_lan.xls



For how to use each element in ctrl_lan.xls, refer to the following description.

1. Enter the version number of WinSock API in the cell to the right side of "Winsock Version." The version number is obtained by multiplying

256 by the major version and then adding the minor version. For example, when the version of your Winsock API is 1.1, the version number is obtained as follows: 256×1+1=257. Enter the IP address of the E5061B in the cell to the right side of "IP Address." This VBA macro will not work properly without the correct values in these two cells.

2. In part 2, the sweep range (start and stop points) and the number of measurement points are set. Click **Set** to execute the setting operation as specified with the setting table, while clicking the button labeled "Query" retrieves the current settings of the E5061B.
3. Part 3 is dedicated to setting the trigger mode.
4. Part 4 sets the measurement parameters and data format for trace 1 in channel 1. Click **Set** to execute the setting operation as specified with the setting table, while clicking the button labeled "Query" retrieves the current settings of the E5061B.
5. In part 5, click **Auto Scale** to execute auto scaling for trace 1 in channel 1.
6. Click **Read Trace** in part 6 to retrieve the formatted data of trace 1 in channel 1. The data is displayed in tabular.
7. Click **Preset** to execute the presetting operation.

Description of operation in VBA macro

This section describes the operation of the VBA macro, focusing on the part related to control with WinSock API.

In order to use WinSock API, you must declare functions and define variables with a definition file of WinSock API, as shown in Definition file of WinSock API.

Definition file of WinSock API

```
'This is the Winsock API definition file for Visual Basic

'Setup the variable type 'hostent' for the WSAStartup command
Type Hostent
h_name As Long
h_aliases As Long
h_addrtype As String * 2
h_length As String * 2
h_addr_list As Long
End Type
```

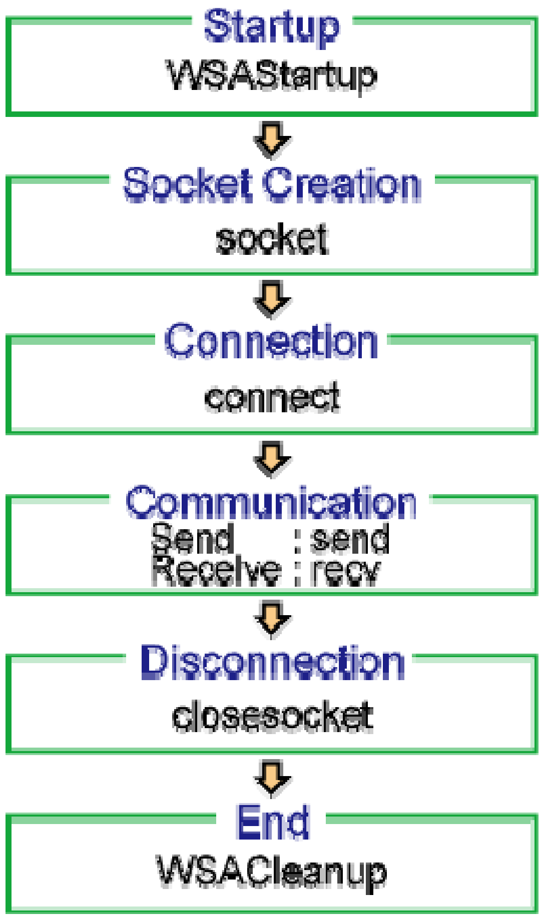
```
Public Const SZHOSTENT = 16

'Set the Internet address type to a long integer (32-bit)
Type in_addr
s_addr As Long
End Type

'A note to those familiar with the C header file for Winsock
'Visual Basic does not permit a user-defined variable type
'to be used as a return structure. In the case of the
'variable definition below, sin_addr must
'be declared as a long integer rather than the user-defined
'variable type of in_addr.
Type sockaddr_in
sin_family As Integer
sin_port As Integer
sin_addr As Long
sin_zero As String * 8
End Type

Public Const WSADESCRIPTION_LEN = 256
Public Const WSASYS_STATUS_LEN = 128
Public Const WSA_DescriptionSize = WSADESCRIPTION_LEN + 1
Public Const WSA_SysStatusSize = WSASYS_STATUS_LEN + 1

'Setup the structure for the information returned from
'the WSAStartup() function.
Type WSAData
wVersion As Integer
wHighVersion As Integer
szDescription As String * WSA_DescriptionSize
szSystemStatus As String * WSA_SysStatusSize
iMaxSockets As Integer
iMaxUdpDg As Integer
lpVendorInfo As String * 200
End Type
```

```
'Define socket return codes
Public Const INVALID_SOCKET = &HFFFF
Public Const SOCKET_ERROR = -1

'Define socket types
Public Const SOCK_STREAM = 1 'Stream socket
Public Const SOCK_DGRAM = 2 'Datagram socket
Public Const SOCK_RAW = 3 'Raw data socket
Public Const SOCK_RDM = 4 'Reliable Delivery socket
Public Const SOCK_SEQPACKET = 5 'Sequenced Packet socket

'Define address families
Public Const AF_UNSPEC = 0 'unspecified
Public Const AF_UNIX = 1 'local to host (pipes, portals)
Public Const AF_INET = 2 'internetwork: UDP, TCP, etc.
Public Const AF_IMPLINK = 3 'arpanet imp addresses
Public Const AF_PUP = 4 'pup protocols: e.g. BSP
Public Const AF_CHAOS = 5 'mit CHAOS protocols
Public Const AF_NS = 6 'XEROX NS protocols
Public Const AF_ISO = 7 'ISO protocols
Public Const AF_OSI = AF_ISO 'OSI is ISO
Public Const AF_ECMA = 8 'european computer manufacturers
Public Const AF_DATAKIT = 9 'datakit protocols
Public Const AF_CCITT = 10 'CCITT protocols, X.25 etc
Public Const AF_SNA = 11 'IBM SNA
Public Const AF_DECnet = 12 'DECnet
Public Const AF_DLI = 13 'Direct data link interface
Public Const AF_LAT = 14 'LAT
Public Const AF_HYLINK = 15 'NSC Hyperchannel
Public Const AF_APPLETALK = 16 'AppleTalk
Public Const AF_NETBIOS = 17 'NetBios-style addresses
Public Const AF_MAX = 18 'Maximum # of address families

'Setup sockaddr data type to store Internet addresses
Type sockaddr
sa_family As Integer
sa_data As String * 14
```

```
End Type
Public Const SADDRLEN = 16

'Declare Socket functions

Public Declare Function closesocket Lib "wsock32.dll" (ByVal s As Long) As Long

Public Declare Function connect Lib "wsock32.dll" (ByVal s As Long, addr As sockaddr_in, ByVal
namelen As Long) As Long

Public Declare Function htons Lib "wsock32.dll" (ByVal hostshort As Long) As Integer

Public Declare Function inet_addr Lib "wsock32.dll" (ByVal cp As String) As Long

Public Declare Function recv Lib "wsock32.dll" (ByVal s As Long, ByVal buf As Any, ByVal buflen As
Long, ByVal flags As Long) As Long

Public Declare Function recvB Lib "wsock32.dll" Alias "recv" (ByVal s As Long, buf As Any, ByVal buflen
As Long, ByVal flags As Long) As Long

Public Declare Function send Lib "wsock32.dll" (ByVal s As Long, buf As Any, ByVal buflen As Long,
ByVal flags As Long) As Long

Public Declare Function socket Lib "wsock32.dll" (ByVal af As Long, ByVal socktype As Long, ByVal
protocol As Long) As Long

Public Declare Function WSAStartup Lib "wsock32.dll" (ByVal wVersionRequired As Long, lpWSAData
As WSAData) As Long

Public Declare Function WSACleanup Lib "wsock32.dll" () As Long

Public Declare Function WSAUnhookBlockingHook Lib "wsock32.dll" () As Long

Public Declare Sub CopyMemory Lib "kernel32" Alias "RtlMoveMemory" (hpvDest As Any, hpvSource As
Any, ByVal cbCopy As Long)
```

The basic control flow with WinSock API is shown in the figure below:

Control flow with WinSock API



e9071c372

The procedures of each step in Control flow with WinSock API are described below.

Startup

The procedure corresponding to Startup is StartIt. StartIt launches and initializes WinSock API with **WSAStartup**, whose version is shown in part 1 of ctrl_ lan.xls. The function WSAStartup should always be used when initiating WinSock. This function takes the version number (input) and launching information (output) as its parameters.

StartIt

```
Sub StartIt()

Dim StartUpInfo As WSAData
'Version 1.1 (1*256 + 1) = 257
```

```
'version 2.0 (2*256 + 0) = 512
'Get WinSock version
Sheets("Sheet1").Select
Range("C2").Select
version = ActiveCell.FormulaR1C1
'Initialize Winsock DLL
x = WSAStartup(version, StartUpInfo)

End Sub
```

Socket Creation and Connection

The procedure for Socket Creation and Connection is OpenSocket. OpenSocket makes a connection to an instrument associated with the IP address specified with the input parameter Hostname. It uses a socket of the port specified with the input parameter PortNumber. Each functional part of OpenSocket is described below.

In (1), the inet_aadr function of WinSock API is used to convert an IP address delimited by "." to an Internet address.

In (2), a new socket is created with the **socket** function of WinSock API and its socket descriptor is obtained. If an error occurs, the control returns to the main program with a message. The socket function takes an address family (input), a socket type (input), and a protocol number (input) as its parameters.

In (3), the socket address is specified. Note that htons, which is used for specifying the port number, is a function of WinSock API. This function converts a 2-byte integer from the Windows byte order (little endian) to the network byte order (big endian).

In (4), a connection to the E5061B is made by using the **connect** function of WinSock API. If an error occurs, the control returns to the main program with a message. The connect function takes a socket descriptor (input), a socket address (input), and the size of the socket address (input) as its parameters.

OpenSocket

```
Function OpenSocket(ByVal Hostname As String, ByVal PortNumber As Intege r) As Integer
Dim I_SocketAddress As sockaddr_in
Dim ipAddress As Long
ipAddress = inet_addr(Hostname) '...........(1)
```

```
'Create a new socket
socketId = socket(AF_INET, SOCK_STREAM, 0) '
If socketId = SOCKET_ERROR Then '
MsgBox ("ERROR: socket = " + Str$(socketId)) '...........(2)
OpenSocket = COMMAND_ERROR '
Exit Function '
End If '

'Open a connection to a server

I_SocketAddress.sin_family = AF_INET '
I_SocketAddress.sin_port = htons(PortNumber) '...........(3)
I_SocketAddress.sin_addr = ipAddress '
I_SocketAddress.sin_zero = String$(8, 0) '

x = connect(socketId, I_SocketAddress, Len(I_SocketAddress)) '
If socketId = SOCKET_ERROR Then '
MsgBox ("ERROR: connect = " + Str$(x)) '..(4)
OpenSocket = COMMAND_ERROR '
Exit Function '
End If '

OpenSocket = socketId

End Function
```

Communication

The procedure corresponding to Communication is SendCommand. SendCommand transmits a message (SCPI command) specified with the input parameter "command" to the E5061B using the **send** function of WinSock API. The send function takes a socket descriptor (input), a message to be transmitted (input), message length (input) and a flag (input) as its parameters.

SendCommand

```
Function SendCommand(ByVal command As String) As Integer

Dim strSend As String
strSend = command + vbCrLf
```

```
count = send(socketId, ByVal strSend, Len(strSend), 0)
If count = SOCKET_ERROR Then
MsgBox ("ERROR: send = " + Str$(count))
SendCommand = COMMAND_ERROR
Exit Function
End If
SendCommand = NO_ERROR

End Function
```

The procedure corresponding to the Receiving part of communication is RecvAscii and other functions. RecvAscii receives a message in ASCII format and stores it in the dataBuf output parameter. Maximum length of the message is specified with the maxLength input parameter. Each functional part of RecvAscii is described below.

In (1), a message (a response to a query for a SCPI command) is received from the E5061B as a series of characters using the **recv** function of WinSock API. If an error occurs, the control returns to the main program with a message. The recv function takes a socket descriptor (input), a message to be received (input), message length (input) and a flag (input) as its parameters.

In (2), it is determined whether each received character is LF (ASCII code: 10). When it is LF, receiving is terminated by adding NULL (ASCII code: 0) to the end of the dataBuf string and the control returns to the main program.

In (3), the number of the last characters that were read out is added to the count value for checking the number of received characters, and the characters are appended to the end of the dataBuf string.

RecvAscii

```
Function RecvAscii(dataBuf As String, ByVal maxLength As Integer) As Integer

Dim c As String * 1
Dim length As Integer
dataBuf = ""
While length < maxLength
DoEvents
count = recv(socketId, c, 1, 0) '
If count < 1 Then '
RecvAscii = RECV_ERROR '............(1)
dataBuf = Chr$(0) '
```

```
Exit Function '
End If '
If c = Chr$(10) Then '
dataBuf = dataBuf + Chr$(0) '............(2)
RecvAscii = NO_ERROR '
Exit Function '
End If '
length = length + count '............(3)
dataBuf = dataBuf + c '
Wend
RecvAscii = RECV_ERROR
End Function
```

Disconnection

The procedure corresponding to Disconnection is CloseConnection. CloseConnection disconnects communication and removes a socket using the **closesocket** function of WinSock API. The closesocket function takes a socket descriptor (input) as its parameter.

CloseConnection

```
Sub CloseConnection()

x = closesocket(socketId)
If x = SOCKET_ERROR Then
MsgBox ("ERROR: closesocket = " + Str$(x))
Exit Sub
End If

End Sub
```

End

The procedure corresponding to End is EndIt. EndIt disconnects WinSock API using the **WSACleanup** function of WinSock API. The function WSACleanup should always be used when terminating WinSock.

EndIt

```
Sub EndIt()

'Shutdown Winsock DLL
```

```
x = WSACleanup()

End Sub
```

Example of control

The E5061B can be controlled by executing the above procedures in order, following the control flow in Control flow with WinSock API. This is demonstrated by the procedure autoscale (a procedure that is executed when the Auto Scale button is clicked) as described in autoscale.

autoscale

```
Sub autoscale()
'
' auto scaling
'
Call StartIt
Call get_hostname
x = OpenSocket(Hostname$, ScpiPort)
x = SendCommand(":DISP:WIND1:TRAC1:Y:AUTO")
Call CloseConnection
Call EndIt
End Sub
```

When you execute more than one command by connecting and disconnecting a socket for every command, the sequence of execution may change.

Control LCD Update Timing

- Overview
- Sample Program in Excel VBA using VISA
- Sample Program in HT Basic

Other topics about Sample Programs

Overview

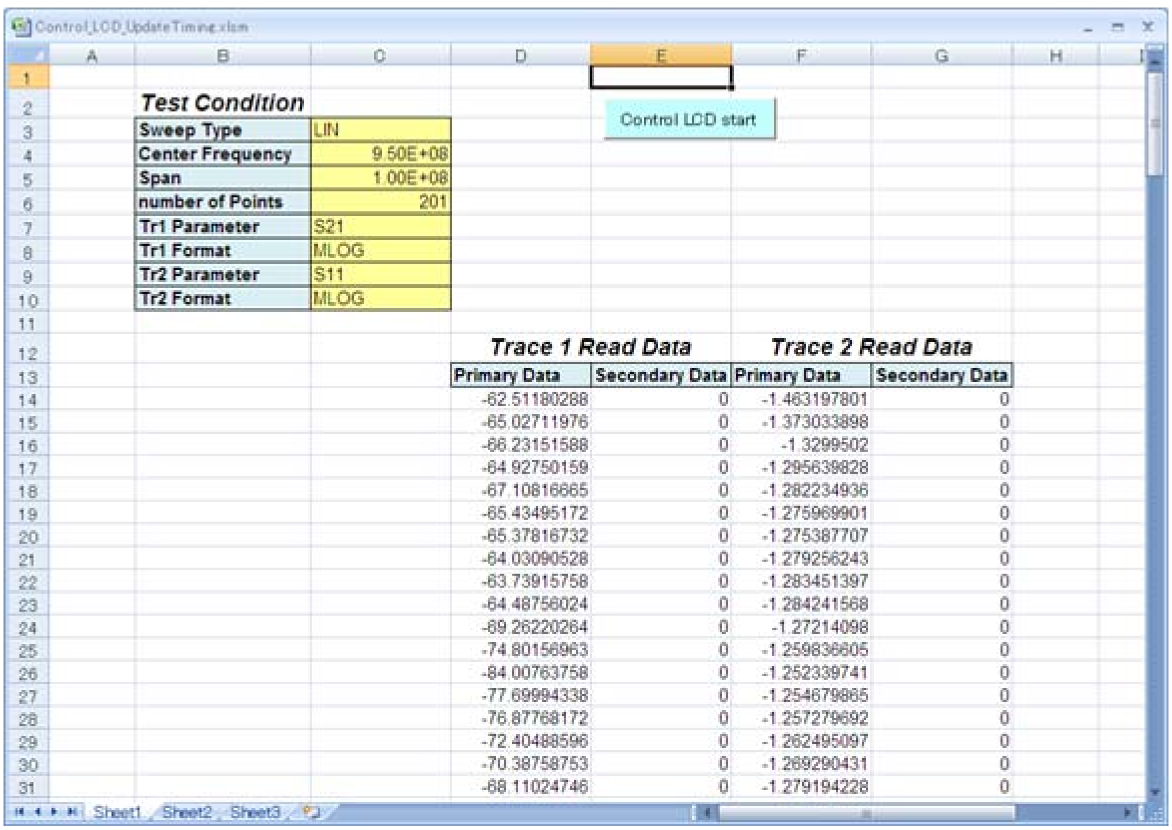
This sample program is provided in this section where the command processing time is improved by controlling the update timing of the LCD display.

NOTE This sample program correctly runs when the maximum number of channels/traces is set to 4 channels/4 traces.

This program sets the necessary measurement conditions and then turns OFF the updating of the LCD display. Next, it performs measurement, reads out the result, and updates the screen once. This program repeats this measurement procedure ten times.

Sample Program in Excel VBA using VISA

Example of excel sheet with control LCD update timing program


Control_LCD_UpdateTiming.xlsm

Control LCD start

Test Condition

Sweep Type	LIN
Center Frequency	9.50E+08
Span	1.00E+08
number of Points	201
Tr1 Parameter	S21
Tr1 Format	MLOG
Tr2 Parameter	S11
Tr2 Format	MLOG

Trace 1 Read Data		Trace 2 Read Data	
Primary Data	Secondary Data	Primary Data	Secondary Data
-62.51180288	0	-1.463197801	0
-65.02711976	0	-1.373033898	0
-66.23151588	0	-1.3299502	0
-64.92750159	0	-1.295639828	0
-67.10816665	0	-1.282234936	0
-65.43495172	0	-1.275969901	0
-65.37816732	0	-1.275387707	0
-64.03090528	0	-1.279256243	0
-63.73915758	0	-1.283451397	0
-64.48756024	0	-1.284241568	0
-69.26220264	0	-1.27214098	0
-74.80156963	0	-1.259836605	0
-84.00763758	0	-1.252339741	0
-77.69994338	0	-1.254679865	0
-76.87768172	0	-1.257279692	0
-72.40488596	0	-1.262495097	0
-70.38758753	0	-1.269290431	0
-68.11024746	0	-1.279194228	0

Sheet1 Sheet2 Sheet3

```
Private Sub Ctrl_LCD_Click()
    Dim defrm As Long
    Dim Age506x As Long

    Dim SwType As String
    Dim Cent As Double, Span As Double
    Dim Param(1) As String, Fmt(1) As String
    Dim NumPoin As Integer, NumData As Integer

    Const SwTime = 2#

    Dim Dummy As String * 20

    Dim Tr1ptr(3) As Long
    Dim Tr1Data() As Double
    Dim Tr2ptr(3) As Long
    Dim Tr2Data() As Double

    Dim WrtFmt As String

    '***
    '*** Open session.
    '***

    Call viOpenDefaultRM(defrm)
    Call viOpen(defrm, "GPIB0::17::INSTR", 0, 0, Age506x)
    Call viSetAttribute(Age506x, VI_ATTR_TMO_VALUE, 10000)

    '***
    '*** Set variable of measurement condition.
    '***

    SwType = Trim(Cells(3, 3).Value)
    Cent = CDbl(Cells(4, 3).Value)
    Span = CDbl(Cells(5, 3).Value)
    NumPoin = CInt(Cells(6, 3).Value)
```

```
Param(0) = Trim(Cells(7, 3).Value)
Fmt(0) = Trim(Cells(8, 3).Value)
Param(1) = Trim(Cells(9, 3).Value)
Fmt(1) = Trim(Cells(10, 3).Value)

'***
'*** Send measurement condition to E5061B.
'***

Call viVPrintf(Age506x, ":SENS1:SWE:TYPE " + Trim(SwType) + vbLf, 0)
Call viVPrintf(Age506x, ":SENS1:FREQ:CENT " + CStr(Cent) + vbLf, 0)
Call viVPrintf(Age506x, ":SENS1:FREQ:SPAN " + CStr(Span) + vbLf, 0)
Call viVPrintf(Age506x, ":SENS1:SWE:POIN " + CStr(NumPoin) + vbLf, 0)

Call viVPrintf(Age506x, ":TRIG:SOUR BUS" + vbLf, 0)
Call viVPrintf(Age506x, ":INIT1:CONT ON" + vbLf, 0)
Call viVPrintf(Age506x, ":SENS1:SWE:TIME:AUTO OFF" + vbLf, 0)
Call viVPrintf(Age506x, ":SENS1:SWE:TIME " + CStr(SwTime) + vbLf, 0)
Call viVPrintf(Age506x, "*OPC?" + vbLf, 0)
Call viVScanf(Age506x, "%t", Dummy)

For i = 2 To 4
   Call viVPrintf(Age506x, ":INIT" + CStr(i) + ":CONT OFF" + vbLf, 0)
Next i

Call viVPrintf(Age506x, ":DISP:SPL D1" + vbLf, 0)
Call viVPrintf(Age506x, ":DISP:WIND1:SPL D1_2" + vbLf, 0)
Call viVPrintf(Age506x, "*OPC?" + vbLf, 0)
Call viVScanf(Age506x, "%t", Dummy)

'***
'*** Setting Trace 1 and 2.
'***

Call viVPrintf(Age506x, ":CALC1:PAR:COUN 2" + vbLf, 0)
Call viVPrintf(Age506x, ":CALC1:PAR1:DEF " + Trim(Param(0)) + vbLf, 0)
Call viVPrintf(Age506x, ":CALC1:PAR1:SEL" + vbLf, 0)
```

```
Call viVPrintf(Age506x, ":CALC1:FORM " + Trim(Fmt(0)) + vbLf, 0)
Call viVPrintf(Age506x, ":CALC1:PAR2:DEF " + Trim(Param(1)) + vbLf, 0)
Call viVPrintf(Age506x, ":CALC1:PAR2:SEL" + vbLf, 0)
Call viVPrintf(Age506x, ":CALC1:FORM " + Trim(Fmt(1)) + vbLf, 0)

Call viVPrintf(Age506x, ":DISP:ENAB OFF" + vbLf, 0)
Call viVPrintf(Age506x, ":FORM:DATA REAL" + vbLf, 0)
Call viVPrintf(Age506x, "*OPC?" + vbLf, 0)
Call viVScanf(Age506x, "%t", Dummy)

'***
'*** redim for read data.
'***

NumData = NumPoin * 2

ReDim Tr1Data(NumData - 1)
Tr1ptr(0) = VarPtr(NumData)
Tr1ptr(1) = VarPtr(Tr1Data(0))

ReDim Tr2Data(NumData - 1)
Tr2ptr(0) = VarPtr(NumData)
Tr2ptr(1) = VarPtr(Tr2Data(0))
WrtFmt = "%#Zb%1t"

'***
'*** Cycling trigger, read data, and screen update.
'***

For i = 1 To 10

    '***
    '*** Trigger.
    '***

    Call viVPrintf(Age506x, ":TRIG:SING" + vbLf, 0)
    Call viVPrintf(Age506x, "*OPC?" + vbLf, 0)
```

```
        Call viVScanf(Age506x, "%t", Dummy)

        '***
        '*** Read trace data.
        '***

        Call viVPrintf(Age506x, ":CALC1:PAR1:SEL" + vbLf, 0)
        Call viVPrintf(Age506x, ":CALC1:DATA:FDAT?" + vbLf, 0)
        Call viVScanf(Age506x, WrtFmt, Tr1ptr(0))

        For j = 0 To NumData / 2 - 1
            Cells(14 + j, 4).Value = Tr1Data(j * 2)
            Cells(14 + j, 5).Value = Tr1Data(j * 2 + 1)
        Next j

        Call viVPrintf(Age506x, ":CALC1:PAR2:SEL" + vbLf, 0)
        Call viVPrintf(Age506x, ":CALC1:DATA:FDAT?" + vbLf, 0)
        Call viVScanf(Age506x, WrtFmt, Tr2ptr(0))

        For j = 0 To NumData / 2 - 1
            Cells(14 + j, 6).Value = Tr2Data(j * 2)
            Cells(14 + j, 7).Value = Tr2Data(j * 2 + 1)
        Next j

        '***
        '*** screen update.
        '***

        Call viVPrintf(Age506x, ":DISP:UPD" + vbLf, 0)
    Next i

    Call viVPrintf(Age506x, ":FORM:DATA ASC" + vbLf, 0)
    Call viClose(Age506x)
    Call viClose(defrm)

End Sub
```

Sample Program in HT Basic (cont_upd.htb)

```
10 REAL Trace1(1:201,1:2),Trace2(1:201,1:2)
20 DIM Buff$[9],Img$[30]
30 INTEGER Nop,I
40 !
50 ASSIGN @Agte506x TO 717
60 ASSIGN @Binary TO 717;FORMAT OFF
70 !
80 OUTPUT @Agte506x;":SENS1:SWE:TYPE LIN"
90 OUTPUT @Agte506x;":SENS1:FREQ:CENT 950E6"
100 OUTPUT @Agte506x;":SENS1:FREQ:SPAN 100E6"
110 OUTPUT @Agte506x;":SENS1:SWE:POIN 201"
120 OUTPUT @Agte506x;":TRIG:SOUR BUS"
130 OUTPUT @Agte506x;":INIT1:CONT ON"
140 FOR I=2 TO 4
150 OUTPUT @Agte506x;":INIT"&VAL$(I)&":CONT OFF"
160 NEXT I
170 !
180 OUTPUT @Agte506x;":DISP:SPL D1"
190 OUTPUT @Agte506x;":DISP:WIND1:SPL D1_2"
200 !
210 OUTPUT @Agte506x;":CALC1:PAR:COUN 2"
220 OUTPUT @Agte506x;":CALC1:PAR1:DEF S21"
230 OUTPUT @Agte506x;":CALC1:PAR1:SEL"
240 OUTPUT @Agte506x;":CALC1:FORM MLOG"
250 OUTPUT @Agte506x;":CALC1:PAR2:DEF S11"
260 OUTPUT @Agte506x;":CALC1:PAR2:SEL"
270 OUTPUT @Agte506x;":CALC1:FORM MLOG"
280 !
290 OUTPUT @Agte506x;":DISP:ENAB OFF"
300 OUTPUT @Agte506x;":FORM:DATA REAL"
310 !
320 FOR I=1 TO 10
330 OUTPUT @Agte506x;":TRIG:SING"
340 OUTPUT @Agte506x;"*OPC?"
350 ENTER @Agte506x;Buff$
360 !
```

```
370 ! Read Trace Data
380 !
390 OUTPUT @Agte506x;":CALC1:PAR1:SEL"
400 OUTPUT @Agte506x;":CALC1:DATA:FDAT?"
410 ENTER @Agte506x USING "#,8A";Buff$
420 ENTER @Binary;Trace1(*)
430 ENTER @Agte506x USING "#,1A";Buff$
440 !
450 OUTPUT @Agte506x;":CALC1:PAR2:SEL"
460 OUTPUT @Agte506x;":CALC1:DATA:FDAT?"
470 ENTER @Agte506x USING "#,8A";Buff$
480 ENTER @Binary;Trace2(*)
490 ENTER @Agte506x USING "#,1A";Buff$
500 !
510 ! Update Display
520 !
530 OUTPUT @Agte506x;":DISP:UPD"
540 NEXT I
550 END
```

Description

Lines 50 to 60

Assigns a GPIB address to the I/O pass.

Lines 80 to 110

These lines set the sweep type to linear sweep, the sweep center value to 950 MHz, the sweep span value to 100 MHz, and the number of measurement points to 201.

Lines 120 to 160

These lines set the trigger source to bus trigger, turn ON Continuous Activation mode for channel 1, and turn the mode OFF for channels 2 through 4.

Lines 180 to 190

These lines display the window for channel 1 only and arrange two graphs tiled horizontally.

Lines 210 to 270

These lines set the number of traces for channel 1 to 2, the measurement parameter and its data format for trace 1 to S21 and Log Mag, respectively, and those for trace 2 to S11 and Log Mag, respectively.

Line 290

This line turns OFF the updating of the LCD screen.

Line 300

This line sets the data transfer format to binary.

Lines 320 to 540

These lines repeat the following procedure ten times.

Lines 340 to 360: These lines trigger the instrument and wait until the measurement cycle finishes.

Lines 400 to 440: Reads out the formatted data array of trace 1 in channel 1.

Lines 460 to 500: Reads out the formatted data array of trace 2 in channel 1.

Line 540: This line updates the LCD screen once.

Handler Interface

- Overview
- Program Code

Other topics about Sample Programs

Overview

The sample program communicates with an external instrument through the handler I/O port.

This program outputs 5 (sets bit 2 and bit 0 to Low, and the other bits to High) to the port A of the handler I/O port, then waits until the bit 3 of the port C is set to Low.

See Inputting/Outputting Data for this programming.

Program Code

Excel VBA

```
Sub Handler_Click()
    Dim defrm As Long          'Session to Default Resource Manager.
    Dim vi As Long             'Session to instrument.
    Dim Out_Data As Integer    'Decimal value.
    Dim In_Data As Long
    Dim Bit_stat As Integer
    Dim Flag_bit As Integer
    Dim Out_Data_Bin As String
    Dim Ret As Long            'Return value.
    Dim i As Long
    Dim X As Long
    Const TimeOutTime = 40000    'timeout time.
    Out_Data_Bin = "00000101"     'Store the output data on the port A (binaly).
    Flag_bit = 3                  'Bit location (bit 3).

    Call viOpenDefaultRM(defrm)    'Initializes the VISA system.
    Call viOpen(defrm, "GPIB0::17::INSTR", 0, 0, vi)   'Opens the session to the specified instrument.
    Call viSetAttribute(vi, VI_ATTR_TMO_VALUE, TimeOutTime)   'The state of an attribute for the
specified session.

    Call viVPrintf(vi, "*RST" & vbLf, 0)    'Presets the setting state of the ENA-L.
    Call viVPrintf(vi, "*CLS" & vbLf, 0)   'Clears the all status register.
```

```
   Call viVPrintf(vi, ":CONT:HAND:C:MODE INP" & vbLf, 0)     'Configures the port C to input port.
   Call viVPrintf(vi, ":CONT:HAND:IND:STAT ON" & vbLf, 0)    'Line enable /INDEX signal.
   Call viVPrintf(vi, ":CONT:HAND:RTR:STAT ON" & vbLf, 0)   'Line enable /READY FOR TRIGGER
signal.

   For i = 1 To Len(Out_Data_Bin)    'Convert Out_Dara_Bin to a decimal value.
      If Mid(Out_Data_Bin, Len(Out_Data_Bin) - i + 1, 1) = "1" Then
         X = 2 ^ (i - 1)
         Ret = Ret + X
      End If
   Next i
   Out_Data = Ret       'Sets the decimal value.
   Call viVPrintf(vi, ":CONT:HAND:A " & Ret & vbLf, 0)    'Sets to the port A.

   Call viVPrintf(vi, ":CONT:HAND:C?" & vbLf, 0)   'Outputs data to output port C.
   Call viVScanf(vi, "%t", In_Data)    'Reads data from the port C.

   Call ErrorCheck(vi)      'Checking the error.

   Call viClose(vi)           'Closes the resource manager session.
   Call viClose(defrm)     'Breaks the communication and terminates the VISA system.

   End
End Sub
Sub ErrorCheck(vi As Long)
   Dim err As String * 50, ErrNo As Variant, Response

   Call viVQueryf(vi, ":SYST:ERR?" & vbLf, "%t", err)    'Reads error message.
   ErrNo = Split(err, ",")  'Gets the error code.

   If Val(ErrNo(0)) <> 0 Then
      Response = MsgBox(CStr(ErrNo(1)), vbOKOnly)  'Display the message box.
   End If
End Sub
```

HT Basic (handler.htb)

```
10 INTEGER Out_data,In_data,Bit_stat
20 DIM Out_data_bin$[9]
```

```
30 !
40 ASSIGN @Agte506x TO 717
50 !
60 Out_data_bin$="00000101"
70 Flag_bit=3
80 !
90 OUTPUT @Agte506x;":CONT:HAND:C:MODE INP"
100 OUTPUT @Agte506x;":CONT:HAND:IND:STAT ON"
110 OUTPUT @Agte506x;":CONT:HAND:RTR:STAT ON"
120 !
130 Out_data=IVAL(Out_data_bin$,2)
140 OUTPUT @Agte506x;":CONT:HAND:A ";Out_data
150 !
160 REPEAT
170 OUTPUT @Agte506x;":CONT:HAND:C?"
180 ENTER @Agte506x;In_data
190 Bit_stat=BIT(In_data,Flag_bit)
200 UNTIL Bit_stat=1
210 END
```

VBA Programming

VBA Programming (Embedded VBA)

Introduction to VBA Programming

Introduction to VBA Programming

- Introduction of the E5061B Macro Function
- An Overview of a Control System Based on the Macro Function
- Overview of E5061B COM Object

Introduction of the E5061B Macro Function

- Overview
- Macro Function

Other topics about Introduction to VBA Programming

Overview

The E5061B has a built-in macro function that allows a single instruction to substitute for multiple instructions. You can have the E5061B to automatically execute your own macro program that contains a series of VBA (Visual Basic for Application) statements. The macro function allows you to run a variety of applications; you can control not only the E5061B but also various peripherals from your own macro code.

The VBA is based on the VB (Visual Basic) programming language. Although the VBA is similar to the VB, they are not the same. The VBA has decreased some of the VB's features and added characteristic features to each application. The E5061B VBA is an added feature for controlling the E5061B. For details on the difference between the VBA and the VB, refer to Microsoft official guides, and various books on VBA.

For information on the basic operating procedures for the E5061B's VBA, see Operation Basics of the E5061B's VBA. This manual is not meant to be an in-depth guide to VBA programming basics and the syntax of VBA functions and commands. Such in-depth information is covered in VBA Help, Microsoft official guides, and various books on VBA.

Macro Function

The macro function allows you to control the E5061B itself as well as various peripherals. You can do the following:

1. Automate repetitive tasks
   You can use the E5061B's macro function to combine several processes into one. Automating repetitive tasks provides higher efficiency and eliminates human error. Once you have contained repetitive tasks in Sub procedures, you can later call the procedures from other programs, thus allowing effective reuse of programming assets.
2. Implement a user interface
   The E5061B VBA supports user forms that simplify visual user interface creation. User forms guide users through common tasks such as performing measurement and entering data, without requiring familiarity with the E5061B, thus minimizing the possibility of human error.

An Overview of a Control System Based on the Macro Function

Other topics about Introduction to VBA Programming

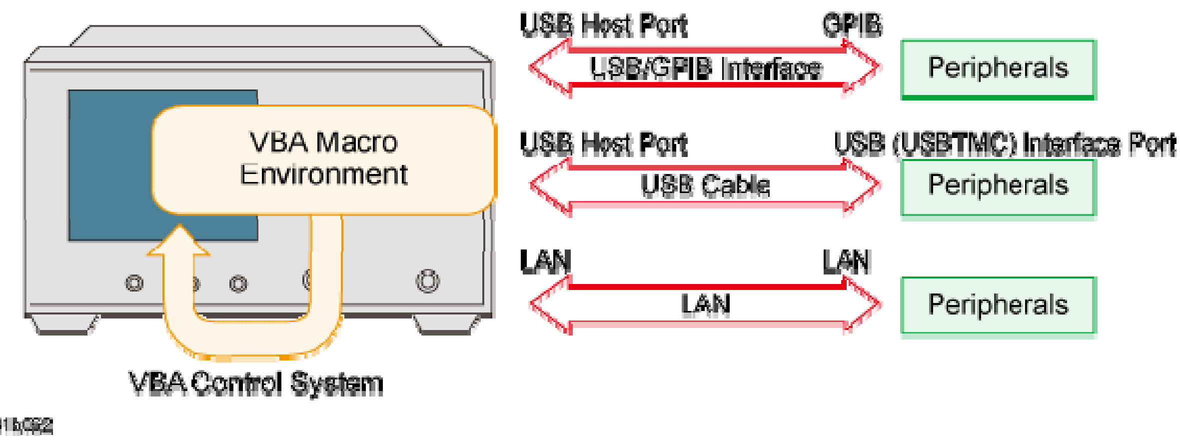
Overview

This section describes how you can use the E5061B's built-in VBA macro function to implement a system that controls the E5061B and peripherals, and what command sets are available for such purposes.

Implementing a Control System

Macro-based control systems are classified into two types: As shown in the following figure, a VBA control system controls the E5061B itself while a VBA remote control system controls peripherals. When you use the macro function to control the peripherals, you must connect the E5061B with the peripherals through USB/GPIB interface, USB or LAN, and configure them to communicate over VISA (Virtual Instrument Software Architecture). For information on programming using the VISA library, refer to Programming with VISA.

Configuration example of control system using macro environment



Required Equipment

- E5061B
- Peripherals and/or other instruments that serve your purpose
- USB/GPIB interface, USB Cable, or LAN

Control Methods

The command set you can use differs depending on whether you use the macro function to control the E5061B or a peripheral.

When you want to control the E5061B itself, you can create a program using COM objects within the E5061B VBA environment. COM objects that come with the E5061B include seven objects specific to the COM interface and COM objects that correspond to SCPI commands.

Controlling a Peripheral

When you want to control a peripheral, you can create a program using VISA library functions within the E5061B VBA environment.

For information on using the VISA library, see Controlling Peripherals. For a complete description on VISA functions, refer to the VISA library's online help.

For information on the GPIB commands that is available with a particular peripheral, refer to the documentation that comes with the peripheral.

Overview of E5061B COM Object

Other topics about Introduction to VBA Programming

Overview

The E5061B VBA environment provides COM objects that support the E5061B control. This section provides an overview of COM objects as well as considerations for using the E5061B's COM objects.

The definitions and specifications of COM are beyond the scope of this guide. Such in-depth information is covered in various books on COM.

About COM Object

When you control the E5061B through the macro function, you can use COM objects as components of your application. The functionality of the E5061B's COM objects is exposed through properties and methods.

Property

A property allows you to read or write a setting or attribute of an object. With the E5061B, you can use properties to set or read the settings of the E5061B.

You can find properties in the list of object types in COM Object Reference.

Method

A method allows you to manipulate an object in a particular way. With the E5061B, you can use methods to perform specific tasks.

You can find methods in the list of object types in COM Object Reference.

Event

An event means an operation from outside that the program can recognize such as clicking a mouse. The E5061B detects events when a specific softkey is pressed using the UserMenu_OnPress(ByVal Key_id As Long) procedure to execute the assigned procedure.

You can find events in the list of object types in COM Object Reference.

Using COM Object to Control E5061B

When you want to control the E5061B, you can use COM objects alone or in conjunction with SCPI commands and the Parse object. The latter method is a little slower than the former method because the Parse object is used to parse the messages of SCPI commands. For instructions on using the E5061B's VBA Editor to create a program that uses COM objects, refer to Operation Basics of the E5061B's VBA.

Major Control Difference between COM Object and SCPI Command

While the control using SCPI commands allows SRQ (Service Request) interruptions through the status reporting mechanism, the control using COM objects does not support SRQ interruptions. Instead of SRQ interrupts, you can use the WaitOnSRQ object to suspend the program until the E5061B is placed into the desired state.

Operation Basics

Operation Basics

- Displaying Visual Basic Editor
- Closing Visual Basic Editor
- Switching to the E5061B Measurement Screen
- Making a Preparation Before Coding
- Coding a VBA Program
- Saving a VBA program
- Loading a VBA Program
- Running a VBA Program
- Stopping a VBA Program
- Errors and Debugging
- Printing Output Values in the Echo Window
- Uses Advanced Techniques
- Using VBA Online Help

Displaying Visual Basic Editor

Other topics about Operation Basics

Overview

This section describes how to launch Visual Basic Editor.

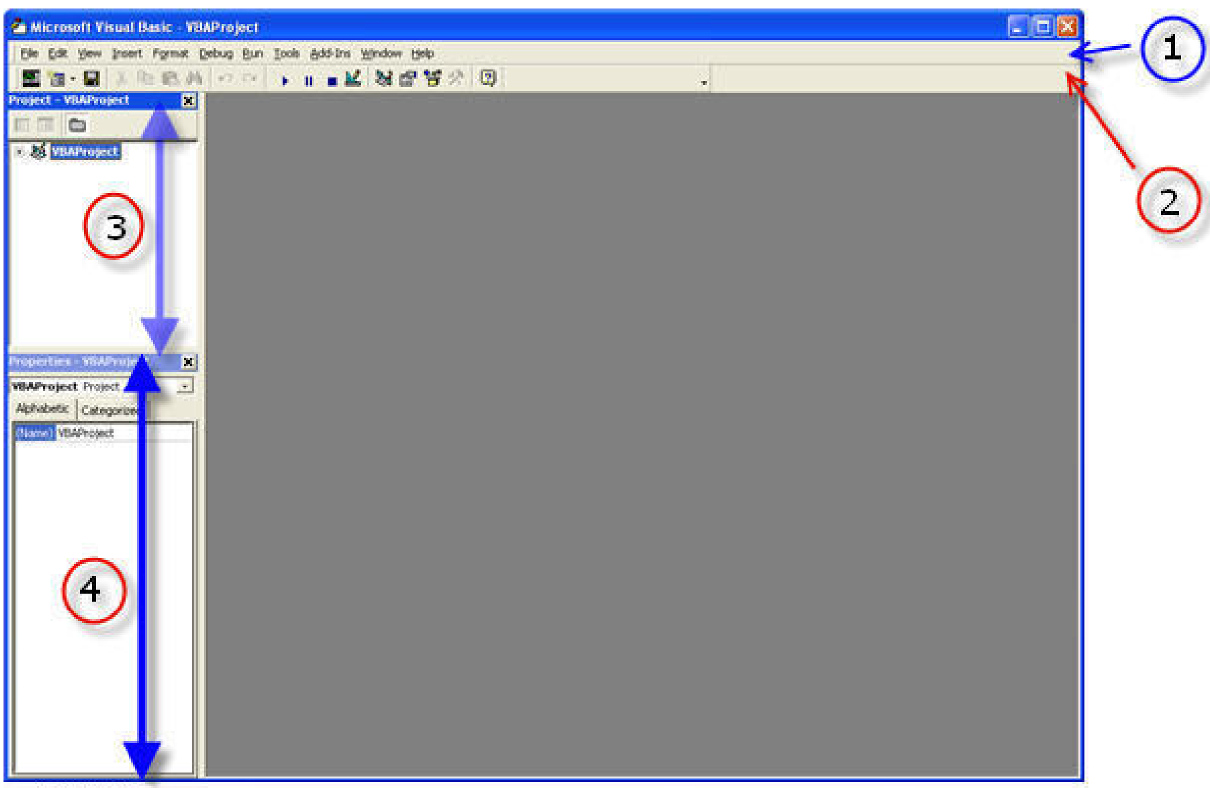
From the E5061B measurement screen, launch Visual Basic Editor using one of the following methods:

1. **Macro Setup** > **VBA Editor**
2. Press **Alt + F11** keys on the keyboard.

Initial Screen of Visual Basic Editor

When you launch Visual Basic Editor, it displays the initial screen, which contains a number of windows as shown in the following figure. The initial screen provides the following GUI elements:

Example of Visual Basic Editor initial screen



e5071c119

1. Menu bar
2. Toolbar
3. Project Explorer

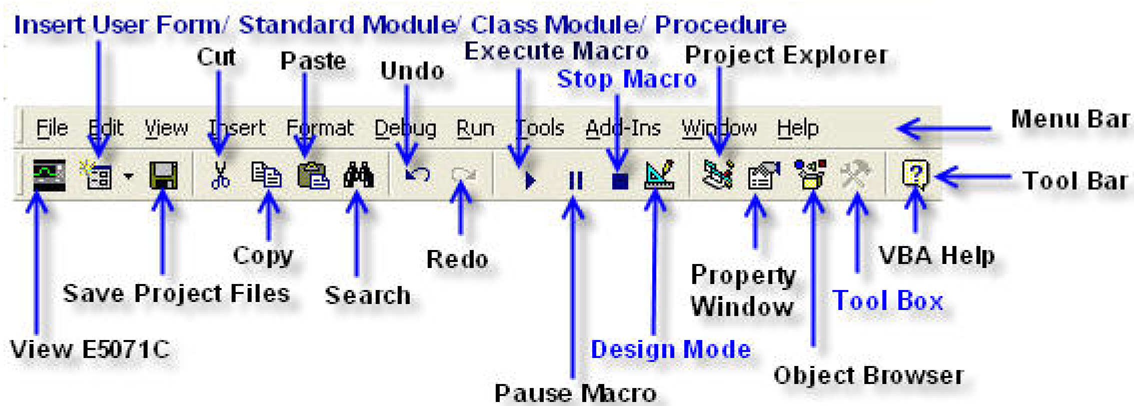
4. Property Window

Menu Bar

Clicking one of the menu labels brings up the corresponding menu. The menu bar can be used as the primary method to navigate through E5061B's VBA environment.

Toolbar

The toolbar provides access to commonly used commands via icon buttons; these commands are a subset of the commands accessible from the menu bar.



e5071c120

Project Explorer

Within the E5061B's VBA environment, you can develop your application as a project that consists of a number of files (modules). Project Explorer shows a list of all files (modules) that make up a project. The list also includes files (modules) created or loaded in Visual Basic Editor. For information on modules, refer to A Project and Three Types of Module.

To display the project explorer, do one of the following:

1. On the **View** menu, click **Project Explorer**.
2. Press **Ctrl** **+** **R** keys on the keyboard.
3. On the toolbar, click .

Property Window

A property window shows the settings (label, font, color, size, etc.) of a control (such as a command button or text box) placed on the user form. For information on user forms, refer to User Form.

You can also set properties by programming in the code window.

To display the properties window, do one of the following:

1. On the **View** menu, click **Properties Window**.
2. Press **F4** key on the keyboard.
3. On the toolbar, click .

Closing Visual Basic Editor

This section describes how to quit Visual Basic Editor. Close the Visual Basic Editor using either one of the following methods:

- On Visual Basic Editor's **File** menu, click **Close and Return to E5061**.
- Within Visual Basic Editor, press **Alt** + **Q** keys on the keyboard.
- **Macro Setup** > **Close Editor** (E5061B measurement screen)

NOTE Whenever you launch Visual Basic Editor, it automatically displays the project files you were working with in the previous session. However, once you turn OFF the power to the E5061B, the project files kept in memory will be lost; therefore, it is strongly recommended to save your VBA programs before you turn OFF the power.

Other topics about Operation Basics

Switching to the E5061B Measurement Screen

You can switch to the E5061B measurement screen without closing Visual Basic Editor.

- On the **View** menu, click **E5061**.
- Press **Alt** + **F11** keys on the keyboard.
- On the toolbar, click "E5061B" icon.
- Press **Foc** key on the E5061B front panel.

Other topics about Operation Basics

Making a Preparation Before Coding

Other topics about Operation Basics

A Project and Three Type of Modules

Project Explorer displays a list of files (modules) that are used in the E5061B VBA. This section describes a project composed of a number of files (modules) and three types of modules ("user form", "standard," and "class"). Each type of module serves its own purpose as described below.

Project

When you develop an application within the E5061B's VBA environment, you use a number of VBA program files (modules), and manage them as one project. The project is saved with the file extension ".vba".

User Form

A user form contains controls such as buttons and text boxes. You can code event-driven procedures that are invoked when a particular event occurs on a particular control, thereby creating a user interface. The user form is saved with the file extension ".frm".

Standard module

A standard module contains a collection of one or more procedures (subprograms enclosed between Sub and End Sub). One typical use of a standard module is to contain shared subroutines and globally called functions. The standard module is saved with the file extension ".bas".

Class Module

A class module contains both data and procedures and acts as one object. Once you have created a class module that serves as an object, you can create any number of instances of that object by naming each instance as an object variable. While each procedure must be unique in a standard module, you can have multiple instances of an object created through a class module. The class module is saved with the file extension ".cls".

Displaying a Code Window

The code window appear on the Visual Basic Editor by inserting the modules in a project. You can practically do coding (programming) on this code window.

The E5061B's VBA environment does not allow you to manage multiple projects. When the current project is exists in the Visual Basic Editor by loading the saved project file, you can replace the current project with a new project by the following method shown the E5061B measurement screen.

1. **Macro Setup** > **New Project**

   NOTE When you replace the current project with a new project, a message prompts to save the current project. If you want to save the project, click **Yes** button to display a dialog box for saving. For saving the project, see Saving a Project.

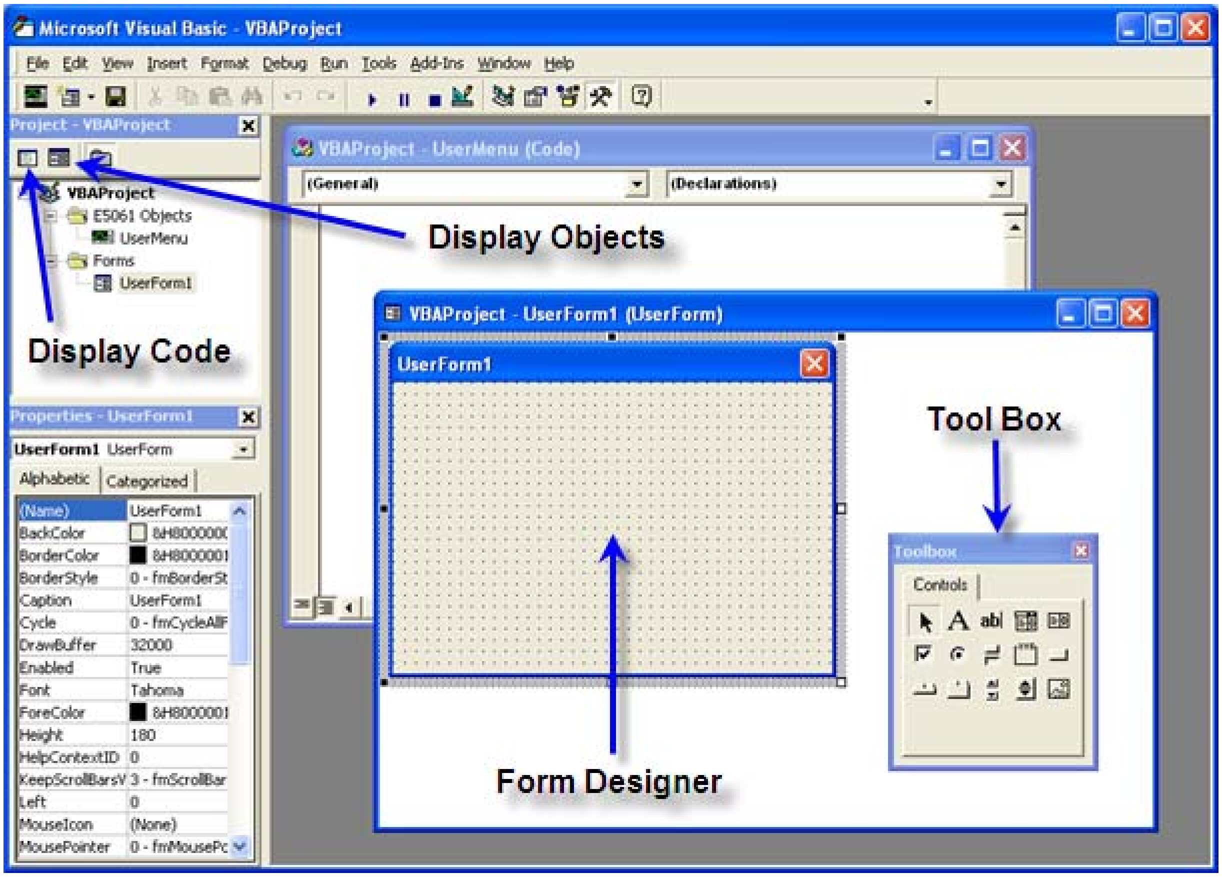
Inserting the User Form

Within Visual Basic Editor, do one of the following to add a user form to your project.

1. On the **Insert** menu, click **UserForm.**
2. On the toolbar, click "Insert User Form/Standard Module/Class Module/Procedure" icon, and click **UserForm**.
3. In Project Explorer, right-click the "VBAProject" icon, and click **Insert** > **UserForm**.

   NOTE Adding a user form does not automatically open the code window for that user form. To open the code window, click the "Display Code" icon on Project Explorer in the following figure or double-click a control placed on the user form.

Adding a user form

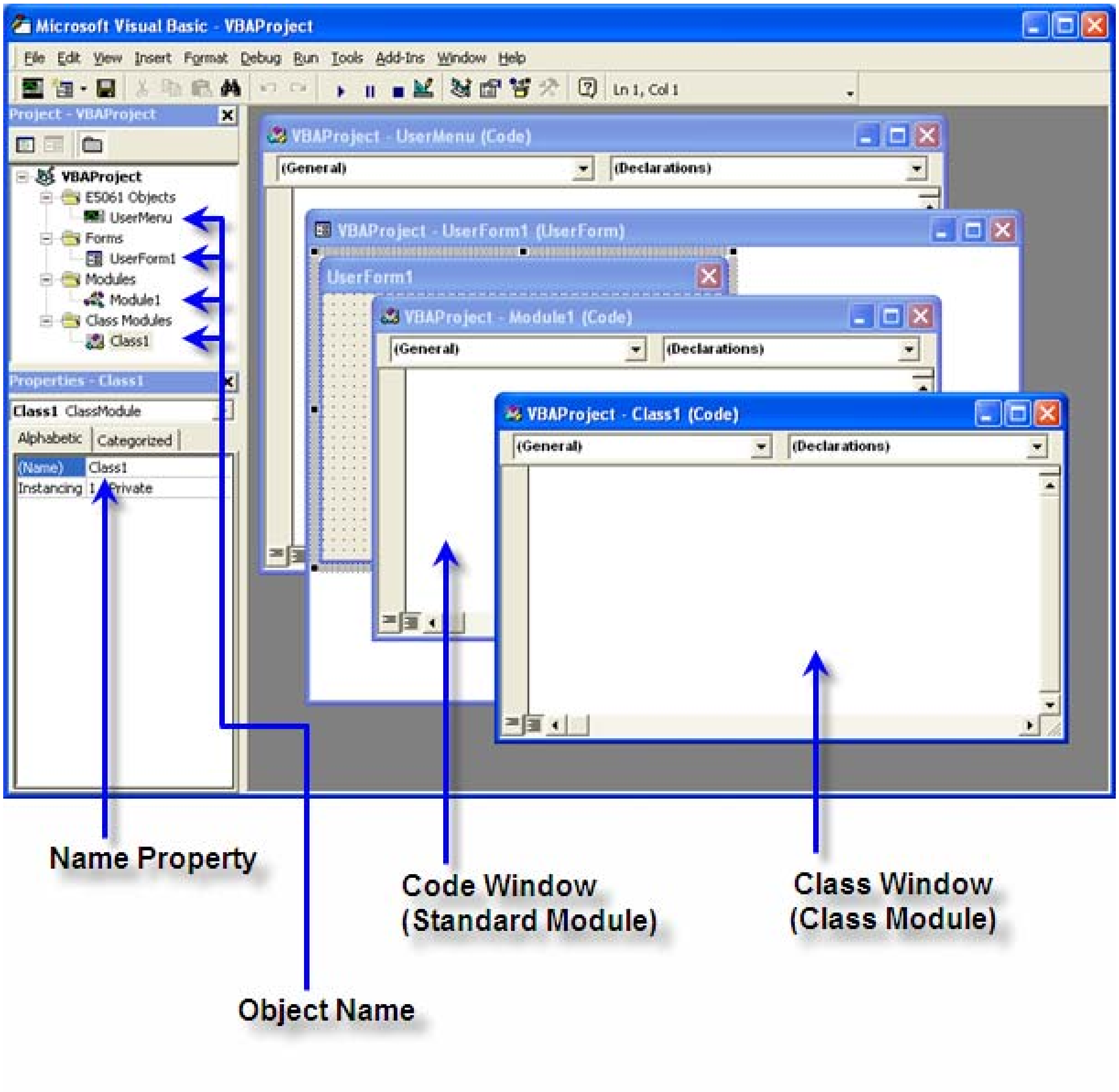


Inserting the Standard Module

Within Visual Basic Editor, do one of the following to add a standard module to your project.

1. On the **Insert** menu, click **Module**.
2. On the toolbar, click "Insert User Form/Standard Module/Class Module/Procedure" icon, and click **Module**.
3. In Project Explorer, right-click the "VBAProject" icon, and click **Insert** > **Module**.

Adding a standard module/class module



Inserting the Class Module

Within Visual Basic Editor, do one of the following to add a class module to your project.

1. On the **Insert** menu, click **ClassModule**.
2. On the toolbar, click "Insert User Form/Standard Module/Class Module/Procedure" icon, and click **ClassModule.**
3. In Project Explorer, right-click the "VBAProject" icon, and click **Insert** > **ClassModule.**

Deleting Modules

You can delete any unnecessary module from the project within Visual Basic Editor. The following procedure assumes that you want to delete a class module named "Class1".

1. In Project Explorer, click the "Class1" class module under the "Class Modules" icon to highlight it.
2. Delete the "Class1" class module using one of the following methods:
    a. On the **File** menu, click **Remove Class1...**.
    b. Click the right mouse button, and click **Remove Class1...**.
3. When you are prompted to confirm whether to export (save) "Class1", click **No**. Alternatively, you can click **Yes** if you want to save the module.

Coding a VBA Program

Other topics about Operation Basics

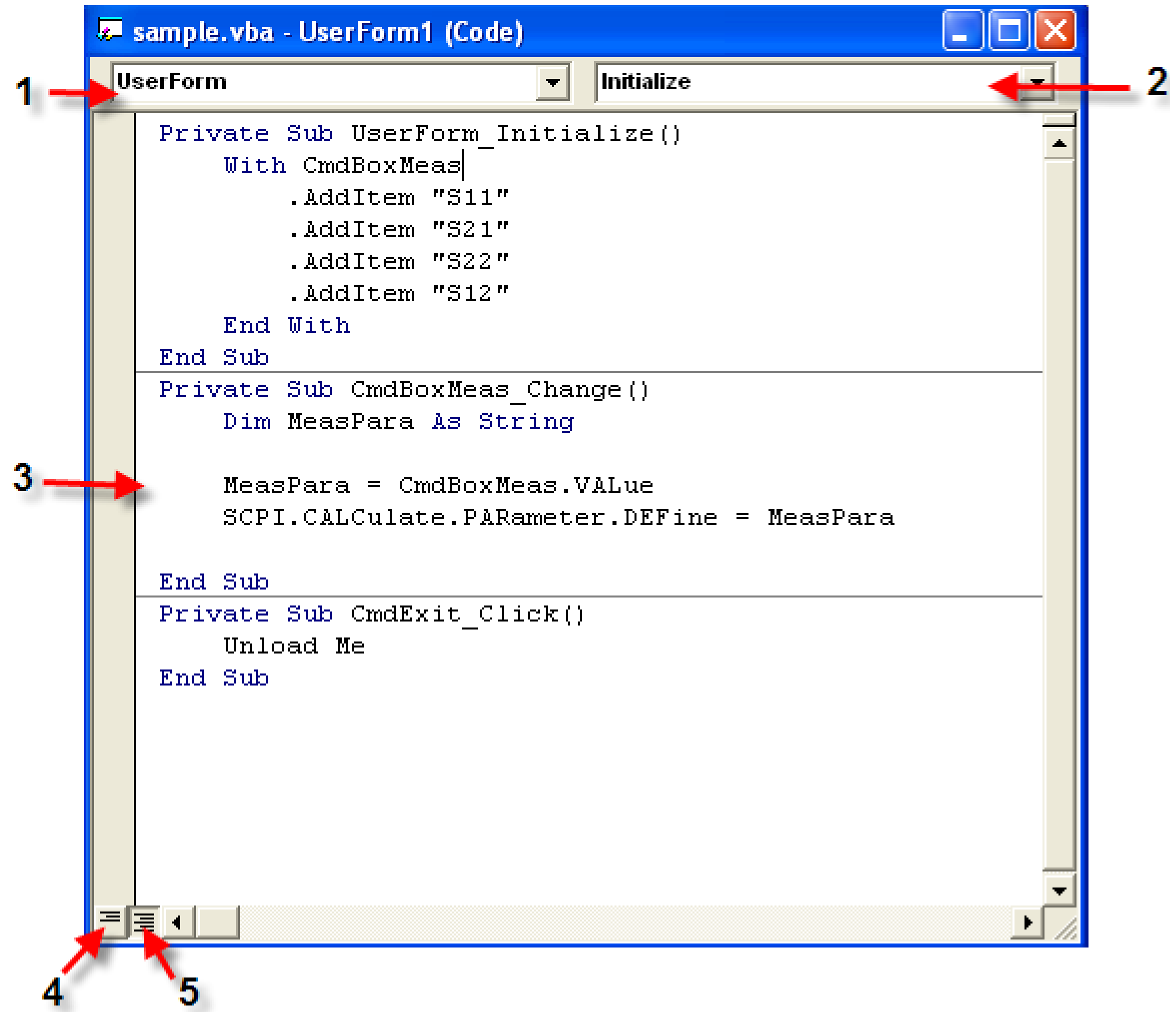
Overview

This section provides descriptive information on the user interface elements of a code window that lets you code a VBA program, and walks through a sample program (procedure) that finds the maximum value contained in an array so you can gain insight into how to create your own programs.

User Interface Elements of a Code Window

A code window is where you code a VBA program. When you are working with a user form, you can open the code window for that user form by double-clicking a control (such as a button or text box) placed on the form. Similarly, when you are working with a standard or class module, you can open the code window associated with that module by double-clicking the module's icon in Project Explorer.

Code window for a standard module



e5061b070

1. Object box

Provides a list of objects currently used within the code window.

2. Procedure box

Provides a list of procedures that reside within the code window. When you are working with a user form, this provides a list of events (actions such as click or double-click).

3. Margin indicator bar

Primarily intended for use when debugging a program.

4. Show Procedure button

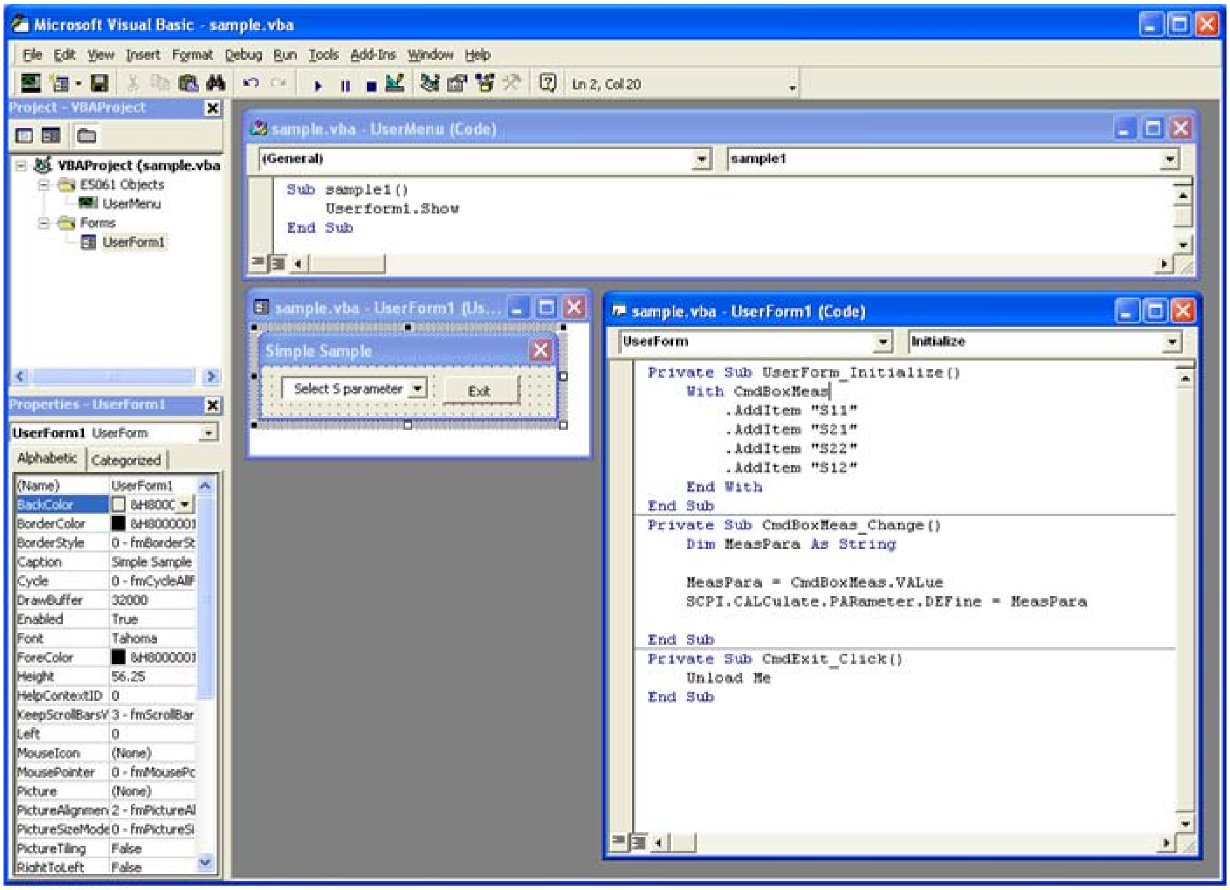
Displays only the procedure at the cursor position.

5. Show Module button

Displays the entire program contained in the code window.

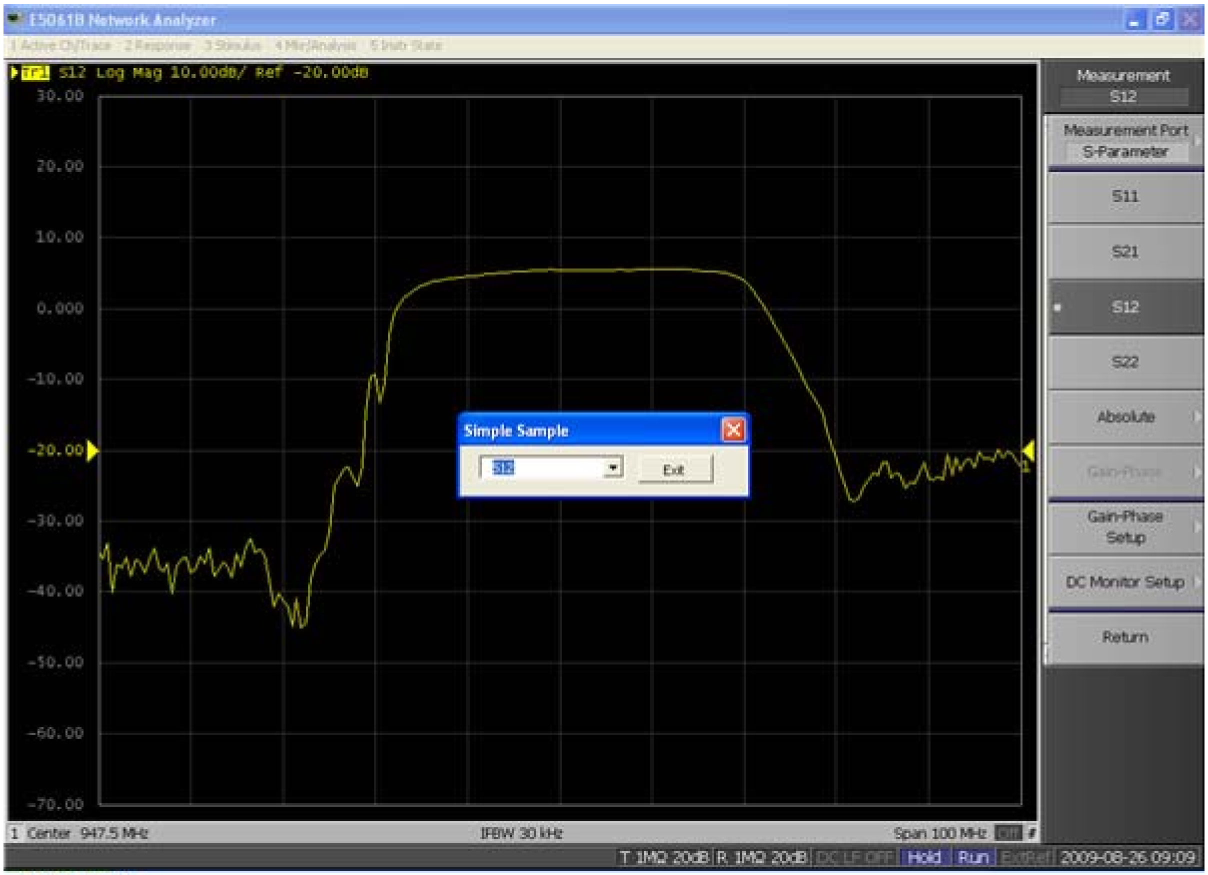
Creating a Simple VBA Program

The following figure shows a simple sample program. This program allows you to select the measurement parameter from S11, S21, S22 and S12.



e5061b069

When you run this program, the following dialog box is displayed and you can change measurement parameter by selecting parameter from combo box. Clicking the exit button quit the program.



e5061b068

Auto-complete Feature

When you use COM objects in Visual Basic Editor, the editor's auto-complete feature allows you to easily type in keywords without misspelling them.

The following procedure assumes that you are entering the SCPI.INITiate(Ch).CONTinuous object.

1. In a standard module, type **sub main** and press the **Enter** key. **End Sub** is automatically added.
2. Typing **scpi** followed by a dot (.) brings up a list of classes under the SCPI class.
3. Typing **in** automatically moves focus to **INITiate** in the list box.
4. Typing **(** brings up a list of indexes.
5. Typing **1).** brings up a list of classes under the INITiate class.
6. Typing **c** automatically moves focus to **CONTinuous** in the list box.
7. Typing **=** brings up a list box for setting a Boolean value (**True/False**).
8. Typing **t** automatically moves focus to **True**.

9. Pressing the **Enter** key completes the statement:
   SCPI.INITiate(1).CONTinuous = True.

Saving a VBA program

Other topics about Operation Basics

Overview

You can save VBA programs either as one complete project or on a module by module basis.

Saving a Project

When you opt to save your program as one complete project, you can have the files (modules) making up the project into a single package. A project is saved as a .vba file. You can save your program to a project file using one of the following two methods:

Saving a Project from Visual Basic Editor

1. Open the Save As dialog box by doing one of the following:
   - On the **File** menu, click **Save xxx.VBA**. "**xxx**" represents the file name.
   - On the toolbar, click "Save Project File" icon.
   - Press **Ctrl** + **S** keys on the keyboard.
2. The Save As dialog box appears. Specify the file name and location (drive or folder) and click **Save**.

E5061B Saving a Project from the E5061B Measurement Screen

1. Display the E5061B measurement screen following the instructions given in Switching to the E5061B Measurement Screen.
2. Open the Save As dialog box using the following key sequence:
   - **Macro Setup** > **Save Project**
3. The **Save As dialog box** appears. Specify the file name and location (drive or folder) and click Save.

Saving a Module (Exporting)

Alternatively, you can save each module (user form, standard, or class) of your VBA program individually. To save a module, you must use Visual Basic Editor. User forms are saved as .frm files, standard modules as .bas files, and class modules as .cls files.

a. In Project Explorer, click the file name that appears under the desired module icon to highlight it.

b. Open the Export File dialog box by doing one of the following:
   - On the **File** menu, click **Export File...**.
   - Click the right mouse button, and click **Export File...**.
   - Press **Ctrl** + **E** keys on the keyboard.

c. The **Export File dialog box** appears. Specify the file name and location (drive or folder) and click **Save.**

Loading a VBA Program

- Overview
- Loading a Project

Other topics about Operation Basics

Overview

Once you have saved a project or module file, you can load it later whenever necessary.

Loading a Project

You can load a saved project file either from the E5061B measurement screen or by specifying that the project file be automatically loaded when the power is turned ON.

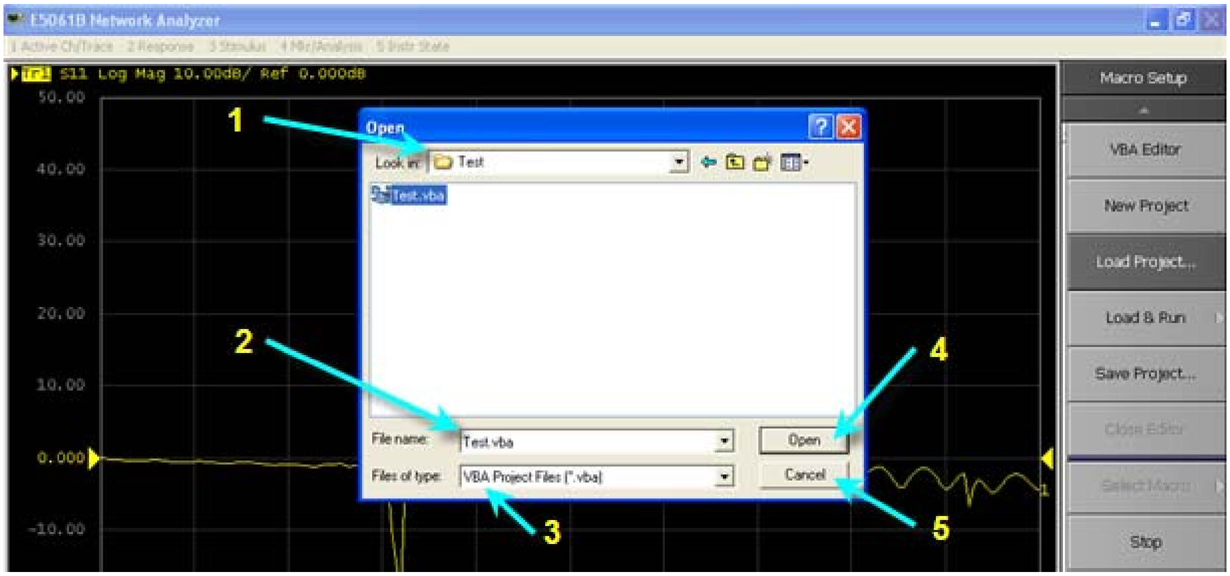
Loading a Project from the E5061B Measurement Screen

1. Press **Macro Setup** key, then click **Load Project.**

   NOTE When another project has already been loaded on the Visual Basic Editor, the message prompts to save the current project. If you want to save the project, click **Yes** button to display a dialog box for saving. For saving the project, see Saving a Project.

2. The Open dialog box appears. Specify the file name and location (drive or folder) of the file you want to load and click **Open**.

Open dialog box



The Open dialog box has the following user interface elements:

1. **Look in**: Specify the location (drive or folder) where the project resides.
2. **File name**: Specify the file name of the project you want to load.
3. **Files of type**: Select the type of the file you want to load. Normally, you should select **VBA Project Files [*.vba]**.
4. **Open**: Clicking this button loads the project.
5. **Cancel**: Clicking this button closes the Open dialog box and brings you back to the main screen.

Automatically Loading a Project at Power-On

Once you have saved a project file that satisfies the following conditions, the project loads automatically whenever the power is turned ON.

Auto-loaded project	Conditions
Directory where the project resides.	**A**:\ or **D**:\
Project file name	autoload.vba

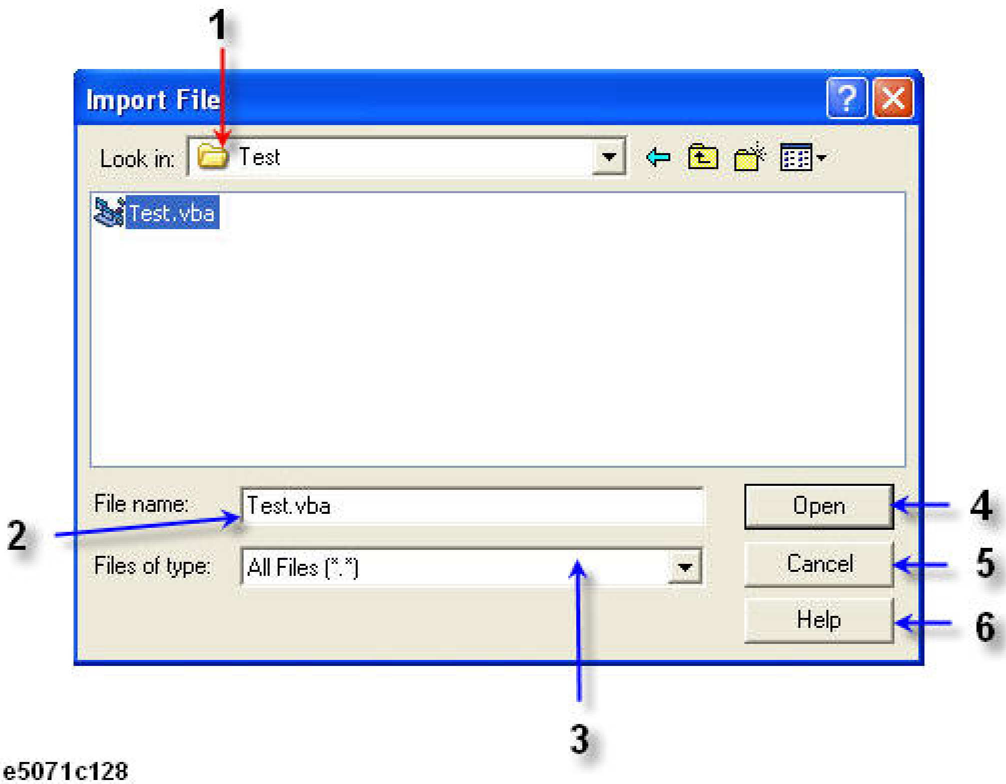
If there is the file named "autoload.vba" in both the A drive and the D drive, the file in the A drive is used.

Loading a Module (Importing)

To load a saved module into a project, you must use Visual Basic Editor.

1. In Project Explorer, click the file name that appears under the desired module icon to highlight it.
2. Open the Import File dialog box by doing one of the following:
   - On the **File** menu, click **Import File...**.
   - In Project Explorer, right-click the "VBAProject" icon, and click **Import File...**.
   - Press **Ctrl** + **M** keys on the keyboard.
3. The Import File dialog box appears. Specify the file name and location (drive or folder) of the file (module) you want to load and click **Open**.
4. The Import File dialog box has the following user interface elements:

Import File dialog box



e5071c128

The Import File dialog box has the following user interface elements:

1. **Look in**: Specify the location (drive or folder) where the module resides.
2. **File name**: Specify the file name of the module you want to load.
3. **Files of type**: Select the type of the file you want load. Normally, you should select **VB Files [*.frm,*.bas,*.cls]**.
4. **Open**: Clicking this button loads the module.
5. **Cancel**: Clicking this button closes the Import File dialog box and brings you back to the main screen.
6. **Help**: Clicking this button brings up VBA Online Help.

Running a VBA Program

Other topics about Operation Basics

Overview

The E5061B provides 2 methods to execute a VBA program: executing a program that you loaded previously and loading and executing a program in a batch process. The execution status of the VBA program is indicated in the instrument status bar, as shown in the following figure. "Run" indicates that the program is running while "Stop" indicates that the program has stopped.

Instrument status bar indicating the status of the VBA program

Running a Previously Loaded VBA Program

Running a Program from Visual Basic Editor

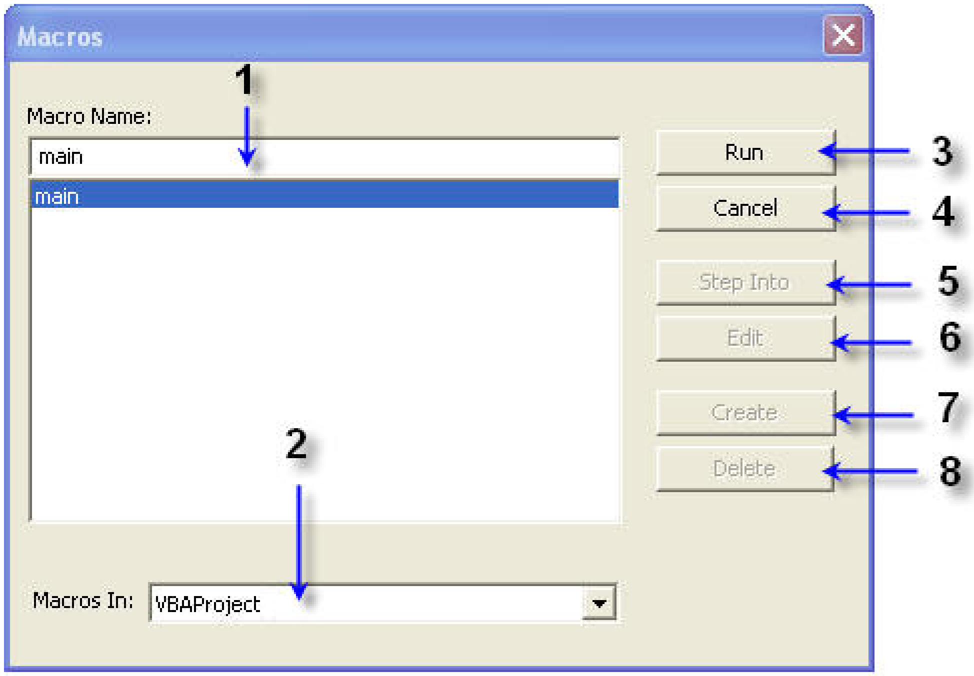
The E5061B allows you to run a previously loaded VBA program using one of the four methods listed below.

1. Open the Macros dialog box by doing either one of the following:
   - On the **Run** menu, click **Run Macro**.
   - On the **Tools** menu, click **Macros...**.
   - On the toolbar, click "Run Macro" icon.
   - Press **F5** key on the keyboard.

   NOTE Doing the above steps with the cursor positioned within a procedure in the code window runs the program immediately without displaying the Macros dialog box.

1. In the Macros dialog box, select the VBA program (procedure name) you want to run, and click the **Run** button.

Macros dialog box



e5071c133

The Macros dialog box has the following user interface elements:

1. **Macro Name**: Select the VBA program (procedure name) you want to run from the list box so that its name appears here.
2. **Macro In**: Specify the project that contains the VBA program you want to run. Normally, use the default.
3. **Run**: Clicking this button runs the selected VBA program (procedure).
4. **Cancel**: Clicking this button closes the Macros dialog box and brings you back to the main screen.
5. **Step Into**: Clicking this button brings up Visual Basic Editor and put it into step-in mode, where the selected VBA program is run step by step. This mode is primarily intended for use when debugging a VBA program. For more information on step-in mode, see Debug Toolbar.
6. **Edit**: Displays the code of the selected VBA program. You can use this for re-editing your code.
7. **Create**: This button is normally dimmed.
8. **Delete**: Clicking this button deletes the selected VBA program. Take care not to inadvertently delete your VBA program before saving it.

The Macros dialog provides access to subprograms (procedures enclosed between Sub and End Sub) created in a standard module.

Running a Program from the E5061B Measurement Screen

The E5061B allows you to run a program from E5061B screen using one of the four methods listed below.

1. Display the E5061B measurement screen following the instructions given in Switching to the E5061B Measurement Screen.
2. Run the VBA program (procedure) using the following key sequence:

- **Macro Setup** > **Select Macro - Module xxx**
  where "**Module**" is the object name (Name property shown in the property window) and "**xxx**" is the procedure name.
- Press the **Macro Run** key on the E5061B front panel. For a program to run from the measurement screen, its procedure name must be "Main" (subprogram enclosed between Sub Main() and End Sub), and its object name (Name property as displayed in the property window) must be "Module1".

  NOTE When you are working with the E5061B measurement screen, the E5061B's macro environment only provides access to those VBA programs that are created as subprograms (enclosed between Sub and End Sub) in a standard module.

Loading and Executing Programs in Batch Process

This section describes how to load and execute a program (VBA project) in a batch process by pressing the softkey corresponding to the program name.

1. Save the VBA program (VBA project file) into the following folder.
   D:\VBA

   NOTE This feature is available only for programs saved in **D:\VBA**. This feature is not available for programs saved in subfolders of **D:\VBA**.

   NOTE When copying a VBA program to D:\VBA from another folder, copy all the files necessary to execute the program to appropriate folders. When copying a factory-installed VBA program into D:\VBA, choose only its VBA project file.
2. Press **Macro Seup** key.
3. Click **Load & Run**.
4. Press the softkey corresponding to the VBA project file name of the program you want to execute. The pressed VBA project is loaded and the program of which the procedure name is set to "Main" (subprogram enclosed between Sub Main() and End Sub) and of which the object name (Name property as displayed in the property window) is set to "**Module1**" is executed.

NOTE There is no limit to the number of VBA project files that can be saved in **D:\VBA**. However, the maximum number of programs that can be displayed as softkeys is 50.

- File names of the VBA projects saved in **D:\VBA** are displayed as softkeys in alphabetical order.
- The maximum number of characters that can be displayed in a softkey is 12. If a file name has 13 or more characters, "..." is added to the 12th character from the beginning of the program name and displayed. In this case a .vba extension is omitted.

Stopping a VBA Program

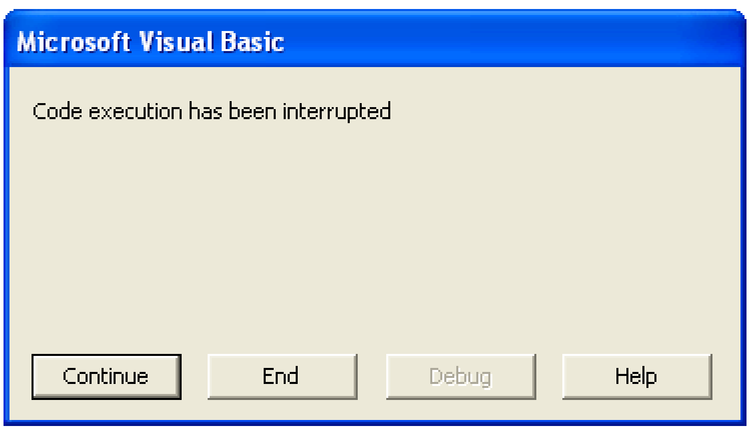
- Stopping a Procedure
- Abruptly Terminating a VBA Program

Other topics about Operation Basics

Stopping a Procedure

This section describes how to break a procedure during the execution of a VBA program.

1. To break the running VBA program, do one of the following:
    - On the **Run** menu, click **Break**.
    - On the toolbar, click "Break Macro" icon.
    - Press **Ctrl** + **Break** keys on the keyboard.
    - **Macro Setup** > **Stop** (E5061B measurement screen)
    - Press **Macro Break** key on the E5061B front panel.
2. A dialog box is displayed through forced interrupts, and the program is suspended.



e5071c250

Select one of the following:

- **Continue**: Resumes the execution of the program.
- **End**: Terminates the VBA program.
- **Debug**: Displays a run-time error.
- **Help**: Brings up VBA Online Help.

Abruptly Terminating a VBA Program

This section describes how to abruptly terminate a running procedure. When abruptly terminating the VBA program by the below methods, the "Program interrupted" message is shown on the instrument status bar at the bottom of the LCD display.

To terminate the running VBA program, perform one of the following:

- On the **Run** menu, click **Reset**.
- On the toolbar, click "Reset Macro" icon.
- Insert an *End* statement into your code.

Errors and Debugging

- Type of Errors
- Using a Debug Tool

Other topics about Operation Basics

Type of Errors

Errors in VBA programs are classified into the following two types:

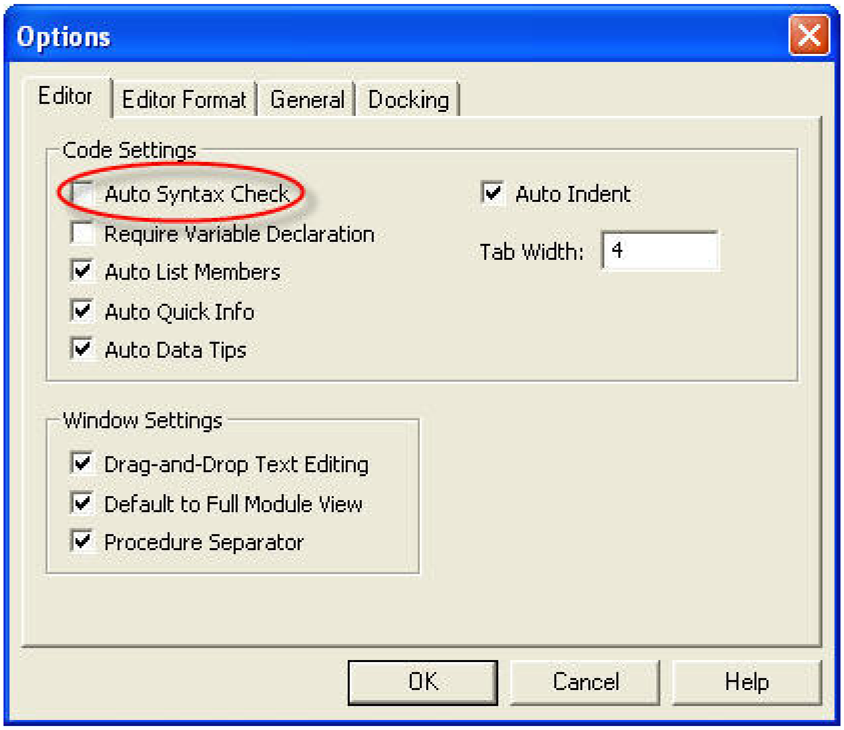
Syntax errors

A syntax error is generated when Visual Basic Editor detects an invalid statement that violates the Visual Basic syntax rules. For example, misspelled keywords generate syntax errors. An error dialog box appears that indicates the error message, and highlights the invalid statement in red. To get detailed information on the error, click the **HELP** button in the error dialog box to display the help topic on the error. You cannot run the macro until you correct the syntax error.

The E5061B VBA environment is configured by default to automatically check for syntax errors, but you can disable the auto syntax check feature using the following steps:

1. On the **Tools** menu, click **Options...**.

2. On the **Editor** tab, clear the **Auto Syntax Check** check box.



e5071c121

3. Click the **OK** button.

Run-time Errors

A run-time error is generated when a VBA program attempts to execute an invalid statement at run time. When a run-time error is generated, the program is stopped at the invalid statement, and an error dialog box appears. You can terminate the program by clicking the END button in the error dialog box. You can also click the DEBUG button in the error dialog box to identify the statement that caused the error. In this case, the statement in question is highlighted in yellow.

NOTE Some run-time errors occur under particular conditions, even though a program could run without getting errors under normal conditions. For example, the "Target value not found" error that occurs when a program that analyzes the results using the Marker Bandwidth Search feature, fails to perform bandwidth search because the marker is not in the appropriate position. The "Ecal module not in RF path" error that occurs when a program that performs calibrations using a ECal module, fails to measure the calibration data because the ECal module is not appropriately connected to test ports, and so on.

Using a Debug Tool

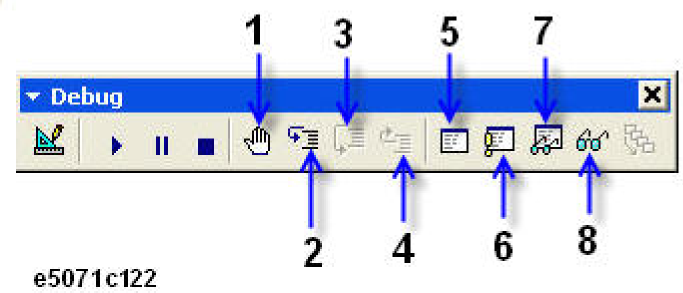
The E5061B's VBA environment provides a variety of debug tools that help you identify logical errors. Detailed information on using the debug tools is covered in VBA Online Help and books on VBA.

Debug Toolbar

The debug toolbar provides tool buttons that allow you to easily access various debug tools.

On the **View** menu, click **Toolbars** > **Debug**.

Debug toolbar



1. Set/clear break points (keyboard **F9**): Places a break point at the cursor position or clears an existing break point.
2. Step-in (keyboard: **F8**): Runs the VBA program step by step. If the current program contains a call to another procedure, that procedure is also run step by step.
3. Step-over (keyboard: **Shift** + **F8**): Runs the VBA program step by step. If the current program contains a call to another procedure, that procedure is run as one line.
4. Step-out (keyboard: **Ctrl** + **Shift** + **F8**): Executes the remaining lines of the function where the execution point is currently placed.
5. Local window: Opens the local window that shows the current values of local variables.
6. Immediate window (keyboard: **Ctrl** + **G**): Opens the immediate window that evaluates entered values of variables or expressions.
7. Watch window: Opens the watch window that displays the current value of a specified expression.
8. Quick window (keyboard: **Shift** + **F9**): Displays the current value of a specified expression in a dialog box.

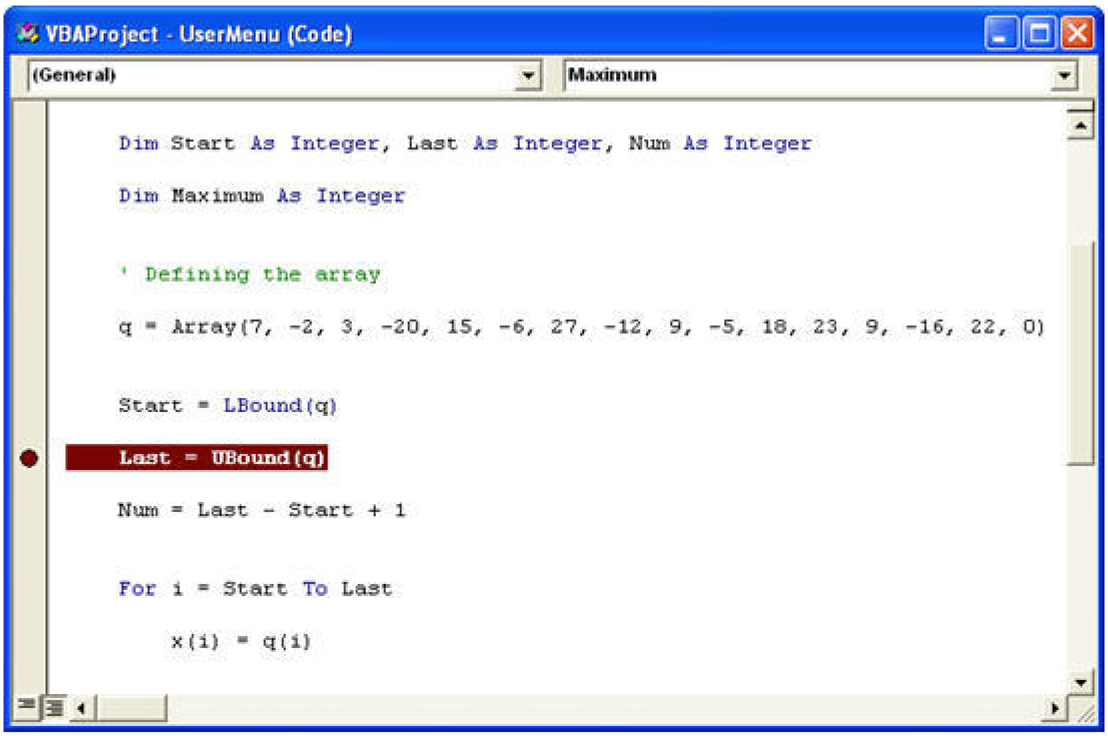
Setting a Break Point

By placing a break point at a particular statement in a VBA program, you can automatically suspend the program when it is executed to that statement.

When you put a break point at a line, the line is highlighted in umber as shown in the following figure. To set a break point, perform one of the following:

- Place the cursor at the desired line of code, and click the "Set/clear break points" button on the debug toolbar.
- Click anywhere in the margin indicator bar of the code window.

Setting a break point



e5071c123

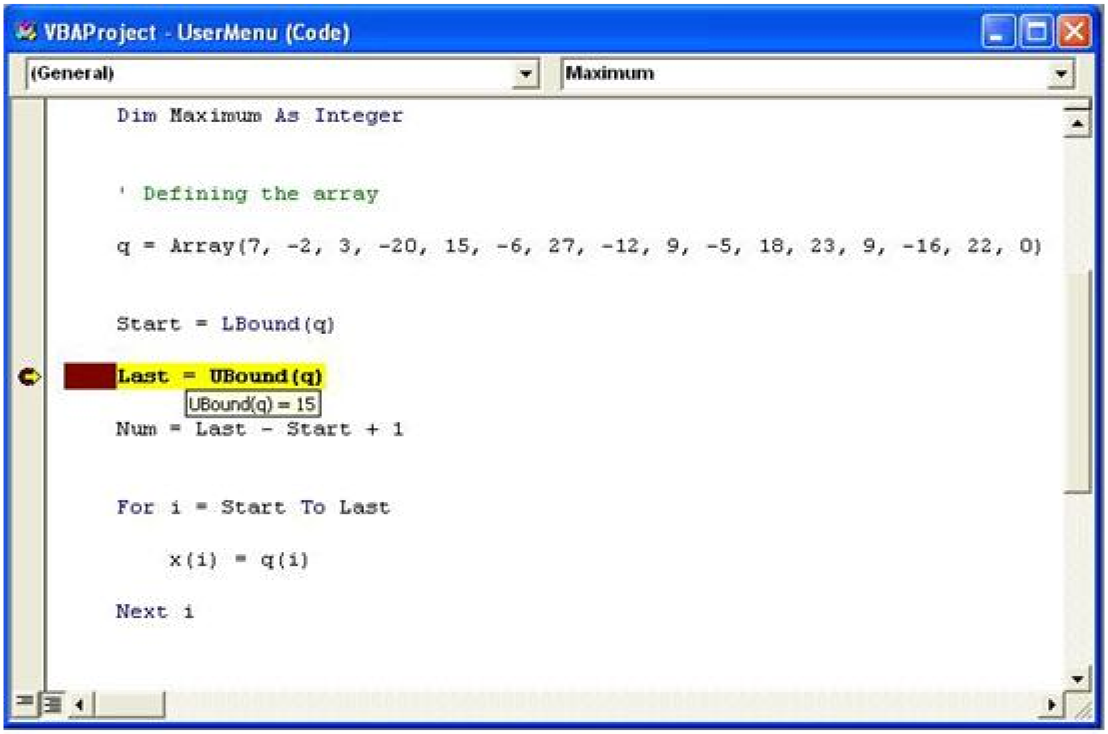
Monitoring Variable or Property Values

With your VBA program suspends, you can use the following debug tool to monitor variables or properties. To do this, you must set a break point, run the VBA program, and suspend it.

Data Hint

When you point to the variable or expression of interest, Data Hint shows the current value as shown in the following figure.

Data Hint



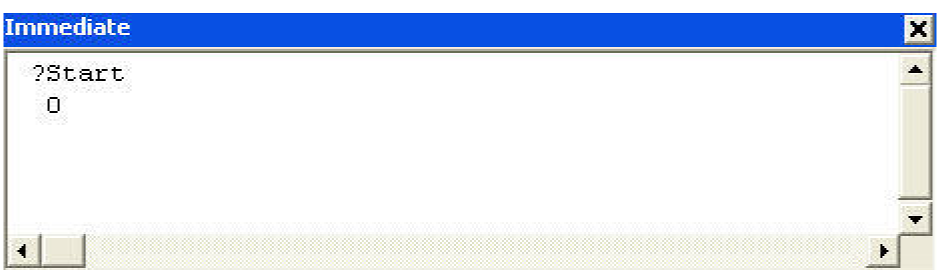
e5071c127

Immediate Window

To display the immediate window, click  on the debug toolbar.

In the immediate window, enter a question mark (?) followed by the variable or expression whose value you want to check, and press the Enter key, as shown in the following figure. The current value appears in the line that follows.

Immediate window

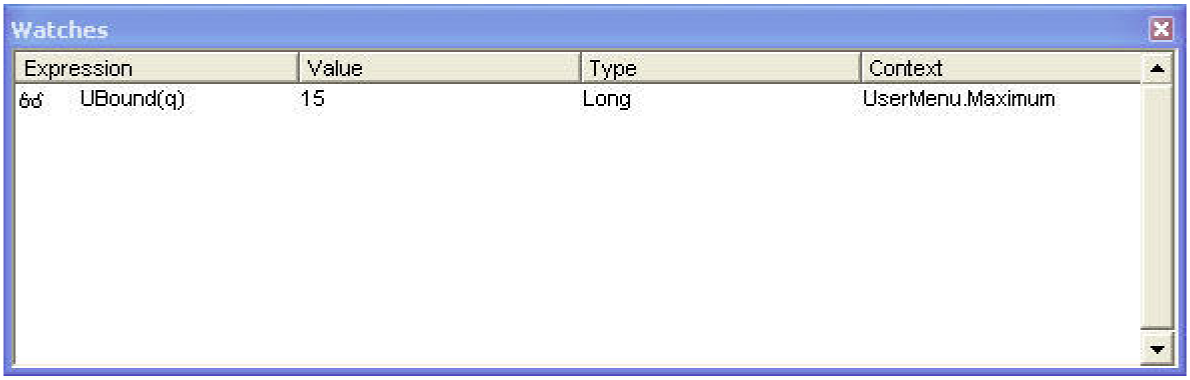


e5071c124

Watch Window

To display the watch window, click the "Watch Window" button on the debug toolbar.

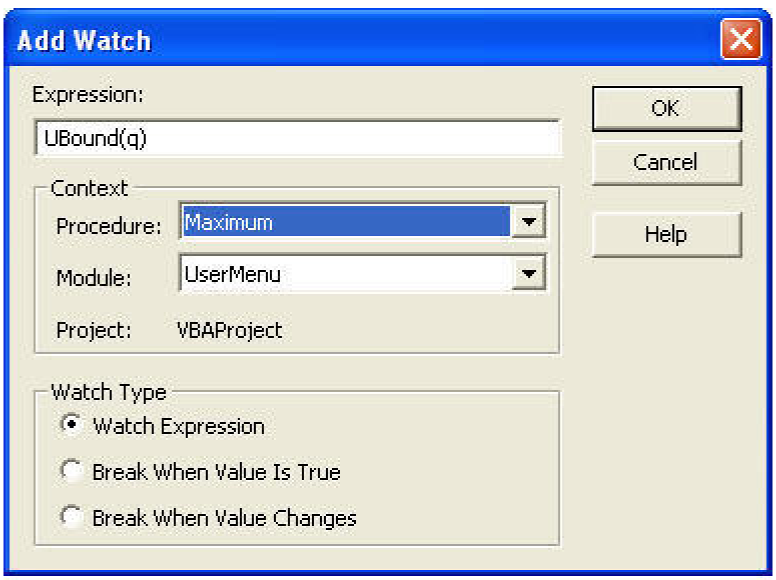
Watch window



e5071c126

1. On the **Debug** menu, click **Add Watch....** to open the Add Watch dialog box.
2. As shown in the following figure, you can specify an expression of interest as a watch expression to always monitor its value.
3. Click the **OK** button.

Add Watch dialog box



e5071c125

Quick Watch

In the code window, select a variable or expression whose value you want to watch. On the debug toolbar, click the "Quick Watch" button to open the Quick Watch dialog box. The dialog box displays the current value of your specified variable or expression.

Also, you can click the **Add** button in the Quick Watch dialog box to specify the current expression as a watch expression.

Printing Output Values in the Echo Window

Other topics about Operation Basics

Overview

The echo window, which appears at the lower section of the E5061B measurement screen, can be used to display a message or the return value (data) of an object.

Entering Values Output to Echo Window

You can use the COM objects listed below to enter values output to the echo window.

- ECHO
- SCPI.DISPlay.ECHO.DATA

Opening Echo Window

You can use the COM objects listed below to open the echo window.

- SCPI.DISPlay.TABLe.TYPE
- SCPI.DISPlay.TABLe.STATe

Alternatively, you can also open the echo window using the following key sequence:

Macro Setup > **Echo Window (ON)**

Clearing Values Output from Echo Window

You can use the COM object shown below to clear values output from the echo window.

- SCPI.DISPlay.ECHO.CLEar

Alternatively, you can also clear values output from the echo window using the following key sequence:

Macro Setup > **Clear Echo**

Uses Advanced Techniques

- Accessing a List of E5061B COM Objects
- Using Automatic Library References

Other topics about Operation Basics

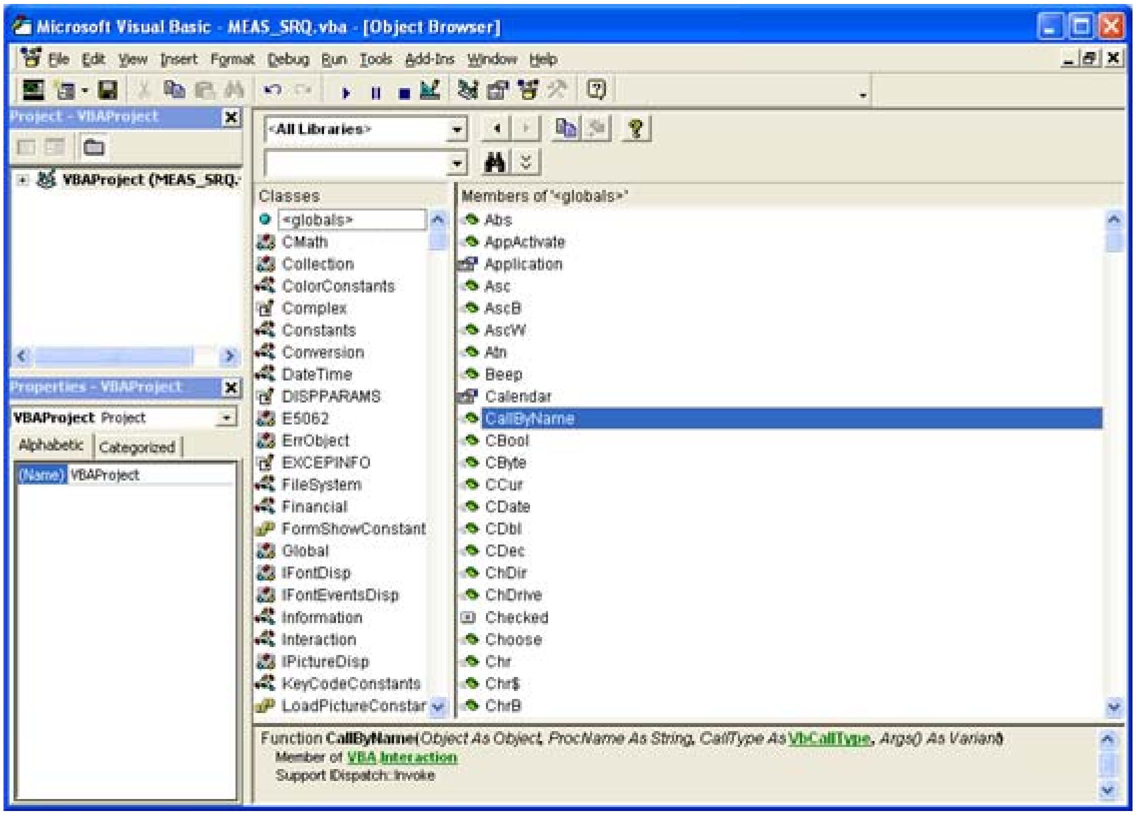
Accessing a List of E5061B COM Objects

The E5061B VBA environment provides COM objects that support the control of E5061B. When you are developing a program using E5061B COM objects, you can access a list of E5061B COM objects by opening Object Browser within Visual Basic Editor.

1. To open Object Browser, do one of the following:
   - On the **View** menu, click **Object Browser**.
   - On the toolbar, click "Object Browser" icon.
2. Select **E5062Lib** from the Project/Library box to display the E5061B library as shown in the following figure.

   NOTE There are some COM objects NOT used in controlling with E5061B VBA in the list of the E5061B COM objects displayed on the Object Browser. The COM objects NOT used in controlling with E5061B VBA are not described in the COM object reference.

How to use Object Browser



Using Automatic Library References

For libraries that satisfy the following conditions, the library reference automatically sets whenever a new project is created and loaded (**Macro Setup** > **New Project**).

Automatically referenced libraries	Conditions
Directory where the library resides.	**D:\Agilent**
Extensions of libraries	olb, tlb, dll, or ocx

To check the library reference setting, you must use Visual Basic Editor.

Follow these steps to check the library reference setting.

- On the **Tools** menu, click **References...**.

NOTE A project sets the library reference when the project is created. Therefore, if the existing project is loaded, libraries added after the development of the project are not automatically set in the library reference.

Using VBA Online Help

- [Overview](#overview)
- Accessing VBA Online Help

Other topics about Operation Basics

Overview

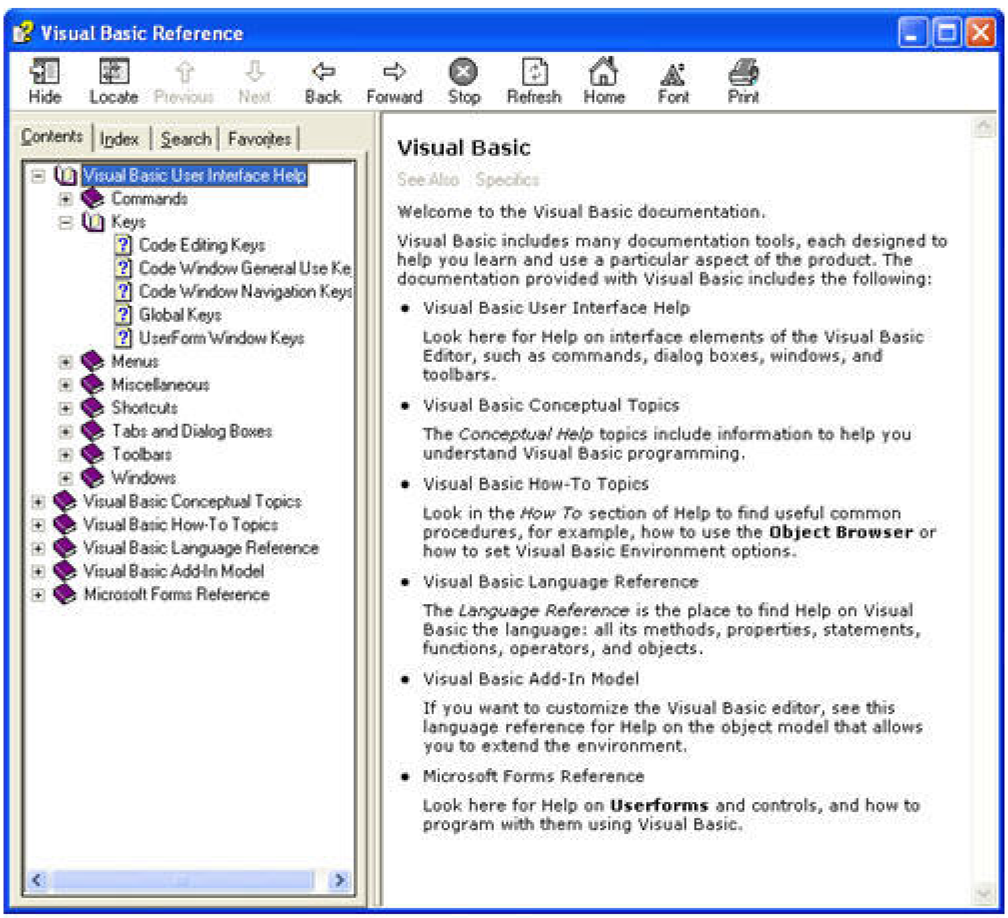
VBA Online Help provides useful topics, such as the VBA terminology or how to use a particular feature. In VBA Online Help, you can find a topic of interest through the Contents or by entering specific keywords.

Accessing VBA Online Help

From Visual Basic Editor, do one of the following to access the VBA Online Help screen.

- On the **Help** menu, click **Microsoft Visual Basic Help**.
- Press **F1** key on the keyboard.
- On the toolbar, click "VBA Help" icon.

VBA Online Help screen



e5071c135

Using the Contents Tab

Clicking the **Contents** tab in the VBA Online Help screen brings up the items listed below. The E5061B VBA Online Help has a hierarchical table of contents. Click an item to expand it, and then find a topic of interest.

- Visual Basic User Interface Help
- Visual Basic Conceptual Topics
- Visual Basic How-To Topics
- Visual Basic Language Reference
- Visual Basic Add-In Model
- Microsoft Forms Reference

When you need information on using Visual Basic Editor, use User Interface Help and How-To Topics as primary sources of information. Format of VBA program is covered in Visual Basic Conceptual Topics. Properties and methods supported by VBA are covered in Visual Basic Language Reference and Visual Basic Add-In Model. Information on using user forms is covered in Microsoft Forms Reference.

Using the Index Tab

In the VBA Online Help screen, click the **Index** tab, and enter a keyword(s) into the text box. For example, you may wish to search for "Sub" or "With" when you are writing your own code.

Looking up a Keyword in the Code within Visual Basic Editor

When you want to know the usage or meaning of a keyword in a sample program or some other code, you can quickly access the help topic on that keyword by moving the cursor to the keyword and pressing **F1** key.

Controlling E5061B

Controlling E5061B

- Detecting the End of Measurement
- Reading/Writing Measurement Data
- Executing a Procedure with a Softkey (User Menu Function)

Detecting End of Measurement

- Overview
- Using Status Register
- Using SCPI.TRIGger.SEQuence.SINGle Object

Other topics about Controlling E5061B

Overview

This chapter uses sample programs to demonstrate how to trigger the instrument to start a new measurement cycle and how to detect the end of a measurement cycle. The trigger system is responsible for such tasks as detecting the start of a measurement cycle (triggering) and enabling/disabling measurement on each channel.

You can detect the end of measurement by using either the status register or the SCPI.TRIGger.SEQuence.SINGle object.

Using Status Register

The status of the E5061B can be detected through the status register. If your program is based on SPCI commands, you can use SRQ (Service Request) interruptions to detect the end of measurement.

On the other hand, if your program is based on COM objects, SRQ interruptions are not available; instead, you can use the following object to suspend the program until SRQs are generated upon completion of measurement.

- WaitOnSRQ

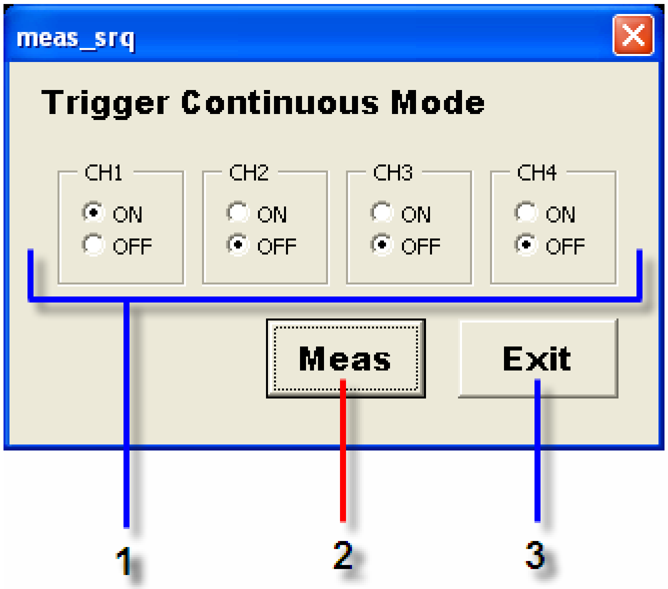
Sample program is available to download from the Agilent Support page, named **meas_srq.vba.** It demonstrates how to use the status register to suspend the program until the end of measurement. This VBA program consists of the following modules:

Object name	Module type	Content
frmSrqMeas	UserForm	Uses the status register to wait for the end of measurement.
mdlSrqMeas	Standard module	Invokes a UserForm.

NOTE This sample program correctly runs when the maximum number of channels/traces is set to 4 channels/4 traces.

When you run this VBA program, the following UserForm appears.

The UserForm when running the meas_srq.vba program



For how to use each element, see the following description.

1. The program turns ON/OFF Continuous Activation mode for each channel and determines whether to enable or disable each channel for measurement.
2. The program triggers the instrument to start a new measurement cycle, waits for the end of measurement, and then displays a message. For detail, see the description of the code window.
3. The program exits, and the UserForm disappears.

In Visual Basic Editor, open the UserForm (object name: frmSrqMeas), and double-click the **Meas** or **Exit** button to bring up the code window. The following is the description of the subprograms associated with the respective buttons.

Using SRQs to detect the end of measurement (object name: frmSrqMeas)

```
''''
' Procedure called when the user clicks the Exit button on the UserForm.
''''
private Sub cmdExit_Click()
'
' Unloads the UserForm from the memory, and terminates the program.
'
```

```
    Unload Me
'
End Sub
'
'
''
' Procedure called when the user clicks the Meas button on the UserForm.
''
Private Sub cmdMeas_Click()
    Dim Cond As Boolean
'
' Hides the UserForm (object name: frmSrqMeas) from the screen.
'
    frmSrqMeas.Hide
'
' Displays 4 channel windows.
'
    SCPI.DISPlay.Split = "d12_34"
'
' Sets the trigger source to "bus".
'
    SCPI.TRIGger.SEQuence.Source = "bus"
'
' These lines turn on or off Continuous Activation mode for each channel
' depending on whether the corresponding option buttons are on or off.
' By default, the mode is turned on for channel 1 only.
'
    SCPI.INITiate(1).CONTinuous = optOn1.Value
    SCPI.INITiate(2).CONTinuous = optOn2.Value
    SCPI.INITiate(3).CONTinuous = optOn3.Value
    SCPI.INITiate(4).CONTinuous = optOn4.Value
'
' These lines configure the instrument so that operation status event
' register's bit 4 is set to 1 only when operation status condition
' register's bit 4 is changed from 1 to 0 (negative transition).
'
    SCPI.STATus.OPERation.PTRansition = 0
```

```
   SCPI.STATus.OPERation.NTRansition = 16
'
' Enables the operation status event register's bit 4.
'
   SCPI.STATus.OPERation.ENABle = 16
'
' Enables the status byte register's bit 7.
'
   SCPI.IEEE4882.SRE = 128
'
' Clears the status byte register and operation status event register.
'
   SCPI.IEEE4882.CLS
'
' Triggers the instrument to start a measurement cycle.
'
   SCPI.IEEE4882.TRG
'
' Verifies that the instrument is in a measurement cycle, and suspends
' the program until the end of measurement.
' The time-out is set to 100 seconds (maximum value).
'
   WaitOnSRQ Cond, 100000
'
' These lines display a measurement completion message upon detecting
' the end of measurement.
'
   If Cond = True Then
     MsgBox "Measurement Completion"
   End If
'
' Displays the UserForm (object name :frmSrqMeas) on the screen.
'
   frmSrqMeas.Show
'
End Sub
```

Using the SCPI.TRIGger.SEQuence.SINGle Object

When you trigger the instrument by issuing the SCPI.TRIGger.SEQuence.SINGle object, you can use the SCPI.IEEE4882.OPC object to suspend the program until the end of measurement.

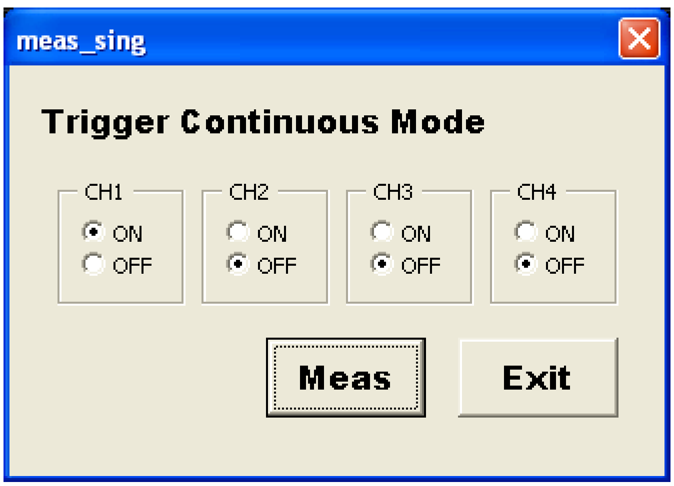
The sample program is available to download from the Agilent Support page, named **meas_sing.vba.** It demonstrates how to use the SCPI.TRIGger.SEQuence.SINGle object to suspend the program until the end of measurement. This VBA program consists of the following modules:

Object name	Module type	Content
frmSingMeas	UserForm	Uses the SCPI.TRIGger.SEQuence.SINGle and SCPI.IEEE4882.OPC objects to suspend the program until the end of measurement.
mdlSingMeas	Standard module	Invokes a UserForm.

NOTE This sample program correctly runs when the maximum number of channels/traces is set 4 channels/4 traces.

When you run this VBA program, a Userform appears.

The UserForm when running the meas_sing.vba program



In Visual Basic Editor, open the UserForm (object name:frmSingMeas), and double-click the **Meas** or **Exit** button to bring up the code window. The following is the description of the subprograms associated with the respective buttons.

Using the SCPI.TRIGger.SEQuence.SINGle object to suspend the program until the end of measurement (object name:frmSingMeas)

```
''
' Procedure called when the user clicks the Exit button on the UserForm.
''
Private Sub cmdExit_Click()
'
' Unloads the UserForm from the memory, and terminates the program.
'
    Unload Me
'
End Sub
'
'
''
' Procedure called when the user clicks the Meas button on the UserForm.
''
Private Sub cmdMeas_Click()
    Dim Dmy As Long
'
' Hides the UserForm (object name: frmSingMeas) from the screen.
'
    frmSingMeas.Hide
'
' Displays 4 channel windows.
'
    SCPI.DISPlay.Split = "d12_34"
'
' Sets the trigger source to "bus".
'
    SCPI.TRIGger.SEQuence.Source = "bus"
'
```

```
' These lines turn on or off Continuous Activation mode for each channel
' depending on whether the corresponding option buttons are on or off.
' By default, the mode is turned on for channel 1 only.
'
    SCPI.INITiate(1).CONTinuous = optOn1.Value
    SCPI.INITiate(2).CONTinuous = optOn2.Value
    SCPI.INITiate(3).CONTinuous = optOn3.Value
    SCPI.INITiate(4).CONTinuous = optOn4.Value
'
' Triggers the instrument to start a measurement cycle.
'
    SCPI.TRIGger.SEQuence.SINGle
'
' Executes the SCPI.IEEE4882.OPC object to suspend the program until
' the value of 1 is returned indicating the end of measurement.
'
    Dmy = SCPI.IEEE4882.OPC
'
' Displays a measurement completion message.
'
    MsgBox "Measurement Completion"
'
' Displays the UserForm (object name: frmSingMeas) on the screen.
'
    frmSingMeas.Show
'
End Sub
```

Reading / Writing Measurement Data

- Overview
- Sample Program

Other topics about Controlling E5061B

Overview

This section describes how to process the E5061B's internal data. You can use these internal data arrays: corrected data arrays, corrected memory arrays, formatted data arrays, formatted memory arrays, and stimulus data arrays. For more information on the internal data arrays, see Internal Data Processing.

To read/write a formatted data array, formatted memory array, corrected data array, or corrected memory array use the following objects:

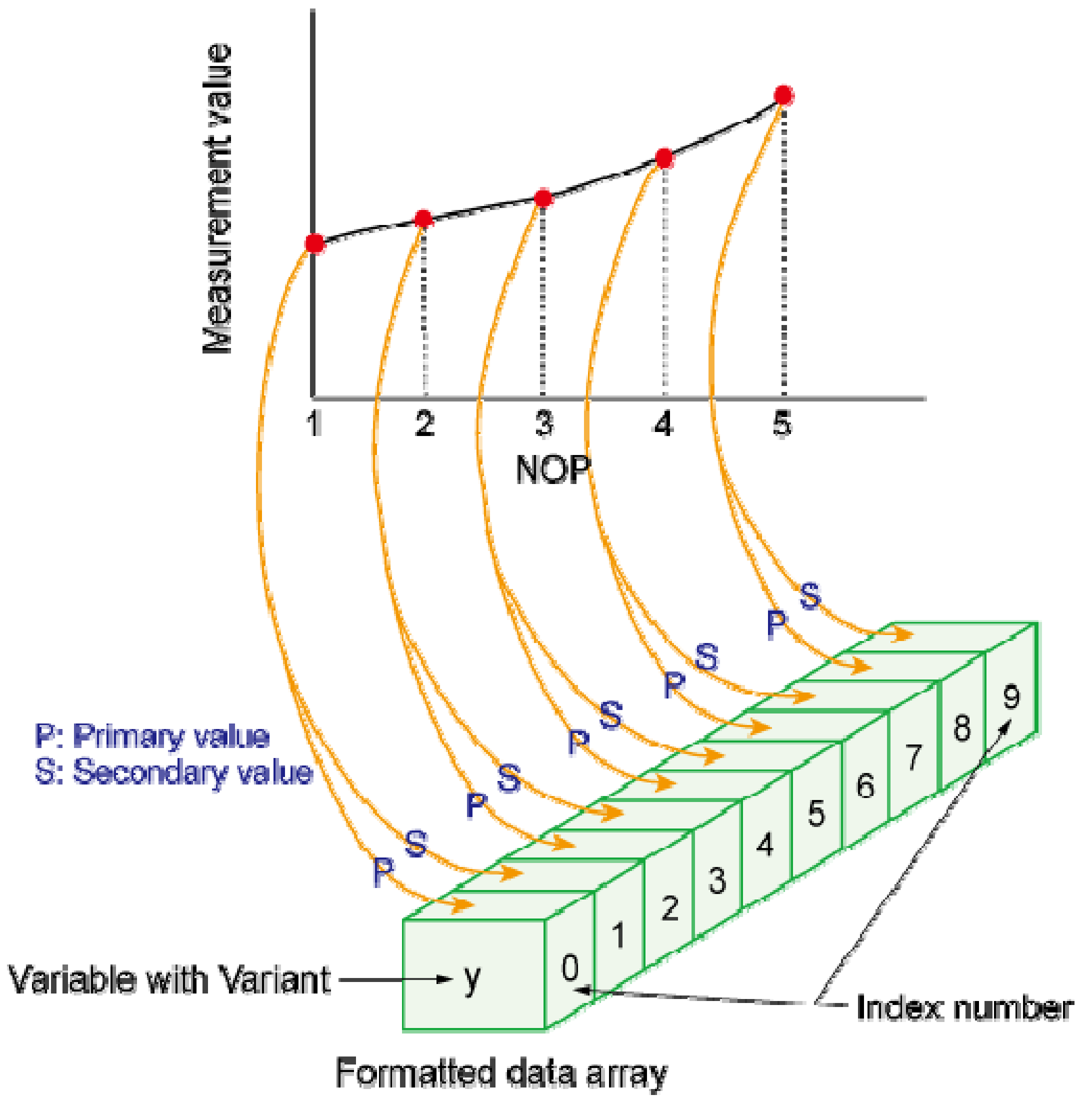
- SCPI.CALCulate(Ch).SELected.DATA.FDATa
- SCPI.CALCulate(Ch).SELected.DATA.FMEMory
- SCPI.CALCulate(Ch).SELected.DATA.SDATa
- SCPI.CALCulate(Ch).SELected.DATA.SMEMory

To read a stimulus data array, use the following objects:

- SCPI.SENSe(Ch).FREQuency.DATA

The E5061B VBA allows you to deal with multiple pieces of data through variables of Variant type. Variant variables can contain any type of data, allowing you to deal with array data without being aware of the number of elements. For example, a formatted data array that includes 5 measurement points is stored as shown in the following figure. Note that a formatted data array always contains 2 data items per measurement point, whichever data format is used. For more information on contained data, see Internal Data Processing. you can find a table that describes the relationship between contained data items and data formats.

Example storing data into a Variant variable



e5071c436

NOTE When you use one of the objects listed above, the base index number of the array is always 0 even if the declaration section contains the "Option Base 1" statement, which specifies the use of the base array index of 1.

For example, you may wish to read the formatted data array for a particular trace in its entirety (including all measurement points), display the data in the echo window, and then write the data into another trace. How to implement such a process can be better understood with the aid of a sample program.

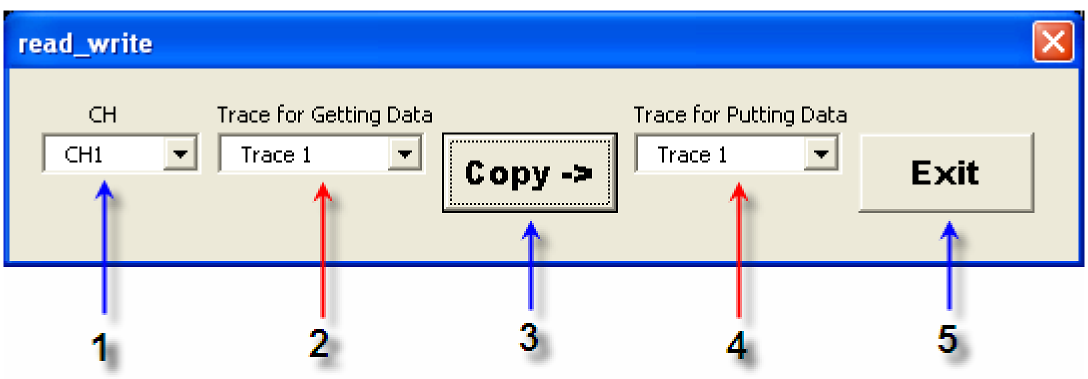
Sample program is available for download from the Agilent Support page, named "**read_write.vba**", that demonstrates how to read and write measurement data. This VBA program consists of the following modules:

Object name	Module type	Content
frmReadWrite	UserForm	Reads, displays, and writes a formatted data array.
mdlReadWrite	Standard module	Invokes a UserForm.

NOTE This sample program runs correctly when the maximum number of channels/traces is set to 4 channels/4 traces.

When you run this VBA program, a following window appears.

UserForm of read_write.vba program



The program lets the user specify the channel to be controlled.

1. The program lets the user specify which trace's formatted data array to read (source trace).
2. The program reads the formatted data array for the trace specified by the user, display the measurement results in the echo window, and write the data into the trace specified by the user. For detail, see the description of the code window.
3. The program lets the user specify which trace's formatted data array to overwrite (target trace).
4. The program exits, and the window disappears.

In Visual Basic Editor, open the UserForm (object name: frmReadWrite), and double-click the entire UserForm or the **Copy ->** or **Exit** button to bring up the code window. The following is the description of the subprograms associated with the respective buttons.

Sample Program

Reading/displaying/writing a formatted data array (read_write.frm)

```
''
' Procedure called when the user clicks the Copy button on the UserForm.
''
'
Private Sub cmdCopy_Click()
   Dim X As Integer, Y As Integer, Z As Integer, I As Integer
   Dim ActCh As Long, TrGet As Long, TrPut As Long
   Dim TrCont As Long, Nop As Long
   Dim FmtData As Variant, Freq As Variant
   Dim Fmt As String
'
' These lines identify the selected items in each list and store them into
' the variables TrGet, TrPut, and ActCh.
'
   X = cboCh.ListIndex
   ActCh = X + 1
   Y = cboGet.ListIndex
   TrGet = Y + 1
   Z = cboPut.ListIndex
   TrPut = Z + 1
'
' If the specified target trace is not displayed, these lines display that trace.
'
   TrCont = SCPI.CALCulate(ActCh).PARameter.Count
   If TrCont < TrPut Then
     SCPI.CALCulate(ActCh).PARameter.Count = TrPut
   End If
'
' These lines make active the specified trace (TrGet: source trace) in
```

```
' the specified channel(ActCh) and hold the sweep.
'
   SCPI.CALCulate(ActCh).PARameter(TrGet).SELect
   SCPI.INITiate(ActCh).CONTinuous = False
   SCPI.ABORt
'
' Reads the number of measurement points for the specified channel (ActCh)
' and stores that number into the Nop variable.
'
   Nop = SCPI.SENSe(ActCh).SWEep.POINts
'
' Reads the formatted data array for the active trace (source trace)
' and store the data into the FmtData variable.
'
   FmtData = SCPI.CALCulate(ActCh).SELected.Data.FDATa
'
' Reads the stimulus array for the specified channel (ActCh)
' and stores the data into the Freq variable.
'
   Freq = SCPI.SENSe(ActCh).FREQuency.Data
'
' Reads the data format for the active trace (source trace) and
' store it into the Fmt variable.
'
   Fmt = SCPI.CALCulate(ActCh).SELected.Format
'
' These lines display the echo window in the lower part of
' the LCD screen.
'
   SCPI.DISPlay.TABLe.TYPE = "ECHO"
   SCPI.DISPlay.TABLe.STATe = True
'
' The lines display, in the echo window, each point along with
' one measured value (the odd part of the index is always 0) and
' a frequency if the Fmt is "MLOG", "PHAS", "GDEL", "MLIN", "SWR", "REAL",
' "IMAG", or "UPH"; or along with two measured values and a frequency
' if Fmt returns any other string.
```

```
'
    Select Case Fmt
    Case "MLOG", "PHAS", "GDEL", "MLIN", "SWR", "REAL", "IMAG", "UPH"
      ECHO "Nop", "Frequency(GHz)", "Data"
      For I = 0 To Nop - 1
        ECHO I + 1, Freq(I) / 1000000000#, FmtData(2 * I)
      Next I
    Case Else
      ECHO "Nop", "Frequency(GHz)", "Data1", "Data2"
      For I = 0 To Nop - 1
        ECHO I + 1, Freq(I) / 1000000000#, FmtData(2 * I), FmtData(2 * I + 1)
      Next I
    End Select
'
' Makes active the specified trace (TrPut: target trace) in the specified
' channel(ActCh).
'
    SCPI.CALCulate(ActCh).PARameter(TrPut).SELect
'
' Writes the formatted data array (FmtData) into the active trace (target trace).
'
    SCPI.CALCulate(ActCh).SELected.Data.FDATa = FmtData
'
End Sub
'
'
''''
' Procedure called when the user clicks the Exit button on the UserForm.
''''
Private Sub cmdExit_Click()
'
' Unloads the UserForm from the memory, and terminates the program.
'
    Unload Me
'
End Sub
'
```

```
'
''''
' Procedure that initializes the UserForm
''''
Private Sub UserForm_Initialize()
'
' When the program is launched, these lines add each list item and set the default value
' for each list.
'
    With cboCh
      .AddItem "CH1"
      .AddItem "CH2"
      .AddItem "CH3"
      .AddItem "CH4"
    End With
'
    With cboGet
      .AddItem "Trace 1"
      .AddItem "Trace 2"
      .AddItem "Trace 3"
      .AddItem "Trace 4"
    End With
'
    With cboPut
      .AddItem "Trace 1"
      .AddItem "Trace 2"
      .AddItem "Trace 3"
      .AddItem "Trace 4"
    End With
'
    cboCh.ListIndex = 0
    cboGet.ListIndex = 0
    cboPut.ListIndex = 0
'
End Sub
```

Executing a Procedure with a Softkey (User Menu Function)

- Overview
- Preparing for User Menu Function
- Using User Menu Function
- Sample Program

Other topics about Controlling E5061B

Overview

The E5061B lets you perform procedures assigned to specific softkeys (**Macro Setup** > **User Menu** > **Button 1/2/3/4/5/6/7/8/9/10**), without using user forms, when that softkey is pressed. This function is called the user menu function.

NOTE You do not have to execute any VBA program when using the user menu function.

Preparing for User Menu Function

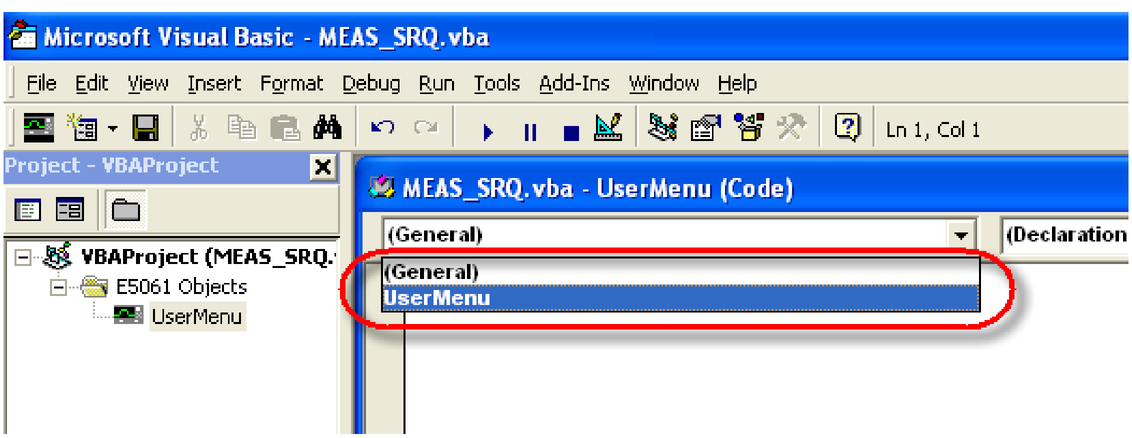
Before using the user menu function, perform the following preparation.

Coding of a Procedure Assigned to a Softkey

Follow these steps to create a procedure assigned to a specific softkey in the "UserMenu" object in the "E5061 Objects" folder.

1. Double-click 

to open the code window.
2. In the object box in the code window, select **UserMenu** as shown below:



3. In the UserMenu_OnPress(ByVal Key_id As Long) procedure, create a program you want to assign to a specific softkey (specify with the *id* variable). For actual use example, see Line 70 to 430 in the Sample program using user menu UserMenu object.

NOTE During processing an event (during execution of a procedure for a key pressed), another event (an interrupt by a procedure for another softkey pressed) cannot be accepted.

NOTE You cannot save (export) the "UserMenu" object by module basis; save it by project basis.

Settings for Softkey Label and Softkey Enabled/Disabled

When you want to change the softkey labels for the user menu function, use the following COM object.

- UserMenu.Item(Key_id).Caption

When you want to set the softkey enabled/disabled for the user menu function, use the following COM object.

- UserMenu.Item(Key_id).Enabled

Moreover, when you want to preset the above settings for the user menu function, use the following COM object.

- UserMenu.PRESet

  NOTE The above user menu setting is also preset by pressing **Macro Setup** > **Preset User Menu** on the E5061B front panel.

Using User Menu Function

To execute a procedure assigned to a softkey, you need to generate an event of pressing the softkey. To generate an event, the manual method and the COM object method are available.

Method by Manual Operation

Click the specific softkey as follows:

Macro Setup > **User Menu** > **Button <No>.**

"**No.**" represents a button number. You can set the label for "**Button No.**" as you like. For detail, refer to the "Settings for Softkey Label and Softkey Enabled/Disabled." section.

Method by COM Object

You can use the following COM object to perform the same operation as pressing a specific softkey.

- UserMenu.Press(Key_id)

Sample Program

Sample program is available to download from the Agilent Support page, named **meas_user.vba**, that demonstrates how to use the user menu function. This VBA program consists of the following standard module and the "UserMenu" object.

Object name	Module type	Content
mdlUserMenu	Standard module	Sets the softkey labels and enables interrupts from the softkeys.

The program (object name: mdlUserMenu) is described in detail in below code:

Sample program using user menu (object name: mdlUserMenu)

```
''
' This VBA program consists of the standard module and the "UserMenu" object.
''
'
' A common variable is declared.
'
   Public State As Boolean
'
''
' The program (object name: mdlUserMenu) is set the softkey labels
' and enables interrupts from the softkeys.
''
'
Sub Main()
   Dim I As Long, J As Long
'
' Stores True into the State variable.
'
   State = True
'
' Sets the first to third softkey (id: 1 to 3) enabled, and sets
' the fourth to tenth softkey (id: 4 to 10) disabled.
'
   For I = 1 To 3
     UserMenu.Item(I).Enabled = True
   Next I
```

```
'
   For J = 4 To 10
     UserMenu.Item(J).Enabled = False
   Next J
'
' Sets the first softkey label (id: 1) to "Setup", the second softkey
' label (id: 2) to "Meas", the third softkey label (id: 3) to "Exit".
'
   UserMenu.Item(1).Caption = "Setup"
   UserMenu.Item(2).Caption = "Meas"
   UserMenu.Item(3).Caption = "Exit"
'
' Displays the buttons for the user menu function in the softkey area.
'
   UserMenu.Show
'
' Processing repeated until the State variable is True (State = True).
' Detects an event that a specific softkey is pressed and
' enables the interrupt from the event.
'
   Do While State
     DoEvents
   Loop
'
End Sub
```

UserMenu object

The procedures of the "UserMenu" object are described in detail in below code.

Sample program using user menu ("UserMenu" object)

```
''''
' Procedure called when the specific softkey is pressed.
''''
'
Private Sub UserMenu_OnPress(ByVal id As Long)
   Dim I As Integer
   Dim Nop As Long, Dmy As Long
```

```
    Dim FmtData As Variant
'
    If id = 1 Then
'
' The procedure when the first softkey (id: 1) is pressed.
'
' Returns the E5061B to the preset state.
'
      SCPI.SYSTem.PRESet
'
' For channel 1, sets the sweep start value to 1.73 GHz,
' the sweep stop value to 1.83 GHz, and the number of
' measurement points to 51.
'
      SCPI.SENSe(1).FREQuency.STARt = 1730000000#
      SCPI.SENSe(1).FREQuency.STOP = 1830000000#
      SCPI.SENSe(1).SWEep.POINts = 51
'
' After aborting the measurement, sets the trigger source
' to the bus trigger and turns on the continuous trigger startup mode
' for channel 1.
'
      SCPI.ABORt
      SCPI.TRIGger.SEQuence.Source = "BUS"
      SCPI.INITiate(1).CONTinuous = True
'
' Displays the buttons for the user menu function in the softkey area.
'
      UserMenu.Show
'
    ElseIf id = 2 Then
'
' The procedure when the second softkey (id: 2) is pressed.
'
' Generates a trigger to start a single sweep and waits until the measurement
' finishes (1 is read out with the SCPI.IEEE4882.OPC object).
'
```

```
    SCPI.TRIGger.SEQuence.SINGle
    Dmy = SCPI.IEEE4882.OPC
'
' Retrieves the number of points in channel 1 and stores that number
' into the Nop variable.
'
    Nop = SCPI.SENSe(1).SWEep.POINts
'
' Specifies trace 1 of channel 1 to the active trace, retrieves the formatted
' data array, and stores the data into the FmtData variable.
'
    SCPI.CALCulate(1).PARameter(1).SELect
    FmtData = SCPI.CALCulate(1).SELected.DATA.FDATa
'
' Displays the echo window in the lower part of the LCD screen.
'
    SCPI.DISPlay.TABLe.TYPE = "ECHO"
    SCPI.DISPlay.TABLe.State = True
'
' Displays 2 measurement data values (primary value and secondary value)
' for each measurement point in the echo window.
'
    For I = 1 To Nop - 1
      ECHO FmtData(2 * I - 2), FmtData(2 * I - 1)
    Next I
'
  ElseIf id = 3 Then
'
' The procedure when the third softkey (id: 3) is pressed.
'
' Displays a program closing message, and stores False into the state
' variable to terminate the main program.
'
    MsgBox "Program ended"'
    State = False
'
  End If
```

```
'
End Sub
```

User Defined Variable

The E5061B is having an area in which a User can set any value. These area are divided by the different data format of the values. A maximum of ten (1 to 10) such areas can be used by each command.

For example, after setting the value (data) obtained using VBA of the E5061B to the User defined variable, this value is available to an external controller (program) through the use of these User defined variables.

NOTE Turning E5061B power ON/OFF initializes the User defined variables. They are not initialized through executing Preset.

These commands does not refers to or change the results of the E5061B.

- SCPI.PROGram.VARiable.ARRay(Vnum).DATA
- SCPI.PROGram.VARiable.ARRay(Vnum).SIZE
- SCPI.PROGram.VARiable.DOUBle(Vnum).DATA
- SCPI.PROGram.VARiable.LONG(Vnum).DATA
- SCPI.PROGram.VARiable.STRing(Vnum).DATA

Controlling Peripherals

Controlling Peripherals

- Overview
- Programming with VISA

Overview

- Overview
- Preparation

Other topics about Controlling Peripherals

Overview

The E5061B macro function (E5061B VBA) can be used not only to automate measurements but also to control external measurement instruments connected via USB/GPIB interface by acting as a self-contained system controller (see An Overview of a Control System Based on the Macro Function).

The E5061B macro function (E5061B VBA) performs communications via the COM interface when controlling the E5061B itself, and it communicates via VISA (Virtual Instrument Software Architecture) when controlling external measurement instruments.

Preparation

Importing Definition Files

To use the VISA library in the E5061B macro (E5061B VBA), you need to import two definition files into your project with the Visual Basic editor to define the VISA functions and perform other tasks. The definition files are stored on the sample programs disk under the following filenames (for information on importing modules, refer to Saving a Module (Exporting).

- **visa32.bas**
- **vpptype.bas**

Programming with VISA

- Overview
- Starting VISA
- Connection
- Communication
- Disconnection
- Sample Program

Other topics about Controlling Peripherals

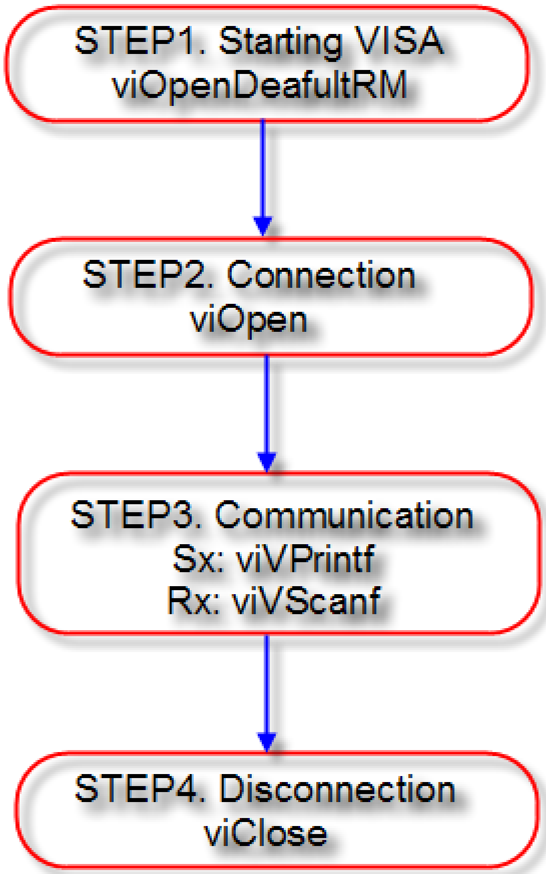
Overview

The following figure shows the flow of controlling the instrument with VISA. When developing a VISA program in the Visual Basic language, a special programing notice (in the readme text file listed below) must be reviewed.

For details on the use of the VISA library and the programing notice for using the VISA library with the E5061B macro (E5061B VBA), refer to the following files contained in I/O library CD-ROM.

- **visa.hlp** (on-line help for the VISA library)
- **vbreadme.txt** (notes on using the VISA library with VB)

Flow of instrument control with VISA



e5071c183

STEP 1. Starting VISA

The VISA system startup session is viOpenDefaultRM function in the sample program **ctrl_ext.vba**. VISA's viOpenDefaultRM function initializes and starts up the VISA system. The viOpenDefaultRM function must be executed before other VISA functions are called, and the parameter of this function has the startup information (Defrm in **ctrl_ext.vba**).

Syntax

viOpenDefaultRM(*param*)

Parameter

Parameter	(*param*)
Description	Startup information (output)
Data type	Long integer type

STEP 2. Connection

The connection session is viOpen function. VISA's viOpen function makes connection with the specified instrument. The viOpen function returns a value so that the VISA functions can apply it to the specified instrument. The parameters of this function contain the startup information (Defrm), the address information of the specified instrument ("GPIB0::17::INSTR" in **ctrl_ext.vba**), access mode (0 in **ctrl_ext.vba**), timeout (0 in **ctrl_ext.vba**), and connection information (Equip in **ctrl_ext.vba**).

Syntax

viOpen(*param1*, *param2*, *param3*, *param4*, *param5*)

Parameters

Parameter	(*param1*)
Description	Startup information (input)
Data type	Long integer type

Parameter	(*param2*)
Description	Address information of the specified instrument (input)
Data type	Character string type
Syntax	"GPIB0::*gpib address*::INSTR" "USB0::*manufacturer ID*::*model code*::*serial number*::0::INSTR" (ex. "USB0::2391::2312::MY12345678::0::INSTR") "TCPIP0::*IP address*::inst0::INSTR"

Parameter	(*param3*)
Description	Access mode (Enter 0)

Parameter	(*param4*)
Description	Timeout (Enter 0)

Parameter	(*param5*)
Description	Connection information (output)

Data type	Long integer type

STEP 3. Communication

The communication session is viVPrintf function. VISA's viVPrintf function sends a program message (GPIB command) to the specified instrument. The parameters of this function contain the connection information (Equip), the program message (*IDN?), and the variable to be formatted (0 in **ctrl_ext.vba**).

NOTE To input/output GPIB commands, the viVPrintf function and the viVScanf function are mainly used, but other VISA functions are also available. For more information, refer to **visa.hlp** (online help for the VISA library).

Syntax

viVPrintf(*param1*, *param2*, *param3*)

Parameters

Parameter	(*param1*)
Description	Connection information (input)
Data type	Long integer type

Parameter	(*param2*)
Description	Program message (input) When sending a program message of the GPIB command, a message terminator is required at the end of the message (Chr$(10) in **ctrl_ext.vba**)
Data type	Character string type

Parameter	(*param3*)
Description	A variable to be formatted. If not applicable, enter 0.
Data type	Specified data type

The receiving session is viVScanf function. VISA's viVScanf function receives the result from the specified instrument and stores it in the output variable. The parameters of this function contain the connection information (Equip in **ctrl_ext.vba**), the format parameter for the output variable (%t in **ctrl_ext.vba**), and the output variable (Prod in **ctrl_ext.vba**).

Syntax

viVScanf(*param1*, *param2*, *param3*)

Parameters

Parameter	(*param1*)
Description	Connection information (input)
Data type	Long integer type

Parameter	(*param2*)
Description	Format parameter for the output variable
Data type	Character string type

Parameter	(*param3*)
Description	Output variable (output)
Data type	Character string type

STEP 4. Disconnection

The disconnection session is viClose function. VISA's viClose function disconnects communication and terminates the VISA system. The parameter of this function has startup information (Defrm in **ctrl_ext.vba**).

Syntax

viClose(*param*)

Parameter

Parameter	(*param*)
Description	Startup information (input)
Data type	Long integer type

Sample Program to Read Out the Product Information of Peripheral (Instrument)

The **ctrl_ext.vba** is a sample program that controls instruments connected through USB/GPIB interface cable using the E5061B as the system controller. This VBA program consists of the following modules.

Object name	Module type	Content
mdlCtrlExt	Standard module	Reads out the product information of external instrument.
Visa32	Standard module	Definition file to use VISA library (Visa32.bas)

NOTE When you control peripherals from E5061B VBA, use the GPIB commands provided for the instrument to communicate with VISA. On the other hand, when you control the E5061B itself with E5061B VBA, use the COM objects provided for the E5061B to communicate.

Read out the product information (ctrl_ext.vba)

```
''''
' This program is sample of controlling peripherals.
''''
'
Sub Main()
   Dim status As Long
   Dim Defrm As Long
   Dim Equip As Long
   Dim Prod As String * 100
'
' Initializes and starts up the VISA system and outputs the startup information to the Defrm variable.
' During this process, if an error occurs, the program goes to the error handling routine.
'
   status = viOpenDefaultRM(Defrm)
   If (status <> VI_SUCCESS) Then GoTo VisaErrorHandler
'
' Establishes the connection to the external instrument (GPIB address: 17) connected via GPIB and
' outputs the connection information to the Equip variable. During this process, if an error occurs,
' the program goes to the error handling routine.
'
   status = viOpen(Defrm, "GPIB0::17::INSTR", 0, 0, Equip)
   If (status <> VI_SUCCESS) Then GoTo VisaErrorHandler
'
```

```
' Queries the product information of the external instrument connected via USB/GPIB interface cable
' using VISA. During this process, if an error occurs, the program goes to the error handling routine.
'
   status = viVPrintf(Equip, "*IDN?" & Chr$(10), 0)
   If (status <> VI_SUCCESS) Then GoTo VisaErrorHandler
'
' Retrieves the product information through VISA and outputs it into the Prod variable. Displays the
' read-out result in the message box. During this process, if an error occurs, the program goes to
' the error handling routine.
'
   status = viVScanf(Equip, "%t", Prod)
   If (status <> VI_SUCCESS) Then GoTo VisaErrorHandler
   MsgBox Prod
'
' Breaks the communication and terminates the VISA system.
'
   Call viClose(Defrm)
'
   GoTo Prog_end
'
' If an error occurs in a VISA function, displays the detail of the error and terminates the program.
'
VisaErrorHandler:
   Dim VisaErr As String * 200
   Call viStatusDesc(Defrm, status, VisaErr)
   MsgBox "Error : " & VisaErr, vbExclamation
   Exit Sub
'
Prog_end:
'
End Sub
```

Application Programs

Application Programs

- Basic Measurement (measuring a band-pass filter)
- Connecting Hard Disk of External PC (shared folder)

Basic Measurement (measuring a band-pass filter)

Other topics about Application Programs

Overview

The **apl_bsc.vba** shows a sample program (VBA program) that demonstrates how to perform the basic measurement of the band-pass filter. This VBA program consists of the following standard module.

Object name	Module type	Content
mdlBscMeas	Standard module	Performs basic measurement of band-pass filter

Overview of the Program

The sample program performs full 2-port calibration using the 85032F calibration kit, measures a band-pass filter (center frequency: 947.5 MHz), and calculates and displays its bandwidth, insertion loss, and so on. This measurement is the same as Measurement Example of a Bandpass Filter in the Quick Start.

Description of the Program

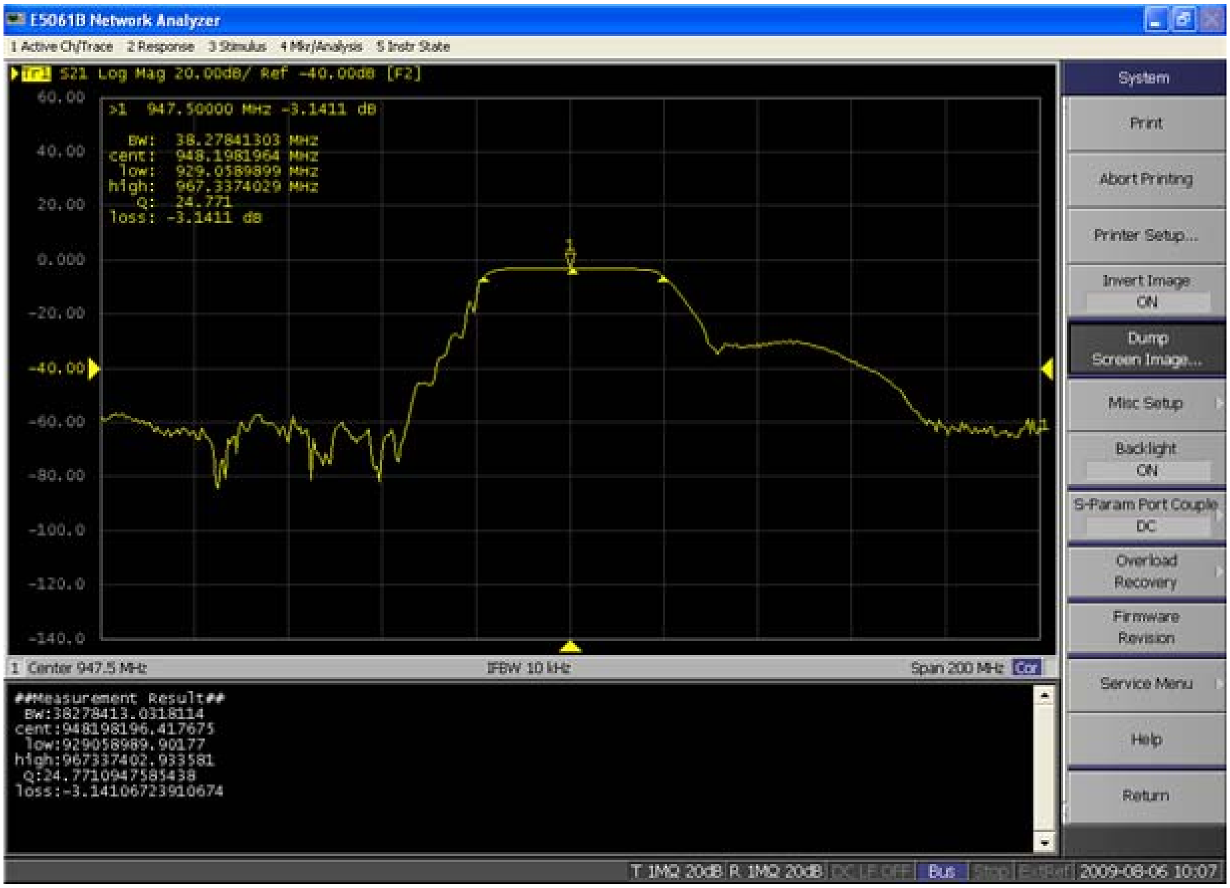
When you run this VBA program, reset is performed, the measurement conditions are automatically set, and the message "Perform the full 2-port calibration" is displayed. To perform the full 2-port calibration, click **Yes**; otherwise click **No**.

To perform the calibration, follow the onscreen messages to connect each standard of the Agilent 85032F calibration kit to the specified port, and then click **OK** to measure the calibration data. Click **Cancel** to return to the beginning of the calibration. You cannot skip the isolation calibration. When the calibration data measurement for all standards is complete, the message "All calibration data completion" is displayed, and the calibration coefficient is calculated.

NOTE When you cancel the calibration data measurement before completing the measurement of necessary calibration data, the settings condition may not be returned to its former state.

Then, the message "Connect DUT, and then press **Macro Setup** > **Continue**" is displayed in the instrument status bar in the lower part of the LCD display. Connect a DUT and perform **Macro Setup** > **Continue**. After the measurement, the search result is displayed in the echo window, as shown below. If no bandwidth search target is found, only the result of the insertion loss obtained with the marker is displayed.

Example of display after executing the program in "apl_bsc.vba"



The basic measurement program (object name: mdlBscMeas) is described in detail below. Line numbers are added for description purpose only and do not appear in the actual program source code.

Measuring a band-pass filter (object name: mdlBscMeas)

```
''''
' This VBA program is sample program to perform the basic measurement of the band-pass filter.
''''
'
Sub Main()
   Dim Par As String, Fmt As String, File As String
   Dim Center As Double, Span As Double, IfBw As Double, Pow As Double
   Dim Bw As Double, Cent As Double
   Dim CutLow As Double, CutHigh As Double
```

```
    Dim Qfac As Double, Loss As Double
    Dim MkrVal As Variant, BwData As Variant
    Dim Nop As Long, NumTrac As Long, CalKit As Long, Buff As Long
    Dim Port As Variant, Error As Variant
'
' Store the sweep center value (947.5 MHz), the sweep span value (200 MHz), the number of measurement points (401),
' the IF bandwidth (10 kHz), and the power level (-10 dBm) into the variables Center, Span, Nop, IfBw, and Pow, respectively.
'
    Center = 947500000#
    Span = 200000000#
    Nop = 401
    IfBw = 10000#
    Pow = -10
'
' Store the number of traces (1), the measurement parameter (S21), the data format (log amplitude), the calibration kit
' number (4: 85032F), and the save file name (State08.sta) into the variables, NumTrac, Par, Fmt, CalKit, and File, respectively.
'
    NumTrac = 1
    Par = "S21"
    Fmt = "MLOG"
    CalKit = 4
    File = "State08.sta"
'
' Returns the E5061B to the preset state.
'
    SCPI.SYSTem.PRESet
'
' For channel 1, turn on the continuous trigger startup mode to On and set the trigger source to the bus trigger.
'
    SCPI.INITiate(1).CONTinuous = True
    SCPI.TRIGger.SEQuence.Source = "BUS"
'
' For channel 1, set the sweep center value to the Center variable, the sweep span value to the Span variable,
```

```
' the number of measurement points to the Nop variable, the IF bandwidth to the IfBw variable, and the power level
' to the Pow variable.
'
    SCPI.SENSe(1).FREQuency.Center = Center
    SCPI.SENSe(1).FREQuency.Span = Span
    SCPI.SENSe(1).SWEep.POINts = Nop
    SCPI.SENSe(1).BANDwidth.RESolution = IfBw
    SCPI.Source(1).POWer.LEVel.IMMediate.AMPLitude = Pow
'
' For channel 1, set the number of traces to the NumTrac variable, the measurement parameter to the Par variable,
' and the data format to the Fmt variable.
'
    SCPI.CALCulate(1).PARameter.Count = NumTrac
    SCPI.CALCulate(1).PARameter(1).DEFine = Par
    SCPI.CALCulate(1).PARameter(1).Select
    SCPI.CALCulate(1).SELected.Format = Fmt
'
' Stores the calibration kit number for channel 1 into the CalKit variable, and stores 1 and 2 into the Port variable
' that indicates ports used for the full 2-port calibration. Then, calls the Calib_Solt procedure.
'
    SCPI.SENSe(1).CORRection.COLLect.CKIT.Select = CalKit
    Port = Array(1, 2)
    Calib_Solt 1, 2, Port
'
' Save the instrument setting and the calibration coefficient into a file whose name is specified with the File variable.
'
    SCPI.MMEMory.STORe.STYPe = "CST"
    SCPI.MMEMory.STORe.STATe = File
'
' Displays a message that prompts you to connect a DUT (Device Under Test) in the instrument status bar in the lower part
' of the LCD display and waits for the operation of [Macro Setup > Continue] after the connection.
'
Meas_Start:
```

```
    Prompt ("Connect DUT, and then press [Macro Setup]-Continue button.")
'
' Generate a trigger to start a single sweep and wait until the measurement finishes (1 is read out with
' the SCPI.IEEE4882.OPC object).
'
    SCPI.TRIGger.SEQuence.SINGle
    Dmy = SCPI.IEEE4882.OPC
'
' For trace 1 of channel 1, executes auto scale to set the optimum scale.
'
    SCPI.DISPlay.WINDow(1).TRACe(1).Y.SCALe.AUTO
'
' Display marker 1 and move it so that the stimulus value becomes equal to the value of the Center
variable.
' Then, these lines read out the response value of marker 1 and store it into the MkrVal variable.
'
    SCPI.CALCulate(1).SELected.MARKer(1).STATe = True
    SCPI.CALCulate(1).SELected.MARKer(1).X = Center
    MkrVal = SCPI.CALCulate(1).SELected.MARKer(1).Y
'
' Enables the error handling routine starting from Bw_Err. If a runtime error occurs, the program goes
' to the error handling routine.
'
    On Error GoTo Bw_Err
'
' Set the bandwidth definition value to -3 dB and the bandwidth search result display to on, read out
' the bandwidth search result (bandwidth, center frequency, Q value, and insertion loss), and
' store it into the BwData variable.
'
    SCPI.CALCulate(1).SELected.MARKer(1).BWIDth.THReshold = -3
    SCPI.CALCulate(1).SELected.MARKer(1).BWIDth.STATe = True
    BwData = SCPI.CALCulate(1).SELected.MARKer(1).BWIDth.DATA
'
' Based on the bandwidth search result, these lines store the bandwidth to the Bw variable, the center
frequency
' to the Cent variable, the Q value to the Qfac variable, and the insertion loss to the Loss variable.
' Then, the program goes to the processing starting from Skip_Bw_Err.
'
```

```
   Bw = BwData(0)
   Cent = BwData(1)
   Qfac = BwData(2)
   Loss = BwData(3)
   GoTo Skip_Bw_Err
'
' Define a runtime error handler. These lines read out and display the error number and error message of the
' error that occurred and store 0 to the Bw, Cent, and Qfac variables and the response value of marker 1
' (MkrVal(0) variable) to the Loss variable. Then, the program finishes the error handling and
' proceeds to the next processing.
'
Bw_Err:
   Error = SCPI.SYSTem.Error
   MsgBox "Error No:" & Error(0) & " , Description:" & Error( 1)
   Bw = 0
   Cent = 0
   Qfac = 0
   Loss = MkrVal(0)
   Resume Skip_Bw_Err
'
' Calculate the 2 (higher and lower) cutoff frequencies from the values in the Bw and Cent variables
' and store them into the CutLow and CutHigh variables.
'
Skip_Bw_Err:
   CutLow = Cent - Bw / 2
   CutHigh = Cent + Bw / 2
'
' Display the search result (the values of the Bw, Cent, CutLow, CutHigh, Qfac, and Loss variables)
' in the echo window.
'
   ECHO "##Measurement Result##"
   ECHO " BW:" & Bw
   ECHO "cent:" & Cent
   ECHO " low:" & CutLow
   ECHO "high:" & CutHigh
   ECHO " Q:" & Qfac
```

```
    ECHO "loss:" & Loss
    SCPI.DISPlay.TABLe.TYPE = "ECHO"
    SCPI.DISPlay.TABLe.STATe = True
'
' Display the message asking whether you want to perform measurement again. Click Yes to return to
' the DUT connection section, otherwise click No to terminate the program.
'
    Buff = MsgBox("Do you make another measurement?", vbYesNo, "Bandpass fileter measurement")
    If Buff = vbYes Then
      GoTo Meas_Start
    End If
'
End Sub
'
'
''''
' The following code is Calib_Solt procedure.
''''
'
Private Sub Calib_Solt(Chan As Long, SoltType As Long, Port As Variant)
    Dim Dmy As Long, I As Long, J As Long, Buff As Long
'
' Display the message that prompts for the execution of the full n-port calibration (specified with the
SoltType variable).
' Click Cancel to cancel the calibration.
'
Cal_Start:
    Buff = MsgBox("Perform the full " & SoltType & "-port cali bration.", vbOKCancel, "Full" & SoltType & "-
port calibration")
    If Buff = vbCancel Then
      GoTo Cal_Skip
    End If
'
' Set the calibration type to the full n-port calibration for the port specified with the Port variable.
'
    Select Case SoltType
    Case 1
      SCPI.SENSe(Chan).CORRection.COLLect.METHod.SOLT1 = Port(0)
```

```
  Case 2
   SCPI.SENSe(Chan).CORRection.COLLect.METHod.SOLT2 = Port
  End Select
'
' Processing loops according to the selected calibration type.
'
  For I = 1 To SoltType
'
' Display the message that prompts for connecting the open standard to the specified port. These lines
' start the measurement of the open calibration data initiated by clicking OK after the connection and
' wait for the completion of the measurement. Click Cancel to return to the beginning of the calibration.
'
   Buff = MsgBox("Connect the Open standard to Port " & CS tr(Port(I - 1)) & ".", vbOKCancel, "Full" &
SoltType & "-port calibration")
   If Buff = vbOK Then
    SCPI.SENSe(Chan).CORRection.COLLect.ACQuire.OPEN = Port(I - 1)
    Dmy = SCPI.IEEE4882.OPC
   Else
    GoTo Cal_Start
   End If
'
' Display the message that prompts for connecting the short standard to the specified port. These lines
' start the measurement of the short calibration data initiated by clicking OK after the connection and
' wait for the completion of the measurement. Click Cancel to return to the beginning of the calibration.
'
   Buff = MsgBox("Connect the Short standard to Port " & CStr(Port(I - 1)) & ".", vbOKCancel, "Full" &
SoltType & "-port calibration")
   If Buff = vbOK Then
    SCPI.SENSe(Chan).CORRection.COLLect.ACQuire.Short = Port(I - 1)
    Dmy = SCPI.IEEE4882.OPC
   Else
    GoTo Cal_Start
   End If
'
' Display the message that prompts for connecting the load standard to the specified port. These lines
' start the measurement of the load calibration data initiated by clicking OK after the connection and
' wait for the completion of the measurement. Click Cancel to return to the beginning of the calibration.
'
```

```
    Buff = MsgBox("Connect the Load standard to Port " & CStr(Port(I - 1)) & ".", vbOKCancel, "Full" &
SoltType & "-port calibration")
    If Buff = vbOK Then
      SCPI.SENSe(Chan).CORRection.COLLect.ACQuire.Load = Port(I - 1)
      Dmy = SCPI.IEEE4882.OPC
    Else
      GoTo Cal_Start
    End If
'
  Next I
'
' Display the message that prompts for connecting the thru standard between the specified ports.
' These lines start the measurement of the thru calibration data initiated by clicking OK after the
' connection and wait for the completion of the measurement. Click Cancel to return to the
' beginning of the calibration.
'
  For I = 1 To SoltType - 1
    For J = I + 1 To SoltType
      Buff = MsgBox("Connect the Thru standard between Por t " & CStr(Port(I - 1)) & _
      " and Port " & CStr(Port(J - 1)) & ".", vbOKCancel, "Full" & SoltType & "-port calibration")
      If Buff = vbOK Then
        SCPI.SENSe(Chan).CORRection.COLLect.ACQuire.THRU = Array(Port(I - 1 ), Port(J - 1))
        Dmy = SCPI.IEEE4882.OPC
        SCPI.SENSe(Chan).CORRection.COLLect.ACQuire.THRU = Array(Port(J - 1 ), Port(I - 1))
        Dmy = SCPI.IEEE4882.OPC
      Else
        GoTo Cal_Start
      End If
    Next J
  Next I
'
' When the calibration type is not the 1-port calibration (a value other than 1 is specified for the
' SoltType variable), displays the message asking you whether you want to measure the isolation
' calibration data. When Yes is clicked, displays the message that prompts for connecting
' the load standard to the specified two ports (specified with the Port(I-1) and Port(J-1) variables).
' These lines start the measurement of the isolation calibration data initiated by clicking OK after the
' connection and wait for the completion of the measurement. Click Cancel to return to the
```

```
' beginning of the calibration.
'
    If SoltType <> 1 Then
      Buff = MsgBox("Do you measure the Isolation (Optional) ?", vbYesNo, "Full" & SoltType & "-port
calibration")
      If Buff = vbYes Then
        For I = 1 To SoltType - 1
          For J = I + 1 To SoltType
            Buff = MsgBox("Connect the Load standard to Port " & Port(I - 1) & " and Port " & Port(J - 1) & ".",
_
            vbOKCancel, "Full" & Solt Type & "-port calibration")
            If Buff = vbOK Then
              SCPI.SENSe(Chan).CORRection.COLLect.ACQuire.ISOLation = Array(Port(I - 1), Port(J - 1))
              Dmy = SCPI.IEEE4882.OPC
              SCPI.SENSe(Chan).CORRection.COLLect.ACQuire.ISOLation = Array(Port(J - 1), Port(I - 1))
              Dmy = SCPI.IEEE4882.OPC
            Else
              GoTo Cal_Start
            End If
          Next J
        Next I
      End If
    End If
'
' Calculate the calibration coefficients from the measured calibration data and turn on the
' error correction function. Then, these lines display a calibration completion message.
'
    SCPI.SENSe(1).CORRection.COLLect.SAVE
    MsgBox "All calibration data completion."
'
Cal_Skip:
'
End Sub
```

Connecting Hard Disk of External PC (shared folder)

- Overview
- Using VBA Program
- Description of VBA Program

Other topics about Application Programs

Overview

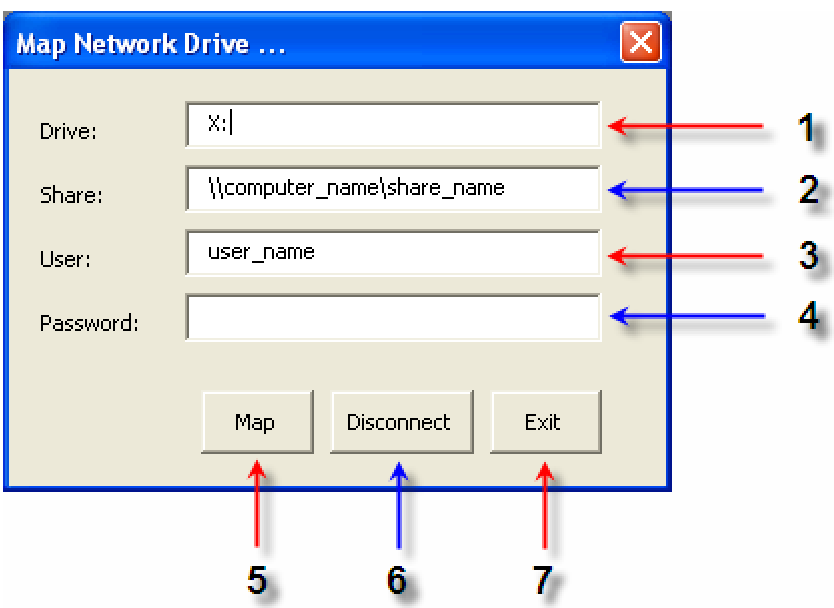
The **map_drive.vba** shows a sample program (VBA program) that demonstrates how to connect a hard disk (a shared folder) of an external PC to the E5061B. This VBA program consists of the following modules:

Object name	Module type	Description
frmMapDrive	User form	Connects or disconnects a hard disk
Module1	Standard module	Displays frmMapDrive

Using VBA Program

Load the **map_drive.vba** and press **Macro Run** key. The following macro appears.

Shared folder connection macro



Connecting (Mapping)

Enter the drive letter for the shared folder (1), share name of the shared folder (2), user name (3), password (4), and then click **Map** (5).

NOTE Consult your network administrator and enter the settings in the same way as done in the Windows PC operating system. If you enter an incorrect setting, an error might occur and the program might be interrupted.

Disconnecting

1. Enter the drive letter for the shared folder (1), and then click **Disconnect** (6).
2. Click **Exit** (7) to exit from the program.

Description of VBA Program

The program (object name: frmMapDrive) is described in detail below. The following description is included as a comment in the source code.

Sub CommandButton1_Click

This procedure is called when the user clicks the **Map** button. It checks whether the drive letter is used by using the IsDriveNameInUse procedure. Then the procedure connects the shared folder using the MapDrive procedure if the drive letter is not used or otherwise it displays a message to show the drive in use.

Sub CommandButton2_Click

This procedure is called when the user clicks the **Disconnect** button. The procedure disconnects the shared folder by using the DisconnectDrive procedure.

Function IsDriveNameInUse

This procedure checks if the txtDrive.Text (the drive letter specified by 1) is used.

Sub MapDrive

This procedure connects the shared folder as the txtDrive.Text (the drive letter specified by 1) drive by using the parameters: txtShare.Text (the share name specified by 2), txtUser.Text (the user name specified by 3), and txtPasswd.Text (the password specified by 4).

Sub DisconnectDrive

This procedure disconnects the txtDrive.Text (the drive letter specified by 1) drive.

Sub CommandButton3_Click

This procedure is called when the user clicks the **Exit** button. This procedure ends the program.

Connecting the hard disk of an external PC (Object name: frmMapDrive)

```
'
' This procedure is called when the user clicks the Map button. It checks whether
' the drive letter is used by using the IsDriveNameInUse procedure.
' Then the procedure connects the shared folder using the MapDrive procedure
' if the drive letter is not used or otherwise displays a message to show
' the drive letter is used.
'
Private Sub CommandButton1_Click()
  If Not IsDriveNameInUse Then
    Call MapDrive
  Else
    MsgBox "Drive """ & txtDrive.Text & """ is Already used", vb Critical
  End If
End Sub
'
' This procedure is called when the user clicks the Disconnect button.
' The procedure disconnects the shared folder by using the DisconnectDrive procedure.
'
Private Sub CommandButton2_Click()
  Call DisconnectDrive
End Sub
'
' This procedure checks if the txtDrive.Text is used.
'
Private Function IsDriveNameInUse() As Boolean
  Set fso = CreateObject("Scripting.FileSystemObject")
  IsDriveNameInUse = fso.DriveExists(txtDrive.Text)
End Function
'
' This procedure connects the shared folder as the txtDrive.Text drive by using the parameters:
' txtShare.Text, txtUser.Text, and txtPasswd.Text.
```

```
'
Private Sub MapDrive()
  Set network = CreateObject("wscript.network")
  Call network.MapNetworkDrive(txtDrive.Text, txtShare.Text, vbFal se, txtUser.Text, txtPasswd.Text)
End Sub
'
' This procedure disconnects the txtDrive.Text drive.
'
Private Sub DisconnectDrive()
  Set network = CreateObject("wscript.network")
  network.RemoveNetworkDrive txtDrive.Text
End Sub
'
' This procedure is called when the user clicks the Exit button. This procedure ends the program.
'
Private Sub CommandButton3_Click()
  Unload Me
End Sub
```

Complex Operation Library

Complex Operation Library

By using the complex operation library, you can perform operations of complex numbers.

Data of the complex type

In the complex operation library, you can use the complex type (Complex) as a data type. Data of the complex type consists of a real part (.real) and an imaginary part (.imag) as shown in the following example.

```
Dim Num as Complex
Num.real=1.0
Num.imag=2.0
```

List of procedures

The following table lists the procedures included in the complex operation library.

Procedure name	Function
ComplexSet(x,y)	Sets a complex number. (Specify a real part and an imaginary part.)
ComplexPolar(x,y)	Sets a complex number. (Specify an absolute value and a phase angle.)
ComplexSetArray(x)	Converts a variant type or double floating point type array to a complex type array.
ComplexAdd(x,y)	Returns the result of the addition.
ComplexSub(x,y)	Returns the result of the subtraction.
ComplexMul(x,y)	Returns the result of the multiplication.
ComplexDiv(x,y)	Returns the result of the division.
ComplexAbs(x)	Returns the absolute value.
ComplexArg(x)	Returns the phase angle.
ComplexNorm(x)	Returns the square of the absolute value.
ComplexConj(x)	Returns the conjugate complex number.
ComplexCos(x)	Returns the cosine.

ComplexCosh(x)	Returns the hyperbolic cosine.
ComplexSin(x)	Returns the sine.
ComplexSinh(x)	Returns the hyperbolic sine.
ComplexExp(x)	Returns e^x.
ComplexLog(x)	Returns the natural logarithm.
ComplexLog10(x)	Returns the common logarithm.
ComplexSqrt(x)	Returns the square root.

Sample Program

```
'
'     :
'     :
' The source code in this part is omitted.
'     :
'     :
'
  Dim Dmy As Long
  Dim s21_raw As Variant
  Dim s12_raw As Variant
  Dim s21_Comp As Complex
  Dim s12_Comp As Complex
  Dim trAce_ratio_comp As Complex
  Dim trAce_ratio(401) As Double
'
  SCPI.DISPlay.Split = "D1"
  SCPI.DISPlay.WINDow(1).Split = "D12_34"
  SCPI.CALCulate(1).PARameter.Count = 2
  SCPI.CALCulate(1).PARameter(1).DEFine = "s21"
  SCPI.CALCulate(1).PARameter(2).DEFine = "s12"
  SCPI.SENSe(1).SWEep.POINts = 201
'
  SCPI.TRIGger.SEQuence.Source = "bus"
  SCPI.TRIGger.SEQuence.SINGle
  Dmy = SCPI.IEEE4882.OPC
```

```
'
'** Get corrected data array of S21 and S12.
'
  SCPI.CALCulate(1).PARameter(1).SELect
  s21_raw = SCPI.CALCulate(1).SELected.DATA.SDATa
  SCPI.CALCulate(1).PARameter(2).SELect
  s12_raw = SCPI.CALCulate(1).SELected.DATA.SDATa
'
  For i = 0 To 200
'
'** Copy corrected data array to the complex data array
'** to take advantage of complex operation library
'
    s21_Comp = ComplexSet(s21_raw(2 * i), s21_raw(2 * i + 1))
    s12_Comp = ComplexSet(s12_raw(2 * i), s12_raw(2 * i + 1))
'
'** Calculate the ratio of S12 and S21
'** S12/S21
'
    trAce_ratio_comp = ComplexDiv(s12_Comp, s21_Comp)
    trAce_ratio(2 * i) = trAce_ratio_comp.real
    trAce_ratio(2 * i + 1) = trAce_ratio_comp.imag
'
  Next i
'
'
'
  SCPI.CALCulate(1).PARameter.Count = 4
'
'** Write "S12/S21" data to corrected data array for the trace 3 (LogMag)
'
  SCPI.CALCulate(1).PARameter(3).SELect
  SCPI.CALCulate(1).SELected.Format = "MLOG"
  SCPI.CALCulate(1).SELected.DATA.SDATa = trAce_ratio
'
'** Write "S12/S21" data to corrected data array for the trace 4 (Phase)
'
```

```
  SCPI.CALCulate(1).PARameter(4).SELect
  SCPI.CALCulate(1).SELected.Format = "PHASe"
  SCPI.CALCulate(1).SELected.DATA.SDATa = trAce_ratio
'
'      :
'      :
' The source code in this part is omitted.
'      :
'      :
'
```

Complex Operation Library

By using the complex operation library, you can perform operations of complex numbers.

Data of the complex type

In the complex operation library, you can use the complex type (Complex) as a data type. Data of the complex type consists of a real part (.real) and an imaginary part (.imag) as shown in the following example.

```
Dim Num as Complex
Num.real=1.0
Num.imag=2.0
```

List of procedures

The following table lists the procedures included in the complex operation library.

Procedure name	Function
ComplexSet(x,y)	Sets a complex number. (Specify a real part and an imaginary part.)
ComplexPolar(x,y)	Sets a complex number. (Specify an absolute value and a phase angle.)
ComplexSetArray(x)	Converts a variant type or double floating point type array to a complex type array.
ComplexAdd(x,y)	Returns the result of the addition.
ComplexSub(x,y)	Returns the result of the subtraction.
ComplexMul(x,y)	Returns the result of the multiplication.
ComplexDiv(x,y)	Returns the result of the division.
ComplexAbs(x)	Returns the absolute value.
ComplexArg(x)	Returns the phase angle.
ComplexNorm(x)	Returns the square of the absolute value.
ComplexConj(x)	Returns the conjugate complex number.
ComplexCos(x)	Returns the cosine.
ComplexCosh(x)	Returns the hyperbolic cosine.

ComplexSin(x)	Returns the sine.
ComplexSinh(x)	Returns the hyperbolic sine.
ComplexExp(x)	Returns e^x.
ComplexLog(x)	Returns the natural logarithm.
ComplexLog10(x)	Returns the common logarithm.
ComplexSqrt(x)	Returns the square root.

Sample Program

```
'
'    :
'    :
' The source code in this part is omitted.
'    :
'    :
'
  Dim Dmy As Long
  Dim s21_raw As Variant
  Dim s12_raw As Variant
  Dim s21_Comp As Complex
  Dim s12_Comp As Complex
  Dim trAce_ratio_comp As Complex
  Dim trAce_ratio(401) As Double
'
  SCPI.DISPlay.Split = "D1"
  SCPI.DISPlay.WINDow(1).Split = "D12_34"
  SCPI.CALCulate(1).PARameter.Count = 2
  SCPI.CALCulate(1).PARameter(1).DEFine = "s21"
  SCPI.CALCulate(1).PARameter(2).DEFine = "s12"
  SCPI.SENSe(1).SWEep.POINts = 201
'
  SCPI.TRIGger.SEQuence.Source = "bus"
  SCPI.TRIGger.SEQuence.SINGle
  Dmy = SCPI.IEEE4882.OPC
'
'** Get corrected data array of S21 and S12.
```

```
'
  SCPI.CALCulate(1).PARameter(1).SELect
  s21_raw = SCPI.CALCulate(1).SELected.DATA.SDATa
  SCPI.CALCulate(1).PARameter(2).SELect
  s12_raw = SCPI.CALCulate(1).SELected.DATA.SDATa
'
  For i = 0 To 200
'
'** Copy corrected data array to the complex data array
'** to take advantage of complex operation library
'
    s21_Comp = ComplexSet(s21_raw(2 * i), s21_raw(2 * i + 1))
    s12_Comp = ComplexSet(s12_raw(2 * i), s12_raw(2 * i + 1))
'
'** Calculate the ratio of S12 and S21
'** S12/S21
'
    trAce_ratio_comp = ComplexDiv(s12_Comp, s21_Comp)
    trAce_ratio(2 * i) = trAce_ratio_comp.real
    trAce_ratio(2 * i + 1) = trAce_ratio_comp.imag
'
  Next i
'
'
'
  SCPI.CALCulate(1).PARameter.Count = 4
'
'** Write "S12/S21" data to corrected data array for the trace 3 (LogMag)
'
  SCPI.CALCulate(1).PARameter(3).SELect
  SCPI.CALCulate(1).SELected.Format = "MLOG"
  SCPI.CALCulate(1).SELected.DATA.SDATa = trAce_ratio
'
'** Write "S12/S21" data to corrected data array for the trace 4 (Phase)
'
  SCPI.CALCulate(1).PARameter(4).SELect
  SCPI.CALCulate(1).SELected.Format = "PHASe"
```

```
  SCPI.CALCulate(1).SELected.DATA.SDATa = trAce_ratio
'
'      :
'      :
' The source code in this part is omitted.
'      :
'      :
'
```

Procedure Reference

ComplexAbs(*x*)

Syntax

Result = ComplexAbs(*x*)

Description

Returns the absolute value of complex number *x*.

Data type

x : Complex type (Complex)

Result : Double precision floating point type (Double)

Example of use

```
Dim a As Complex, b As Double
a = ComplexSet(1.5, 2.0)
b = ComplexAbs(a)
```

ComplexAdd(*x*,*y*)

Syntax

Result = ComplexAdd(*x*,*y*)

Description

Returns the result (x+y) of the addition to complex number *x* and another *y*.

Data type

x : Complex type (Complex)

y : Complex type (Complex)

Result : Complex type (Complex)

Example of use

```
Dim a As Complex, b As Complex, c As Complex
a = ComplexSet(1.5, 2.0)
b = ComplexSet(0.5, 3.5)
c = ComplexAdd(a, b)
```

ComplexArg(*x*)

Syntax

Result = ComplexArg(*x*)

Description

Returns the phase angle (radian) of complex number *x*.

Data type

x : Complex type (Complex)

Result : Double precision floating point type (Double)

Example of use

```
Dim a As Complex, b As Double, c As Double, pi As Double
a = ComplexSet(1.5, 2.0)
b = ComplexArg(a)
pi = 3.14159265
c = b * 180 / pi      ‘ radian -> degree
```

ComplexConj(*x*)

Syntax

Result = ComplexConj(*x*)

Description

Returns the conjugate complex number of complex number *x*.

Data type

x : Complex type (Complex)

Result : Complex type (Complex)

Example of use

```
Dim a As Complex, b As Complex
a = ComplexSet(1.5, 2.0)
b = ComplexConj(a)
```

ComplexCos(*x*)

Syntax

Result = ComplexCos(*x*)

Description

Returns the cosine (cos(*x*)) of complex number *x*.

Data type

x : Complex type (Complex)

Result : Complex type (Complex)

Example of use

```
Dim a As Complex, b As Complex
a = ComplexSet(1.5, 2.0)
b = ComplexCos(a)
```

ComplexCosh(*x*)

Syntax

Result = ComplexCosh(*x*)

Description

Returns the hyperbolic cosine (cosh(*x*)) of complex number *x*.

Data type

x : Complex type (Complex)

Result : Complex type (Complex)

Example of use

```
Dim a As Complex, b As Complex
a = ComplexSet(1.5, 2.0)
b = ComplexCosh(a)
```

ComplexDiv(*x*,*y*)

Syntax

Result = ComplexDiv(*x*,*y*)

Description

Returns the result (x/y) of the division of complex number *x* and another *y*.

Data type

x : Complex type (Complex)

y : Complex type (Complex)

Result : Complex type (Complex)

Example of use

```
Dim a As Complex, b As Complex, c As Complex
a = ComplexSet(1.5, 2.0)
b = ComplexSet(0.5, 3.5)
c = ComplexDiv(a, b)
```

ComplexExp(*x*)

Syntax

Result = ComplexExp(*x*)

Description

Returns the involution of the complex number *x*.

Data type

x : Complex type (Complex)

Result : Complex type (Complex)

Example of use

```
Dim a As Complex, b As Complex
a = ComplexSet(1.5, 2.0)
b = ComplexExp(a)
```

ComplexLog(*x*)

Syntax

Result = ComplexLog(*x*)

Description

Returns the natural logarithm (log(*x*)) of complex number *x*.

Data type

x : Complex type (Complex)

Result : Complex type (Complex)

Example of use

```
Dim a As Complex, b As Complex
a = ComplexSet(1.5, 2.0)
b = ComplexLog(a)
```

ComplexLog10(*x*)

Syntax

Result = ComplexLog(*x*)

Description

Returns the common logarithm (log 10(*x*)) of complex number *x*.

Data type

x : Complex type (Complex)

Result : Complex type (Complex)

Example of use

```
Dim a As Complex, b As Complex
a = ComplexSet(1.5, 2.0)
b = ComplexLog10(a)
```

ComplexMul(*x*,*y*)

Syntax

Result = ComplexMul(*x*,*y*)

Description

Returns the result (x\ y) of the multiplication of complex number *x* and another *y*.

Data type

x : Complex type (Complex)

y : Complex type (Complex)

Result : Complex type (Complex)

Example of use

```
Dim a As Complex, b As Complex, c As Complex
a = ComplexSet(1.5, 2.0)
b = ComplexSet(0.5, 3.5)
c = ComplexMul(a, b)
```

ComplexNorm(*x*)

Syntax

Result = ComplexNorm(*x*)

Description

Returns the square of the absolute value of complex number *x*.

Data type

x : Complex type (Complex)

Result : Double precision floating point type (Double)

Example of use

```
Dim a As Complex, b As Double
a = ComplexSet(1.5, 2.0)
b = ComplexNorm(a)
```

ComplexPolar(*x*,*y*)

Syntax

z = ComplexPolar(x,y)

Description

Sets a complex number to a complex type variable z. Specifies a complex number with an absolute value x and a phase angle y *(radian)*.

Data type

x : Double precision floating point type (Double)

y : Double precision floating point type (Double)

z : Complex type (Complex)

Example of use

```
Dim a As Complex, pi As Double
pi = 3.14159265
a = ComplexPolar(2.5, 60 * pi / 180)
```

ComplexSet(*x*,*y*)

Syntax

z = ComplexSet(*x*,*y*)

Description

Sets a complex number to a complex type variable *z*. Specifies a complex number with a real part *x* and an imaginary part *y*. (Sets *x* and *y* to *z.real* and *z.imag* respectively.)

Data type

x : Double precision floating point type (Double)

y : Double precision floating point type (Double)

z : Complex type (Complex)

Example of use

```
Dim a as Complex
a = ComplexSet(1.5, 2.0)
```

ComplexSetArray(*x*)

Syntax

y = ComplexSetArray(*x*)

Description

Converts a variant type or double floating point type array *x* that contains complex numbers using 2 elements to store each complex number in the order of the real part and imaginary part to complex type array *y*.

Data type

x : Variant type (Variant) array or Double precision floating point type (Double) array

y : Complex type (Complex) array

Example of use

```
Dim a as Variant, b as Complex
a = SCPI.CALCulate(1).SELected.DATA.SDATa
b = ComplexSetArray(a)
```

ComplexSin(*x*)

Syntax

Result = ComplexSin(*x*)

Description

Returns the sine (sin(*x*)) of complex number *x*.

Data type

x : Complex type (Complex)

Result : Complex type (Complex)

Example of use

```
Dim a As Complex, b As Complex
a = ComplexSet(1.5, 2.0)
b = ComplexSin(a)
```

ComplexSinh(*x*)

Syntax

Result = ComplexSinh(*x*)

Description

Returns the hyperbolic sine (sinh(*x*)) of complex number *x*.

Data type

x : Complex type (Complex)

Result : Complex type (Complex)

Example of use

```
Dim a As Complex, b As Complex
a = ComplexSet(1.5, 2.0)
b = ComplexSinh(a)
```

ComplexSqrt(*x*)

Syntax

Result = ComplexSqrt(*x*)

Description

Returns the square root ( ) of the complex number *x*.

Data type

x : Complex type (Complex)

Result : Complex type (Complex)

Example of use

```
Dim a As Complex, b As Complex
a = ComplexSet(1.5, 2.0)
b = ComplexSqrt(a)
```

ComplexSub(*x*,*y*)

Syntax

Result = ComplexSub(*x*,*y*)

Description

Returns the result (x - y) of the subtraction complex number *x* and another *y*.

Data type

x : Complex type (Complex)

y : Complex type (Complex)

Result : Complex type (Complex)

Example of use

```
Dim a As Complex, b As Complex, c As Complex
a = ComplexSet(1.5, 2.0)
b = ComplexSet(0.5, 3.5)
c = ComplexSub(a, b)
```

Waveform Analysis Library

Waveform (Ripple) Analysis Library

Other topics about VBA Programming

Overview

By combining the COM objects provided with the E5061B and the ripple analysis library, you can easily perform the ripple analysis of waveforms.

Flow of Programming using Ripple Analysis Library

Below table shows the flow of the program development using the ripple analysis library. First, set up the analysis range and peak definition to use the procedures for ripple analysis.

1. Condition settings before using the ripple analysis library:
    - Specifying the analysis range
    - Setting the peak definition
2. Using the ripple analysis library

Condition Setting before using Ripple Analysis Library

Since the analysis conditions are not specified in the ripple analysis library, before using the procedure for ripple analysis, set up the analysis range and the peak definition using COM objects.

Specifying the Analysis Range

Use the following COM objects to specify the analysis range for ripple analysis.

- SCPI.CALCulate(Ch).SELected.FUNCtion.DOMain.STARt
- SCPI.CALCulate(Ch).SELected.FUNCtion.DOMain.STOP
- SCPI.CALCulate(Ch).SELected.FUNCtion.DOMain.STATe
- SCPI.CALCulate(Ch).SELected.FUNCtion.DOMain.COUPle

Setting the Peak Definition

Use the following COM objects to set up the peak definition for ripple analysis.

- SCPI.CALCulate(Ch).SELected.FUNCtion.PEXCursion

- SCPI.CALCulate(Ch).SELected.FUNCtion.PPOLarity

List of Ripple Analysis Library

Use the provided procedures for ripple analysis to analyze the ripple of waveforms and output the result. All procedures perform analysis only within the stimulus range for the specified channel.

Function Name	Description
MaxPeakToPeak(Chan)	Returns the maximum value of the difference between a positive peak and a negative peak
MaxRightGap(Chan)	Returns the maximum value of the difference between a positive peak and its right adjacent negative peak.
MaxLeftGap(Chan)	Returns the maximum value of the difference between a positive peak and its left adjacent negative peak.
MaxGap(Chan)	Returns the maximum value of the difference between a positive peak and its adjacent negative peak.
MaxEnvelopeGap(Chan)	Returns the maximum value of the vertical distance between a line segment connecting 2 adjacent positive peaks and the negative peak between them.
GapMean(Chan)	Returns the mean value of the differences between a negative peak and its right and left adjacent positive peaks.
MaxRippleValue(Chan)	Returns the maximum value of the total of the differences between a negative peak and its right and left adjacent positive peaks.
MaxRipplePoint(Chan,Stim)	Returns the maximum value of the total of the differences between a negative peak and its right and left adjacent positive peaks and the

	stimulus value (*Stim*) of the valley of the ripple.
Pole(Chan,D,LeftStim,LeftValue, RightStim,RightValue)	Returns the values (*LeftValue* and *RightValue*) and the stimulus values (*LeftStimulus* and *RightStimulus*) of the right and left negative peaks detected first below the specified value (*D*) relative to the maximum value.
FirstRightGap(Chan)	Returns the difference between the positive peak detected first when searching from the left edge toward the right edge and its right adjacent negative peak.
FirstLeftGap(Chan)	Returns the difference between the positive peak detected first when searching from the right edge toward the left edge and its left adjacent negative peak.
FirstRightInterval(Chan)	Returns the difference of the stimulus value between the positive peak detected first when searching from the left edge toward the right edge and its right adjacent negative peak.
FirstLeftInterval(Chan)	Returns the difference of the stimulus value between the positive peak detected first when searching from the left edge toward the right edge and its left adjacent negative peak.

Sample Program

Here is a simple sample program using the ripple analysis procedures.

```
Sub Sample()

Dim Val As Double    (1)

SCPI.CALCulate(1).SELected.FUNCtion.PEXCursion = 1.5    (2)
SCPI.CALCulate(1).SELected.FUNCtion.PPOLarity = "BOTH"    (2)
SCPI.CALCulate(1).SELected.FUNCtion.DOMain.STARt = 935E6    (3)
```

```
SCPI.CALCulate(1).SELected.FUNCtion.DOMain.STOP = 960E6    (3)
SCPI.CALCulate(1).SELected.FUNCtion.DOMain.STATe = True    (3)
  .
  .
Val = MaxPeakToPeak(1)     (4)

End Sub
```

Let us break down the code into a number of blocks and see what they do.

1. Defines a variable Val as Double.
2. Sets the lower limit of the peak excursion value and polarity of the peak search to 1.5 and both positive peak and negative peak, respectively.
3. Sets the analysis range of channel 1 to 935 MHz to 960 MHz.
4. For channel 1, substitute the return value from the MaxPeakToPeak function (procedure) in the ripple analysis library to the Val variable.

Waveform (Ripple) Analysis Library

Other topics about VBA Programming

Overview

By combining the COM objects provided with the E5061B and the ripple analysis library, you can easily perform the ripple analysis of waveforms.

Flow of Programming using Ripple Analysis Library

Below table shows the flow of the program development using the ripple analysis library. First, set up the analysis range and peak definition to use the procedures for ripple analysis.

1. Condition settings before using the ripple analysis library:
    - Specifying the analysis range
    - Setting the peak definition
2. Using the ripple analysis library

Condition Setting before using Ripple Analysis Library

Since the analysis conditions are not specified in the ripple analysis library, before using the procedure for ripple analysis, set up the analysis range and the peak definition using COM objects.

Specifying the Analysis Range

Use the following COM objects to specify the analysis range for ripple analysis.

- SCPI.CALCulate(Ch).SELected.FUNCtion.DOMain.STARt
- SCPI.CALCulate(Ch).SELected.FUNCtion.DOMain.STOP
- SCPI.CALCulate(Ch).SELected.FUNCtion.DOMain.STATe
- SCPI.CALCulate(Ch).SELected.FUNCtion.DOMain.COUPle

Setting the Peak Definition

Use the following COM objects to set up the peak definition for ripple analysis.

- SCPI.CALCulate(Ch).SELected.FUNCtion.PEXCursion
- SCPI.CALCulate(Ch).SELected.FUNCtion.PPOLarity

List of Ripple Analysis Library

Use the provided procedures for ripple analysis to analyze the ripple of waveforms and output the result. All procedures perform analysis only within the stimulus range for the specified channel.

Function Name	Description
MaxPeakToPeak(Chan)	Returns the maximum value of the difference between a positive peak and a negative peak
MaxRightGap(Chan)	Returns the maximum value of the difference between a positive peak and its right adjacent negative peak.
MaxLeftGap(Chan)	Returns the maximum value of the difference between a positive peak and its left adjacent negative peak.
MaxGap(Chan)	Returns the maximum value of the difference between a positive peak and its adjacent negative peak.
MaxEnvelopeGap(Chan)	Returns the maximum value of the vertical distance between a line segment connecting 2 adjacent positive peaks and the negative peak between them.
GapMean(Chan)	Returns the mean value of the differences between a negative peak and its right and left adjacent positive peaks.
MaxRippleValue(Chan)	Returns the maximum value of the total of the differences between a negative peak and its right and left adjacent positive peaks.
MaxRipplePoint(Chan,Stim)	Returns the maximum value of the total of the differences between a negative peak and its right and left adjacent positive peaks and the stimulus value (*Stim*) of the valley

	of the ripple.
Pole(Chan,D,LeftStim,LeftValue, RightStim,RightValue)	Returns the values (*LeftValue* and *RightValue*) and the stimulus values (*LeftStimulus* and *RightStimulus*) of the right and left negative peaks detected first below the specified value (*D*) relative to the maximum value.
FirstRightGap(Chan)	Returns the difference between the positive peak detected first when searching from the left edge toward the right edge and its right adjacent negative peak.
FirstLeftGap(Chan)	Returns the difference between the positive peak detected first when searching from the right edge toward the left edge and its left adjacent negative peak.
FirstRightInterval(Chan)	Returns the difference of the stimulus value between the positive peak detected first when searching from the left edge toward the right edge and its right adjacent negative peak.
FirstLeftInterval(Chan)	Returns the difference of the stimulus value between the positive peak detected first when searching from the left edge toward the right edge and its left adjacent negative peak.

Sample Program

Here is a simple sample program using the ripple analysis procedures.

```
Sub Sample()

Dim Val As Double    (1)

SCPI.CALCulate(1).SELected.FUNCtion.PEXCursion = 1.5    (2)
SCPI.CALCulate(1).SELected.FUNCtion.PPOLarity = "BOTH"    (2)
SCPI.CALCulate(1).SELected.FUNCtion.DOMain.STARt = 935E6    (3)
SCPI.CALCulate(1).SELected.FUNCtion.DOMain.STOP = 960E6    (3)
```

```
SCPI.CALCulate(1).SELected.FUNCtion.DOMain.STATe = True    (3)
 .
 .
Val = MaxPeakToPeak(1)     (4)

End Sub
```

Let us break down the code into a number of blocks and see what they do.

1. Defines a variable Val as Double.
2. Sets the lower limit of the peak excursion value and polarity of the peak search to 1.5 and both positive peak and negative peak, respectively.
3. Sets the analysis range of channel 1 to 935 MHz to 960 MHz.
4. For channel 1, substitute the return value from the MaxPeakToPeak function (procedure) in the ripple analysis library to the Val variable.

Procedure Reference

FirstLeftGap(*Chan*)

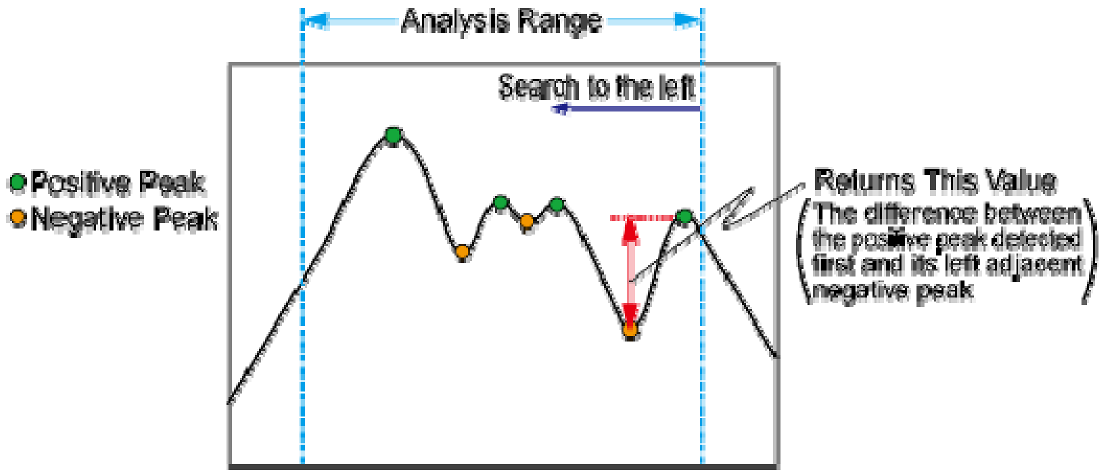
Syntax

Value = FirstLeftGap(*Chan*)

Description

Returns the response difference between the positive peak detected first when searched from the right edge toward the left edge within the analysis range and its left adjacent negative peak.

FirstLeftGap



e5071c431

Variable

Parameter	*Chan*
Description	Specifies the channel number.
Data type	Integer type (Integer)
Range	1 to 9
Note	If the specified variable is out of the allowable setup range, an error occurs when executed.

Return value

Parameter	*Value*
Description	Returns the response difference between the fist detected positive peak when searching from the right edge toward the left edge within the analysis range and its left adjacent negative peak.
Data type	Double precision floating point type (Double)
Note	If no applicable point is detected, 0 is returned.

Example of use

```
Dim Value As Double

Value = FirstLeftGap(1)
MsgBox "First Left Gap =" & Value
```

FirstLeftInterval(*Chan*)

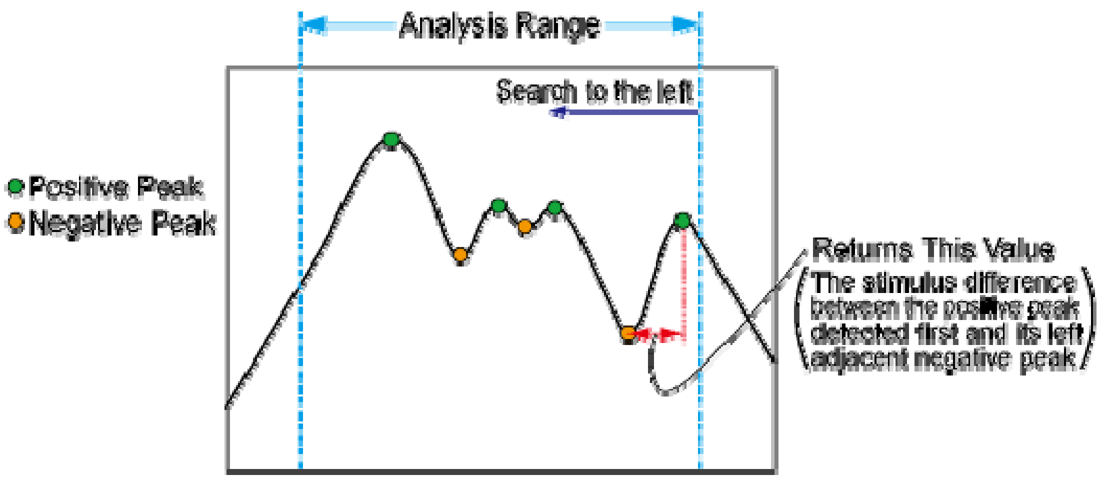
Syntax

Value = FirstLeftInterval(*Chan*)

Description

Returns the stimulus difference between the first detected positive peak when searching from the right edge toward the left edge within the analysis range and its left adjacent negative peak.

FirstLeftInterval



e5071c432

Variable

Parameter	*Chan*
Description	Specifies the channel number.
Data type	Integer type (Integer)
Range	1 to 9
Note	If the specified variable is out of the allowable setup range, an error occurs when executed.

Return value

Parameter	*Value*
Description	Returns the stimulus difference between the positive peak detected first when searched from the right edge toward the left edge within the analysis range and its left adjacent negative peak.
Data type	Double precision floating point type (Double)
Note	If no applicable point is detected, 0 is returned.

Example of use

```
Dim Value As Double

Value = FirstLeftInterval(1)
MsgBox "First Left Interval =" & Value
```

FirstRightGap(*Chan*)

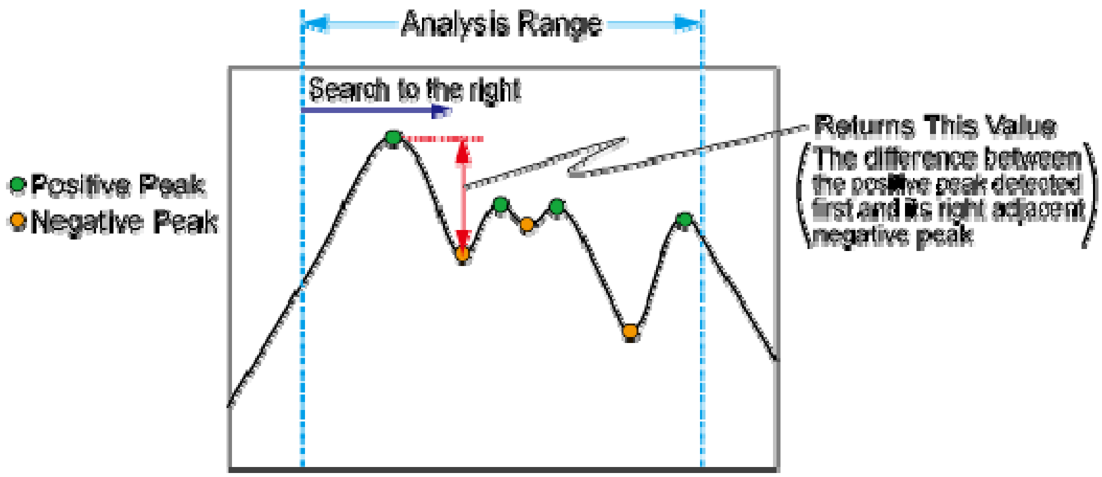
Syntax

Value = FirstRightGap(*Chan*)

Description

Returns the response difference between the fist detected positive peak when searching from the left edge toward the right edge within the analysis range and its right adjacent negative peak.

FirstRightGap



e5071c434

Variable

Parameter	*Chan*
Description	Specifies the channel number.
Data type	Integer type (Integer)
Range	1 to 9
Note	If the specified variable is out of the allowable setup range, an error occurs when executed.

Return value

Parameter	*Value*
Description	Returns the response difference between the first detected positive peak when searching from the left edge toward the right edge within the analysis range and its right adjacent negative peak.
Data type	Double precision floating point type (Double)
Note	If no applicable point is detected, 0 is returned.

Example of use

```
Dim Value As Double
Value = FirstRightGap(1)
MsgBox "First Right Gap =" & Value
```

FirstRightInterval(*Chan*)

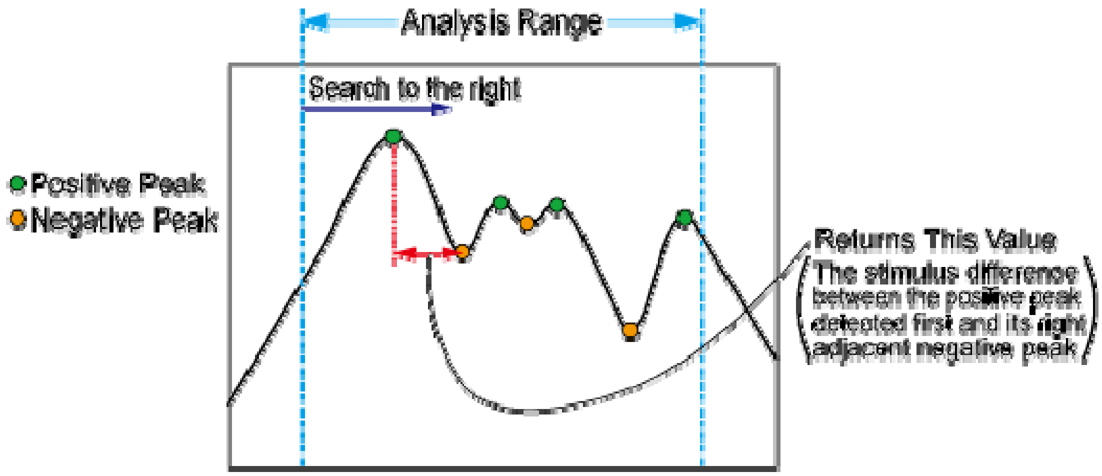
Syntax

Value = FirstRightInterval(*Chan*)

Description

Returns the stimulus difference between the positive peak detected first when searched from the left edge toward the right edge within the analysis range and its right adjacent negative peak.

FirstRightInterval



e5071c433

Variable

Parameter	*Chan*
Description	Specifies the channel number.
Data type	Integer type (Integer)
Range	1 to 9
Note	If the specified variable is out of the allowable setup range, an error occurs when executed.

Return value

Parameter	*Value*
Description	Returns the stimulus difference between the positive peak detected first when searched from the left edge toward the right edge within the analysis range and its right adjacent negative peak.
Data type	Double precision floating point type (Double)
Note	If no applicable point is detected, 0 is returned.

Example of use

```
Dim Value As Double

Value = FirstRightInterval(1)
MsgBox "First Right Interval =" & Value
```

GapMean(*Chan*)

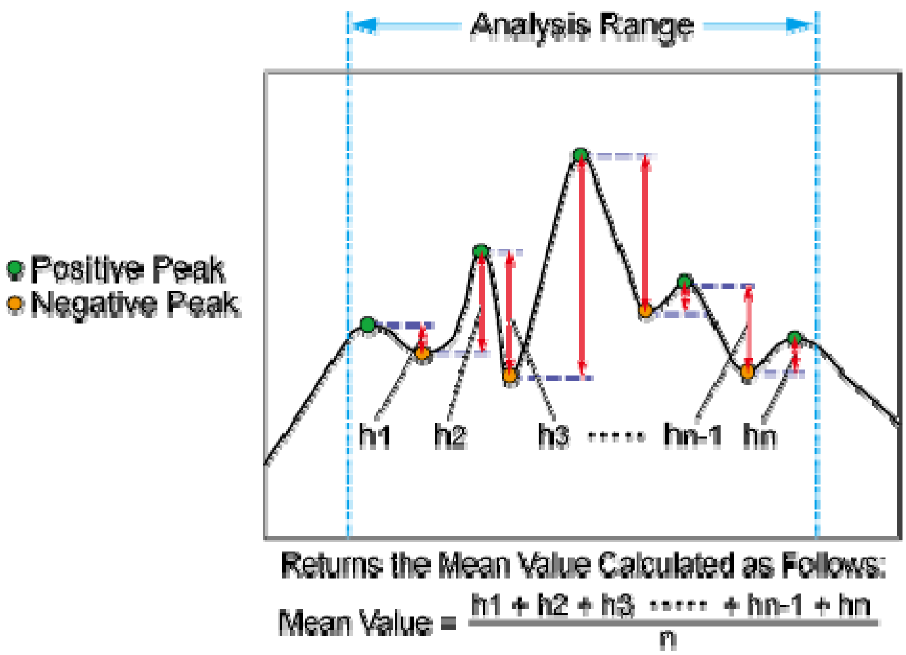
Syntax

Value = GapMean(*Chan*)

Description

Returns the mean value of the response differences between the negative peaks and its adjacent positive peaks within the analysis range.

GapMean



e5071c427

Variable

Parameter	*Chan*
Description	Specifies the channel number.
Data type	Integer type (Integer)
Range	1 to 9

Note	If the specified variable is out of the allowable setup range, an error occurs when executed.

Return value

Parameter	*Value*
Description	Returns the mean value of the response differences between the negative peaks and its right and left adjacent positive peaks.
Data type	Double precision floating point type (Double)
Note	If no applicable point is detected, 0 is returned.

Example of use

```
Dim Value As Double
Value = GapMean(1)
MsgBox "Gap Mean =" & Value
```

MaxEnvelopeGap(*Chan*)

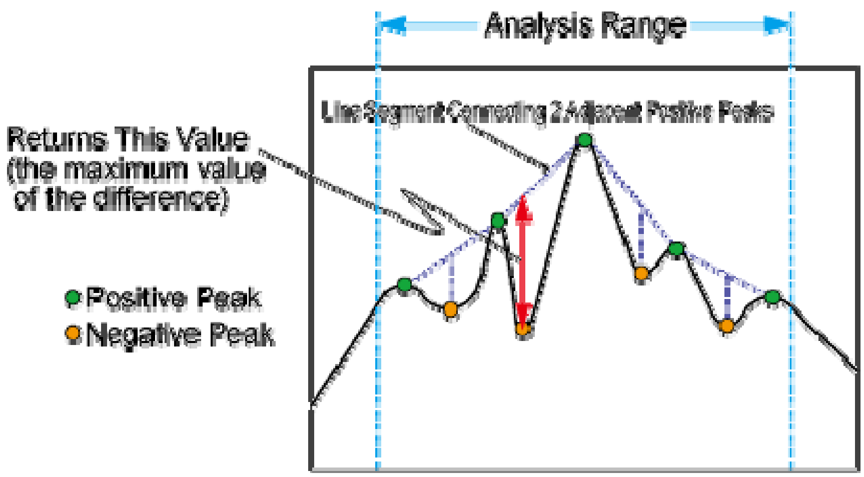
Syntax

Value = MaxEnvelopeGap(*Chan*)

Description

Returns the maximum value of the vertical distance between the line segments connecting 2 adjacent positive peaks and the negative peaks between them within the analysis range.

MaxEnvelopeGap



e5071c426

Variable

Parameter	*Chan*
Description	Specifies the channel number.
Data type	Integer type (Integer)
Range	1 to 9
Note	If the specified variable is out of the allowable setup range, an error occurs when executed.

Return value

Parameter	*Value*
Description	Returns the maximum value of the vertical distance between the line segments connecting 2 adjacent positive peaks and the negative peaks between them.
Data type	Double precision floating point type (Double)
Note	If no applicable point is detected, 0 is returned.

Example of use

```
Dim Value As Double
Value = MaxEnvelopeGap(1)
MsgBox "Max Envelope Gap =" & Value
```

MaxGap(*Chan*)

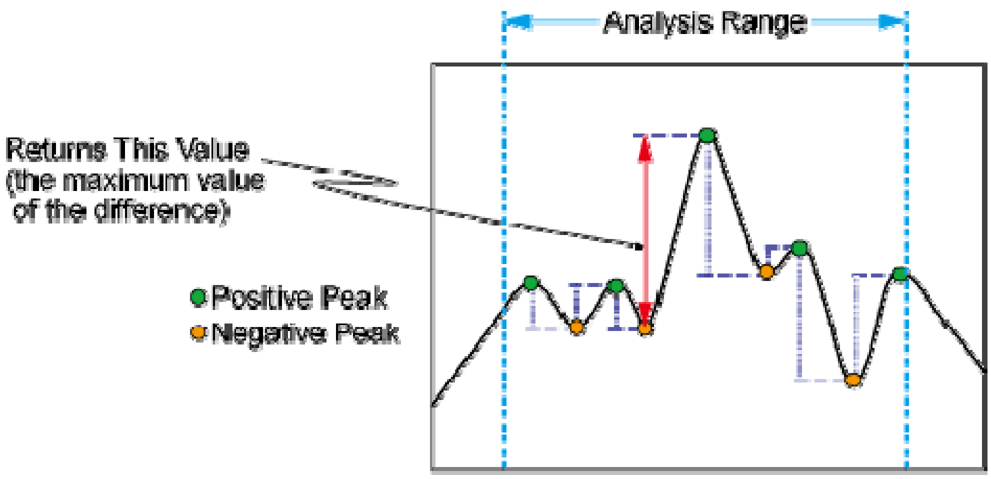
Syntax

Value = MaxGap(*Chan*)

Description

Returns the maximum value of the response differences between the positive peaks and its adjacent negative peaks within the analysis range.

MaxGap



e5071c425

Variable

Parameter	*Chan*
Description	Specifies the channel number.
Data type	Integer type (Integer)
Range	1 to 9
Note	If the specified variable is out of the allowable setup range, an error occurs when executed.

Return value

Parameter	*Value*
Description	Returns the maximum value of the response differences between the positive peaks and its adjacent negative peaks.
Data type	Double precision floating point type (Double)
Note	If no applicable point is detected, 0 is returned.

Example of use

```
Dim Value As Double
Value = MaxGap(1)
MsgBox "Max Gap =" & Value
```

MaxLeftGap(*Chan*)

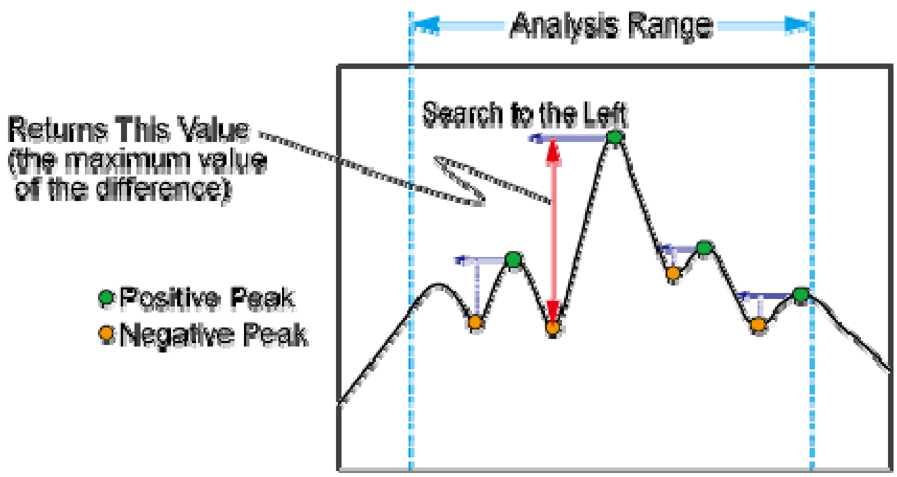
Syntax

Value = MaxLeftGap(*Chan*)

Description

Returns the maximum value of the response differences between the positive peaks and its left adjacent negative peaks within the analysis range.

MaxLeftGap



e5071c424

Variable

Parameter	*Chan*
Description	Specifies the channel number.
Data type	Integer type (Integer)
Range	1 to 9
Note	If the specified variable is out of the allowable setup range, an error occurs when executed.

Return value

Parameter	*Value*
Description	Returns the maximum value of the response differences between the positive peaks and its left adjacent negative peaks.
Data type	Double precision floating point type (Double)
Note	If no applicable point is detected, 0 is returned.

Example of use

```
Dim Value As Double
Value = MaxLeftGap(1)
MsgBox "Max Left Gap =" & Value
```

MaxPeakToPeak(*Chan*)

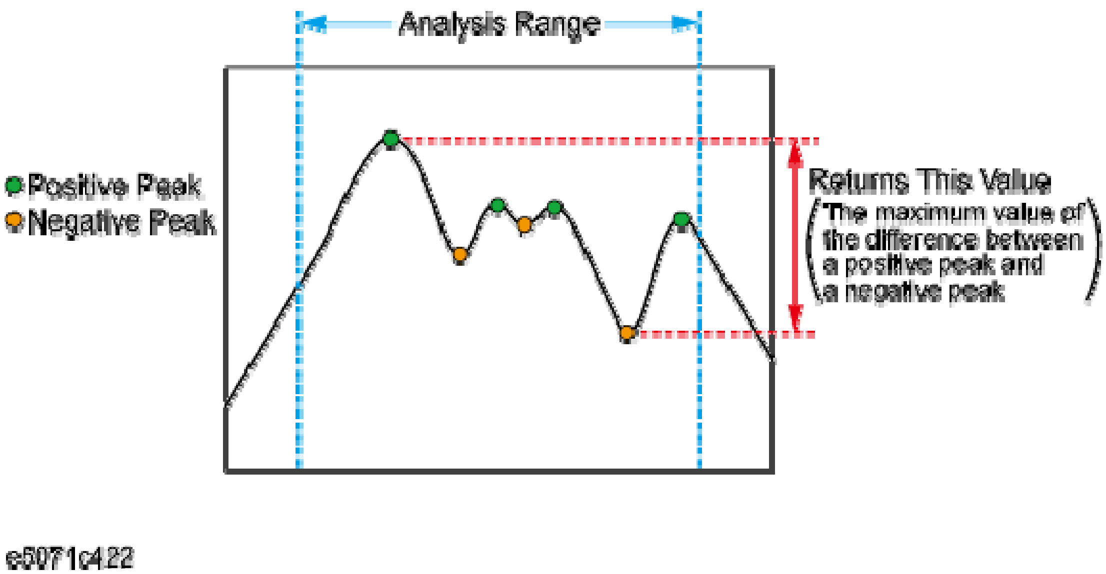
Syntax

Value = MaxPeakToPeak(*Chan*)

Description

Returns the maximum value of the response differences between the positive peaks and the negative peaks within the analysis range.

MaxPeakToPeak



Variable

Parameter	*Chan*
Description	Specifies the channel number.
Data type	Integer type (Integer)
Range	1 to 9
Note	If the specified variable is out of the allowable setup range, an error occurs when executed.

Return value

Parameter	*Value*
Description	Returns the maximum value of the response differences between the positive peaks and the negative peaks.
Data type	Double precision floating point type (Double)
Note	If no applicable point is detected, 0 is returned.

Example of use

```
Dim Value As Double
Value = MaxPeakToPeak(1)
MsgBox "Max Peak To Peak =" & Value
```

MaxRightGap(*Chan*)

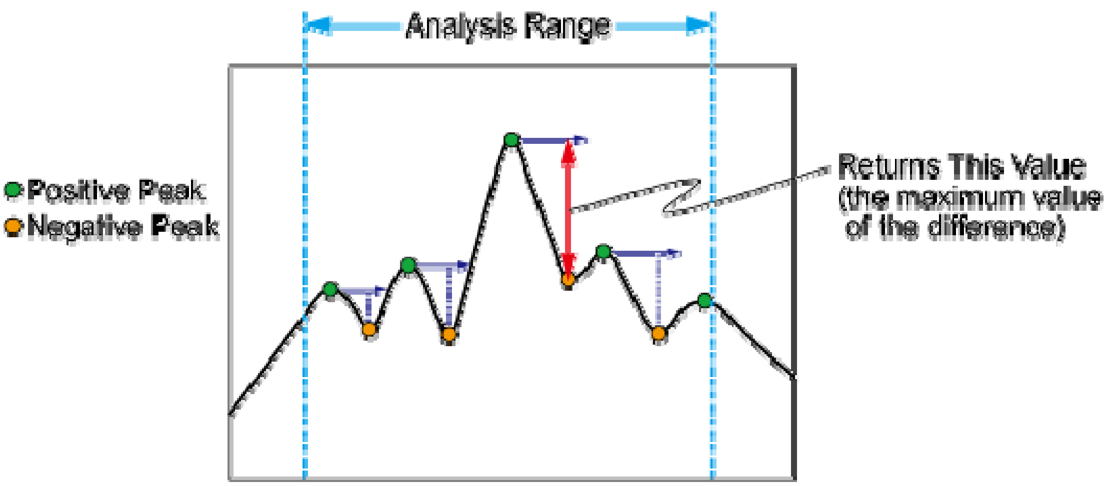
Syntax

Value = MaxRightGap(*chan*)

Description

Returns the maximum value of the response differences between the positive peaks and its right adjacent negative peaks within the analysis range.

MaxRightGap



e5071c423

Variable

Parameter	*Chan*
Description	Specifies the channel number.
Data type	Integer type (Integer)
Range	1 to 9
Note	If the specified variable is out of the allowable setup range, an error occurs when executed.

Return value

Parameter	*Value*
Description	Returns the maximum value of the response differences between the positive peaks and its right adjacent negative peaks.
Data type	Double precision floating point type (Double)
Note	If no applicable point is detected, 0 is returned.

Example of use

```
Dim Value As Double

Value = MaxRightGap(1)
MsgBox "Max Right Gap =" & Value
```

MaxRipplePoint(*Chan*,*Stim*)

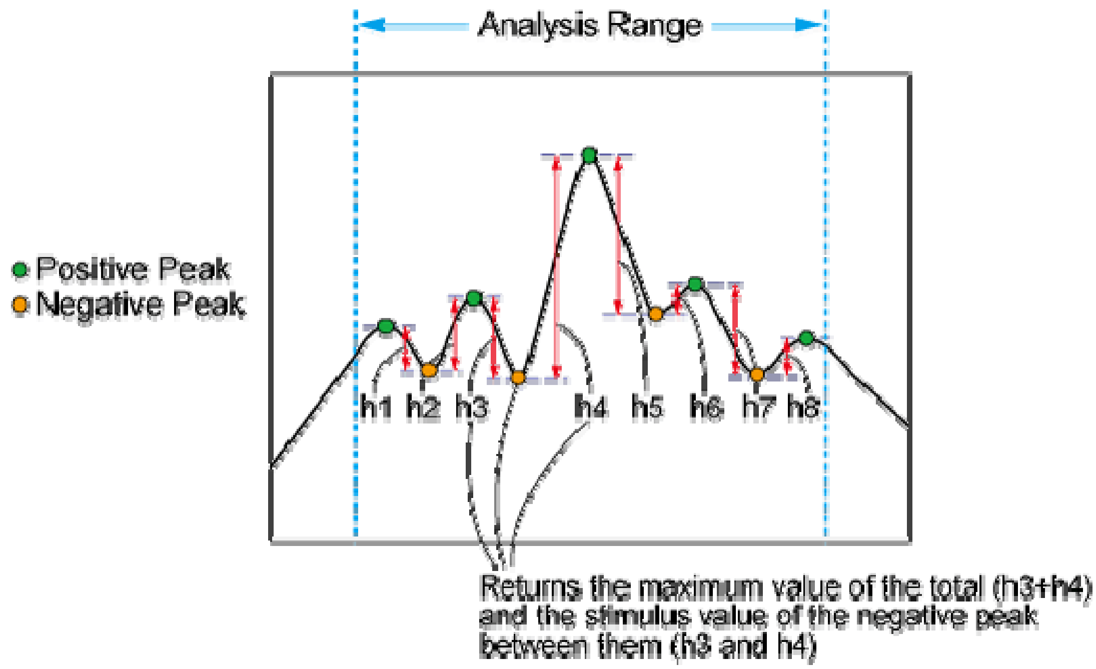
Syntax

Value = MaxRipplePoint(*Chan,Stim*)

Description

Returns the maximum value of the sum of the response differences between the negative peaks and its adjacent positive peaks and the stimulus value of the applicable negative peaks within the analysis range.

MaxRipplePoint



e5071c428

Variable

Parameter	*Chan*
Description	Specifies the channel number.
Data type	Integer type (Integer)
Range	1 to 9

Note	If the specified variable is out of the allowable setup range, an error occurs when executed.

Return value

Parameter	*Value*
Description	Returns the maximum value of the sum of the response differences between the negative peaks and its adjacent positive peaks.
Data type	Double precision floating point type (Double)
Note	If no applicable point is detected, 0 is returned.

Parameter	*Stim*
Description	Returns the stimulus value of the negative peak at which the sum of the response differences between the negative peak and its adjacent positive peaks is maximum.
Data type	Double precision floating point type (Double)
Note	If no applicable point is detected, 0 is returned.

Example of use

```
Dim Value As Double
Dim Stim As Double

Value = MaxRipplePoint(1, Stim)
MsgBox "Max Ripple Value =" & Value & " , Stimulus =" & Stim
```

MaxRippleValue(*Chan*)

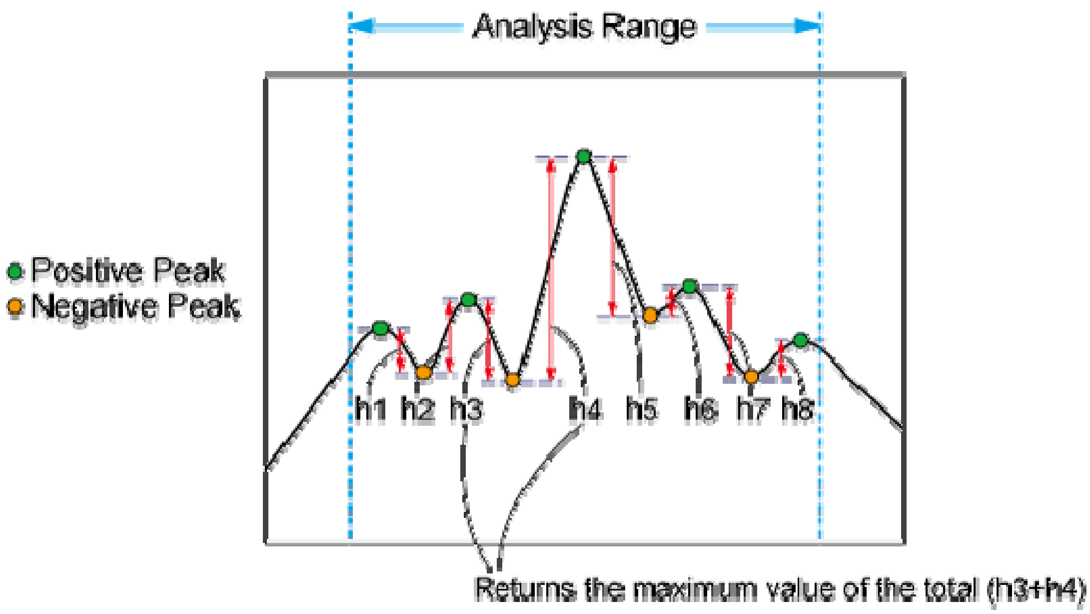
Syntax

Value = MaxRippleValue(*Chan*)

Description

Returns the maximum value of the sum of the response differences between the negative peaks and its adjacent positive peaks within the analysis range.

MaxRippleValue



e5071c429

Variable

Parameter	*Chan*
Description	Specifies the channel number.
Data type	Integer type (Integer)
Range	1 to 9

Note	If the specified variable is out of the allowable setup range, an error occurs when executed.

Return value

Parameter	*Value*
Description	Returns the maximum value of the sum of the response differences between the negative peaks and its adjacent positive peaks.
Data type	Double precision floating point type (Double)
Note	If no applicable point is detected, 0 is returned.

Example of use

```
Dim Value As Double
Value = MaxRippleValue(1)
MsgBox "Max Ripple Value =" & Value
```

Pole(*Chan*,*D*,*LeftStim*,*LeftValue*,*RightStim*,*RightValue*)

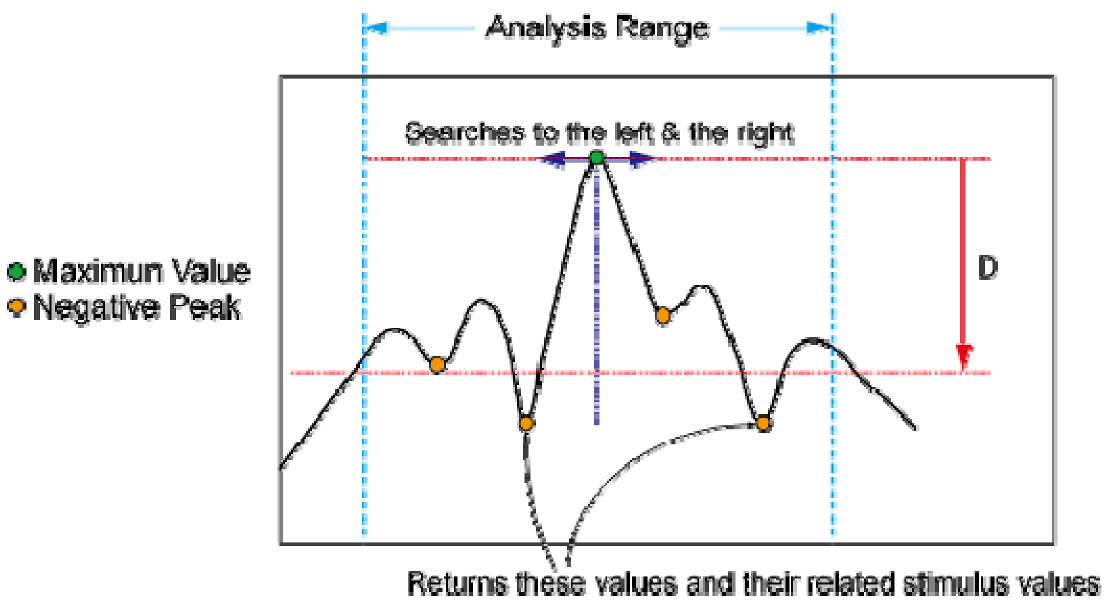
Syntax

Call Pole(*Chan*,*D*,*LeftStim*,*LeftValue*,*RightStim*,*RightValue*)

Description

For the negative peaks below the specified value (*D*) relative to the maximum value of the positive peaks within the analysis range, the response value (*LeftValue*) and stimulus value (*LeftStimulus*) of the negative peak first detected when searching to the left from the maximum value of the positive peaks, and the response value (*RightValue*) and stimulus value (*RightStimulus*) of the first detected negative peak when searching to the right from the maximum value of the positive peaks are returned.

Pole



e5071c430

Variable

Parameter	*Chan*
Description	Specifies the channel number.

Data type	Integer type (Integer)
Range	1 to 9
Note	If the specified variable is out of the allowable setup range, an error occurs when executed.

Parameter	*D*
Description	Specifies the difference from the maximum value.
Data type	Double precision floating point type (Double)

Return value (arguments)

Parameter	*LeftStim*
Description	Returns the stimulus value of the first detected negative peak to the left from the maximum value of the positive peaks.
Data type	Double precision floating point type (Double)
Note	If no applicable point is detected, 0 is returned.

Parameter	*LeftValue*

Description	Returns the response value of the first detected negative peak to the left from the maximum value of the positive peaks.
Data type	Double precision floating point type (Double)
Note	If no applicable point is detected, 0 is returned.

Parameter	*RightStim*
Description	Returns the stimulus value of the first detected negative peak to the right from the maximum value of the positive peaks.
Data type	Double precision floating point type (Double)
Note	If no applicable point is detected, 0 is returned.

Parameter	*RightValue*
Description	Returns the response value of the first detected negative peak to the right from the maximum value of the positive peaks.
Data type	Double precision floating point type (Double)
Note	If no applicable point is detected, 0 is returned.

Example of use

```
Dim LeftStim As Double
Dim LeftValue As Double
Dim RightStim As Double
```

```
Dim RightValue As Double

Call Pole(1, 1, LeftStim, LeftValue, RightStim, RightValue)

MsgBox "Left Pole =" & LeftStim & ":" & LeftValue
MsgBox "Right Pole =" & RightStim & ":" & RightValue
```

Command Reference

Notational Conventions

This section describes the notational conventions used for the description of commands reference.

Object Type

Object type describes different types of E5061B COM objects. The E5061B provides properties and methods as COM objects. COM objects which set (send)/read (return) the state of the E5061B using variables are defined as property and COM objects which does other processing are defined as method.

COM objects which are only used to read the state of the E5061B are indicated with "**Read-only**" and the ones used only to set the state of the E5061B are indicated by "**Write-only**" . COM object that can both read and write data to the E5061B are indicated by '**Read-Write**'.

Syntax

Syntax describes the syntax for sending a COM object from the E5061B VBA to the E5061B. The syntax consists of two parts: the object part and the set/read part, with an equal "=" inserted between them. Variables are indicated by italicized letters. Variables with () are indices. For indices with () having their preset values, you can omit "(*variable*)," and, if omitted, the preset values are automatically set.

There are some commands of which indices are not used. For example ,SCPI.CALCulate(Ch).PARameter.COUNt has no index for PARameter in the manual. However, PARameter can have an index like SCPI.CALCulate(Ch).PARameter(Tr).COUNt. In this case, specifying the trace number in brackets (Tr) has no meaning (same operation for any number).

The following table describes the 3 types of syntax for coding using objects:

Type	Description
"Object (property) = *variable*":	Set the stat of the E5061B.
variable=object (property):	Read the stat of the E5061B.
"Object (method)":	Perform some processing in the E5061B.

Description

Description describes how to use the COM object or the operation when executed.

Variable

Variable provides description about different variables that can be used with the COM objects. It gives the description, data type, allowable range, preset value, unit, resolution, and notes for *variable (italic)* shown in the syntax.

NOTE Variables declared as the string data type (String) are not case-sensitive. For variables of the string type that indicate arguments (written as *Param* in the syntax), you can omit lower-case letters.

The data types of the E5061B COM objects include 5 types as shown in the following table. Before using variables, declare the data type of each variable. If you do not declare the data type of a variable, it is automatically processed as a variant type.

Data type	Name	Consumed memory	Range
Long	Long integer type	4 bytes	-2,147,483,648 to 2,147,483,647 (-2^31 to 2^31 - 1)
Double	Double precision floating point type	8 bytes	For a negative value: -1.797693134862231E+308 to -4.940656458412465E-324 For a positive value: 4.940656458412465E-324 to 1.797693134862231E+308
Boolean	Boolean type	2 bytes	For COM: True or False (For SCPI: ON or OFF)
String	Character string type	1 byte / alphanumeric character	Up to approximately 2 billion characters
Variant	Variant type	16 bytes	No limitation
Binary	Byte		

Examples

Examples provide a sample of using the object through coding with the E5061B VBA.

Related Objects

Related objects provide information about other objects that are similar/related with the object.

Equivalent Key

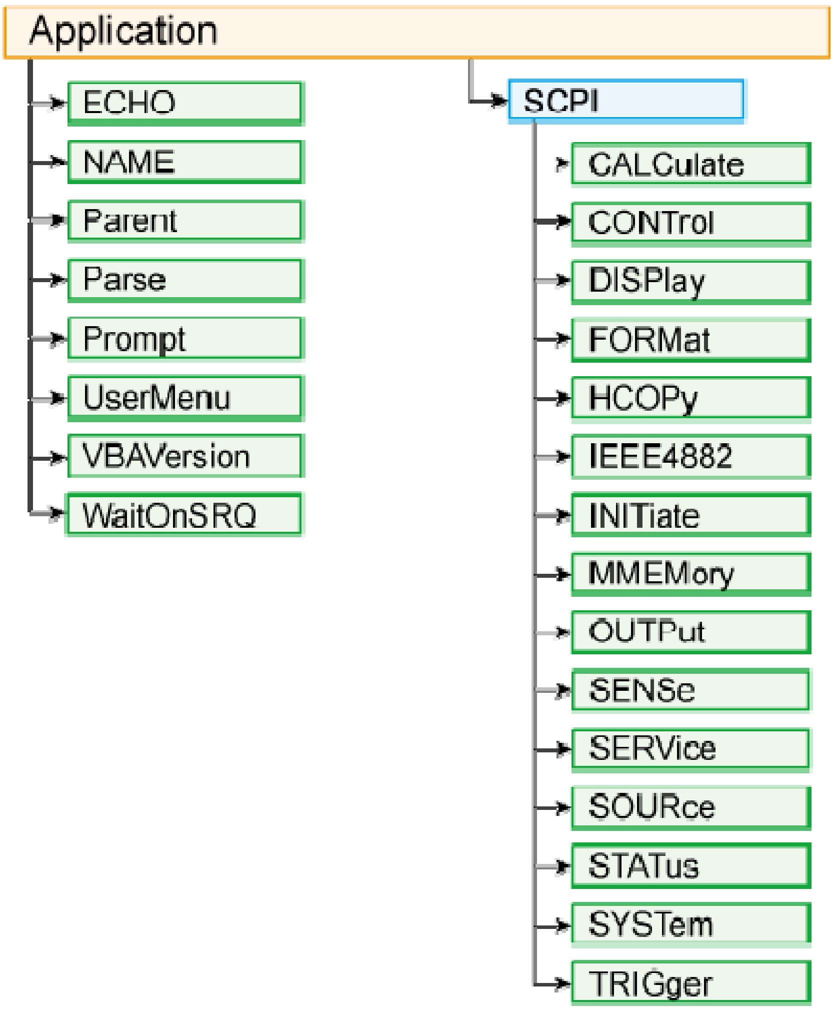
Equivalent key shows the operational procedure of the front panel keys that has the same effect as this object.

Equivalent SCPI command

Equivalent SCPI command shows the SCPI command to execute from an external controller. Its syntax, query response, and example of use are provided.

COM Object Model

The COM objects provided for the E5061B are structured hierarchically as shown below.



e5071c473

Application Objects

The Application objects are at the top of the hierarchy of the E5061B COM object model. They consist of 7 objects dedicated to the COM interface and SCPI objects corresponding to SCPI commands.

SCPI Objects

The SCPI objects are created to realize the SCPI commands of the E5061B with the COM interface.

In writing SCPI object messages, the conversion rules from the SCPI commands are as follows:

- SCPI. must be at the beginning. Notice that the IEEE common commands start with SCPI.IEEE4882. and "*" is omitted.
- Replace colons (:) used as the hierarchical separator symbol with dots (.).
- The number written in the object message is specified with ( ).
- You cannot omit the command message in the syntax.

SCPI command	COM object
OUTPUT 717;":SOUR1:POW -10"	SCPI.SOURce(1).POWer.LEVel.IMMediate.AMPLitude = -10
OUTPUT 717;":SENS1:CORR:COLL:METH:TYPE?" ENTER 717;A$	A = SCPI.SENSe(1).CORRection.COLLect.METHod:TYPE
OUTPUT 717;"*CLS"	SCPI.IEEE4882.CLS

Application Objects

ECHO

Object type

Method (**Write-only**)

Syntax

ECHO *V1,V2, ... ,V10*

ECHO *SCPI object*

Description

Displays in the echo window. (No read)

This command is different from SCPI.DISPlay.ECHO.DATA.

- Up to 10 data items can be displayed.
- Data is displayed as the declared data type without a cast.

Parameter	*V1,V2,* **.....** *,V10*
Description	Data you want to display in the echo window.
Data type	Variant type (Variant)

Examples

```
Dim Nop As Long
Dim i As Integer
Dim Fdata As Variant
Nop = SCPI.SENSe(1).SWEep.POINts
Fdata = SCPI.CALCulate(1).SELected.DATA.FDATa
ECHO "Test Results"
For i=1 to Nop
    ECHO i, Fdata(2*i-2), Fdata(2*i-1)
Next i


ECHO SCPI.SYSTem.ERRor
```

Related objects

SCPI.DISPlay.ECHO.DATA

Equivalent key

No equivalent key is available on the front panel.

NAME

Object type

Property

Syntax

App = NAME

Description

Reads out the application name of VBA. E5061B is always read out. (Read only)

Variable

Parameter	*App*
Description	Application name
Data type	Character string type (String)

Examples

```
Dim Inst As String
Inst = NAME
ECHO Inst
```

Equivalent key

No equivalent key is available on the front panel.

Parse

Object type

Method (**Write-only**)

Syntax

Parse(*Scpi*)

Return = Parse(*Scpi?*)

Description

Executes an SCPI command of the E5061B.

The Parse object is a little slower in the execution speed than the COM object which has the same function as the SCPI command because it must parse the message string of the SCPI command.

Variable

Parameter	*Scpi*
Description	SCPI command
Data type	Character string type (String)

Parameter	*Return*
Description	Response (query) of the SCPI command
Data type	Character string type (String)

Examples

```
Dim Start As String
Parse(":SENS1:FREQ:STAR 100E6")
Start = Parse(":SENS1:FREQ:STAR?")
```

```
Dim TtlLbl As String
Parse(":DISP:WIND1:TITL:DATA ""filter""")
TtlLbl = Parse(":DISP:WIND1:TITL:DATA?")
```

```
Dim Fmt As String
Parse(":CALC1:PAR2:SEL")
```

```
Parse(":CALC1:FORM SMIT")
Fmt = Parse(":CALC1:FORM?")
```

```
Dim BckLght As String
Parse(":SYST:BACK OFF")
BckLght = Parse(":SYST:BACK?")
```

Equivalent key

No equivalent key is available on the front panel.

Prompt

Object type

Method (**Write-only**)

Syntax

Prompt(*Mes*)

Description

Displays the message you specify on the instrument status bar (at the bottom of the LCD display) and suspends the program until the **Macro Setup** > **Continue** button is pressed. (No read)

NOTE When using this object, execute the program with the Visual Basic application closed since you need to press the **Macro Setup** > **Continue**. For more information, see Running a Program from the E5061B Measurement Screen. If you need to abort the program, see Stopping with the Dialog Box Appeared.

Variable

Parameter	*Mes*
Description	Message
Data type	Character string type (String)

Examples

Prompt ("Connect DUT, and then press [Continue]")

Equivalent key

No equivalent key is available on the front panel.

UserMenu.Item(*Key_id*).Caption

Object type

Property

Syntax

UserMenu.Item(*Key_id*).Caption = *Lbl*

Lbl = UserMenu.Item(*Key_id*).Caption

Description

Sets the label name of the user menu function softkeys 1 to 10 (*Key_id*).

Variable

Parameter	*Key_id*
Description	Softkey number for the user menu function
Data type	Long integer type (Long)
Range	1 to 10
Note	You cannot omit this because it does not have a preset value. If the specified variable is out of the valid setting range, an error occurs when executed.

Parameter	*Lbl*
Description	Softkey label name for the user menu function
Data type	Character string type (String)
Preset value	Varies depending on the specified softkey number.

Examples

```
Dim KeyLbl As String
UserMenu.Item(1).Caption = "Meas"
KeyLbl = UserMenu.Item(1).Caption
```

Equivalent key

No equivalent key is available on the front panel.

UserMenu.Item(*Key_id*).Enabled

Object type

Property

Syntax

UserMenu.Item(*Key_id*).Enabled = *Status*

Status = UserMenu.Item(*Key_id*).Enabled

Description

Makes the user menu function softkeys 1 to 10 (*Key_id*) enabled/disabled. The softkey label disabled is displayed in grey color and its softkey cannot be pressed.

Variable

Parameter	*Status*
Description	Enabled/disabled for the user menu function softkey
Data type	Boolean type (Boolean)
Range	Select from the following. • True or -1 Makes the softkey enabled. • False or 0 Makes the softkey enabled.
Preset value	True or -1

For information on the variable (*Key_id*), see UserMenu.Item.Caption.

Examples

```
Dim KeyEna As Boolean
UserMenu.Item(10).Enabled = False
KeyEna = UserMenu.Item(10).Enabled
```

Related objects

UserMenu.Press

Equivalent key

No equivalent key is available on the front panel.

UserMenu.PRESet

Object type

Method (**Write-only**)

Syntax

UserMenu.PRESet

Description

Presets the label name and enables/disables settings for the user menu softkeys. (No read)

Examples

```
UserMenu.PRESet
```

Related objects

UserMenu.Item.Caption

UserMenu.Item.Enabled

Equivalent key

Macro Setup > **Preset User Menu**

UserMenu.Press(*Key_id*)

Object type

Method (**Write-only**)

Syntax

UserMenu.Press(*Key_id*)

Description

Presses one of the user menu function softkeys 1 to 10 (*id*). (No read)

Variable

For information on the variable (*Key_id*), see UserMenu.Item.Caption.

Examples

```
UserMenu.Press(1)
```

Related objects

UserMenu.Item.Enabled

Equivalent key

Macro Setup > **User Menu** > **Button 1** to **Button 10**

UserMenu.Show

Object type

Method (**Write-only**)

Syntax

UserMenu.Show

Description

Displays the user menu function softkeys in the softkey area. (No read)

Examples

```
UserMenu.Show
```

Equivalent key

Macro Setup > **User Menu**

UserMenu_OnPress(ByVal *Key_id* As Long)

Object type

Event

Description

Executes the processing when one of the user menu function softkeys 1 to 10 (*Key_id*) are pressed. Write the processing in the "UserMenu" object. For more information on its use, see Executing a Procedure with a Softkey (User Menu Function).

Variable

For information on the variable (*Key_id*), see UserMenu.Item.Caption.

Examples

```
Private Sub UserMenu_OnPress(ByVal id As Long)
If id = 1 Then
MsgBox "Button 1 was pressed."
ElseIf id = 10 Then
MsgBox "Button 10 was pressed."
End If
End Sub
```

Equivalent key

No equivalent key is available on the front panel.

VBAVersion

Object type

Property

Syntax

Vers = VBAVersion

Description

Reads out the version information of VBA installed in the E5061B. (Read only)

Variable

Parameter	*Vers*
Description	VBA version information
Data type	Character string type (String)

Examples

```
Dim Version As String
Version = VBAVersion
ECHO Version
```

Equivalent key

From the **Help** menu of the Visual Basic editor, click **About Microsoft Visual Basic...**.

WaitOnSRQ

Object type

Method (**Write-only**)

Syntax

WaitOnSRQ *Status*, *Timeout*

Description

Suspends the program for a specified time until the RQS/MSS bit (bit 6) of the status byte register changes to 1. (No read)

Variable

Parameter	*Status*
Description	State of the RQS/MSS bit (read only)
Data type	Boolean type (Boolean)
Range	One of the following is returned. • True or -1 1 has been received within the specified time. • False or 0 1 has not been received within the specified time due to time-out or abort.

Parameter	*Timeout*
Description	Time-out time
Data type	Long integer type (Long)
Range	0 to 2,147,483,647
Preset value	-1 (infinity)
Unit	ms (millisecond)

Note	If the specified variable 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.

Examples

```
Dim Stat As Boolean
SCPI.IEEE4882.CLS
SCPI.STATus.OPERation.PTRansition = 0
SCPI.STATus.OPERation.NTRansition = 16
SCPI.STATus.OPERation.ENABle = 16
SCPI.IEEE4882.SRE = 128
SCPI.TRIGger.SEQuence.SOURce = "bus"
SCPI.INITiate(1).CONTinuous = True
SCPI.TRIGger.SEQuence.IMMediate
WaitOnSRQ Stat, 10000
If Stat = True Then
MsgBox "Done"
End If
```

Equivalent key

No equivalent key is available on the front panel.

ABORT

SCPI.ABORt

Object type

Method (**Write-only**)

Syntax

SCPI.ABORt

Description

This command aborts the measurement and changes the trigger sequence for all channels to idle state.

The channels for which the continuous startup mode is set to *ON* (setting to start up the trigger system continuously), changes immediately from idle to start-up state. See Trigger System for details.

Examples

```
SCPI.ABORt
```

Related objects

SCPI.INITiate(Ch).IMMediate

SCPI.INITiate(Ch).CONTinuous

Equivalent key

Trigger > **Restart**

Equivalent SCPI command

Syntax

:ABORt

Example of use

```
10 OUTPUT 717;":ABORt"
```

CALCULATE

SCPI.CALCulate(*Ch*).EPARameters.CIRCuit.A.C1

Object Type

Property (**Read-Write**)

Syntax

SCPI.CALCulate(*Ch*).EPARameters.CIRCuit.A.C1 = *Value*

Value = SCPI.CALCulate(*Ch*).EPARameters.CIRCuit.A.C1

Description

This command sets or gets C1 value of equivalent circuit parameters.

Variable

Parameter	*Value*
Description	C1
Data Type	Double precision floating point type (Double)
Range	-1E ~ 1E
Preset Value	0
Unit	Farad (F)
Resolution	1E-21 (1z)

Examples

```
Dim Var as Double
Var= 0
SCPI.CALCulate(1).EPARameters.CIRCuit.A.C1 = Var
Var = SCPI.CALCulate(1).EPARameters.CIRCuit.A.C1
```

Related Objects

SCPI.CALCulate(*Ch*).EPARameters.CIRCuit.A.R1

SCPI.CALCulate(*Ch*).EPARameters.CIRCuit.A.L1

Equivalent Key

Analysis > **Equivalent Circuit** > **Select Circuit** > **A**

Equivalent Circuit > **C1**

Using either the keyboard or ENTRY keys on the front panel, enter the value of C1.

Equivalent SCPI Command

Syntax

:CALCulate{[1]-4}:EPARameters:CIRCuit:A:C1 <numeric>

:CALCulate{[1]-4}:EPARameters:CIRCuit:A:C1?

Query Response

<numeric><newline><^END>

Example of use

```
10 OUTPUT 717;":CALC1:EPAR:CIRC:A:C1 0"
20 OUTPUT 717;":CALC1:EPAR:CIRC:A:C1?"
30 ENTER 717;A
```

SCPI.CALCulate(*Ch*).EPARameters.CIRCuit.A.L1

Object Type

Property (**Read-Write**)

Syntax

SCPI.CALCulate(*Ch*).EPARameters.CIRCuit.A.L1 = *Value*

Value = SCPI.CALCulate(*Ch*).EPARameters.CIRCuit.A.L1

Description

This command sets or gets L1 value of equivalent circuit parameters.

Variable

Parameter	*Value*
Description	L1
Data Type	Double precision floating point type (Double)
Range	-1E ~ 1E
Preset Value	0
Unit	Henry (H)
Resolution	1E-18 (1a)

Examples

```
Dim Var as Double
Var= 0
SCPI.CALCulate4.EPARameters.CIRCuit.A.L1 = Var
Var = SCPI.CALCulate4.EPARameters.CIRCuit.A.L1
```

Related Objects

SCPI.CALCulate(*Ch*).EPARameters.CIRCuit.A.R1

SCPI.CALCulate(*Ch*).EPARameters.CIRCuit.A.C1

Equivalent Key

Analysis > **Equivalent Circuit** > **Select Circuit** > **A**

Equivalent Circuit > **L1**

Using either the keyboard or ENTRY keys on the front panel, enter the value of L1.

Equivalent SCPI Command

Syntax

:CALCulate{[1]-4}:EPARameters:CIRCuit:A:L1 <numeric>

:CALCulate{[1]-4}:EPARameters:CIRCuit:A:L1?

Query Response

<numeric><newline><^END>

Example of use

```
10 OUTPUT 717;":CALC1:EPAR:CIRC:A:L1 0"
20 OUTPUT 717;":CALC1:EPAR:CIRC:A:L1?"
30 ENTER 717;A
```

SCPI.CALCulate(*Ch*).EPARameters.CIRCuit.A.R1

Object Type

Property (**Read-Write**)

Syntax

SCPI.CALCulate(*Ch*).EPARameters.CIRCuit.A.R1 = *Value*

Value = SCPI.CALCulate(*Ch*).EPARameters.CIRCuit.A.R1

Description

This command sets and gets R1 value of equivalent circuit parameters.

Variable

Parameter	*Value*
Description	R1
Data Type	Double precision floating point type (Double)
Range	-1E ~ 1E
Preset Value	0
Unit	Ohm (Ω)
Resolution	1E-18 (1a)

Examples

```
Dim Var as Double
Var= 0
SCPI.CALCulate4.EPARameters.CIRCuit.A.R1 = Var
Var = SCPI.CALCulate4.EPARameters.CIRCuit.A.R1
```

Related Objects

SCPI.CALCulate(*Ch*).EPARameters.CIRCuit.A.C1

SCPI.CALCulate(*Ch*).EPARameters.CIRCuit.A.L1

Equivalent Key

Analysis > **Equivalent Circuit** > **Select Circuit** > **A**

Equivalent Circuit > **R1**

Using either the keyboard or ENTRY keys on the front panel, enter the value of R1.

Equivalent SCPI Command

Syntax

:CALCulate{[1]-4}:EPARameters:CIRCuit:A:R1 <numeric>

:CALCulate{[1]-4}:EPARameters:CIRCuit:A:R1?

Query Response

<numeric><newline><^END>

Example of use

```
10 OUTPUT 717;":CALC1:EPAR:CIRC:A:R1 0"
20 OUTPUT 717;":CALC1:EPAR:CIRC:A:R1?"
30 ENTER 717;A
```

SCPI.CALCulate(*Ch*).EPARameters.CIRCuit.C.C1

Object Type

Property (**Read-Write**)

Syntax

SCPI.CALCulate(*Ch*).EPARameters.CIRCuit.C.C1 = *Value*

Value = SCPI.CALCulate(*Ch*).EPARameters.CIRCuit.C.C1

Description

This command sets or gets C1 value of equivalent circuit parameters.

Variable

Parameter	*Value*
Description	C1
Data Type	Double precision floating point type (Double)
Range	-1E ~ 1E
Preset Value	0
Unit	Farad (F)
Resolution	1E-21 (1z)

Examples

```
Dim Var as Double
Var= 0
SCPI.CALCulate(1).EPARameters.CIRCuit.C.C1 = Var
Var = SCPI.CALCulate(1).EPARameters.CIRCuit.C.C1
```

Related Objects

SCPI.CALCulate(*Ch*).EPARameters.CIRCuit.C.R1

SCPI.CALCulate(*Ch*).EPARameters.CIRCuit.C.L1

Equivalent Key

Analysis > **Equivalent Circuit** > **Select Circuit** > **C**

Equivalent Circuit > **C1**

Using either the keyboard or ENTRY keys on the front panel, enter the value of C1.

Equivalent SCPI Command

Syntax

:CALCulate{[1]-4}:EPARameters:CIRCuit:C:C1 <numeric>

:CALCulate{[1]-4}:EPARameters:CIRCuit:C:C1?

Query Response

<numeric><newline><^END>

Example of use

```
10 OUTPUT 717;":CALC1:EPAR:CIRC:C:C1 0"
20 OUTPUT 717;":CALC1:EPAR:CIRC:C:C1?"
30 ENTER 717;A
```

SCPI.CALCulate(*Ch*).EPARameters.CIRCuit.C.L1

Object Type

Property (**Read-Write**)

Syntax

SCPI.CALCulate(*Ch*).EPARameters.CIRCuit.C.L1 = *Value*

Value = SCPI.CALCulate(*Ch*).EPARameters.CIRCuit.C.L1

Description

This command sets or gets L1 value of equivalent circuit parameters.

Variable

Parameter	*Value*
Description	L1
Data Type	Double precision floating point type (Double)
Range	-1E ~ 1E
Preset Value	0
Unit	Henry (H)
Resolution	1E-18 (1a)

Examples

```
Dim Var as Double
Var= 0
SCPI.CALCulate(1).EPARameters.CIRCuit.C.L1 = Var
Var = SCPI.CALCulate(1).EPARameters.CIRCuit.C.L1
```

Related Objects

SCPI.CALCulate(*Ch*).EPARameters.CIRCuit.C.R1

SCPI.CALCulate(*Ch*).EPARameters.CIRCuit.C.C1

Equivalent Key

Analysis > **Equivalent Circuit** > **Select Circuit** > **C**

Equivalent Circuit > **L1**

Using either the keyboard or ENTRY keys on the front panel, enter the value of L1.

Equivalent SCPI Command

Syntax

:CALCulate{[1]-4}:EPARameters:CIRCuit:C:L1 <numeric>

:CALCulate{[1]-4}:EPARameters:CIRCuit:C:L1?

Query Response

<numeric><newline><^END>

Example of use

```
10 OUTPUT 717;":CALC1:EPAR:CIRC:C:L1 0"
20 OUTPUT 717;":CALC1:EPAR:CIRC:C:L1?"
30 ENTER 717;A
```

SCPI.CALCulate(*Ch*).EPARameters.CIRCuit.C.R1

Object Type

Property (**Read-Write**)

Syntax

SCPI.CALCulate(*Ch*).EPARameters.CIRCuit.C.R1 = *Value*

Value = SCPI.CALCulate(*Ch*).EPARameters.CIRCuit.C.R1

Description

This command sets or gets R1 value of equivalent circuit parameters.

Variable

Parameter	*Value*
Description	R1
Data Type	Double precision floating point type (Double)
Range	-1E ~ 1E
Preset Value	0
Unit	Ohm (Ω)
Resolution	1E-18 (1a)

Examples

```
Dim Var as Double
Var= 0
SCPI.CALCulate(1).EPARameters.CIRCuit.C.R1 = Var
Var = SCPI.CALCulate(1).EPARameters.CIRCuit.C.R1
```

Related Objects

SCPI.CALCulate(*Ch*).EPARameters.CIRCuit.C.C1

SCPI.CALCulate(*Ch*).EPARameters.CIRCuit.C.L1

Equivalent Key

Analysis > **Equivalent Circuit** > **Select Circuit** > **C**

Equivalent Circuit > **R1**

Using either the keyboard or ENTRY keys on the front panel, enter the value of R1.

Equivalent SCPI Command

Syntax

:CALCulate{[1]-4}:EPARameters:CIRCuit:C:R1 <numeric>

:CALCulate{[1]-4}:EPARameters:CIRCuit:C:R1?

Query Response

<numeric><newline><^END>

Example of use

```
10 OUTPUT 717;":CALC1:EPAR:CIRC:C:R1 0"
20 OUTPUT 717;":CALC1:EPAR:CIRC:C:R1?"
30 ENTER 717;A
```

SCPI.CALCulate(*Ch*).EPARameters.CIRCuit.B.C1

Object Type

Property (**Read-Write**)

Syntax

SCPI.CALCulate(*Ch*).EPARameters.CIRCuit.B.C1 = *Value*

Value = SCPI.CALCulate(*Ch*).EPARameters.CIRCuit.B.C1

Description

This command sets or gets C1 value of equivalent circuit parameters.

Variable

Parameter	*Value*
Description	C1
Data Type	Double precision floating point type (Double)
Range	-1E ~ 1E
Preset Value	0
Unit	Farad (F)
Resolution	1E-21 (1z)

Examples

```
Dim Var as Double
Var= 0
SCPI.CALCulate(1).EPARameters.CIRCuit.B.C1 = Var
Var = SCPI.CALCulate(1).EPARameters.CIRCuit.B.C1
```

Related Objects

SCPI.CALCulate(*Ch*).EPARameters.CIRCuit.B.R1

SCPI.CALCulate(*Ch*).EPARameters.CIRCuit.B.L1

Equivalent Key

Analysis > **Equivalent Circuit** > **Select Circuit** > **B**

Equivalent Circuit > **C1**

Using either the keyboard or ENTRY keys on the front panel, enter the value of C1.

Equivalent SCPI Command

Syntax

:CALCulate{[1]-4}:EPARameters:CIRCuit:B:C1 <numeric>

:CALCulate{[1]-4}:EPARameters:CIRCuit:B:C1?

Query Response

<numeric><newline><^END>

Example of use

```
10 OUTPUT 717;":CALC1:EPAR:CIRC:B:C1 0"
20 OUTPUT 717;":CALC1:EPAR:CIRC:B:C1?"
30 ENTER 717;A
```

SCPI.CALCulate(*Ch*).EPARameters.CIRCuit.B.L1

Object Type

Property (**Read-Write**)

Syntax

SCPI.CALCulate(*Ch*).EPARameters.CIRCuit.B.L1 = *Value*

Value = SCPI.CALCulate(*Ch*).EPARameters.CIRCuit.B.L1

Description

This command sets or gets L1 value of equivalent circuit parameters.

Variable

Parameter	*Value*
Description	L1
Data Type	Double precision floating point type (Double)
Range	-1E ~ 1E
Preset Value	0
Unit	Henry (H)
Resolution	1E-18 (1a)

Examples

```
Dim Var as Double
Var= 0
SCPI.CALCulate(1).EPARameters.CIRCuit.B.L1 = Var
Var = SCPI.CALCulate(1).EPARameters.CIRCuit.B.L1
```

Related Objects

SCPI.CALCulate(*Ch*).EPARameters.CIRCuit.B.R1

SCPI.CALCulate(*Ch*).EPARameters.CIRCuit.B.C1

Equivalent Key

Analysis > **Equivalent Circuit** > **Select Circuit** > **B**

Equivalent Circuit > **L1**

Using either the keyboard or ENTRY keys on the front panel, enter the value of L1.

Equivalent SCPI Command

Syntax

:CALCulate{[1]-4}:EPARameters:CIRCuit:B:L1 <numeric>

:CALCulate{[1]-4}:EPARameters:CIRCuit:B:L1?

Query Response

<numeric><newline><^END>

Example of use

```
10 OUTPUT 717;":CALC1:EPAR:CIRC:B:L1 0"
20 OUTPUT 717;":CALC1:EPAR:CIRC:B:L1?"
30 ENTER 717;A
```

SCPI.CALCulate(*Ch*).EPARameters.CIRCuit.B.R1

Object Type

Property (**Read-Write**)

Syntax

SCPI.CALCulate(*Ch*).EPARameters.CIRCuit.B.R1 = *Value*

Value = SCPI.CALCulate(*Ch*).EPARameters.CIRCuit.B.R1

Description

This command sets and gets R1 value of equivalent circuit parameters.

Variable

Parameter	*Value*
Description	R1
Data Type	Double precision floating point type (Double)
Range	-1E ~ 1E
Preset Value	0
Unit	Ohm (Ω)
Resolution	1E-18 (1a)

Examples

```
Dim Var as Double
Var= 0
SCPI.CALCulate(1).EPARameters.CIRCuit.B.R1 = Var
Var = SCPI.CALCulate(1).EPARameters.CIRCuit.B.R1
```

Related Objects

SCPI.CALCulate(*Ch*).EPARameters.CIRCuit.B.C1

SCPI.CALCulate(*Ch*).EPARameters.CIRCuit.B.L1

Equivalent Key

Analysis > **Equivalent Circuit** > **Select Circuit** > **B**

Equivalent Circuit > **R1**

Using either the keyboard or ENTRY keys on the front panel, enter the value of R1.

Equivalent SCPI Command

Syntax

:CALCulate{[1]-4}:EPARameters:CIRCuit:B:R1 <numeric>

:CALCulate{[1]-4}:EPARameters:CIRCuit:B:R1?

Query Response

<numeric><newline><^END>

Example of use

```
10 OUTPUT 717;":CALC1:EPAR:CIRC:B:R1 0"
20 OUTPUT 717;":CALC1:EPAR:CIRC:B:R1?"
30 ENTER 717;A
```

SCPI.CALCulate(*Ch*).EPARameters.CIRCuit.D.C1

Object Type

Property (**Read-Write**)

Syntax

SCPI.CALCulate(*Ch*).EPARameters.CIRCuit.D.C1 = *Value*

Value = SCPI.CALCulate(*Ch*).EPARameters.CIRCuit.D.C1

Description

This command sets or gets C1 value of equivalent circuit parameters.

Variable

Parameter	*Value*
Description	C1
Data Type	Double precision floating point type (Double)
Range	-1E ~ 1E
Preset Value	0
Unit	Farad (F)
Resolution	1E-21 (1z)

Examples

```
Dim Var as Double
Var= 0
SCPI.CALCulate(1).EPARameters.CIRCuit.D.C1 = Var
Var = SCPI.CALCulate(1).EPARameters.CIRCuit.D.C1
```

Related Objects

SCPI.CALCulate(*Ch*).EPARameters.CIRCuit.D.R1

SCPI.CALCulate(*Ch*).EPARameters.CIRCuit.D.L1

Equivalent Key

Analysis > **Equivalent Circuit** > **Select Circuit** > **D**

Equivalent Circuit > **C1**

Using either the keyboard or ENTRY keys on the front panel, enter the value of C1.

Equivalent SCPI Command

Syntax

:CALCulate{[1]-4}:EPARameters:CIRCuit:D:C1 <numeric>

:CALCulate{[1]-4}:EPARameters:CIRCuit:D:C1?

Query Response

<numeric><newline><^END>

Example of use

```
10 OUTPUT 717;":CALCulate1:EPAR:CIRC:D:C1 0"
20 OUTPUT 717;":CALCulate1:EPAR:CIRC:D:C1?"
30 ENTER 717;A
```

SCPI.CALCulate(*Ch*).EPARameters.CIRCuit.D.L1

Object Type

Property (**Read-Write**)

Syntax

SCPI.CALCulate(*Ch*).EPARameters.CIRCuit.D.L1 = *Value*

Value = SCPI.CALCulate(*Ch*).EPARameters.CIRCuit.D.L1

Description

This command sets or gets L1 value of equivalent circuit parameters.

Variable

Parameter	*Value*
Description	L1
Data Type	Double precision floating point type (Double)
Range	-1E ~ 1E
Preset Value	0
Unit	Henry (H)
Resolution	1E-18 (1a)

Examples

```
Dim Var as Double
Var= 0
SCPI.CALCulate(1).EPARameters.CIRCuit.D.L1 = Var
Var = SCPI.CALCulate(1).EPARameters.CIRCuit.D.L1
```

Related Objects

SCPI.CALCulate(*Ch*).EPARameters.CIRCuit.D.R1

SCPI.CALCulate(*Ch*).EPARameters.CIRCuit.D.C1

Equivalent Key

Analysis > **Equivalent Circuit** > **Select Circuit** > **D**

Equivalent Circuit > **L1**

Using either the keyboard or ENTRY keys on the front panel, enter the value of L1.

Equivalent SCPI Command

Syntax

```
:CALCulate{[1]-4}:EPARameters:CIRCuit:D:L1 <numeric>
:CALCulate{[1]-4}:EPARameters:CIRCuit:D:L1?
```

Query Response

```
<numeric><newline><^END>
```

Example of use

```
10 OUTPUT 717;":CALC1:EPAR:CIRC:D:L1 0"
20 OUTPUT 717;":CALC1:EPAR:CIRC:D:L1?"
30 ENTER 717;A
```

SCPI.CALCulate(*Ch*).EPARameters.CIRCuit.D.R1

Object Type

Property (**Read-Write**)

Syntax

SCPI.CALCulate(*Ch*).EPARameters.CIRCuit.D.R1 = *Value*

Value = SCPI.CALCulate(*Ch*).EPARameters.CIRCuit.D.R1

Description

This command sets or gets R1 value of equivalent circuit parameters.

Variable

Parameter	*Value*
Description	R1
Data Type	Double precision floating point type (Double)
Range	-1E ~ 1E
Preset Value	0
Unit	Ohm (Ω)
Resolution	1E-18 (1a)

Examples

```
Dim Var as Double
Var= 0
SCPI.CALCulate(1).EPARameters.CIRCuit.D.R1 = Var
Var = SCPI.CALCulate(1).EPARameters.CIRCuit.D.R1
```

Related Objects

SCPI.CALCulate(*Ch*).EPARameters.CIRCuit.D.C1

SCPI.CALCulate(*Ch*).EPARameters.CIRCuit.D.L1

Equivalent Key

Analysis > **Equivalent Circuit** > **Select Circuit** > **D**

Equivalent Circuit > **R1**

Using either the keyboard or ENTRY keys on the front panel, enter the value of R1.

Equivalent SCPI Command

Syntax

:CALCulate{[1]-4}:EPARameters:CIRCuit:D:R1 <numeric>

:CALCulate{[1]-4}:EPARameters:CIRCuit:D:R1?

Query Response

<numeric><newline><^END>

Example of use

```
10 OUTPUT 717;":CALC1:EPAR:CIRC:D:R1 0"
20 OUTPUT 717;":CALC1:EPAR:CIRC:D:R1?"
30 ENTER 717;A
```

SCPI.CALCulate(*Ch*).EPARameters.CIRCuit.E.C0

Object Type

Property (**Read-Write**)

Syntax

SCPI.CALCulate(*Ch*).EPARameters.CIRCuit.E.C0 = *Value*

Value = SCPI.CALCulate(*Ch*).EPARameters.CIRCuit.E.C0

Description

This command sets or gets C0 value of equivalent circuit parameters.

Variable

Parameter	*Value*
Description	C0
Data Type	Double precision floating point type (Double)
Range	-1E ~ 1E
Preset Value	0
Unit	Farad (F)
Resolution	1E-18 (1a)

Examples

```
Dim Var as Double
Var= 0
SCPI.CALCulate(1).EPARameters.CIRCuit.E.C0 = Var
Var = SCPI.CALCulate(1).EPARameters.CIRCuit.E.C0
```

Related Objects

SCPI.CALCulate(*Ch*).EPARameters.CIRCuit.E.R1

SCPI.CALCulate(*Ch*).EPARameters.CIRCuit.E.C1

SCPI.CALCulate(*Ch*).EPARameters.CIRCuit.E.L1

Equivalent Key

Analysis > **Equivalent Circuit** > **Select Circuit** > **E**

Equivalent Circuit > **C0**

Using either the keyboard or ENTRY keys on the front panel, enter the value of C0.

Equivalent SCPI Command

Syntax

:CALCulate{[1]-4}:EPARameters:CIRCuit:E:C0 <numeric>

:CALCulate{[1]-4}:EPARameters:CIRCuit:E:C0?

Query Response

<numeric><newline><^END>

Example of use

```
10 OUTPUT 717;":CALC1:EPAR:CIRC:E:C0 0"
20 OUTPUT 717;":CALC1:EPAR:CIRC:E:C0?"
30 ENTER 717;A
```

SCPI.CALCulate(*Ch*).EPARameters.CIRCuit.E.C1

Object Type

Property (**Read-Write**)

Syntax

SCPI.CALCulate(*Ch*).EPARameters.CIRCuit.E.C1 = *Value*

Value = SCPI.CALCulate(*Ch*).EPARameters.CIRCuit.E.C1

Description

This command sets or gets C1 value of equivalent circuit parameters.

Variable

Parameter	*Value*
Description	C1
Data Type	Double precision floating point type (Double)
Range	-1E ~ 1E
Preset Value	0
Unit	Farad (F)
Resolution	1E-21 (1z)

Examples

```
Dim Var as Double
Var= 0
SCPI.CALCulate(1).EPARameters.CIRCuit.E.C1 = Var
Var = SCPI.CALCulate(1).EPARameters.CIRCuit.E.C1
```

Related Objects

SCPI.CALCulate(*Ch*).EPARameters.CIRCuit.E.R1

SCPI.CALCulate(*Ch*).EPARameters.CIRCuit.E.L1

SCPI.CALCulate(Ch).EPARameters.CIRCuit.E.C0

Equivalent Key

Analysis > **Equivalent Circuit** > **Select Circuit** > **E**

Equivalent Circuit > **C1**

Using either the keyboard or ENTRY keys on the front panel, enter the value of C1.

Equivalent SCPI Command

Syntax

:CALCulate{[1]-4}:EPARameters:CIRCuit:E:C1 <numeric>

:CALCulate{[1]-4}:EPARameters:CIRCuit:E:C1?

Query Response

<numeric><newline><^END>

Example of use

```
10 OUTPUT 717;":CALC1:EPAR:CIRC:E:C1 0"
20 OUTPUT 717;":CALC1:EPAR:CIRC:E:C1?"
30 ENTER 717;A
```

SCPI.CALCulate(*Ch*).EPARameters.CIRCuit.E.L1

Object Type

Property (**Read-Write**)

Syntax

SCPI.CALCulate(*Ch*).EPARameters.CIRCuit.E.L1 = *Value*

Value = SCPI.CALCulate(*Ch*).EPARameters.CIRCuit.E.L1

Description

This command sets or gets L1 value of equivalent circuit parameters.

Variable

Parameter	*Value*
Description	L1
Data Type	Double precision floating point type (Double)
Range	-1E ~ 1E
Preset Value	0
Unit	Henry (H)
Resolution	1E-18 (1a)

Examples

```
Dim Var as Double
Var= 0
SCPI.CALCulate(1).EPARameters.CIRCuit.E.L1 = Var
Var = SCPI.CALCulate(1).EPARameters.CIRCuit.E.L1
```

Related Objects

SCPI.CALCulate(*Ch*).EPARameters.CIRCuit.E.R1

SCPI.CALCulate(*Ch*).EPARameters.CIRCuit.E.C1

SCPI.CALCulate(Ch).EPARameters.CIRCuit.E.C0

Equivalent Key

Analysis > **Equivalent Circuit** > **Select Circuit** > **E**

Equivalent Circuit > **L1**

Using either the keyboard or ENTRY keys on the front panel, enter the value of L1.

Equivalent SCPI Command

Syntax

:CALCulate{[1]-4}:EPARameters:CIRCuit:E:L1 <numeric>

:CALCulate{[1]-4}:EPARameters:CIRCuit:E:L1?

Query Response

<numeric><newline><^END>

Example of use

```
10 OUTPUT 717;":CALC1:EPAR:CIRC:E:L1 0"
20 OUTPUT 717;":CALC1:EPAR:CIRC:E:L1?"
30 ENTER 717;A
```

SCPI.CALCulate(*Ch*).EPARameters.CIRCuit.E.R1

Object Type

Property (**Read-Write**)

Syntax

SCPI.CALCulate(*Ch*).EPARameters.CIRCuit.E.R1 = *Value*

Value = SCPI.CALCulate(*Ch*).EPARameters.CIRCuit.E.R1

Description

This command sets and gets R1 value of equivalent circuit parameters.

Variable

Parameter	*Value*
Description	R1
Data Type	Double precision floating point type (Double)
Range	-1E ~ 1E
Preset Value	0
Unit	Ohm (Ω)
Resolution	1E-18 (1a)

Examples

```
Dim Var as Double
Var= 0
SCPI.CALCulate(1).EPARameters.CIRCuit.E.R1 = Var
Var = SCPI.CALCulate(1).EPARameters.CIRCuit.E.R1
```

Related Objects

SCPI.CALCulate(*Ch*).EPARameters.CIRCuit.E.C1

SCPI.CALCulate(*Ch*).EPARameters.CIRCuit.E.L1

SCPI.CALCulate(Ch).EPARameters.CIRCuit.E.C0

Equivalent Key

Analysis > **Equivalent Circuit** > **Select Circuit** > **E**

Equivalent Circuit > **R1**

Using either the keyboard or ENTRY keys on the front panel, enter the value of R1.

Equivalent SCPI Command

Syntax

:CALCulate{[1]-4}:EPARameters:CIRCuit:E:R1 <numeric>

:CALCulate{[1]-4}:EPARameters:CIRCuit:E:R1?

Query Response

<numeric><newline><^END>

Example of use

```
10 OUTPUT 717;":CALC1:EPAR:CIRC:E:R1 0"
20 OUTPUT 717;":CALC1:EPAR:CIRC:E:R1?"
30 ENTER 717;A
```

SCPI.CALCulate(*Ch*).EPARameters.CIRCuit.TYPE

Object Type

Property (**Read-Write**)

Syntax

SCPI.CALCulate(*Ch*).EPARameters.CIRCuit.TYPE = *Value*

Value = SCPI.CALCulate(*Ch*).EPARameters.CIRCuit.TYPE

Description

This command sets or gets Select Eqv Ckt Type.

Variable

Parameter	*Value*
Description	A|B|C|D|E
Data Type	Character string type (String)
Range	A|B|C|D|E
Preset Value	A
Unit	-
Resolution	-

Examples

```
Dim Var as String
Var= "A"
SCPI.CALCulate(1).EPARameters.CIRCuit.TYPE = Var
Var = SCPI.CALCulate(1).EPARameters.CIRCuit.TYPE
```

Equivalent Key

Analysis > **Equivalent Circuit** > **Select Circuit** > **A** or

Analysis > **Equivalent Circuit** > **Select Circuit** > **B** or

Analysis > **Equivalent Circuit** > **Select Circuit** > **C** or

Analysis > **Equivalent Circuit** > **Select Circuit** > **D** or

Analysis > **Equivalent Circuit** > **Select Circuit** > **E**

Equivalent SCPI Command

Syntax

:CALCulate{[1]-4}:EPARameters:CIRCuit[:TYPE] {A|B|C|D|E}

:CALCulate{[1]-4}:EPARameters:CIRCuit[:TYPE]?

Query Response

{A|B|C|D|E}<newline><^END>

Example of use

```
10 OUTPUT 717;":CALC1:EPAR:CIRC A"
20 OUTPUT 717;":CALC1:EPAR:CIRC?"
30 ENTER 717;A$
```

SCPI.CALCulate(*Ch*).EPARameters.DISPlay.STATe

Object Type

Property (**Read-Write**)

Syntax

SCPI.CALCulate(*Ch*).EPARameters.DISPlay.STATe = *Value*

Value = SCPI.CALCulate(*Ch*).EPARameters.DISPlay.STATe

Description

This command displays the equivalent circuit parameter values.

Variable

Parameter	*Status*
Description	ON|OFF of the display of the equivalent circuit parameter
Data Type	Boolean type (Boolean)
Range	Select from the following: • True or ON: Turns ON the display of the equivalent circuit parameter. • False or OFF: Turns OFF the display of the equivalent circuit parameter.
Preset Value	False or OFF
Unit	-
Resolution	-

Examples

```
Dim Var as Boolean
SCPI.CALCulate(1).EPARameters.DISPlay.STATe = True
Var = SCPI.CALCulate(1).EPARameters.DISPlay.STATe
```

Equivalent Key

Analysis > **Equivalent Circuit** > **Display**

Equivalent SCPI Command

Syntax

:CALCulate{[1]-4}:EPARameters:DISPlay[:STATe] {ON|OFF|1|0}

:CALCulate{[1]-4}:EPARameters:DISPlay[:STATe]?

Query Response

{1|0}<newline><^END>

Example of use

```
10 OUTPUT 717;":CALCulate1:EPAR:DISP ON"
20 OUTPUT 717;":CALCulate1:EPAR:DISP?"
30 ENTER 717;A
```

SCPI.CALCulate(*Ch*).EPARameters.EXECute

Object Type

Method (**Write Only**)

Syntax

SCPI.CALCulate(*Ch*).EPARameters.EXECute

Description

This command executes the equivalent circuit analysis in the selected equivalent circuit model in the partial search range.

Examples

SCPI.CALCulate(1).EPARameters.EXECute

Equivalent Key

Analysis > **Equivalent Circuit** > **Calculate**

Equivalent SCPI Command

Syntax

:CALCulate{[1]-4}:EPARameters:EXECute

Example of use

10 OUTPUT 717;":CALC1:EPAR:EXEC"

SCPI.CALCulate(*Ch*).EPARameters.SIMulate.AUTO

Object Type

Property (**Read-Write**)

Syntax

SCPI.CALCulate(*Ch*).EPARameters.SIMulate.AUTO = *Value*

Value = SCPI.CALCulate(*Ch*).EPARameters.SIMulate.AUTO

Description

This command executes the equivalent circuit analysis simulation function automatically when parameters are changed. When this is set at ON, if one of the parameter is changed or set, the simulation calculation is done automatically. If its set at OFF, the simulation calculation is not executed but the simulated Memory Trace is still shown.

Variable

Parameter	*Status*
Description	ON|OFF of the equivalent circuit analysis simulation function
Data Type	Boolean type (Boolean)
Range	Select from the following: • True or ON: Turns ON the equivalent circuit analysis simulation function. • False or OFF: Turns OFF the equivalent circuit analysis simulation function.
Preset Value	False or OFF
Unit	-
Resolution	-

Examples

```
Dim Var as Boolean
Var= True
SCPI.CALCulate(1).EPARameters.SIMulate.AUTO = Var
Var = SCPI.CALCulate(1).EPARameters.SIMulate.AUTO
```

Related Objects

SCPI.CALCulate(*Ch*).EPARameters.SIMulate.IMMediate

Equivalent Key

There is no equivalent key is available on the front panel. However, the similar key is:

Analysis > **Equivalent Circuit** > **Simulate** > **ON|OFF**

When this softkey is turned ON, its equivalent to:

SCPI.CALCulate[1-4].EPARameters.SIMulate.AUTO ON +
SCPI.CALCulate(*Ch*).EPARameters.SIMulate.IMMediate

When this softkey is turned OFF, its equivalent to:

SCPI.CALCulate[1-4].EPARameters.SIMulate.AUTO OFF

Equivalent SCPI Command

Syntax

:CALCulate{[1]-4}:EPARameters:SIMulate:AUTO {ON|OFF|1|0}

:CALCulate{[1]-4}:EPARameters:SIMulate:AUTO?

Query Response

{1|0}<newline><^END>

Example of use

```
10 OUTPUT 717;":CALC1:EPAR:SIM:AUTO ON"
20 OUTPUT 717;":CALC1:EPAR:SIM:AUTO?"
30 ENTER 717;A
```

SCPI.CALCulate(*Ch*).EPARameters.SIMulate.IMMediate

Object Type

Method (**Write Only**)

Syntax

SCPI.CALCulate(*Ch*).EPARameters.SIMulate.IMMediate

Description

This command writes the simulation results into the memory trace, then set "Data & Mem" for the display setting. All traces related with impedance measurement are updated, even if CALCulate [1-4]: EPARameters: SIMulate: AUTO is set at OFF.

Examples

SCPI.CALCulate(1).EPARameters.SIMulate.IMMediate

Related Objects

SCPI.CALCulate(*Ch*).EPARameters.SIMulate.AUTO

Equivalent Key

There is no equivalent key is available on the front panel. However, the similar key is:

Analysis > **Equivalent Circuit** > **Simulate** > **ON|OFF**

When this softkey is turned ON, its equivalent to:

SCPI.CALCulate[1-4].EPARameters.SIMulate.AUTO ON +
SCPI.CALCulate(*Ch*).EPARameters.SIMulate.IMMediate

When this softkey is turned OFF, its equivalent to:

SCPI.CALCulate[1-4].EPARameters.SIMulate.AUTO OFF

Equivalent SCPI Command

Syntax

:CALCulate{[1]-4}:EPARameters:SIMulate[:IMMediate]

Example of use

10 OUTPUT 717;":CALC1:EPAR:SIM"

SCPI.CALCulate(*Ch*).PARameter(*Tr*).DEFine

Object type

Property (**Read-Write**)

Syntax

SCPI.CALCulate(*Ch*).PARameter(*Tr*).DEFine = *Param*

Param = SCPI.CALCulate(*Ch*).PARameter(*Tr*).DEFine

Description

This command sets/gets the measurement parameter of the selected trace (*Tr*), for the selected channel (*Ch*).

Variable

Parameter	*Param*
Description	Measurement parameter
Data Type	Character string type (String)
Range	Select either one of the following: •S-Parameter: o S11|S21|S12|S22 o A|B|R1|R2 o Gain-Phase o TR|T|R o Z (Impedance Measurement)
Preset Value	"S11"

Examples

```
Dim MeasPara As String
SCPI.CALCulate(1).PARameter(1).DEFine = "s21"
MeasPara = SCPI.CALCulate(1).PARameter(1).DEFine
```

Related Objects

SCPI.CALCulate(Ch).SELected.ZPARameter.DEFine

SCPI.CALCulate(Ch).SELected.FORMat

Equivalent key

Meas > **S<XY>** {X=1-2;Y=1-2}

Meas > **Absolute** > **R1|R2|A|B**

Meas > **Gain-Phase** > **T/R|R|T**

Meas > **Impedance Analysis Menu**

Equivalent SCPI command

Syntax

:CALCulate{[1]-4}:PARameter{[1]-4}:DEFine {S11|S21|S12|S22|A|B|R1|R2|TR|T|R}

:CALCulate{[1]-4}:PARameter{[1]-4}:DEFine?

Query response

{S11|S21|S12|S22|A|B|R1|R2|TR|T|R}<newline><^END>

Example of use

```
10 OUTPUT 717;":CALC1:PAR1:DEF S21"
20 OUTPUT 717;":CALC1:PAR1:DEF?"
30 ENTER 717;A$
```

SCPI.CALCulate(*Ch*).PARameter(*Tr*).SELect

Object type

Method (**Write only**)

Syntax

SCPI.CALCulate(*Ch*).PARameter(*Tr*).SELect

Description

This command sets the selected trace (*Tr*) of selected channel (*Ch*) to the active trace.

You can set a trace to be displayed only to the active trace. If this object is used to set a trace displayed to the non-active trace, an error occurs during execution and the object is ignored. (No read)

Variable

None

Examples

```
SCPI.CALCulate(2).PARameter(2).SELect
```

Related objects

SCPI.CALCulate(Ch).SELected.CORRection.EDELay.TIME

SCPI.DISPlay.WINDow(Ch).ACTivate

SCPI.SENSe(Ch).CORRection.EXTension.AUTO.DCOFfset

Equivalent key

Trace Prev

Trace Next

Equivalent SCPI command

Syntax

:CALCulate{[1]-4}:PARameter{[1]-4}:SELect

Example of use

```
10 OUTPUT 717;":CALC2:PAR2:SEL"
```

SCPI.CALCulate(*Ch*).PARameter(*Tr*).SPORt

Type of object

Property (**Read-Write**)

Syntax

SCPI.CALCulate(*Ch*).PARameter(*Tr*).SPORt = *Value*

Value = SCPI.CALCulate(*Ch*).PARameter(*Tr*).SPORt

Description

This command sets/gets the output port used for absolute for the selected trace (*Tr*) of the selected channel (*Ch*).

Variable

Parameter	*Value*
Description	Setting of the output port
Data type	Long integer type (Long)
Range	1 to 2
Preset value	1
Note	You need to set the measurement parameter for absolute measurements with the SCPI.CALCulate(Ch).PARameter(Tr).DEFine command.

Example of use

```
Dim Sport As Long
SCPI.CALCulate(1).PARameter(1).DEFine = "B"
SCPI.CALCulate(1).PARameter(1).SPORt = 2
Sport = SCPI.CALCulate(1).PARameter(1).SPORt
```

Related objects

SCPI.CALCulate(Ch).PARameter(Tr).DEFine

Equivalent key

Meas > **Absolute** > **A(x)** , **B(x)**, **R1(x)**, **R2(x)** (x: 1 to 2)

Equivalent SCPI command

Syntax

:CALCulate{[1]-4}:PARameter{[1]-4}:SPORt <numeric>

:CALCulate{[1]-4}:PARameter{[1]-4}:SPORt?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":CALC1:PAR1:DEF B"
20 OUTPUT 717;":CALC1:PAR1:SPOR 2"
30 OUTPUT 717;":CALC1:PAR1:SPOR?"
30 ENTER 717;A
```

SCPI.CALCulate(*Ch*).PARameter.COUNt

Object type

Property (**Read-Write**)

Syntax

SCPI.CALCulate(*Ch*).PARameter.COUNt = *Value*

Value = SCPI.CALCulate(*Ch*).PARameter.COUNt

Description

This command sets/gets the number of traces of selected channel (*Ch*).

Variable

Parameter	*Value*
Description	Number of traces
Data type	Long integer type (Long)
Range	1 to 8, Varies depending on the upper limit setting for the channel/trace number
Preset value	1
Note	If the specified variable 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.

Examples

```
Dim TraceNum As Long
SCPI.CALCulate(1).PARameter.COUNt = 4
TraceNum = SCPI.CALCulate(1).PARameter.COUNt
```

Equivalent key

Display > **Num of Traces**

Equivalent SCPI command

Syntax

:CALCulate{[1]-4}:PARameter:COUNt <numeric>

:CALCulate{[1]-4}:PARameter:COUNt?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":CALC1:PAR:COUN 4"
20 OUTPUT 717;":CALC1:PAR:COUN?"
30 ENTER 717;A
```

SCPI.CALCulate(*Ch*).SELected.BLIMit.DB

Object type

Property (**Read-Write**)

Syntax

SCPI.CALCulate(*Ch*).SELected.BLIMit.DB = Value

Value = SCPI.CALCulate(*Ch*).SELected.BLIMit.DB

Description

This command sets/gets the bandwidth threshold value (attenuation from the peak) of the bandwidth test, for the selected channel (specified with the SCPI.CALCulate(Ch).PARameter(Tr).SELect command).

Variable

Parameter	*Value*
Description	Bandwidth N dB points.
Data type	Double precision floating point type (Double)
Range	0 to 5E8
Preset value	3
Unit	Varies depending on the selected data format. • Log magnitude (MLOG): dB (decibel) • Phase (PHAS), Expanded phase (UPH) or Positive phase (PPH): ° (degree) • Group delay (GDEL): s (second) • Others: No unit

Examples

```
Dim BLimDB As Double
SCPI.CALCulate(1).SELected.BLIMit.DB = 3
BLimDB = SCPI.CALCulate(1).SELected.BLIMit.DB
```

Related objects

SCPI.CALCulate(Ch).PARameter(Tr).SELect

SCPI.CALCulate(Ch).SELected.BLIMit.STATe

Equivalent key

Analysis > Bandwidth Limit > N dB Points

Equivalent SCPI command

Syntax

:CALCulate{[1]-4}[:SELected]:BLIMit:DB <numeric>

:CALCulate{[1]-4}[:SELected]:BLIMit:DB?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":CALC1:BLIM:DB 3"
20 OUTPUT 717;":CALC1:BLIM:DB?"
30 ENTER 717;A
```

SCPI.CALCulate(*Ch*).SELected.BLIMit.DISPlay.MARKer

Object type

Property (**Read-Write**)

Syntax

SCPI.CALCulate(*Ch*).SELected.BLIMit.DISPlay.MARKer = Status

Status = SCPI.CALCulate(*Ch*).SELected.BLIMit.DISPlay.MARKer

Description

This command turns ON/OFF the marker display of the bandwidth test, for the active trace of selected channel (specified with the SCPI.CALCulate(Ch).PARameter(Tr).SELect command).

Variable

Parameter	*Status*
Description	ON/OFF of the bandwidth marker.
Data type	Boolean type (Boolean)
Range	Select from the following. • True or ON: Turns ON the bandwidth marker. • False or OFF: Turns OFF the bandwidth marker.
Preset value	False or OFF

Examples

```
Dim BLimMk As Boolean
SCPI.CALCulate(1).PARameter(1).SELect
SCPI.CALCulate(1).SELected.BLIMit.DISPlay.MARKer = True
BLimMk = SCPI.CALCulate(1).SELected.BLIMit.DISPlay.MARKer
```

Related objects

SCPI.CALCulate(Ch).PARameter(Tr).SELect

SCPI.CALCulate(Ch).SELected.BLIMit.STATe

SCPI.CALCulate(Ch).SELected.BLIMit.DISPlay.VALue

Equivalent key

Analysis > **Bandwidth Limit** > **BW Marker**

Equivalent SCPI command

Syntax

:CALCulate{[1]-4}[:SELected]:BLIMit:DISPlay:MARKer {ON|OFF|1|0}

:CALCulate{[1]-4}[:SELected]:BLIMit:DISPlay:MARKer?

Query response

{1|0}<newline><^END>

Example of use

```
10 OUTPUT 717;":CALC1:BLIM:DISP:MARK ON"
20 OUTPUT 717;":CALC1:BLIM:DISP:MARK?"
30 ENTER 717;A
```

SCPI.CALCulate(*Ch*).SELected.BLIMit.DISPlay.VALue

Object type

Property (**Read-Write**)

Syntax

SCPI.CALCulate(*Ch*).SELected.BLIMit.DISPlay.VALue = Status

Status = SCPI.CALCulate(*Ch*).SELected.BLIMit.DISPlay.VALue

Description

This command turns ON/OFF the bandwidth value display of the bandwidth test, for the active trace of selected channel (specified with the SCPI.CALCulate(Ch).PARameter(Tr).SELect command).

Variable

Parameter	*Status*
Description	ON/OFF of the bandwidth display of the bandwidth test.
Data type	Boolean type (Boolean)
Range	Select from the following. • True or ON: Turns ON the bandwidth display. • False or OFF: Turns OFF the bandwidth display.
Preset value	False or OFF

Examples

```
Dim BLimVal As Boolean
SCPI.CALCulate(1).PARameter(1).SELectSCPI.CALCulate(1).SELected.BLIMit.DISPlay.VALue = True
BLimVal = SCPI.CALCulate(1).SELected.BLIMit.DISPlay.VALue
```

Related objects

SCPI.CALCulate(Ch).PARameter(Tr).SELect

SCPI.CALCulate(Ch).SELected.BLIMit.STATe

SCPI.CALCulate(Ch).SELected.BLIMit.DISPlay.MARKer

Equivalent key

Analysis > **Bandwidth Limit** > **BW Display**

Equivalent SCPI command

Syntax

:CALCulate{[1]-4}[:SELected]:BLIMit:DISPlay:VALue {ON|OFF|1|0}

:CALCulate{[1]-4}[:SELected]:BLIMit:DISPlay:VALue?

Query response

{1|0}<newline><^END>

Example of use

```
10 OUTPUT 717;":CALC1:BLIM:DISP:VAL ON"
20 OUTPUT 717;":CALC1:BLIM:DISP:VAL?"
30 ENTER 717;A
```

SCPI.CALCulate(*Ch*).SELected.BLIMit.FAIL

Object type

Property (**Read Only**)

Syntax

Status = SCPI.CALCulate(*Ch*).SELected.BLIMit.FAIL

Description

This command gets the bandwidth limit test results, for the active trace of the selected channel (specified with the SCPI.CALCulate(Ch).PARameter(Tr).SELect command).

Variable

Parameter	*Status*
Description	The bandwidth limit test result
Data type	Long integer type (Long)
Range	Returns from the following. • 0: The bandwidth limit test result is PASS. • 1: The bandwidth limit test result is FAIL of wide. • -1: The bandwidth limit test result is FAIL of narrow.
Note	When the bandwidth limit test is set to OFF, the 0 is always read out.

Examples

```
Dim Result As Integer
SCPI.CALCulate(1).PARameter(1).SELect
SCPI.CALCulate(1).SELected.BLIMit.STATe = 1
Result = SCPI.CALCulate(1).SELected.BLIMit.FAIL
```

Related objects

SCPI.CALCulate(Ch).PARameter(Tr).SELect

SCPI.CALCulate(Ch).SELected.BLIMit.STATe

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

:CALCulate{[1]-4}[:SELected]:BLIMit:FAIL?

Query response

{0|1|-1}<newline><^END>

(0: Pass, 1: Fail (Wide), -1:Fail (Narrow), When the bandwidth limit test is set to OFF, 0 is always read out.)

Example of use

```
10 OUTPUT 717;":CALC1:BLIM:FAIL?"
20 ENTER 717;A
```

SCPI.CALCulate(*Ch*).SELected.BLIMit.MAXimum

Object type

Property (**Read-Write**)

Syntax

SCPI.CALCulate(*Ch*).SELected.BLIMit.MAXimum = Value

Value = SCPI.CALCulate(*Ch*).SELected.BLIMit.MAXimum

Description

This command sets/gets the upper limit value of the bandwidth test, for the selected channel (specified with the SCPI.CALCulate(Ch).PARameter(Tr).SELect command).

Variable

Parameter	*Value*
Description	Maximum bandwidth
Data type	Double precision floating point type (Double)
Range	0 to 1E12
Preset value	3E5
Unit	Hz (hertz), dBm or second

Examples

```
Dim BLimMax As Double
SCPI.CALCulate(1).SELected.BLIMit.MAXimum = 1E9
BLimMax = SCPI.CALCulate(1).SELected.BLIMit.MAXimum
```

Related objects

SCPI.CALCulate(Ch).PARameter(Tr).SELect

SCPI.CALCulate(Ch).SELected.BLIMit.STATe

SCPI.CALCulate(Ch).SELected.BLIMit.MINimum

Equivalent key

Analysis > **Bandwidth Limit** > **Max Bandwidth**

Equivalent SCPI command

Syntax

:CALCulate{[1]-4}[:SELected]:BLIMit:MAXimum <numeric>

:CALCulate{[1]-4}[:SELected]:BLIMit:MAXimum?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":CALC1:BLIM:MAX 3E5"
20 OUTPUT 717;":CALC1:BLIM:MAX?"
30 ENTER 717;A
```

SCPI.CALCulate(*Ch*).SELected.BLIMit.MINimum

Object type

Property (**Read-Write**)

Syntax

SCPI.CALCulate(*Ch*).SELected.BLIMit.MINimum = Value

Value = SCPI.CALCulate(*Ch*).SELected.BLIMit.MINimum

Description

This command sets/gets the lower limit value of the bandwidth test, for the selected channel (specified with the SCPI.CALCulate(Ch).PARameter(Tr).SELect command).

Variable

Parameter	*Value*
Description	Minimum bandwidth
Data type	Double precision floating point type (Double)
Range	0 to 1E12
Preset value	1E4
Unit	Hz (hertz), dBm or second

Examples

```
Dim BLimMin As Double
SCPI.CALCulate(1).SELected.BLIMit.MINimum = 1E6
BLimMin = SCPI.CALCulate(1).SELected.BLIMit.MINimum
```

Related objects

SCPI.CALCulate(Ch).PARameter(Tr).SELect

SCPI.CALCulate(Ch).SELected.BLIMit.STATe

SCPI.CALCulate(Ch).SELected.BLIMit.MAXimum

Equivalent key

Analysis > **Bandwidth Limit** > **Min Bandwidth**

Equivalent SCPI command

Syntax

:CALCulate{[1]-4}[:SELected]:BLIMit:MINimum <numeric>

:CALCulate{[1]-4}[:SELected]:BLIMit:MINimum?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":CALC1:BLIM:MIN 1E4"
20 OUTPUT 717;":CALC1:BLIM:MIN?"
30 ENTER 717;A
```

SCPI.CALCulate(*Ch*).SELected.BLIMit.REPort.DATA

Object type

Property (**Read Only**)

Syntax

Data = SCPI.CALCulate(*Ch*).SELected.BLIMit.REPort.DATA

Description

This command reads the bandwidth value of the bandwidth test, for the active trace of selected channel (specified with the SCPI.CALCulate(Ch).PARameter(Tr).SELect command).

Variable

Parameter	*Data*
Description	The bandwidth value of the bandwidth
Data type	Double precision floating point type (Double)

Examples

```
Dim BWData As Double
SCPI.CALCulate(1).PARameter(1).SELect
BWData = SCPI.CALCulate(1).SELected.BLIMit.REPort.DATA
```

Related objects

SCPI.CALCulate(Ch).PARameter(Tr).SELect

SCPI.CALCulate(Ch).SELected.BLIMit.STATe

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

:CALCulate{[1]-4}[:SELected]:BLIMit:REPort[:DATA]?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":CALC1:BLIM:REP?"
20 ENTER 717;A
```

SCPI.CALCulate(*Ch*).SELected.BLIMit.STATe

Object type

Property (**Read-Write**)

Syntax

SCPI.CALCulate(*Ch*).SELected.BLIMit.STATe = *Status*

Status = SCPI.CALCulate(*Ch*).SELected.BLIMit.STATe

Description

This command turns ON/OFF the bandwidth test function, for the active trace of selected channel (specified with the SCPI.CALCulate(Ch).PARameter(Tr).SELect command).

Variable

Parameter	*Status*
Description	ON/OFF the bandwidth test function.
Data type	Boolean type (Boolean)
Range	Select from the following: • True or ON: Turns ON the bandwidth test function. • False or OFF: Turns OFF the bandwidth test function.
Preset value	False or OFF

Examples

```
Dim BLimTest As Boolean
SCPI.CALCulate(1).PARameter(1).SELect
SCPI.CALCulate(1).SELected.BLIMit.STATe = True
BLimTest = SCPI.CALCulate(1).SELected.BLIMit.STATe
```

Related objects

SCPI.CALCulate(Ch).PARameter(Tr).SELect

SCPI.CALCulate(Ch).SELected.BLIMit.DB

SCPI.CALCulate(Ch).SELected.BLIMit.DISPlay.MARKer

SCPI.CALCulate(Ch).SELected.BLIMit.DISPlay.VALue

SCPI.CALCulate(Ch).SELected.BLIMit.FAIL

SCPI.CALCulate(Ch).SELected.BLIMit.MAXimum

SCPI.CALCulate(Ch).SELected.BLIMit.MINimum

SCPI.CALCulate(Ch).SELected.BLIMit.REPort.DATA

Equivalent key

Analysis > **Bandwidth Limit** > **BW Test**

Equivalent SCPI command

Syntax

:CALCulate{[1]-4}[:SELected]:BLIMit[:STATe] {ON|OFF|1|0}

:CALCulate{[1]-4}[:SELected]:BLIMit[:STATe]?

Query response

{1|0}<newline><^END>

Example of use

```
10 OUTPUT 717;":CALC1:BLIM ON"
20 OUTPUT 717;":CALC1:BLIM?"
30 ENTER 717;A
```

SCPI.CALCulate(*Ch*).SELected.CONVersion.FUNCtion

Object type

Property (**Read-Write**)

Syntax

SCPI.CALCulate(*Ch*).SELected.CONVersion.FUNCtion = *Param*

Param = SCPI.CALCulate(*Ch*).SELected.CONVersion.FUNCtion

Description

This command sets/gets the parameter after conversion using the parameter conversion function, for the active trace of selected channel (*Ch*).

Variable

Parameter	*Param*
Description	The parameter after conversion
Data type	Character string type (String)
Range	Select from the following: • "ZREFlection": Specifies the equivalent impedance in reflection measurement. • "ZTRansmit": Specifies the equivalent impedance (series) in transmission measurement. • "YREFlection": Specifies the equivalent admittance in reflection measurement. • "YTRansmit": Specifies the equivalent admittance (series) in transmission measurement. • "INVersion": Specifies the inverse S-parameter. • "ZTSHunt": Specifies the equivalent impedance (shunt) in transmission measurement. • "YTSHunt": Specifies the equivalent admittance (shunt) in transmission measurement. • "CONJugation": Specifies the conjugate.
Preset value	"ZREFlection"

Examples

```
Dim Func As String
SCPI.CALCulate(1).PARameter(1).SELect
SCPI.CALCulate(1).SELected.CONVersion.FUNCtion = "ztr"
Func = SCPI.CALCulate(1).SELected.CONVersion.FUNCtion
```

Related objects

SCPI.CALCulate(Ch).SELected.CONVersion.STATe

SCPI.CALCulate(Ch).PARameter(Tr).SELect

SCPI.SENSe(Ch).CORRection.EXTension.AUTO.DCOFfset

Equivalent key

Analysis > **Conversion** > **Function** > **Z:Reflection|Z:Transmission|Y:Reflection|Y:Transmission|1/S| Z:Trans-Shunt|Y:Trans-Shunt|Conjugation**

Equivalent SCPI command

Syntax

:CALCulate{[1]-4}[:SELected]:CONVersion:FUNCtion {ZREFlection|ZTRansmit|YREFlection|YTRansmit|INVersion|ZTSHunt|YTSHunt|CONJugation}

:CALCulate{[1]-4}[:SELected]:CONVersion:FUNCtion?

Query response

{ZREF|ZTR|YREF|YTR|INV|ZTSH|YTSH|CONJ}<newline><^END>

Example of use

```
10 OUTPUT 717;":CALC1:CONV:FUNC ZTR"
20 OUTPUT 717;":CALC1:CONV:FUNC?"
30 ENTER 717;A$
```

SCPI.CALCulate(*Ch*).SELected.CONVersion.STATe

Object type

Property (**Read-Write**)

Syntax

SCPI.CALCulate(*Ch*).SELected.CONVersion.STATe = *Status*

Status = SCPI.CALCulate(*Ch*).SELected.CONVersion.STATe

Description

This command turns ON/OFF the parameter conversion function, for the active trace of selected channel (*Ch*).

Variable

Parameter	*Status*
Description	ON/OFF of the parameter conversion function
Data type	Boolean type (Boolean)
Range	Select from the following: • True or ON: Turns ON the parameter conversion function. • False or OFF: Turns OFF the parameter conversion function.
Preset value	False or OFF

Examples

```
Dim Conv As Boolean
SCPI.CALCulate(1).PARameter(1).SELect
SCPI.CALCulate(1).SELected.CONVersion.STATe = True
Conv = SCPI.CALCulate(1).SELected.CONVersion.STATe
```

Related objects

SCPI.CALCulate(Ch).SELected.CONVersion.FUNCtion

SCPI.CALCulate(Ch).PARameter(Tr).SELect

Equivalent key

Analysis > **Conversion** > **Conversion**

Equivalent SCPI command

Syntax

:CALCulate{[1]-4}[:SELected]:CONVersion[:STATe] {ON|OFF|1|0}

:CALCulate{[1]-4}[:SELected]:CONVersion[:STATe]?

Query response

{1|0}<newline><^END>

Example of use

```
10 OUTPUT 717;":CALC1:CONV ON"
20 OUTPUT 717;":CALC1:CONV?"
30 ENTER 717;A
```

SCPI.CALCulate(*Ch*).SELected.CORRection.EDELay.TIME

Object type

Property (**Read-Write**)

Syntax

SCPI.CALCulate(*Ch*).SELected.CORRection.EDELay.TIME = *Value*

Value = SCPI.CALCulate(*Ch*).SELected.CORRection.EDELay.TIME

Description

This command sets/gets the electrical delay time of the active trace of the selected channel (*Ch*).

Variable

Parameter	*Value*
Description	Electrical delay time
Data type	Double precision floating point type (Double)
Range	-10 to 10
Preset value	0
Unit	s (second)
Note	If the specified variable 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.

Examples

```
Dim Edel As Double
SCPI.CALCulate(1).PARameter(1).SELect
SCPI.CALCulate(1).SELected.CORRection.EDELay.TIME = 0.2
Edel = SCPI.CALCulate(1).SELected.CORRection.EDELay.TIME
```

Related objects

SCPI.CALCulate(Ch).PARameter(Tr).SELect

Equivalent key

Scale > **Electrical Delay** > **Electrical Delay**

Equivalent SCPI command

Syntax

:CALCulate{[1]-4}[:SELected]:CORRection:EDELay:TIME <numeric>

:CALCulate{[1]-4}[:SELected]:CORRection:EDELay:TIME?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":CALC1:CORR:EDEL:TIME 0.2"
20 OUTPUT 717;":CALC1:CORR:EDEL:TIME?"
30 ENTER 717;A
```

SCPI.CALCulate(*Ch*).SELected.CORRection.OFFSet.PHASe

Object type

Property (**Read-Write**)

Syntax

SCPI.CALCulate(*Ch*).SELected.CORRection.OFFSet.PHASe = *Value*

Value = SCPI.CALCulate(*Ch*).SELected.CORRection.OFFSet.PHASe

Description

This command sets/gets the phase offset of the active trace of selected channel (*Ch*).

Variable

Parameter	*Value*
Description	Phase offset
Data type	Double precision floating point type (Double)
Range	-360 to 360
Preset value	0
Unit	° (degree)
Note	If the specified variable 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.

Examples

```
Dim Offset As Double
SCPI.CALCulate(2).PARameter(1).SELect
SCPI.CALCulate(2).SELected.CORRection.OFFSet.PHASe = 2.5
Offset = SCPI.CALCulate(2).SELected.CORRection.OFFSet.PHASe
```

Related objects

SCPI.CALCulate(Ch).PARameter(Tr).SELect

Equivalent key

Scale > **Phase Offset**

Equivalent SCPI command

Syntax

:CALCulate{[1]-4}[:SELected]:CORRection:OFFSet:PHASe <numeric>

:CALCulate{[1]-4}[:SELected]:CORRection:OFFSet:PHASe?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":CALC1:CORR:OFFS:PHAS 2.5"
20 OUTPUT 717;":CALC1:CORR:OFFS:PHAS?"
30 ENTER 717;A
```

SCPI.CALCulate(*Ch*).SELected.DATA.FDATa

Object type

Property (**Read-Write**)

Syntax

SCPI.CALCulate(*Ch*).SELected.DATA.FDATa = *Data*

Data = SCPI.CALCulate(*Ch*).SELected.DATA.FDATa

Description

This command sets/gets the formatted data array, for the active trace of selected channel (*Ch*).
The array data element varies in the data format (specified with the SCPI.CALCulate(Ch).SELected.FORMat object). For more information on the formatted data array, see Internal Data Processing.

NOTE If valid data is not calculated because of the invalid measurement, "1.#QNB" is read out.

Variable

Parameter	*Data*
Description	Indicates the array data (formatted data array) of NOP (number of measurement points)×2. Where n is an integer between 1 and NOP. • *Data(n×2-2) :*Data (primary value) at the n-th measurement point. • *Data(n×2-1) :*Data (secondary value) at the n-th measurement point. Always 0 when the data format is not the Smith chart format or the polar format. The index of the array starts from 0.
Data type	Variant type (Variant)
Note	If there is no array data of NOP (number of measurement point))×2 when setting a formatted data array, an error occurs when executed and the object is ignored.

Examples

```
Dim FmtData As Variant
SCPI.SENSe(1).SWEep.POINts = 201
SCPI.CALCulate(1).PARameter(1).SELect
```

```
FmtData = SCPI.CALCulate(1).SELected.DATA.FDATa
SCPI.CALCulate(1).PARameter(2).SELect
SCPI.CALCulate(1).SELected.DATA.FDATa = FmtData
```

Related objects

SCPI.CALCulate(Ch).PARameter(Tr).SELect

SCPI.SENSe(Ch).SWEep.POINts

SCPI.CALCulate(Ch).SELected.FORMat

SCPI.CALCulate(Ch).SELected.DATA.FMEMory

SCPI.CALCulate(Ch).SELected.DATA.SDATa

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

:CALCulate{[1]-4}[:SELected]:DATA:FDATa <numeric1>,… ,<numeric NOP×2>

:CALCulate{[1]-4}[:SELected]:DATA:FDATa?

Query response

{numeric 1},… ,{numeric NOP×2}<newline><^END>

Example of use

```
10 DIM A(1:201,1:2)
20 OUTPUT 717;":CALC1:DATA:FDAT?"
30 ENTER 717;A(*)
```

SCPI.CALCulate(*Ch*).SELected.DATA.FMEMory

Object type

Property (**Read-Write**)

Syntax

SCPI.CALCulate(*Ch*).SELected.DATA.FMEMory = *Data*

Data = SCPI.CALCulate(*Ch*).SELected.DATA.FMEMory

Description

This command sets/gets the formatted memory array, for the active trace of selected channel (*Ch*).

The array data element varies in the data format (specified with the SCPI.CALCulate(Ch).SELected.FORMat object). For more information on the formatted memory array, see Internal Data Processing.

NOTE If valid data is not calculated because of the invalid measurement, “1.#QNB” is read out.

Variable

Parameter	*Data*
Description	Indicates the array data (formatted memory array) of NOP (number of measurement points)×2. Where n is an integer between 1 and NOP. • *Data(n×2-2) :*Data (primary value) at the n-th measurement point. • *Data(n×2-1) :*Data (secondary value) at the n-th measurement point. Always 0 when the data format is not the Smith chart format or the polar format. The index of the array starts from 0.
Data type	Variant type (Variant)
Note	If there is no array data of NOP (number of measurement point))*2 when setting a formatted memory array, an error occurs when executed and the object is ignored.

Examples

```
Dim FmtMem As Variant
SCPI.SENSe(1).SWEep.POINts = 201
```

```
SCPI.CALCulate(1).PARameter(1).SELect
FmtMem = SCPI.CALCulate(1).SELected.DATA.FMEMory
SCPI.CALCulate(1).PARameter(2).SELect
SCPI.CALCulate(1).SELected.DATA.FMEMory = FmtMem
```

Related objects

SCPI.CALCulate(Ch).PARameter(Tr).SELect

SCPI.SENSe(Ch).SWEep.POINts

SCPI.CALCulate(Ch).SELected.FORMat

SCPI.CALCulate(Ch).SELected.DATA.FDATa

SCPI.CALCulate(Ch).SELected.DATA.SMEMory

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

:CALCulate{[1]-4}[:SELected]:DATA:FMEMory <numeric 1>,… ,<numeric NOP×2>

:CALCulate{[1]-4}[:SELected]:DATA:FMEMory?

Query response

{numeric 1},… ,{numeric NOP×2}<newline><^END>

Example of use

```
10 DIM A(1:201,1:2)
20 OUTPUT 717;":CALC1:DATA:FMEM?"
30 ENTER 717;A(*)
```

SCPI.CALCulate(*Ch*).SELected.DATA.SDATa

Object type

Property (**Read-Write**)

Syntax

SCPI.CALCulate(*Ch*).SELected.DATA.SDATa = *Data*

Data = SCPI.CALCulate(*Ch*).SELected.DATA.SDATa

Description

This command sets/gets the corrected data array, for the active trace of selected channel (*Ch*).

For more information on the corrected data array, see Internal Data Processing

NOTE If valid data is not calculated because of the invalid measurement, "1.#QNB" is read out.

Variable

Parameter	*Data*
Description	Indicates the array data (corrected data array) of NOP (number of measurement points)×2. Where n is an integer between 1 and NOP. • *Data(n×2-2) :*Real part of the data (complex number) at the n-th measurement point. • *Data(n×2-1)* :Imaginary part of the data (complex number) at the n-th measurement point. The index of the array starts from 0.
Data type	Variant type (Variant)
Note	If there is no array data of NOP (number of measurement point))× 2 when setting a corrected data array, an error occurs when executed and the object is ignored.

Examples

```
Dim CorData As Variant
SCPI.SENSe(1).SWEep.POINts = 201
CorData = SCPI.CALCulate(1).SELected.DATA.SDATa
SCPI.SENSe(2).SWEep.POINts = 201
SCPI.CALCulate(2).SELected.DATA.SDATa = CorData
```

Related objects

SCPI.CALCulate(Ch).PARameter(Tr).SELect

SCPI.SENSe(Ch).SWEep.POINts

SCPI.CALCulate(Ch).SELected.DATA.SMEMory

SCPI.CALCulate(Ch).SELected.DATA.FDATa

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

:CALCulate{[1]-4}[:SELected]:DATA:SDATa <numeric 1>,… ,<numeric NOP×2>

:CALCulate{[1]-4}[:SELected]:DATA:SDATa?

Query response

<numeric 1>,… ,<numeric NOP×2><^END>

Example of use

```
10 DIM A(1:201,1:2)
20 OUTPUT 717;":CALC1:DATA:SDAT?"
30 ENTER 717;A(*)
```

SCPI.CALCulate(*Ch*).SELected.DATA.SMEMory

Object type

Property (**Read-Write**)

Syntax

SCPI.CALCulate(*Ch*).SELected.DATA.SMEMory = *Data*

Data = SCPI.CALCulate(*Ch*).SELected.DATA.SMEMory

Description

This command sets/gets the corrected memory array, for the active trace of selected channel (*Ch*).

For more information on the corrected memory array, see Section Internal Data Processing.

NOTE If valid data is not calculated because of the invalid measurement, "1.#QNB" is read out.

Variable

Parameter	*Data*
Description	Indicates the array data (corrected memory array) of NOP (number of measurement points)×2. Where n is an integer between 1 and NOP. • *Data(n×2-2)* :Real part of the data (complex number) at the n-th measurement point. • *Data(n×2-1)* :Imaginary part of the data (complex number) at the n-th measurement point. The index of the array starts from 0.
Data type	Variant type (Variant)
Note	If there is no array data of NOP (number of measurement point))×2 when setting a corrected memory array, an error occurs when executed and the object is ignored.

Examples

```
Dim CorMem As Variant
SCPI.SENSe(1).SWEep.POINts = 201
CorMem = SCPI.CALCulate(1).SELected.DATA.SMEMory
SCPI.SENSe(2).SWEep.POINts = 201
SCPI.CALCulate(1).SELected.DATA.SMEMory = CorMem
```

Related objects

SCPI.CALCulate(Ch).PARameter(Tr).SELect

SCPI.SENSe(Ch).SWEep.POINts

SCPI.CALCulate(Ch).SELected.DATA.SDATa

SCPI.CALCulate(Ch).SELected.DATA.FMEMory

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

:CALCulate{[1]-4}[:SELected]:DATA:SMEMory <numeric 1>,… ,<numeric NOP×2>

:CALCulate{[1]-4}[:SELected]:DATA:SMEMory?

Query response

<numeric 1>,… ,<numeric NOP×2><^END>

Example of use

```
10 DIM A(1:201,1:2)
20 OUTPUT 717;":CALC1:DATA:SMEM?"
30 ENTER 717;A(*)
```

SCPI.CALCulate(*Ch*).SELected.DATA.XAXis

Object type

Property (**Read Only**)

Syntax

Data = SCPI.CALCulate(*Ch*).SELected.DATA.XAXis

Description

This command reads the data of measurement points of X axis, for the active trace of selected channel (*Ch*).

Variable

Parameter	*Data*
Description	Indicates the array data (measurement points) of X axis
Data type	Variant type (Variant)

Examples

```
Dim AnaData As Variant
AnaData = SCPI.CALC1.SEL.DATA.XAX
```

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

:CALCulate{[1]-4}[:SELected]:DATA:XAXis?

Query response

<numeric 1>,… ,<numeric N><^END>

Example of use

```
10 OUTPUT 717;":CALC1:DATA:XAX?"
20 ENTER 717;A(*)
```

SCPI.CALCulate(*Ch*).SELected.EQUation.STATE

Object type

Property (**Read-Write**)

Syntax

SCPI.CALCulate(*Ch*).SELected.EQUation.STATE= *Status*

Status = SCPI.CALCulate(*Ch*).SELected.EQUation.STATE

Description

This command enables/disables the Equation Editor of selected channel (*Ch*).

Variable

Parameter	*Status*
Description	Sets/Gets the state of equation in the Equation Editor
Data type	Boolean type (Boolean)
Range	Select from either of the following: • True or ON: Turns ON the Equation Editor. • False or OFF: Turns OFF the Equation Editor.
Preset value	False or OFF

Examples

```
Dim strEq As String
SCPI.CALCulate(1).SELected.EQUation.STATE = True
EqState = SCPI.CALCulate(1).SELected.EQUation.State
SCPI.CALCulate(1).SELected.EQUation.TEXT = "Example=S21/(1-S11)"
strEq = SCPI.CALCulate(1).SELected.EQUation.TEXT
```

Related objects

SCPI.CALCulate(Ch).SELected.EQUation.TEXT

SCPI.CALCulate(Ch).SELected.EQUation.VALid

Equivalent key

Display > **Equation**

Equivalent SCPI command

Syntax

:CALCulate{[1]-4}[:SELected]:EQUation:STATE {ON|OFF|1|0}

:CALCulate{[1]-4}[:SELected]:EQUation:STATE?

Query response

{1|0}<newline><^END>

Example of use

```
10 OUTPUT 717;":CALC1:EQU:STATE ON"
20 OUTPUT 717;":CALC1:EQU:STATE?"
30 ENTER 717;A
```

SCPI.CALCulate(*Ch*).SELected.EQUation.TEXT

Object type

Property (**Read-Write**)

Syntax

SCPI.CALCulate(*Ch*).SELected.EQUation.TEXT = *Data*

Data = SCPI.CALCulate(*Ch*).SELected.EQUation.TEXT

Description

This command sets/gets the equation in the Equation Editor for selected channel (*Ch*). For valid parameters that can be used in this equation, refer to the Equation Editor.

Variable

Parameter	*Data*
Description	Sets/Gets the equation in the Equation Editor
Data type	Character string type (String)

Examples

```
Dim  strEq As String
SCPI.CALCulate(1).SELected.EQUation.STATE = True
EqState = SCPI.CALCulate(1).SELected.EQUation.State
SCPI.CALCulate(1).SELected.EQUation.TEXT = "Example=S21/(1-S11)"
strEq = SCPI.CALCulate(1).SELected.EQUation.TEXT
```

Related objects

SCPI.CALCulate(Ch).SELected.EQUation.STATE

SCPI.CALCulate(Ch).SELected.EQUation.VALid

Equivalent key

No equivalent key is available on the front panel for Equation Text but Equation Editor can be accessed through **Display** > **Equation Editor.**

Equivalent SCPI command

Syntax

:CALCulate{[1]-4}[:SELected]:EQUation:TEXT <string1>

:CALCulate{[1]-4}[:SELected]:EQUation:TEXT?

Query response

<string 1><newline><^END>

Example of use

```
10 OUTPUT 717;":CALC1:EQU:TEXT?"
30 ENTER 717;A$
```

SCPI.CALCulate(*Ch*).SELected.EQUation.VALid

Object type

Property (**Read Only**)

Syntax

Data = SCPI.CALCulate(*Ch*).SELected.EQUation.VALid

Description

This command returns False when the equation expression and label are correct but the required S-parameter data is not measured or if it refers the invalid corrected memory array. Annotation of '**Equ!**' is displayed when this command returns a False value.

NOTE Equation Editor can refer S parameter data and data present in corrected memory array.

Variable

Parameter	*Data*
Description	A boolean value which gets the state of the equation in the Equation Editor of selected channel (*Ch*) as invalid (False) or valid (True)
Data type	Boolean type (Boolean)
Range	True or ON: Valid Spara data False or OFF: Invalid Spara data
Preset value	False or OFF

Examples

```
Dim EqState As Boolean
EqState = SCPI.CALCulate(1).SELected.EQUation.VALid
```

Related objects

SCPI.CALCulate(Ch).SELected.EQUation.STATE

SCPI.CALCulate(Ch).SELected.EQUation.TEXT

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

:CALCulate{[1]-4}[:SELected]:EQUation:VALid?

Query response

<1|0><newline><^END>

Example of use

```
10 OUTPUT 717;":CALC1:EQU:VAL?"
20 ENTER 717;A
```

SCPI.CALCulate(*Ch*).SELected.FILTer.GATE.DISTance.CENTer

Object Type

Property (**Read-Write**)

Syntax

SCPI.CALCulate(*Ch*).SELected.FILTer.GATE.DISTance.CENTer = *Value*

Value = SCPI.CALCulate(*Ch*).SELected.FILTer.GATE.DISTance.CENTer

Description

This command sets/gets the center value of the gate used for the gating function of the fault location display, for the selected channel (*Ch*).

Variable

Parameter	*Value*
Description	The center value of the gate
Data Type	Double precision floating point type (Double)
Range	Varies depending on the frequency span and the number of points.
Preset Value	0 m or 0 ft
Unit	ft (feet) or m (meters)
Resolution	-

Examples

```
Dim Var as Double
SCPI.CALCulate(4).SELected.FILTer.GATE.DISTance.CENTer = Var
Var = SCPI.CALCulate(4).SELected.FILTer.GATE.DISTance.CENTer
```

Related Objects

SCPI.CALCulate(Ch).SELected.FILTer.GATE.DISTance.SHAPe

SCPI.CALCulate(Ch).SELected.FILTer.GATE.DISTance.SPAN

SCPI.CALCulate(Ch).SELected.FILTer.GATE.DISTance.STARt

SCPI.CALCulate(Ch).SELected.FILTer.GATE.DISTance.STATe

SCPI.CALCulate(Ch).SELected.FILTer.GATE.DISTance.STOP

SCPI.CALCulate(Ch).SELected.FILTer.GATE.DISTance.TYPE

SCPI.CALCulate(Ch).SELected.FILTer.GATE.DISTance.UNIT

Equivalent Key

Analysis > Gating > Center

Equivalent SCPI Command

Syntax

:CALCulate{[1]-4}[:SELected]:FILTer:GATE:DISTance:CENTer <numeric>

:CALCulate{[1]-4}[:SELected]:FILTer:GATE:DISTance:CENTer?

Query Response

{numeric} <newline><^END>

Example of use

```
10 OUTPUT 717;":CALC1:FILT:GATE:DIST:CENT 0"
20 OUTPUT 717;":CALC1:FILT:GATE:DIST:CENT?"
30 ENTER 717;A
```

SCPI.CALCulate(*Ch*).SELected.FILTer.GATE.DISTance.SHAPe

Object Type

Property (**Read-Write**)

Syntax

SCPI.CALCulate(*Ch*).SELected.FILTer.GATE.DISTance.SHAPe = *Param*

Param = SCPI.CALCulate(*Ch*).SELected.FILTer.GATE.DISTance.SHAPe

Description

This command sets/gets the shape of the gate used for the gating function of the fault location display, for the active trace of the selected channel (*Ch*).

Variable

Parameter	*Param*
Description	The shape of the gate
Data Type	Character string type (String)
Range	Select from the following: • "MAXimum": Specifies the maximum shape. • "WIDE": Specifies the wide shape. • "NORMal": Specifies the normal shape. • "MINimum": Specifies the minimum shape.
Preset Value	NORMal

Examples

```
Dim Var as String
Var= "MAXimum"
SCPI.CALCulate(4).SELected.FILTer.GATE.DISTance.SHAPe = Var
Var = SCPI.CALCulate(4).SELected.FILTer.GATE.DISTance.SHAPe
```

Related Objects

SCPI.CALCulate(Ch).SELected.FILTer.GATE.DISTance.CENTer

SCPI.CALCulate(Ch).SELected.FILTer.GATE.DISTance.SPAN

SCPI.CALCulate(Ch).SELected.FILTer.GATE.DISTance.STARt

SCPI.CALCulate(Ch).SELected.FILTer.GATE.DISTance.STATe

SCPI.CALCulate(Ch).SELected.FILTer.GATE.DISTance.STOP

SCPI.CALCulate(Ch).SELected.FILTer.GATE.DISTance.TYPE

SCPI.CALCulate(Ch).SELected.FILTer.GATE.DISTance.UNIT

Equivalent Key

Analysis > **Gating** > **Shape** > **Maximum|Wide|Normal|Minimum**

Equivalent SCPI Command

Syntax

:CALCulate{[1]-4}[:SELected]:FILTer:GATE:DISTance:SHAPe {MAXimum|WIDE|NORMal|MINimum}

:CALCulate{[1]-4}[:SELected]:FILTer:GATE:DISTance:SHAPe?

Query Response

{MAX|WIDE|NORM|MIN} <newline><^END>

Example of use

```
10 OUTPUT 717;":CALC1:FILT:GATE:DIST:SHAP WIDE"
20 OUTPUT 717;":CALC1:FILT:GATE:DIST:SHAP?"
30 ENTER 717;A$
```

SCPI.CALCulate(*Ch*).SELected.FILTer.GATE.DISTance.SPAN

Object Type

Property (**Read-Write**)

Syntax

SCPI.CALCulate(*Ch*).SELected.FILTer.GATE.DISTance.SPAN = *Value*

Value = SCPI.CALCulate(*Ch*).SELected.FILTer.GATE.DISTance.SPAN

Description

This command sets/gets the span value of the gate used for the gating function of the fault location display, for the active trace of the selected channel (*Ch)*.

Variable

Parameter	*Value*
Description	the span value of the gate
Data Type	Double precision floating point type (Double)
Range	Varies depending on the frequency span and the number of points.
Preset Value	19.6714211286 ft or 5.99584916 m
Unit	ft (feet) or m (meters)

Examples

```
Dim Var as Double
SCPI.CALCulate(4).SELected.FILTer.GATE.DISTance.SPAN = Var
Var = SCPI.CALCulate(4).SELected.FILTer.GATE.DISTance.SPAN
```

Related Objects

SCPI.CALCulate(Ch).SELected.FILTer.GATE.DISTance.CENTer

SCPI.CALCulate(Ch).SELected.FILTer.GATE.DISTance.SHAPe

SCPI.CALCulate(Ch).SELected.FILTer.GATE.DISTance.STARt

SCPI.CALCulate(Ch).SELected.FILTer.GATE.DISTance.STATe

SCPI.CALCulate(Ch).SELected.FILTer.GATE.DISTance.STOP

SCPI.CALCulate(Ch).SELected.FILTer.GATE.DISTance.TYPE

SCPI.CALCulate(Ch).SELected.FILTer.GATE.DISTance.UNIT

Equivalent Key

Analysis > Gating > Span

Equivalent SCPI Command

Syntax

:CALCulate{[1]-4}[:SELected]:FILTer:GATE:DISTance:SPAN <numeric>

:CALCulate{[1]-4}[:SELected]:FILTer:GATE:DISTance:SPAN?

Query Response

{numeric} <newline><^END>

Example of use

```
10 OUTPUT 717;":CALCulate1:FILT:GATE:DIST:SPAN 0"
20 OUTPUT 717;":CALCulate1:FILT:GATE:DIST:SPAN?"
30 ENTER 717;A
```

SCPI.CALCulate(*Ch*).SELected.FILTer.GATE.DISTance.STARt

Object Type

Property (**Read-Write**)

Syntax

SCPI.CALCulate(*Ch*).SELected.FILTer.GATE.DISTance.STARt = *Value*

Value = SCPI.CALCulate(*Ch*).SELected.FILTer.GATE.DISTance.STARt

Description

This command sets/gets the start value of the gate used for the gating function of the fault location display, for the active trace of the selected channel (*Ch)*.

Variable

Parameter	*Value*
Description	The start value of the gate
Data Type	Double precision floating point type (Double)
Range	Varies depending on the frequency span and the number of points.
Preset Value	-9.8357105643 ft or -2.99792458 m
Unit	ft (feet) or m (meters)

Examples

```
Dim Var as Double
SCPI.CALCulate(4).SELected.FILTer.GATE.DISTance.STARt = Var
Var = SCPI.CALCulate(4).SELected.FILTer.GATE.DISTance.STARt
```

Related Objects

SCPI.CALCulate(Ch).SELected.FILTer.GATE.DISTance.CENTer

SCPI.CALCulate(Ch).SELected.FILTer.GATE.DISTance.SHAPe

SCPI.CALCulate(Ch).SELected.FILTer.GATE.DISTance.SPAN

SCPI.CALCulate(Ch).SELected.FILTer.GATE.DISTance.STATe

SCPI.CALCulate(Ch).SELected.FILTer.GATE.DISTance.STOP

SCPI.CALCulate(Ch).SELected.FILTer.GATE.DISTance.TYPE

SCPI.CALCulate(Ch).SELected.FILTer.GATE.DISTance.UNIT

Equivalent Key

Analysis > Gating > Start

Equivalent SCPI Command

Syntax

:CALCulate{[1]-4}[:SELected]:FILTer:GATE:DISTance:STARt <numeric>

:CALCulate{[1]-4}[:SELected]:FILTer:GATE:DISTance:STARt?

Query Response

{numeric} <newline><^END>

Example of use

```
10 OUTPUT 717;":CALC1:FILT:GATE:DIST:STAR 0"
20 OUTPUT 717;":CALC1:FILT:GATE:DIST:STAR?"
30 ENTER 717;A
```

SCPI.CALCulate(*Ch*).SELected.FILTer.GATE.DISTance.STATe

Object Type

Property (**Read-Write**)

Syntax

SCPI.CALCulate(*Ch*).SELected.FILTer.GATE.DISTance.STATe = *Status*

Status = SCPI.CALCulate(*Ch*).SELected.FILTer.GATE.DISTance.STATe

Description

This command turns ON/OFF the gating function of the fault location display, for the active trace of the selected channel (*Ch)*.

Variable

Parameter	*Status*
Description	ON/OFF of the gating function
Data Type	Boolean type (Boolean)
Range	Select from the following: • True or ON: Turns ON the gating function. • False or OFF: Turns OFF the gating function.
Preset Value	False or OFF

Examples

```
Dim Var as Boolean
Var= True
SCPI.CALCulate(4).SELected.FILTer.GATE.DISTance.STATe = Var
Var = SCPI.CALCulate(4).SELected.FILTer.GATE.DISTance.STATe
```

Related Objects

SCPI.CALCulate(Ch).SELected.FILTer.GATE.DISTance.CENTer

SCPI.CALCulate(Ch).SELected.FILTer.GATE.DISTance.SHAPe

SCPI.CALCulate(Ch).SELected.FILTer.GATE.DISTance.SPAN

SCPI.CALCulate(Ch).SELected.FILTer.GATE.DISTance.STARt

SCPI.CALCulate(Ch).SELected.FILTer.GATE.DISTance.STOP

SCPI.CALCulate(Ch).SELected.FILTer.GATE.DISTance.TYPE

SCPI.CALCulate(Ch).SELected.FILTer.GATE.DISTance.UNIT

Equivalent Key

Analysis > **Gating** > **Gating**

Equivalent SCPI Command

Syntax

:CALCulate{[1]-4}[:SELected]:FILTer:GATE:DISTance:STATe {ON|OFF|1|0}

:CALCulate{[1]-4}[:SELected]:FILTer:GATE:DISTance:STATe?

Query Response

{1|0} <newline><^END>

Example of use

```
10 OUTPUT 717;":CALC1:FILT:GATE:DIST:STAT ON"
20 OUTPUT 717;":CALC1:FILT:GATE:DIST:STAT?"
30 ENTER 717;A
```

(SCPI.CALCulate(*Ch*).SELected.FILTer.GATE.DISTance.STOP

Object Type

Property (**Read-Write**)

Syntax

SCPI.CALCulate(*Ch*).SELected.FILTer.GATE.DISTance.STOP = *Value*

Value = SCPI.CALCulate(*Ch*).SELected.FILTer.GATE.DISTance.STOP

Description

This command sets/gets the stop value of the gate used for the gating function of the fault location display, for the active trace of the selected channel (*Ch).*

Variable

Parameter	*Value*
Description	The stop value of the gate
Data Type	Double precision floating point type (Double)
Range	Varies depending on the frequency span and the number of points.
Preset Value	9.8357105643 ft or 2.99792458 m
Unit	ft (feet) or m (meters)

Examples

```
Dim Var as Double
SCPI.CALCulate(4).SELected.FILTer.GATE.DISTance.STOP = Var
Var = SCPI.CALCulate(4).SELected.FILTer.GATE.DISTance.STOP
```

Related Objects

Equivalent Key

Analysis > **Gating** > **Stop**

Equivalent SCPI Command

Syntax

:CALCulate{[1]-4}[:SELected]:FILTer:GATE:DISTance:STOP <numeric>

:CALCulate{[1]-4}[:SELected]:FILTer:GATE:DISTance:STOP?

Query Response

{numeric} <newline><^END>

Example of use

```
10 OUTPUT 717;":CALC1:FILT:GATE:DIST:STOP 0"
20 OUTPUT 717;":CALC1:FILT:GATE:DIST:STOP?"
30 ENTER 717;A
```

SCPI.CALCulate(*Ch*).SELected.FILTer.GATE.DISTance.TYPE

Object Type

Property (**Read-Write**)

Syntax

SCPI.CALCulate(*Ch*).SELected.FILTer.GATE.DISTance.TYPE = *Param*

Param = SCPI.CALCulate(*Ch*).SELected.FILTer.GATE.DISTance.TYPE

Description

This command sets/gets the gate type used for the gating function of the fault location display, for the active trace of the selected channel (*Ch*).

Variable

Parameter	*Param*
Description	The gate type used for the gating function
Data Type	Character string type (String)
Range	Select from the following. • "BPASs": Specifies the band-pass type. • "NOTCh": Specifies the notch type.
Preset Value	"BPASs"

Examples

```
Dim Var as String
Var= "BPASs"
SCPI.CALCulate(4).SELected.FILTer.GATE.DISTance.TYPE = Var
Var = SCPI.CALCulate(4).SELected.FILTer.GATE.DISTance.TYPE
```

Related Objects

SCPI.CALCulate(Ch).SELected.FILTer.GATE.DISTance.CENTer

SCPI.CALCulate(Ch).SELected.FILTer.GATE.DISTance.SHAPe

SCPI.CALCulate(Ch).SELected.FILTer.GATE.DISTance.SPAN

SCPI.CALCulate(Ch).SELected.FILTer.GATE.DISTance.STARt

SCPI.CALCulate(Ch).SELected.FILTer.GATE.DISTance.STATe

SCPI.CALCulate(Ch).SELected.FILTer.GATE.DISTance.STOP

SCPI.CALCulate(Ch).SELected.FILTer.GATE.DISTance.UNIT

Equivalent Key

Analysis > **Gating** > **Type**

Equivalent SCPI Command

Syntax

:CALCulate{[1]-4}[:SELected]:FILTer:GATE:DISTance:TYPE {BPASs|NOTCh}

:CALCulate{[1]-4}[:SELected]:FILTer:GATE:DISTance:TYPE?

Query Response

{BPAS|NOTC} <newline><^END>

Example of use

```
10 OUTPUT 717;":CALC1:FILT:GATE:DIST:TYPE NOTC"
20 OUTPUT 717;":CALC1:FILT:GATE:DIST:TYPE?"
30 ENTER 717;A$
```

SCPI.CALCulate(*Ch*).SELected.FILTer.GATE.DISTance.UNIT

Object Type

Property (**Read-Write**)

Syntax

SCPI.CALCulate(*Ch*).SELected.FILTer.GATE.DISTance.UNIT = *Value*

Value = SCPI.CALCulate(*Ch*).SELected.FILTer.GATE.DISTance.UNIT

Description

This command sets/gets the distance unit used for the gating function of the fault location display, for the active trace of the selected channel (*Ch*).

Variable

Parameter	*Value*
Description	The distance unit used for the gating function
Data Type	Character string type (String)
Range	Select from the following. • "METers": Specifies the meter. • "FEET": Specifies the feet.
Preset Value	"METers"

Examples

```
Dim Var as String
Var= "METers"
SCPI.CALCulate(4).SELected.FILTer.GATE.DISTance.UNIT = Var
Var = SCPI.CALCulate(4).SELected.FILTer.GATE.DISTance.UNIT
```

Related Objects

SCPI.CALCulate(Ch).SELected.FILTer.GATE.DISTance.CENTer

SCPI.CALCulate(Ch).SELected.FILTer.GATE.DISTance.SHAPe

SCPI.CALCulate(Ch).SELected.FILTer.GATE.DISTance.SPAN

SCPI.CALCulate(Ch).SELected.FILTer.GATE.DISTance.STARt

SCPI.CALCulate(Ch).SELected.FILTer.GATE.DISTance.STATe

SCPI.CALCulate(Ch).SELected.FILTer.GATE.DISTance.STOP

SCPI.CALCulate(Ch).SELected.FILTer.GATE.DISTance.TYPE

Equivalent Key

Analysis > **Gating** > **Unit** > **Meters|Feet|Seconds**

NOTE When performing this operation from the front panel, you select the horizontal axis of the gating function as either time or distance at the same time.

Equivalent SCPI Command

Syntax

:CALCulate{[1]-4}[:SELected]:FILTer:GATE:DISTance:UNIT {METers|FEET}

:CALCulate{[1]-4}[:SELected]:FILTer:GATE:DISTance:UNIT?

Query Response

{MET|FEET} <newline><^END>

Example of use

```
10 OUTPUT 717;":CALC1:FILT:GATE:DIST:UNIT MET"
20 OUTPUT 717;":CALC1:FILT:GATE:DIST:UNIT?"
30 ENTER 717;A$
```

SCPI.CALCulate(*Ch*).SELected.FILTer.GATE.METHod

Object Type

Property (**Read-Write**)

Syntax

SCPI.CALCulate(*Ch*).SELected.FILTer.GATE.METHod = *Param*

Param = SCPI.CALCulate(*Ch*).SELected.FILTer.GATE.METHod

Description

This command sets/gets the horizontal axis of the gating function either time or distance, for the active trace of the selected channel (*Ch*).

Variable

Parameter	*Param*
Description	The horizontal axis of the gating function
Data Type	Character string type (String)
Range	Select from the following. • "TIME": Specifies the time. • "DISTance": Specifies the distance.
Preset Value	"TIME"

Examples

```
Dim Var as String
Var= "TIME"
SCPI.CALCulate(4).SELected.FILTer.GATE.METHod = Var
Var = SCPI.CALCulate(4).SELected.FILTer.GATE.METHod
```

Related Objects

SCPI.CALCulate(Ch).SELected.FILTer.GATE.DISTance.UNIT

Equivalent Key

Analysis > **Gating** > **Unit** > **Meters|Feet|Seconds**

Equivalent SCPI Command

Syntax

:CALCulate{[1]-4}[:SELected]:FILTer:GATE:METHod {TIME|DISTance}

:CALCulate{[1]-4}[:SELected]:FILTer:GATE:METHod?

Query Response

{TIME|DIST} <newline><^END>

Example of use

```
10 OUTPUT 717;":CALC1:FILT:GATE:METH TIME"
20 OUTPUT 717;":CALC1:FILT:GATE:METH?"
30 ENTER 717;A$
```

SCPI.CALCulate(*Ch*).SELected.FILTer.GATE.TIME.CENTer

Object type

Property (**Read-Write**)

Syntax

SCPI.CALCulate(*Ch*).SELected.FILTer.GATE.TIME.CENTer = *Value*

Value = SCPI.CALCulate(*Ch*).SELected.FILTer.GATE.TIME.CENTer

Description

This command sets/gets the center value of the gate used for the gating function of the time domain function, for the active trace of the selected channel (*Ch*).

Variable

Parameter	*Value*
Description	The center value of the gate
Data type	Double precision floating point type (Double)
Range	Varies depending on the frequency span and the number of points.
Preset value	0
Unit	s (second)
Note	If the specified variable 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.

Examples

```
Dim FilCent As Double
SCPI.CALCulate(1).PARameter(1).SELect
SCPI.CALCulate(1).SELected.FILTer.GATE.TIME.CENTer = 1E-8
FilCent = SCPI.CALCulate(1).SELected.FILTer.GATE.TIME.CENTer
```

Related objects

SCPI.CALCulate(Ch).SELected.FILTer.GATE.TIME.SPAN

SCPI.CALCulate(Ch).SELected.FILTer.GATE.TIME.STATe

SCPI.CALCulate(Ch).PARameter(Tr).SELect

Equivalent key

Analysis > Gating > Center

Equivalent SCPI command

Syntax

:CALCulate{[1]-4}[:SELected]:FILTer[:GATE]:TIME:CENTer <numeric>

:CALCulate{[1]-4}[:SELected]:FILTer[:GATE]:TIME:CENTer?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":CALC1:FILT:TIME:CENT 1E-8"
20 OUTPUT 717;":CALC1:FILT:TIME:CENT?"
30 ENTER 717;A
```

SCPI.CALCulate(*Ch*).SELected.FILTer.GATE.TIME.SHAPe

Object type

Property (**Read-Write**)

Syntax

SCPI.CALCulate(*Ch*).SELected.FILTer.GATE.TIME.SHAPe = *Param*

Param = SCPI.CALCulate(*Ch*).SELected.FILTer.GATE.TIME.SHAPe

Description

This command sets/gets the shape of the gate used for the gating function of the time domain function, for the active trace of the selected channel (*Ch*).

Variable

Parameter	*Param*
Description	The shape of the gate
Data type	Character string type (String)
Range	Select from the following: • "MAXimum": Specifies the maximum shape. • "WIDE": Specifies the wide shape. • "NORMal": Specifies the normal shape. • "MINimum": Specifies the minimum shape.
Preset value	"NORMal"

Examples

```
Dim FilShape As String
SCPI.CALCulate(1).PARameter(1).SELect
SCPI.CALCulate(1).SELected.FILTer.GATE.TIME.SHAPe = "wide"
FilShape = SCPI.CALCulate(1).SELected.FILTer.GATE.TIME.SHAPe
```

Related objects

SCPI.CALCulate(Ch).SELected.FILTer.GATE.TIME.TYPE

SCPI.CALCulate(Ch).SELected.FILTer.GATE.TIME.STATe

SCPI.CALCulate(Ch).PARameter(Tr).SELect

Equivalent key

Analysis > Gating > Shape > Maximum|Wide|Normal|Minimum

Equivalent SCPI command

Syntax

:CALCulate{[1]-4}[:SELected]:FILTer[:GATE]:TIME:SHAPe {MAXimum|WIDE|NORMal|MINimum}

:CALCulate{[1]-4}[:SELected]:FILTer[:GATE]:TIME:SHAPe?

Query response

{MAX|WIDE|NORM|MIN}<newline><^END>

Example of use

```
10 OUTPUT 717;":CALC1:FILT:TIME:SHAP WIDE"
20 OUTPUT 717;":CALC1:FILT:TIME:SHAP?"
30 ENTER 717;A$
```

SCPI.CALCulate(*Ch*).SELected.FILTer.GATE.TIME.SPAN

Object type

Property (**Read-Write**)

Syntax

SCPI.CALCulate(*Ch*).SELected.FILTer.GATE.TIME.SPAN = *Value*

Value = SCPI.CALCulate(*Ch*).SELected.FILTer.GATE.TIME.SPAN

Description

This command sets/gets the span value of the gate used for the gating function of the time domain function, for the active trace of the selected channel (*Ch*).

Variable

Parameter	*Value*
Description	The span value of the gate
Data type	Double precision floating point type (Double)
Range	Varies depending on the frequency span and the number of points.
Preset value	2E-8
Unit	s (second)
Note	If the specified variable 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.

Examples

```
Dim FilStar As Double
SCPI.CALCulate(1).PARameter(1).SELect
SCPI.CALCulate(1).SELected.FILTer.GATE.TIME.SPAN = 1E-8
FilStar = SCPI.CALCulate(1).SELected.FILTer.GATE.TIME.SPAN
```

Related objects

SCPI.CALCulate(Ch).SELected.FILTer.GATE.TIME.CENTer

SCPI.CALCulate(Ch).SELected.FILTer.GATE.TIME.STATe

SCPI.CALCulate(Ch).PARameter(Tr).SELect

Equivalent key

Analysis > Gating > Span

Equivalent SCPI command

Syntax

:CALCulate{[1]-4}[:SELected]:FILTer[:GATE]:TIME:SPAN <numeric>

:CALCulate{[1]-4}[:SELected]:FILTer[:GATE]:TIME:SPAN?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":CALC1:FILT:TIME:SPAN 1E-8"
20 OUTPUT 717;":CALC1:FILT:TIME:SPAN?"
30 ENTER 717;A
```

SCPI.CALCulate(*Ch*).SELected.FILTer.GATE.TIME.STARt

Object type

Property (**Read-Write**)

Syntax

SCPI.CALCulate(*Ch*).SELected.FILTer.GATE.TIME.STARt = *Value*

Value = SCPI.CALCulate(*Ch*).SELected.FILTer.GATE.TIME.STARt

Description

This command sets/gets the start value of the gate used for the gating function of the time domain function, for the active trace of the selected channel (*Ch*).

Variable

Parameter	*Value*
Description	The start value of the gate
Data type	Double precision floating point type (Double)
Range	Varies depending on the frequency span and the number of points.
Preset value	-1E-8
Unit	s (second)
Note	If the specified variable 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.

Examples

```
Dim FilCent As Double
SCPI.CALCulate(1).PARameter(1).SELect
SCPI.CALCulate(1).SELected.FILTer.GATE.TIME.STARt = 0
FilCent = SCPI.CALCulate(1).SELected.FILTer.GATE.TIME.STARt
```

Related objects

SCPI.CALCulate(Ch).SELected.FILTer.GATE.TIME.STOP

SCPI.CALCulate(Ch).SELected.FILTer.GATE.TIME.STATe

SCPI.CALCulate(Ch).PARameter(Tr).SELect

Equivalent key

Analysis > Gating > Start

Equivalent SCPI command

Syntax

:CALCulate{[1]-4}[:SELected]:FILTer[:GATE]:TIME:STARt <numeric>

:CALCulate{[1]-4}[:SELected]:FILTer[:GATE]:TIME:STARt?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":CALC1:FILT:TIME:STAR 0"
20 OUTPUT 717;":CALC1:FILT:TIME:STAR?"
30 ENTER 717;A
```

SCPI.CALCulate(*Ch*).SELected.FILTer.GATE.TIME.STATe

Object type

Property (**Read-Write**)

Syntax

SCPI.CALCulate(*Ch*).SELected.FILTer.GATE.TIME.STATe = *Status*

Status = SCPI.CALCulate(*Ch*).SELected.FILTer.GATE.TIME.STATe

Description

This command turns ON/OFF the gating function of the time domain function, for the active trace of selected channel (*Ch*).

You can turn ON the gating function only when the sweep type is the linear sweep and the number of points is 3 or more. If you execute this object to turn ON the gating function when the sweep type is other than linear sweep or the number of points is less than 3, an error occurs and the object is ignored.

When the sweep type is the power sweep, you cannot turn ON the gating function. If you execute this object to turn ON the gating function during the power sweep, an error occurs and the object is ignored.

Variable

Parameter	*Status*
Description	ON/OFF of the gating function
Data type	Boolean type (Boolean)
Range	Select from the following: • True or ON: Turns ON the gating function. • False or OFF: Turns OFF the gating function.
Preset value	False or OFF

Examples

```
Dim Gating As Boolean
SCPI.CALCulate(1).PARameter(1).SELect
SCPI.CALCulate(1).SELected.FILTer.GATE.TIME.STATe = True
Gating = SCPI.CALCulate(1).SELected.FILTer.GATE.TIME.STATe
```

Related objects

SCPI.CALCulate(Ch).PARameter(Tr).SELect

SCPI.SENSe(Ch).SWEep.TYPE

SCPI.SENSe(Ch).SWEep.POINts

Equivalent key

Analysis > **Gating** > **Gating**

Equivalent SCPI command

Syntax

:CALCulate{[1]-4}[:SELected]:FILTer[:GATE]:TIME:STATe {ON|OFF|1|0}

:CALCulate{[1]-4}[:SELected]:FILTer[:GATE]:TIME:STATe?

Query response

{1|0}<newline><^END>

Example of use

```
10 OUTPUT 717;":CALC1:FILT:TIME:STAT ON"
20 OUTPUT 717;":CALC1:FILT:TIME:STAT?"
30 ENTER 717;A
```

SCPI.CALCulate(*Ch*).SELected.FILTer.GATE.TIME.STOP

Object type

Property (**Read-Write**)

Syntax

SCPI.CALCulate(*Ch*).SELected.FILTer.GATE.TIME.STOP = *Value*

Value = SCPI.CALCulate(*Ch*).SELected.FILTer.GATE.TIME.STOP

Description

This command sets/gets the stop value of the gate used for the gating function of the time domain function, for the active trace of the selected channel (*Ch*).

Variable

Parameter	*Value*
Description	The stop value of the gate
Data type	Double precision floating point type (Double)
Range	Varies depending on the frequency span and the number of points.
Preset value	1E-8
Unit	s (second)
Note	If the specified variable 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.

Examples

```
Dim FilStop As Double
SCPI.CALCulate(1).PARameter(1).SELect
SCPI.CALCulate(1).SELected.FILTer.GATE.TIME.STOP = 2E-8
FilStop = SCPI.CALCulate(1).SELected.FILTer.GATE.TIME.STOP
```

Related objects

SCPI.CALCulate(Ch).SELected.FILTer.GATE.TIME.STARt

SCPI.CALCulate(Ch).SELected.FILTer.GATE.TIME.STATe

SCPI.CALCulate(Ch).PARameter(Tr).SELect

Equivalent key

Analysis > Gating > Stop

Equivalent SCPI command

Syntax

```
:CALCulate{[1]-4}[:SELected]:FILTer[:GATE]:TIME:STOP <numeric>
:CALCulate{[1]-4}[:SELected]:FILTer[:GATE]:TIME:STOP?
```

Query response

```
{numeric}<newline><^END>
```

Example of use

```
10 OUTPUT 717;":CALC1:FILT:TIME:STOP 2E-8"
20 OUTPUT 717;":CALC1:FILT:TIME:STOP?"
30 ENTER 717;A
```

SCPI.CALCulate(*Ch*).SELected.FILTer.GATE.TIME.TYPE

Object type

Property (**Read-Write**)

Syntax

SCPI.CALCulate(*Ch*).SELected.FILTer.GATE.TIME.TYPE = *Param*

Param = SCPI.CALCulate(*Ch*).SELected.FILTer.GATE.TIME.TYPE

Description

This command sets/gets the gate type used for the gating function of the time domain function, for the active trace of the selected channel (*Ch*).

Variable

Parameter	*Param*
Description	The gate type
Data type	Character string type (String)
Range	Select from the following. • "BPASs": Specifies the band-pass type. • "NOTCh": Specifies the notch type.
Preset value	"BPASs"

Examples

```
Dim FilType As String
SCPI.CALCulate(1).PARameter(1).SELect
SCPI.CALCulate(1).SELected.FILTer.GATE.TIME.SHAPe = "notc"
FilType = SCPI.CALCulate(1).SELected.FILTer.GATE.TIME.SHAPe
```

Related objects

SCPI.CALCulate(Ch).SELected.FILTer.GATE.TIME.SHAPe

SCPI.CALCulate(Ch).SELected.FILTer.GATE.TIME.STATe

SCPI.CALCulate(Ch).PARameter(Tr).SELect

Equivalent key

Analysis > **Gating** > **Type**

Equivalent SCPI command

Syntax

:CALCulate{[1]-4}[:SELected]:FILTer[:GATE]:TIME[:TYPE] {BPASs|NOTCh}

:CALCulate{[1]-4}[:SELected]:FILTer[:GATE]:TIME[:TYPE]?

Query response

{BPAS|NOTC}<newline><^END>

Example of use

```
10 OUTPUT 717;":CALC1:FILT:TIME NOTC"
20 OUTPUT 717;":CALC1:FILT:TIME?"
30 ENTER 717;A$
```

SCPI.CALCulate(*Ch*).SELected.FORMat

Object type

Property (**Read-Write**)

Syntax

SCPI.CALCulate(*Ch*).SELected.FORMat = *Param*

Param = SCPI.CALCulate(*Ch*).SELected.FORMat

Description

This command sets/gets the data format of the active trace of the selected channel (*Ch*).

Variable

Parameter	*Param*
Description	Data format
Data type	Character string type (String)
Range	Select from the following: • "MLOGarithmic": Specifies the log magnitude format. • "PHASe": Specifies the phase format. • "GDELay": Specifies the group delay format. • "SLINear": Specifies the Smith chart format (Lin/Phase). • "SLOGarithmic": Specifies the Smith chart format (Log/Phase). • "SCOMplex": Specifies the Smith chart format (Re/Im). • "SMITh": Specifies the Smith chart format (R+jX). • "SADMittance": Specifies the Smith chart format (G+jB). • "PLINear": Specifies the polar format (Lin/Phase). • "PLOGarithmic": Specifies the polar format (Log/Phase). • "POLar": Specifies the polar format (Re/Im).

	• "MLINear": Specifies the linear magnitude format. • "SWR": Specifies the SWR format. • "REAL": Specifies the real format. • "IMAGinary": Specifies the imaginary format. • "UPHase": Specifies the expanded phase format. • "PPHase": Specifies the positive phase format.
Preset value	"MLOGarithmic"

Examples

```
Dim Fmt As String
SCPI.CALCulate(1).PARameter(1).SELect
SCPI.CALCulate(1).SELected.FORMat = "smit"
Fmt = SCPI.CALCulate(1).SELected.FORMat
```

Related objects

SCPI.CALCulate(Ch).PARameter(Tr).SELect

Equivalent key

Format > **Log Mag|Phase|Group Delay|Lin Mag|SWR|Real|Imaginary|Expand Phase|Positive Phase**

Format > **Smith** > **Lin/Phase|Log/Phase|Real/Imag|R+jX|G+jB**

Format > **Polor** > **Lin/Phase|Log/Phase|Real/Imag**

Equivalent SCPI command

Syntax

:CALCulate{[1]-4}[:SELected]:FORMat {MLOGarithmic|PHASe|GDELay|SLINear|SLOGarithmic|SCOMplex|SMITh|SADMittance|PLINear|PLOGarithmic|POLar|MLINear|SWR|REAL| IMAGinary|UPHase|PPHase}

:CALCulate{[1]-4}[:SELected]:FORMat?

Query response

{MLOG|PHAS|GDEL|SLIN|SLOG|SCOM|SMIT|SADM|PLIN|PLOG|POL|MLIN|SWR| REAL|IMAG|UPH|PPH}<newline><^END>

Example of use

```
10 OUTPUT 717;":CALC1:FORM SLIN"
20 OUTPUT 717;":CALC1:FORM?"
30 ENTER 717;A$
```

SCPI.CALCulate(*Ch*).SELected.FUNCtion.DATA

Object type

Property (**Read Only**)

Syntax

Data = SCPI.CALCulate(*Ch*).SELected.FUNCtion.DATA

Description

This command reads the analysis result of the SCPI.CALCulate(Ch).SELected.FUNCtion.EXECute object, for the active trace of selected channel (*Ch*).

Variable

Parameter	*Data*
Description	Indicates the array data (analysis result) of N (number of data pairs)×2. N (number of data pairs) can be read out with the SCPI.CALCulate(Ch).SELected.FUNCtion.POINts object. Where n is an integer between 1 and N. • *Data(n×2-2)* :Response value or analysis result of the searched n-th measurement point. • *Data(n×2-1)* :Stimulus value of the searched n-th measurement point. Always 0 for the analysis of the mean value, the standard deviation, and the difference between the maximum value and the minimum value. The index of the array starts from 0.
Data type	Variant type (Variant)

Examples

```
Dim AnaData As Variant
SCPI.CALCulate(1).PARameter(1).SELect
SCPI.CALCulate(1).SELected.FUNCtion.TYPE = "mean"
SCPI.CALCulate(1).SELected.FUNCtion.EXECute
AnaData = SCPI.CALCulate(1).SELected.FUNCtion.DATA
```

Related objects

SCPI.CALCulate(Ch).PARameter(Tr).SELect

SCPI.CALCulate(Ch).SELected.FUNCtion.TYPE

SCPI.CALCulate(Ch).SELected.FUNCtion.EXECute

SCPI.CALCulate(Ch).SELected.FUNCtion.POINts

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

:CALCulate{[1]-4}[:SELected]:FUNCtion:DATA?

Query response

<numeric 1>,… ,<numeric N×2><^END>

Example of use

```
10 OUTPUT 717;":CALC1:FUNC:POIN?"
20 ENTER 717;A
30 REDIM B(1:2*A)
40 OUTPUT 717;":CALC1:FUNC:DATA?"
50 ENTER 717;B(*)
```

SCPI.CALCulate(*Ch*).SELected.FUNCtion.DOMain.COUPle

Object type

Property (**Read-Write**)

Syntax

SCPI.CALCulate(*Ch*).SELected.FUNCtion.DOMain.COUPle = *Status*

Status = SCPI.CALCulate(*Ch*).SELected.FUNCtion.DOMain.COUPle

Description

This command specifies whether to set the coupling of the analysis range of the SCPI.CALCulate(Ch).SELected.FUNCtion.EXECute object for all traces, for the selected channel (*Ch*).

Variable

Parameter	*Status*
Description	ON/OFF the trace coupling of the analysis range.
Data type	Boolean type (Boolean)
Range	Select from the following: • True or ON:Specifies the analysis range with the trace coupling. • False or OFF:Specifies the analysis range for each trace.
Preset value	True or ON

Examples

```
Dim TrCpl As Boolean
SCPI.CALCulate(1).SELected.FUNCtion.DOMain.COUPle = False
TrCpl = SCPI.CALCulate(1).SELected.FUNCtion.DOMain.COUPle
```

Related objects

SCPI.CALCulate(Ch).SELected.FUNCtion.DATA

SCPI.CALCulate(Ch).SELected.FUNCtion.EXECute

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

```
:CALCulate{[1]-4}[:SELected]:FUNCtion:DOMain:COUPle {ON|OFF|1|0}
:CALCulate{[1]-4}[:SELected]:FUNCtion:DOMain:COUPle?
```

Query response

```
{1|0}<newline><^END>
```

Example of use

```
10 OUTPUT 717;":CALC1:FUNC:DOM:COUP OFF"
20 OUTPUT 717;":CALC1:FUNC:DOM:COUP?"
30 ENTER 717;A
```

SCPI.CALCulate(*Ch*).SELected.FUNCtion.DOMain.STARt

Object type

Property (**Read-Write**)

Syntax

SCPI.CALCulate(*Ch*).SELected.FUNCtion.DOMain.STARt = *Value*

Value = SCPI.CALCulate(*Ch*).SELected.FUNCtion.DOMain.STARt

Description

This command sets/gets the start value of the analysis range of the SCPI.CALCulate(Ch).SELected.FUNCtion.EXECute object, for the selected channel (*Ch*).

When the trace coupling is OFF, the active trace is the target to be set.

Variable

Parameter	*Value*
Description	Start value of the analysis range
Data type	Double precision floating point type (Double)
Preset value	0
Unit	Hz (hertz), dBm or s (second)

Examples

```
Dim AnaStar As Double
SCPI.CALCulate(1).SELected.FUNCtion.DOMain.STARt = 1.5E9
AnaStar = SCPI.CALCulate(1).SELected.FUNCtion.DOMain.STARt
```

Related objects

SCPI.CALCulate(Ch).SELected.FUNCtion.DATA

SCPI.CALCulate(Ch).SELected.FUNCtion.DOMain.STOP

SCPI.CALCulate(Ch).SELected.FUNCtion.DOMain.STATe

SCPI.CALCulate(Ch).SELected.FUNCtion.DOMain.COUPle

SCPI.CALCulate(Ch).SELected.FUNCtion.EXECute

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

:CALCulate{[1]-4}[:SELected]:FUNCtion:DOMain:STARt <numeric>

:CALCulate{[1]-4}[:SELected]:FUNCtion:DOMain:STARt?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":CALC1:FUNC:DOM:STAR 1.7E9"
20 OUTPUT 717;":CALC1:FUNC:DOM:STAR?"
30 ENTER 717;A
```

SCPI.CALCulate(*Ch*).SELected.FUNCtion.DOMain.STATe

Object type

Property (**Read-Write**)

Syntax

SCPI.CALCulate(*Ch*).SELected.FUNCtion.DOMain.STATe = *Status*

Status = SCPI.CALCulate(*Ch*).SELected.FUNCtion.DOMain.STATe

Description

This command sets/gets whether to use an arbitrary range when executing the analysis with the SCPI.CALCulate(Ch).SELected.FUNCtion.EXECute object, for the selected channel (*Ch*).

When the trace coupling is OFF, the active trace is the target to be set.

Variable

Parameter	*Status*
Description	Selection of the analysis range
Data type	Boolean type (Boolean)
Range	Select from the following: • True or ON: Specifies an arbitrary range • False or OFF: Specifies the entire sweep range
Preset value	False or OFF

Examples

```
Dim AnaRnge As Boolean
SCPI.CALCulate(1).SELected.FUNCtion.DOMain.STARt = 1.5E9
SCPI.CALCulate(1).SELected.FUNCtion.DOMain.STOP = 1.8E9
SCPI.CALCulate(1).SELected.FUNCtion.DOMain.STATe = True
AnaRnge = SCPI.CALCulate(1).SELected.FUNCtion.DOMain.STATe
```

Related objects

SCPI.CALCulate(Ch).SELected.FUNCtion.DATA

SCPI.CALCulate(Ch).SELected.FUNCtion.DOMain.STARt

SCPI.CALCulate(Ch).SELected.FUNCtion.DOMain.STOP

SCPI.CALCulate(Ch).SELected.FUNCtion.DOMain.COUPle

SCPI.CALCulate(Ch).SELected.FUNCtion.EXECute

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

:CALCulate{[1]-4}[:SELected]:FUNCtion:DOMain[:STATe] {ON|OFF|1|0}

:CALCulate{[1]-4}[:SELected]:FUNCtion:DOMain[:STATe]?

Query response

{1|0}<newline><^END>

Example of use

```
10 OUTPUT 717;":CALC1:FUNC:DOM ON"
20 OUTPUT 717;":CALC1:FUNC:DOM?"
30 ENTER 717;A
```

SCPI.CALCulate(*Ch*).SELected.FUNCtion.DOMain.STOP

Object type

Property (**Read-Write**)

Syntax

SCPI.CALCulate(*Ch*).SELected.FUNCtion.DOMain.STOP = *Value*

Value = SCPI.CALCulate(*Ch*).SELected.FUNCtion.DOMain.STOP

Description

This command sets/gets the stop value of the analysis range of the SCPI.CALCulate(Ch).SELected.FUNCtion.EXECute object, for the selected channel (*Ch*).

When the trace coupling is OFF, the active trace is the target to be set.

Variable

Parameter	*Value*
Description	Stop value of the analysis range
Data type	Double precision floating point type (Double)
Preset value	0
Unit	Hz (hertz), dBm or s (second)

Examples

```
Dim AnaStop As Double
SCPI.CALCulate(1).SELected.FUNCtion.DOMain.STOP = 1.8E9
AnaStop = SCPI.CALCulate(1).SELected.FUNCtion.DOMain.STOP
```

Related objects

SCPI.CALCulate(Ch).SELected.FUNCtion.DATA

SCPI.CALCulate(Ch).SELected.FUNCtion.DOMain.STARt

SCPI.CALCulate(Ch).SELected.FUNCtion.DOMain.STATe

SCPI.CALCulate(Ch).SELected.FUNCtion.DOMain.COUPle

SCPI.CALCulate(Ch).SELected.FUNCtion.EXECute

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

:CALCulate{[1]-4}[:SELected]:FUNCtion:DOMain:STOP <numeric>

:CALCulate{[1]-4}[:SELected]:FUNCtion:DOMain:STOP?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":CALC1:FUNC:DOM:STOP 1.8E9"
20 OUTPUT 717;":CALC1:FUNC:DOM:STOP?"
30 ENTER 717;A
```

SCPI.CALCulate(*Ch*).SELected.FUNCtion.EXECute

Object type

Method (**Write Only**)

Syntax

SCPI.CALCulate(*Ch*).SELected.FUNCtion.EXECute

Description

This command executes the analysis specified with the SCPI.CALCulate(Ch).SELected.FUNCtion.TYPE object, for the active trace of the selected channel (*Ch*).

Examples

```
SCPI.CALCulate(1).PARameter(1).SELect
SCPI.CALCulate(1).SELected.FUNCtion.EXECute
```

Related objects

SCPI.CALCulate(Ch).SELected.FUNCtion.DATA

SCPI.CALCulate(Ch).PARameter(Tr).SELect

SCPI.CALCulate(Ch).SELected.FUNCtion.TYPE

SCPI.CALCulate(Ch).SELected.FUNCtion.DOMain.STATe

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

:CALCulate{[1]-4}[:SELected]:FUNCtion:EXECute

Example of use

```
10 OUTPUT 717;":CALC1:FUNC:EXEC"
```

SCPI.CALCulate(*Ch*).SELected.FUNCtion.PEXCursion

Object type

Property (**Read-Write**)

Syntax

SCPI.CALCulate(*Ch*).SELected.FUNCtion.PEXCursion = *Value*

Value = SCPI.CALCulate(*Ch*).SELected.FUNCtion.PEXCursion

Description

This command sets/gets the lower limit of peak excursion value (the minimum value of the difference relative to the right and left adjacent measurement points) when executing the peak search with the SCPI.CALCulate(Ch).SELected.FUNCtion.EXECute object, for the active trace of the selected channel (*Ch*).

Variable

Parameter	*Value*
Description	Lower limit of peak excursion value
Data type	Double precision floating point type (Double)
Range	0 to 5E8
Preset value	3
Unit	Varies depending on the data format. • Log magnitude (MLOG) : dB (decibel) • Phase (PHAS), Expanded phase (UPH) or Positive phase (PPH) : ° (degree) • Group delay (GDEL) : s (second) • Others : No unit
Note	If the specified variable 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.

Examples

```
Dim PeakExc As Double
SCPI.CALCulate(1).PARameter(1).SELect
SCPI.CALCulate(1).SELected.FUNCtion.TYPE = "peak"
SCPI.CALCulate(1).SELected.FUNCtion.PEXCursion = 1.5
PeakExc = SCPI.CALCulate(1).SELected.FUNCtion.PEXCursion
```

Related objects

SCPI.CALCulate(Ch).SELected.FUNCtion.DATA

SCPI.CALCulate(Ch).PARameter(Tr).SELect

SCPI.CALCulate(Ch).SELected.FUNCtion.TYPE

SCPI.CALCulate(Ch).SELected.FUNCtion.PPOLarity

SCPI.CALCulate(Ch).SELected.FUNCtion.EXECute

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

:CALCulate{[1]-4}[:SELected]:FUNCtion:PEXCursion <numeric>

:CALCulate{[1]-4}[:SELected]:FUNCtion:PEXCursion?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":CALC1:FUNC:PEXC 0.2"
20 OUTPUT 717;":CALC1:FUNC:PEXC?"
30 ENTER 717;A
```

SCPI.CALCulate(*Ch*).SELected.FUNCtion.POINts

Object type

Property (**Read Only**)

Syntax

Value = SCPI.CALCulate(*Ch*).SELected.FUNCtion.POINts

Description

This command reads the number of data pairs of the analysis result of the SCPI.CALCulate(Ch).SELected.FUNCtion.EXECute object, for the active trace of selected channel (*Ch*).

For the analysis of the mean value or the search of the maximum value, 1 is always read out; for the search of all peaks or the search of all targets, the total number of searched measurement points is read out.

Variable

Parameter	*Value*
Description	Number of analyzed data pairs
Data type	Long integer type (Long)
Preset Value	0

Examples

```
Dim AnaPoin As Long
SCPI.CALCulate(1).PARameter(1).SELect
SCPI.CALCulate(1).SELected.FUNCtion.TYPE = "ape"
SCPI.CALCulate(1).SELected.FUNCtion.EXECute
AnaPoin = SCPI.CALCulate(1).SELected.FUNCtion.POINts
```

Related objects

SCPI.CALCulate(Ch).SELected.FUNCtion.DATA

SCPI.CALCulate(Ch).PARameter(Tr).SELect

SCPI.CALCulate(Ch).SELected.FUNCtion.EXECute

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

:CALCulate{[1]-4}[:SELected]:FUNCtion:POINts?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":CALC1:FUNC:POIN?"
20 ENTER 717;A
```

SCPI.CALCulate(*Ch*).SELected.FUNCtion.PPOLarity

Object type

Property (**Read-Write**)

Syntax

SCPI.CALCulate(*Ch*).SELected.FUNCtion.PPOLarity = *Param*

Param = SCPI.CALCulate(*Ch*).SELected.FUNCtion.PPOLarity

Description

This command sets/gets the polarity when performing the peak search with the SCPI.CALCulate(Ch).SELected.FUNCtion.EXECute object, for the active trace of the selected channel (*Ch*).

Variable

Parameter	*Param*
Description	Polarity for peak search
Data type	Character string type (String)
Range	Select from the following: • "POSitive": Specifies the positive peak. • "NEGative": Specifies the negative peak. • "BOTH": Specifies both the positive peak and the negative peak.
Preset value	"POSitive"

Examples

```
Dim PeakPol As String
SCPI.CALCulate(1).PARameter(1).SELect
SCPI.CALCulate(1).SELected.FUNCtion.TYPE = "peak"
SCPI.CALCulate(1).SELected.FUNCtion.PPOLarity = "both"
PeakPol = SCPI.CALCulate(1).SELected.FUNCtion.PPOLarity
```

Related objects

SCPI.CALCulate(Ch).SELected.FUNCtion.DATA

SCPI.CALCulate(Ch).PARameter(Tr).SELect

SCPI.CALCulate(Ch).SELected.FUNCtion.TYPE

SCPI.CALCulate(Ch).SELected.FUNCtion.PEXCursion

SCPI.CALCulate(Ch).SELected.FUNCtion.EXECute

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

:CALCulate{[1]-4}[:SELected]:FUNCtion:PPOLarity {POSitive|NEGative|BOTH}

:CALCulate{[1]-4}[:SELected]:FUNCtion:PPOLarity?

Query response

{POS|NEG|BOTH}<newline><^END>

Example of use

```
10 OUTPUT 717;":CALC1:FUNC:PPOL BOTH"
20 OUTPUT 717;":CALC1:FUNC:PPOL?"
30 ENTER 717;A$
```

SCPI.CALCulate(*Ch*).SELected.FUNCtion.TARGet

Object type

Property (**Read-Write**)

Syntax

SCPI.CALCulate(*Ch*).SELected.FUNCtion.TARGet = *Value*

Value = SCPI.CALCulate(*Ch*).SELected.FUNCtion.TARGet

Description

This command sets/gets the target value when performing the target search with the SCPI.CALCulate(Ch).SELected.FUNCtion.EXECute object, for the active trace of the selected channel (*Ch*).

Variable

Parameter	*Value*
Description	Target value
Data type	Double precision floating point type (Double)
Range	-5E8 to 5E8
Preset value	0
Unit	Varies depending on the data format. • Log magnitude (MLOG) : dB (decibel) • Phase (PHAS), Expanded phase (UPH) or Positive phase (PPH) : ° (degree) • Group delay (GDEL) : s (second) • Others : No unit
Note	If the specified variable 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.

Examples

```
Dim TargVal As Double
SCPI.CALCulate(1).PARameter(1).SELect
SCPI.CALCulate(1).SELected.FUNCtion.TYPE = "atar"
```

```
SCPI.CALCulate(1).SELected.FUNCtion.TARGet = -12.5
TargVal = SCPI.CALCulate(1).SELected.FUNCtion.TARGet
```

Related objects

SCPI.CALCulate(Ch).SELected.FUNCtion.DATA

SCPI.CALCulate(Ch).PARameter(Tr).SELect

SCPI.CALCulate(Ch).SELected.FUNCtion.TYPE

SCPI.CALCulate(Ch).SELected.FUNCtion.TTRansition

SCPI.CALCulate(Ch).SELected.FUNCtion.EXECute

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

:CALCulate{[1]-4}[:SELected]:FUNCtion:TARGet <numeric>

:CALCulate{[1]-4}[:SELected]:FUNCtion:TARGet?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":CALC1:FUNC:TARG -12.5"
20 OUTPUT 717;":CALC1:FUNC:TARG?"
30 ENTER 717;A
```

SCPI.CALCulate(*Ch*).SELected.FUNCtion.TTRansition

Object type

Property (**Read-Write**)

Syntax

SCPI.CALCulate(*Ch*).SELected.FUNCtion.TTRansition = *Param*

Param = SCPI.CALCulate(*Ch*).SELected.FUNCtion.TTRansition

Description

This command sets/gets the transition type when performing the target search with the SCPI.CALCulate(Ch).SELected.FUNCtion.EXECute object, for the active trace of the selected channel (*Ch*).

Variable

Parameter	*Param*
Description	Transition type for search
Data type	Character string type (String)
Range	Select from the following: • "POSitive": Specifies the positive transition. • "NEGative": Specifies the negative transition. • "BOTH": Specifies both the positive transition and the negative transition.
Preset value	"BOTH"

Examples

```
Dim TargTran As String
SCPI.CALCulate(1).PARameter(1).SELect
SCPI.CALCulate(1).SELected.FUNCtion.TYPE = "atar"
SCPI.CALCulate(1).SELected.FUNCtion.TTRansition = "pos"
TargTran = SCPI.CALCulate(1).SELected.FUNCtion.TTRansition
```

Related objects

SCPI.CALCulate(Ch).SELected.FUNCtion.DATA

SCPI.CALCulate(Ch).PARameter(Tr).SELect

SCPI.CALCulate(Ch).SELected.FUNCtion.TYPE

SCPI.CALCulate(Ch).SELected.FUNCtion.TARGet

SCPI.CALCulate(Ch).SELected.FUNCtion.EXECute

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

:CALCulate{[1]-4}[:SELected]:FUNCtion:TTRansition {POSitive|
NEGative|BOTH}

:CALCulate{[1]-4}[:SELected]:FUNCtion:TTRansition?

Query response

{POS|NEG|BOTH}<newline><^END>

Example of use

```
10 OUTPUT 717;":CALC1:FUNC:TTR NEG"
20 OUTPUT 717;":CALC1:FUNC:TTR?"
30 ENTER 717;A$
```

SCPI.CALCulate(*Ch*).SELected.FUNCtion.TYPE

Object type

Property (**Read-Write**)

Syntax

SCPI.CALCulate(*Ch*).SELected.FUNCtion.TYPE = *Param*

Param = SCPI.CALCulate(*Ch*).SELected.FUNCtion.TYPE

Description

This command sets/gets the type of analysis, for the active trace of the selected channel (*Ch*).

Variable

Parameter	*Param*
Description	Analysis type
Data type	Character string type (String)
Range	Select from the following: • "PTPeak": Specifies the analysis of the difference between the maximum value and the minimum value (Peak to Peak). • "STDEV": Specifies the analysis of the standard deviation. • "MEAN": Specifies the analysis of the mean value. • "MAXimum": Specifies the search for the maximum value. • "MINimum": Specifies the search for the minimum value. • "PEAK": Specifies the search for the peak. • "APEak": Specifies the search for all peaks. • "ATARget": Specifies the search for all targets. • "SDEViation": Specifies the search for all deviations.
Preset	"PTPeak"

value	

Examples

```
Dim AnaType As String
SCPI.CALCulate(1).PARameter(1).SELect
SCPI.CALCulate(1).SELected.FUNCtion.TYPE = "atar"
AnaType = SCPI.CALCulate(1).SELected.FUNCtion.TYPE
```

Related objects

SCPI.CALCulate(Ch).SELected.FUNCtion.DATA

SCPI.CALCulate(Ch).PARameter(Tr).SELect

SCPI.CALCulate(Ch).SELected.FUNCtion.PEXCursion

SCPI.CALCulate(Ch).SELected.FUNCtion.PPOLarity

SCPI.CALCulate(Ch).SELected.FUNCtion.TARGet

SCPI.CALCulate(Ch).SELected.FUNCtion.TTRansition

SCPI.CALCulate(Ch).SELected.FUNCtion.EXECute

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

:CALCulate{[1]-4}[:SELected]:FUNCtion:TYPE {PTPeak| STDEV|MEAN| MAXimum|MINimum|PEAK|APEak|ATARget|SDEViation}

:CALCulate{[1]-4}[:SELected]:FUNCtion:TYPE?

Query response

{PTP|STDEV|MEAN|MAX|MIN|PEAK|APE|ATAR|SDEV}<newline><^END>

Example of use

```
10 OUTPUT 717;":CALC1:FUNC:TYPE PEAK"
20 OUTPUT 717;":CALC1:FUNC:TYPE?"
30 ENTER 717;A$
```

SCPI.CALCulate(*Ch*).SELected.LIMit.DATA

Object type

Property (**Read-Write**)

Syntax

SCPI.CALCulate(*Ch*).SELected.LIMit.DATA = *Data*

Data = SCPI.CALCulate(*Ch*).SELected.LIMit.DATA

Description

This command sets/gets the limit table for the limit test, for the active trace of selected channel (*Ch*).

Variable

Parameter	*Data*
Description	Indicates the array data (for limit line) of 1 + Num (number of limit lines) × 5. Where n is an integer between 1 and Num. • *Data(0)* :The number of limit lines you want to set. Specify an integer ranging 0 to 100. When the number of limit lines is set to 0 (clears the limit table), the variable Data is only required with *Data(0)*. • *Data(n×5-4)* :The type of the n-th line. Specify an integer 0 to 2 as follows. 0: OFF 1: Upper limit line 2: Lower limit line • *Data(n×5-3)* :The value on the horizontal axis (frequency/power/time) of the start point of the n-th line. • *Data(n×5-2)* :The value on the horizontal axis (frequency/power/time) of the end point of the n-th line. • *Data(n×5-1)* :The value on the vertical axis of the start point of the n-th line. • *Data(n×5)* :The value on the vertical axis of the end point of the n-th line. The index of the array starts from 0.
Data type	Variant type (Variant)

Note	If there is no array data, an error occurs at execution and the object is ignored. For *Data(n×5-4)* in the array data, if you specify an integer other than 0, 1 or 2, an error occurs at execution. For *Data(n×5-3)*, *Data(n×5-2)*, *Data(n×5-1)*, and *Data(n×5)* in the array data, if the specified value 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.

Examples

```
Dim LimData As Variant
SCPI.CALCulate(1).PARameter(1).SELect
SCPI.CALCulate(1).SELected.LIMit.DATA = Array(1,1,1e6,1e9,0,0)
LimData = SCPI.CALCulate(1).SELected.LIMit.DATA

SCPI.CALCulate(1).PARameter(1).SELect
SCPI.CALCulate(1).SELected.LIMit.DATA = Array(0) 'Clear Limit Table
```

```
Dim LimData(5) As Variant
Dim Ref As Variant
LimData(0) = 1
LimData(1) = 1
LimData(2) = 1e6
LimData(3) = 1e9
LimData(4) = 0
LimData(5) = 0
SCPI.CALCulate(1).PARameter(1).SELect
SCPI.CALCulate(1).SELected.LIMit.DATA = LimData
Ref = SCPI.CALCulate(1).SELected.LIMit.DATA

Dim LimData(0) As Variant
LimData(0) = 0
SCPI.CALCulate(1).PARameter(1).SELect
SCPI.CALCulate(1).SELected.LIMit.DATA = LimData 'Clear Limit Table
```

Related objects

SCPI.CALCulate(Ch).PARameter(Tr).SELect

SCPI.CALCulate(Ch).SELected.LIMit.STATe

SCPI.CALCulate(Ch).SELected.LIMit.DISPlay.STATe

Equivalent key

Analysis > Limit Test > Edit Limit Line

Equivalent SCPI command

Syntax

:CALCulate{[1]-4}[:SELected]:LIMit:DATA <numeric 1>, ... ,<numeric 1+(N×5)>

:CALCulate{[1]-4}[:SELected]:LIMit:DATA?

Query response

{numeric 1}, ... ,{numeric 1+(N×5)}<newline><^END>

Example of use

```
10 DIM B(1:2,1:5)
20 OUTPUT 717;":CALC1:LIM:DATA 2,1,1E9,3E9,0,0,2,1E9,3E9,-3,-3"
30 OUTPUT 717;":CALC1:LIM:DATA?"
40 ENTER 717;A,B(*)
```

```
10 OUTPUT 717;":CALC1:LIM:DATA 0" ! Clear Limit Table
```

SCPI.CALCulate(*Ch*).SELected.LIMit.DISPlay.CLIP

Object Type

Property (**Read-Write**)

Syntax

SCPI.CALCulate(*Ch*).SELected.LIMit.DISPlay.CLIP = *Status*

Status = SCPI.CALCulate(*Ch*).SELected.LIMit.DISPlay.CLIP

Description

This command sets/gets whether to display the part of the limit line that is not used for evaluation, for the selected channel (*Ch*).

Variable

Parameter	*Status*
Description	Displays the clipped limit lines
Data Type	Boolean type (Boolean)
Range	Select from the following: • True or ON: Displays the clipped limit lines. • False or OFF: Displays the entire limit lines.
Preset Value	True or ON

Examples

```
Dim LimClip as Boolean
LimClip = True
SCPI.CALCulate(4).SELected.LIMit.DISPlay.CLIP = LimClip
LimClip = SCPI.CALCulate(4).SELected.LIMit.DISPlay.CLIP
```

Related Objects

SCPI.CALCulate(Ch).SELected.LIMit.DISPlay.STATe

Equivalent Key

Analysis > **Limit Test** > **Clip Lines**

Equivalent SCPI Command

Syntax

:CALCulate{[1]-4}[:SELected]:LIMit:DISPlay:CLIP {ON|OFF|1|0}

:CALCulate{[1]-4}[:SELected]:LIMit:DISPlay:CLIP?

Query Response

{1|0} <newline><^END>

Example of use

```
10 OUTPUT 717;":CALC1:LIM:DISP:CLIP ON"
20 OUTPUT 717;":CALC1:LIM:DISP:CLIP?"
30 ENTER 717;A
```

SCPI.CALCulate(*Ch*).SELected.LIMit.DISPlay.STATe

Object type

Property (**Read-Write**)

Syntax

SCPI.CALCulate(*Ch*).SELected.LIMit.DISPlay.STATe = *Status*

Status = SCPI.CALCulate(*Ch*).SELected.LIMit.DISPlay.STATe

Description

This command turns ON/OFF the limit line display, for the active trace of the selected channel (*Ch*).

Variable

Parameter	*Status*
Description	Limit line display
Data type	Boolean type (Boolean)
Range	Select from the following. • True or ON: Turns ON the limit line display. • False or OFF: Turns OFF the limit line display.
Preset value	False or OFF

Examples

```
Dim LimDisp As Boolean
SCPI.CALCulate(1).PARameter(1).SELect
SCPI.CALCulate(1).SELected.LIMit = True
LimDisp = SCPI.CALCulate(1).SELected.LIMit.DISPlay.STATe
```

Related objects

SCPI.CALCulate(Ch).PARameter(Tr).SELect

SCPI.CALCulate(Ch).SELected.LIMit.STATe

Equivalent key

Analysis > **Limit Test** > **Limit Line**

Equivalent SCPI command

Syntax

:CALCulate{[1]-4}[:SELected]:LIMit:DISPlay[:STATe] {ON|OFF|1|0}

:CALCulate{[1]-4}[:SELected]:LIMit:DISPlay[:STATe]?

Query response

{1|0}<newline><^END>

Example of use

```
10 OUTPUT 717;":CALC1:LIM:DISP ON"
20 OUTPUT 717;":CALC1:LIM:DISP?"
30 ENTER 717;A
```

SCPI.CALCulate(*Ch*).SELected.LIMit.FAIL

Object type

Property (**Read Only**)

Syntax

Status = SCPI.CALCulate(*Ch*).SELected.LIMit.FAIL

Description

This command reads out the limit test result, for the active trace of selected channel (*Ch*).

Variable

Parameter	*Status*
Description	Limit test result
Data type	Boolean type (Boolean)
Range	Select from the following: • True or ON: The limit test result is FAIL. • False or OFF: The limit test result is PASS.
Note	When the limit test is set to OFF, False or OFF is always read out.

Examples

```
Dim Result As Boolean
SCPI.CALCulate(1).PARameter(1).SELect
SCPI.CALCulate(1).SELected.LIMit.STATe = True
Result = SCPI.CALCulate(1).SELected.LIMit.FAIL
```

Related objects

SCPI.CALCulate(Ch).PARameter(Tr).SELect

SCPI.CALCulate(Ch).SELected.LIMit.STATe

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

:CALCulate{[1]-4}[:SELected]:LIMit:FAIL?

Query response

{1|0}<newline><^END>

	Description
1	The limit test result is FAIL.
0	The limit test result is PASS.

When the limit test is set to OFF, 0 is always read out.

Example of use

```
10 OUTPUT 717;":CALC1:LIM:FAIL?"
20 ENTER 717;A
```

SCPI.CALCulate(*Ch*).SELected.LIMit.OFFSet.AMPLitude

Object type

Property (**Read-Write**)

Syntax

SCPI.CALCulate(Ch).SELected.LIMit.OFFSet.AMPLitude = *Value*

Value = SCPI.CALCulate(Ch).SELected.LIMit.OFFSet.AMPLitude

Description

This command sets/gets the limit line offset of response for the active trace of selected channel (*Ch*). The setting of the limit line does not change even if the offset value is changed.

Variable

Parameter	*Value*
Description	The limit line offset of Response (Vertical offset)
Data type	Double precision floating point type (Double)
Range	-5E8 to 5E8
Preset value	0
Unit	Varies depending on the data format. • Log magnitude (MLOG): dB (decibel) • Phase (PHAS), Expanded phase (UPH) or Positive phase (PPH): ° (degree) • Group delay (GDEL): s (second) • Others: No unit

Examples

```
Dim LimOffset As Double
SCPI.CALCulate(1).SELected.LIMit.OFFSet.AMPLitude = -10
LimOffset = SCPI.CALCulate(1).SELected.LIMit.OFFSet.AMPLitude
```

Related objects

SCPI.CALCulate(Ch).SELected.LIMit.STATe

SCPI.CALCulate(Ch).SELected.LIMit.OFFSet.MARKer

SCPI.CALCulate(Ch).SELected.LIMit.OFFSet.STIMulus

Equivalent key

Analysis > **Limit Test** > **Limit Line Offsets** > **Amplitude Offset**

Equivalent SCPI command

Syntax

:CALCulate{[1]-4}[:SELected]:LIMit:OFFSet:AMPLitude <numeric>

:CALCulate{[1]-4}[:SELected]:LIMit:OFFSet:AMPLitude?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":CALC1:LIM:OFFS:AMPL -10"
20 OUTPUT 717;":CALC1:LIM:OFFS:AMPL?"
30 ENTER 717;A
```

SCPI.CALCulate(*Ch*).SELected.LIMit.OFFSet.MARKer

Object type

Method (**Write-only**)

Syntax

SCPI.CALCulate(*Ch*).SELected.LIMit.OFFSet.MARKer

Description

This command sets the active marker value to amplitude offset using the limit line for the active trace of selected channel (*Ch*). The setting of the limit line does not change even if the offset value is changed. When the markers are not displayed, this command does not operate.

Variable

None

Examples

```
SCPI.CALCulate(1).PARameter(1).SELect
SCPI.CALCulate(1).SELected.LIMit.OFFSet.MARKer
```

Related objects

SCPI.CALCulate(Ch).SELected.LIMit.STATe

SCPI.CALCulate(Ch).SELected.LIMit.OFFSet.AMPLitude

SCPI.CALCulate(Ch).SELected.LIMit.OFFSet.STIMulus

Equivalent key

Analysis > **Limit Test** > **Limit Line Offsets** > **Marker -> Amplitude Offset**

Equivalent SCPI command

Syntax

:CALCulate{[1]-4}[:SELected]:LIMit:OFFSet:MARKer

Example of use

```
10 OUTPUT 717;":CALC1:LIM:OFFS:MARK"
```

SCPI.CALCulate(*Ch*).SELected.LIMit.OFFSet.STIMulus

Object type

Property (**Read-Write**)

Syntax

SCPI.CALCulate(*Ch*).SELected.LIMit.OFFSet.STIMulus = *Value*

Value = SCPI.CALCulate(*Ch*).SELected.LIMit.OFFSet.STIMulus

Description

This command sets/gets the stimulus offset of the limit line, for the active trace of the selected channel (*Ch*).

NOTE The setting of the limit line doesn't change even if the offset value is changed.

Variable

Parameter	*Value*
Description	The stimulus offset of the limit line
Data type	Double precision floating point type (Double)
Range	-1E12 to 1E12
Preset value	0
Unit	Hz (hertz), dBm or second

Examples

```
Dim LimOffset As Double
SCPI.CALCulate(1).SELected.LIMit.OFFSet.STIMulus = 1E9
LimOffset = SCPI.CALCulate(1).SELected.LIMit.OFFSet.STIMulus
```

Related objects

SCPI.CALCulate(Ch).SELected.LIMit.STATe

SCPI.CALCulate(Ch).SELected.LIMit.OFFSet.AMPLitude

SCPI.CALCulate(Ch).SELected.LIMit.OFFSet.MARKer

Equivalent key

Analysis > **Limit Test** > **Limit Line Offsets** > **Stimulus Offset**

Equivalent SCPI command

Syntax

:CALCulate{[1]-4}[:SELected]:LIMit:OFFSet:STIMulus <numeric>

:CALCulate{[1]-4}[:SELected]:LIMit:OFFSet:STIMulus?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":CALC1:LIM:OFFS:STIM 5E3"
20 OUTPUT 717;":CALC1:LIM:OFFS:STIM?"
30 ENTER 717;A
```

SCPI.CALCulate(*Ch*).SELected.LIMit.REPort.ALL

Object type

Property (**Read Only**)

Syntax

Data = SCPI.CALCulate(*Ch*).SELected.LIMit.REPort.ALL

Description

This command reads the bandwidth test results (stimulus value, limit test result, upper limit value and lower limit value of all measurement points), for the active trace of selected channel (specified with the SCPI.CALCulate(Ch).PARameter(Tr).SELect command).

Variable

Parameter	*Data*
Description	Indicates the array data (for limit line) of NOP (number of measurement points) × 4. Where n is an integer between 1 and NOP. • *Data(n×4-3)* The stimulus value for the measurement point. • *Data(n×4-2)* The limit test result. Specify an integer -1 to 1 as follows. -1: No limit 0: Fail 1: Pass • *Data(n×4-1)* The upper limit value at the measurement point. (If there is no limit at this point, reads out the 0.) • *Data(n×4)* The lower limit value at the measurement point. (If there is no limit at this point, reads out the 0.) The index of the array starts from 0.
Data type	Variant type (Variant)

Examples

```
Dim LimData As Variant
SCPI.CALCulate(1).PARameter(1).SELect
LimData = SCPI.CALCulate(1).SELected.LIMit.REPort.ALL
```

Related objects

SCPI.CALCulate(Ch).PARameter(Tr).SELect

SCPI.CALCulate(Ch).SELected.LIMit.STATe

SCPI.CALCulate(Ch).SELected.LIMit.REPort.DATA

SCPI.CALCulate(Ch).SELected.LIMit.REPort.POINts

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

:CALCulate{[1]-4}[:SELected]:LIMit:REPort:ALL?

Example of use

```
10 OUTPUT 717;":SENS1:SWE:POIN?"
20 ENTER 717;A
30 REDIM B(1:4*A)
40 OUTPUT 717;":CALC1:LIM:REP:ALL?"
50 ENTER 717;B(*)
```

SCPI.CALCulate(*Ch*).SELected.LIMit.REPort.DATA

Object type

Property (**Read Only**)

Syntax

Data = SCPI.CALCulate(*Ch*).SELected.LIMit.REPort.DATA

Description

This command reads the stimulus values (frequency, power level or time) at all the measurement points that failed the limit test, for the active trace of the selected channel (*Ch*).

Variable

Parameter	*Data*
Description	Indicates the array data for failed measurement points (can be read out with the SCPI.CALCulate(Ch).SELected.LIMit.REPort.POINts object).
Data type	Variant type (Variant)

Examples

```
Dim FailData As Variant
SCPI.CALCulate(1).PARameter(1).SELect
SCPI.CALCulate(1).SELected.LIMit.STATe = True
FailData = SCPI.CALCulate(1).SELected.LIMit.REPort.DATA
```

Related objects

SCPI.CALCulate(Ch).PARameter(Tr).SELect

SCPI.CALCulate(Ch).SELected.LIMit.REPort.POINts

SCPI.CALCulate(Ch).SELected.LIMit.STATe

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

:CALCulate{[1]-4}[:SELected]:LIMit:REPort[:DATA]?

Query response

{numeric 1},… ,{numeric N}<newline><^END>

Where N is the number of the measurement points that failed (can be read out with the :CALC{1-16}:LIM:REP:POIN? command).

Example of use

```
10 OUTPUT 717;":CALC1:LIM:REP:POIN?"
20 ENTER 717;A
30 REDIM B(1:A)
40 OUTPUT 717;":CALC1:LIM:REP?"
50 ENTER 717;B(*)
```

SCPI.CALCulate(*Ch*).SELected.LIMit.REPort.POINts

Object type

Property (**Read Only**)

Syntax

Value = SCPI.CALCulate(*Ch*).SELected.LIMit.REPort.POINts

Description

This command reads the number of the measurement points that failed the limit test, for the active trace of the selected channel (*Ch*).

Variable

Parameter	*Value*
Description	Number of measurement points that failed
Data type	Long integer type (Long)
Preset Value	0

Examples

```
Dim FailPoin As Long
SCPI.CALCulate(1).PARameter(1).SELect
SCPI.CALCulate(1).SELected.LIMit.STATe = True
FailPoin = SCPI.CALCulate(1).SELected.LIMit.REPort.POINts
```

Related objects

SCPI.CALCulate(Ch).PARameter(Tr).SELect

SCPI.CALCulate(Ch).SELected.LIMit.STATe

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

:CALCulate{[1]-4}[:SELected]:LIMit:REPort:POINts?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":CALC1:LIM:REP:POIN?"
20 ENTER 717;A
```

SCPI.CALCulate(*Ch*).SELected.LIMit.STATe

Object type

Property (**Read-Write**)

Syntax

SCPI.CALCulate(*Ch*).SELected.LIMit.STATe = *Status*

Status = SCPI.CALCulate(*Ch*).SELected.LIMit.STATe

Description

This command turns ON/OFF the limit test function, for the active trace of selected channel (*Ch*).

Variable

Parameter	*Status*
Description	ON/OFF of the limit test function
Data type	Boolean type (Boolean)
Range	Select from the following: • True or ON: Turns ON the limit test function. • False or OFF: Turns OFF the limit test function.
Preset value	False or OFF

Examples

```
Dim LimTest As Boolean
SCPI.CALCulate(1).PARameter(1).SELect
SCPI.CALCulate(1).SELected.LIMit.STATe = True
LimTest = SCPI.CALCulate(1).SELected.LIMit.STATe
```

Related objects

SCPI.CALCulate(Ch).PARameter(Tr).SELect

SCPI.CALCulate(Ch).SELected.LIMit.DISPlay.STATe

SCPI.DISPlay.FSIGn

Equivalent key

Analysis > **Limit Test** > **Limit Test**

Equivalent SCPI command

Syntax

:CALCulate{[1]-4}[:SELected]:LIMit[:STATe] {ON|OFF|1|0}

:CALCulate{[1]-4}[:SELected]:LIMit[:STATe]?

Query response

{1|0}<newline><^END>

Example of use

```
10 OUTPUT 717;":CALC1:LIM ON"
20 OUTPUT 717;":CALC1:LIM?"
30 ENTER 717;A
```

SCPI.CALCulate(*Ch*).SELected.MARKer(*Mk*).ACTivate

Object type

Method (**Write Only**)

Syntax

SCPI.CALCulate(*Ch*).SELected.MARKer(*Mk*).ACTivate

Description

This command sets the marker 1 to 9 (*Mk*) and reference marker (*Mk*:10) to the active marker, for the active trace of selected channel (*Ch*).

NOTE If you set a marker as not to be displayed to the active marker, the marker display is automatically set to ON.

Variable

Parameter	*Mk*
Description	Marker number
Data type	Long integer type (Long)
Range	1 to 10 Notice that 10 is for the reference marker.
Note	If the specified variable is out of the allowable setup range, an error occurs when executed.

Examples

```
SCPI.CALCulate(1).PARameter(1).SELect
SCPI.CALCulate(1).SELected.MARKer(1).ACTivate
```

Related objects

SCPI.CALCulate(Ch).PARameter(Tr).SELect

SCPI.DISPlay.WINDow(Ch).ACTivate

Equivalent key

Marker > **Marker 1**|**Marker 2**|**Marker 3**|**Marker 4**|**Ref Marker**

Marker > **More Markers** > **Marker 5**|**Marker 6**|**Marker 7**|**Marker 8**|**Marker 9**

Equivalent SCPI command

Syntax

:CALCulate{[1]-4}[:SELected]:MARKer{[1]-10}:ACTivate

Example of use

```
10 OUTPUT 717;":CALC1:MARK1:ACT"
```

SCPI.CALCulate(*Ch*).SELected.MARKer(*Mk*).BWIDth.DATA

Object type

Property (**Read Only**)

Syntax

Data = SCPI.CALCulate(*Ch*).SELected.MARKer(*Mk*).BWIDth.DATA

Description

This command reads the bandwidth search result of marker 1 to 9 (*Mk*) and reference marker (*Mk*:10), for the active trace of selected channel (*Ch*).

If the bandwidth search is impossible, an error occurs when executed and the object is ignored.

Variable

Parameter	*Data*
Description	Indicates 4-element array data (bandwidth search result). • *Data(0)* :The bandwidth. • *Data(1)* :Center point frequency of the 2 cutoff frequency points. • *Data(2)* :The Q value. • *Data(3)* :Insertion loss The index of the array starts from 0.
Data type	Variant type (Variant)

Examples

```
Dim BandData As Variant
SCPI.CALCulate(1).PARameter(1).SELect
BandData = SCPI.CALCulate(1).SELected.MARKer(1).BWIDth.DATA
```

Related objects

SCPI.CALCulate(Ch).PARameter(Tr).SELect

SCPI.CALCulate(Ch).SELected.MARKer.BWIDth.STATe

SCPI.CALCulate(Ch).SELected.MARKer(Mk).BWIDth.THReshold

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

:CALCulate{[1]-4}[:SELected]:MARKer{[1]-10}:BWIDth:DATA?

Query Response

{numeric1},{numeric2},{numeric3},{numeric4},<newline><^END>

Example of use

```
10 OUTPUT 717;":CALC1:MARK1:BWID:DATA?"
20 ENTER 717;A,B,C,D
```

SCPI.CALCulate(*Ch*).SELected.MARKer(*Mk*).BWIDth.THReshold

Object type

Property (**Read-Write**)

Syntax

SCPI.CALCulate(*Ch*).SELected.MARKer(*Mk*).BWIDth.THReshold = *Value*

Value = SCPI.CALCulate(*Ch*).SELected.MARKer(*Mk*).BWIDth.THReshold

Description

This command sets/gets the bandwidth definition value (the value to define the pass-band of the filter) of marker 1 to 9 (*Mk*) and reference marker (*Mk*:10), for the active trace of the selected channel (*Ch*).

Variable

Parameter	*Value*
Description	Bandwidth definition value (the value to define the pass band of the filter)
Data type	Double precision floating point type (Double)
Range	-5E8 to 5E8
Preset value	-3
Unit	Varies depending on the data format. • Log magnitude (MLOG): dB (decibel) • Phase (PHAS), Expanded phase (UPH) or Positive phase (PPH): ° (degree) • Group delay (GDEL): s (second) • Others: No unit
Note	If the specified variable 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.

Examples

```
Dim BandVal As Double
SCPI.CALCulate(1).PARameter(1).SELect
```

```
SCPI.CALCulate(1).SELected.MARKer(1).BWIDth.THReshold = -6
BandVal = SCPI.CALCulate(1).SELected.MARKer(1).BWIDth.THReshold
```

Related objects

SCPI.CALCulate(Ch).PARameter(Tr).SELect

SCPI.CALCulate(Ch).SELected.MARKer.BWIDth.STATe

Equivalent key
Marker Search > Bandwidth Value
Equivalent SCPI command

Syntax

:CALCulate{[1]-4}[:SELected]:MARKer{[1]-10}:BWIDth:THReshold <numeric>

:CALCulate{[1]-4}[:SELected]:MARKer{[1]-10}:BWIDth:THReshold?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":CALC1:MARK1:BWID:THR 6"
20 OUTPUT 717;":CALC1:MARK1:BWID:THR?"
30 ENTER 717;A
```

SCPI.CALCulate(*Ch*).SELected.MARKer(*Mk*).FUNCtion.EXECute

Object type

Method (**Write Only**)

Syntax

SCPI.CALCulate(*Ch*).SELected.MARKer(*Mk*).FUNCtion.EXECute

Description

This command executes search with marker 1 to 9 (*Mk*) and reference marker (*Mk*:10), for the active trace of the selected channel (*Ch*).

To specify the type of the search, use the SCPI.CALCulate(Ch).SELected.MARKer(Mk).FUNCtion. TYPE object. (No read)

Examples

```
SCPI.CALCulate(1).PARameter(1).SELect
SCPI.CALCulate(1).SELected.MARKer(1).FUNCtion.TYPE = "maximum"
SCPI.CALCulate(1).SELected.MARKer(1).FUNCtion.EXECute
```

Related objects

SCPI.CALCulate(Ch).PARameter(Tr).SELect

SCPI.CALCulate(Ch).SELected.MARKer(Mk).FUNCtion.TYPE

SCPI.CALCulate(Ch).SELected.MARKer.FUNCtion.DOMain.STATe

Equivalent key

Marker Search > **Max|Min**

Marker Search > **Peak** > **Search Peak|Search Left|Search Right**

Marker Search > **Target** > **Search Target|Search Left|Search Right**

When performing the operation from the front panel, you select the search type and execute the search at the same time.

Equivalent SCPI command

Syntax

:CALCulate{[1]-4}[:SELected]:MARKer{[1]-10}:FUNCtion:EXECute

Example of use

```
10 OUTPUT 717;":CALC1:MARK1:FUNC:EXEC"
```

SCPI.CALCulate(*Ch*).SELected.MARKer(*Mk*).FUNCtion.PEXCursion

Object type

Property (**Read-Write**)

Syntax

SCPI.CALCulate(*Ch*).SELected.MARKer(*Mk*).FUNCtion.PEXCursion = *Value*

Value = SCPI.CALCulate(*Ch*).SELected.MARKer(*Mk*).FUNCtion.PEXCursion

Description

This command sets/gets the lower limit of peak excursion value when executing the peak search with marker 1 to 9 (*Mk*) and reference marker (*Mk*:10), for the active trace of selected channel (*Ch*). Peak excursion value is the minimum value of the difference relative to the right and left adjacent measurement points.

Variable

Parameter	*Value*
Description	Lower limit of peak excursion value
Data type	Double precision floating point type (Double)
Range	0 to 5E8
Preset value	3
Unit	Varies depending on the data format. • Log magnitude (MLOG): dB (decibel) • Phase (PHAS), Expanded phase (UPH) or Positive phase (PPH): × (degree) • Group delay (GDEL): s (second) • Others: No unit
Note	If the specified variable 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.

Examples

```
Dim PeakExc As Double
SCPI.CALCulate(1).PARameter(1).SELect
SCPI.CALCulate(1).SELected.MARKer(1).FUNCtion.TYPE = "peak"
SCPI.CALCulate(1).SELected.MARKer(1).FUNCtion.PEXCursion = 0.2
PeakExc = SCPI.CALCulate(1).SELected.MARKer(1).FUNCtion.PEXCursion
```

Related objects

SCPI.CALCulate(Ch).PARameter(Tr).SELect

SCPI.CALCulate(Ch).SELected.MARKer(Mk).FUNCtion.TYPE

SCPI.CALCulate(Ch).SELected.MARKer(Mk).FUNCtion.PPOLarity

Equivalent key

Marker Search > **Peak** > **Peak Excursion**

Equivalent SCPI command

Syntax

:CALCulate{[1]-4}[:SELected]:MARKer{[1]-10}:FUNCtion:PEXCursion <numeric>

:CALCulate{[1]-4}[:SELected]:MARKer{[1]-10}:FUNCtion:PEXCursion?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":CALC1:MARK1:FUNC:PEXC 0.2"
20 OUTPUT 717;":CALC1:MARK1:FUNC:PEXC?"
30 ENTER 717;A
```

SCPI.CALCulate(*Ch*).SELected.MARKer(*Mk*).FUNCtion.PPOLarity

Object type

Property (**Read-Write**)

Syntax

SCPI.CALCulate(*Ch*).SELected.MARKer(*Mk*).FUNCtion.PPOLarity = *Param*

Param = SCPI.CALCulate(*Ch*).SELected.MARKer(*Mk*).FUNCtion.PPOLarity

Description

This comamnd set/get the polarity of the peak search with marker 1 to 9 (*Mk*) and reference marker (*Mk*:10), for the active trace of the selected channel (*Ch*).

Variable

Parameter	*Param*
Description	Polarity for peak search
Data type	Character string type (String)
Range	Select from the following: • "POSitive" Specifies the positive peak. • "NEGative" Specifies the negative peak. • "BOTH" Specifies both the positive peak and the negative peak.
Preset value	"POSitive"

Examples

```
Dim PeakPol As String
SCPI.CALCulate(1).PARameter(1).SELect
SCPI.CALCulate(1).SELected.MARKer(1).FUNCtion.TYPE = "peak"
SCPI.CALCulate(1).SELected.MARKer(1).FUNCtion.PPOLarity = "both"
PeakPol = SCPI.CALCulate(1).SELected.MARKer(1).FUNCtion.PPOLarity
```

Related objects

SCPI.CALCulate(Ch).PARameter(Tr).SELect

SCPI.CALCulate(Ch).SELected.MARKer(Mk).FUNCtion.TYPE

SCPI.CALCulate(Ch).SELected.MARKer(Mk).FUNCtion.PEXCursion

Equivalent key

Marker Search > **Peak** > **Peak Polarity** > **Positive|Negative|Both**

Equivalent SCPI command

Syntax

:CALCulate{[1]-4}[:SELected]:MARKer{[1]-10}:FUNCtion:PPOLarity {POSitive|NEGative|BOTH}

:CALCulate{[1]-4}[:SELected]:MARKer{[1]-10}:FUNCtion:PPOLarity?

Query response

{POS|NEG|BOTH}<newline><^END>

Example of use

```
10 OUTPUT 717;":CALC1:MARK1:FUNC:PPOL NEG"
20 OUTPUT 717;":CALC1:MARK1:FUNC:PPOL?"
30 ENTER 717;A$
```

SCPI.CALCulate(*Ch*).SELected.MARKer(*Mk*).FUNCtion.TARGet

Object type

Property (**Read-Write**)

Syntax

SCPI.CALCulate(*Ch*).SELected.MARKer(*Mk*).FUNCtion.TARGet = *Value*

Value = SCPI.CALCulate(*Ch*).SELected.MARKer(*Mk*).FUNCtion.TARGet

Description

This command sets/gets the target value to be searched with marker 1 to 9 (*Mk*) and reference marker (*Mk*:10, for the active trace of the selected channel (*Ch*).

Variable

Parameter	*Value*
Description	Target value for target search
Data type	Double precision floating point type (Double)
Range	-5E8 to 5E8
Preset value	0
Unit	Varies depending on the data format. • Log magnitude (MLOG): dB (decibel) • Phase (PHAS), Expanded phase (UPH) or Positive phase (PPH): ° (degree) • Group delay (GDEL): s (second) • Others: No unit
Note	If the specified variable 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.

Examples

```
Dim TargVal As Double
SCPI.CALCulate(1).PARameter(1).SELect
```

```
SCPI.CALCulate(1).SELected.MARKer(1).FUNCtion.TARGet = -12.5
TargVal = SCPI.CALCulate(1).SELected.MARKer(1).FUNCtion.TARGet
```

Related objects

SCPI.CALCulate(Ch).PARameter(Tr).SELect

SCPI.CALCulate(Ch).SELected.MARKer(Mk).FUNCtion.TYPE

SCPI.CALCulate(Ch).SELected.MARKer(Mk).FUNCtion.TTRansition

Equivalent key

Marker Search > **Target** > **Target Value**

Equivalent SCPI command

Syntax

:CALCulate{[1]-4}[:SELected]:MARKer{[1]-10}:FUNCtion:TARGet <numeric>

:CALCulate{[1]-4}[:SELected]:MARKer{[1]-10}:FUNCtion:TARGet?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":CALC1:MARK1:FUNC:TARG -12.5"
20 OUTPUT 717;":CALC1:MARK1:FUNC:TARG?"
30 ENTER 717;A
```

SCPI.CALCulate(*Ch*).SELected.MARKer(*Mk*).FUNCtion.TRACking

Object type

Property (**Read-Write**)

Syntax

SCPI.CALCulate(*Ch*).SELected.MARKer(*Mk*).FUNCtion.TRACking = *Status*

Status = SCPI.CALCulate(*Ch*).SELected.MARKer(*Mk*).FUNCtion.TRACking

Description

This command turns ON/OFF the search tracking (function to repeat search for each sweep) for marker 1 to 9 (*Mk*) and reference marker (*Mk*:10), for the active trace of the selected channel (*Ch*).

Variable

Parameter	*Status*
Description	ON/OFF of the marker search tracing
Data type	Boolean type (Boolean)
Range	Select from the following: • True or ON: Turns ON the search tracking. • False or OFF: Turns OFF the search tracking.
Preset value	False or OFF

Examples

```
Dim SrchTrac As Boolean
SCPI.CALCulate(1).PARameter(1).SELect
SCPI.CALCulate(1).SELected.MARKer(1).FUNCtion.TYPE = "targ"
SCPI.CALCulate(1).SELected.MARKer(1).FUNCtion.TRACking = True
SrchTrac = SCPI.CALCulate(1).SELected.MARKer(1).FUNCtion.TRACking
```

Related objects

SCPI.CALCulate(Ch).PARameter(Tr).SELect

SCPI.CALCulate(Ch).SELected.MARKer(Mk).FUNCtion.TYPE

SCPI.CALCulate(Ch).SELected.MARKer(Mk).FUNCtion.EXECute

Equivalent key

Marker Search > **Tracking**

Equivalent SCPI command

Syntax

:CALCulate{[1]-4}[:SELected]:MARKer{[1]-10}:FUNCtion:TRACking {ON|OFF|1|0}

:CALCulate{[1]-4}[:SELected]:MARKer{[1]-10}:FUNCtion:TRACking?

Query response

{1|0}<newline><^END>

Example of use

```
10 OUTPUT 717;":CALC1:MARK1:FUNC:TRAC ON"
20 OUTPUT 717;":CALC1:MARK1:FUNC:TRAC?"
30 ENTER 717;A
```

SCPI.CALCulate(*Ch*).SELected.MARKer(*Mk*).FUNCtion.TTRansition

Object type

Property (**Read-Write**)

Syntax

SCPI.CALCulate(*Ch*).SELected.MARKer(*Mk*).FUNCtion.TTRansition = *Param*

Param = SCPI.CALCulate(*Ch*).SELected.MARKer(*Mk*).FUNCtion.TTRansition

Description

This command selects the transition type of the target search, for marker 1 to 9 (*Mk*) and reference marker (*Mk*:10) of the active trace of the selected channel (*Ch*).

Variable

Parameter	*Param*
Description	Transition type for search
Data type	Character string type (String)
Range	Select from the following: • "POSitive": Specifies the positive transition. • "NEGative": Specifies the negative transition. • "BOTH": Specifies both the positive transition and the negative transition.
Preset value	"BOTH"

Examples

```
Dim TargTran As String
SCPI.CALCulate(1).PARameter(1).SELect
SCPI.CALCulate(1).SELected.MARKer(1).FUNCtion.TYPE = "targ"
SCPI.CALCulate(1).SELected.MARKer(1).FUNCtion.TTRansition = "neg"
TargTran = SCPI.CALCulate(1).SELected.MARKer(1).FUNCtion.TTRansition
```

Related objects

SCPI.CALCulate(Ch).PARameter(Tr).SELect

SCPI.CALCulate(Ch).SELected.MARKer(Mk).FUNCtion.TYPE

SCPI.CALCulate(Ch).SELected.MARKer(Mk).FUNCtion.TARGet

Equivalent key

Marker Search > **Target** > **Target Transition** > **Positive|Negative|Both**

Equivalent SCPI command

Syntax

:CALCulate{[1]-4}[:SELected]:MARKer{[1]-10}:FUNCtion:TTRansition {POSitive| NEGative|BOTH}

:CALCulate{[1]-4}[:SELected]:MARKer{[1]-10}:FUNCtion:TTRansition?

Query response

{POS|NEG|BOTH}<newline><^END>

Example of use

```
10 OUTPUT 717;":CALC1:MARK1:FUNC:TTR NEG"
20 OUTPUT 717;":CALC1:MARK1:FUNC:TTR?"
30 ENTER 717;A$
```

SCPI.CALCulate(*Ch*).SELected.MARKer(*Mk*).FUNCtion.TYPE

Object type

Property (**Read-Write**)

Syntax

SCPI.CALCulate(*Ch*).SELected.MARKer(*Mk*).FUNCtion.TYPE = *Param*

Param = SCPI.CALCulate(*Ch*).SELected.MARKer(*Mk*).FUNCtion.TYPE

Description

This command selects the search type for marker 1 to 9 (*Mk*) and reference marker (*Mk*:10), for the active trace of the selected channel (*Ch*).

Variable

Parameter	*Param*
Description	Search type of marker
Data type	Character string type (String)
Range	Select from the following: • "MAXimum": Sets the search type to the maximum value. • "MINimum": Sets the search type to the minimum value. • "PEAK": Sets the search type to the peak search. • "LPEak": Sets the search type to the peak search to the left from the marker position. • "RPEak": Sets the search type to the peak search to the right from the marker position. • "TARGet": Sets the search type to the target search. • "LTARget": Sets the search type to the target search to the left from the marker position. • "RTARget": Sets the search type to the target search to the right from the marker position.
Preset value	"MAXimum"

Examples

```
Dim SrchType As String
SCPI.CALCulate(1).PARameter(1).SELect
SCPI.CALCulate(1).SELected.MARKer(1).FUNCtion.TYPE = "targ"
SrchType = SCPI.CALCulate(1).SELected.MARKer(1).FUNCtion.TYPE
```

Related objects

SCPI.CALCulate(Ch).PARameter(Tr).SELect

SCPI.CALCulate(Ch).SELected.MARKer(Mk).FUNCtion.PEXCursion

SCPI.CALCulate(Ch).SELected.MARKer(Mk).FUNCtion.PPOLarity

SCPI.CALCulate(Ch).SELected.MARKer(Mk).FUNCtion.TARGet

SCPI.CALCulate(Ch).SELected.MARKer(Mk).FUNCtion.TTRansition

SCPI.CALCulate(Ch).SELected.MARKer(Mk).FUNCtion.EXECute

Equivalent key

Marker Search > **Max|Min**

Marker Search > **Peak** > **Search Peak|Search Left|Search Right**

Marker Search > **Target** > **Search Target|Search Left|Search Right**

NOTE When performing the operation from the front panel, you select the search type and execute the search at the same time.

Equivalent SCPI command

Syntax

:CALCulate{[1]-4}[:SELected]:MARKer{[1]-10}:FUNCtion:TYPE {MAXimum|MINimum|PEAK|LPEak|RPEak|TARGet|LTARget|RTARget}

:CALCulate{[1]-4}[:SELected]:MARKer{[1]-10}:FUNCtion:TYPE?

Query response

{MAX|MIN|PEAK|LPE|RPE|TARG|LTAR|RTAR}<newline><^END>

Example of use

```
10 OUTPUT 717;":CALC1:MARK1:FUNC:TYPE PEAK"
20 OUTPUT 717;":CALC1:MARK1:FUNC:TYPE?"
30 ENTER 717;A$
```

SCPI.CALCulate(*Ch*).SELected.MARKer(*Mk*).NOTCh.DATA

Object type

Property (**Read Only**)

Syntax

Data = SCPI.CALCulate(*Ch*).SELected.MARKer(Mk).NOTCh.DATA

Description

This command reads the notch search result of marker 1 to 9 (*Mk*) and reference marker (*Mk*:10), for the active trace of the selected channel (*Ch*).

If the notch search is impossible, an error occurs and the command is ignored. In this case, no query response is obtained.

Variable

Parameter	*Data*
Description	Indicates 4-element array data (notch bandwidth search result). • *Data(0)* :The bandwidth. • *Data(1)* :Center point frequency of the 2 cutoff frequency points. • *Data(2)* :The Q value. • *Data(3)* :Insertion loss The index of the array starts from 0.
Data type	Variant type (Variant)

Examples

```
Dim NotchData As Variant
SCPI.CALCulate(1).PARameter(1).SELect
BandData = SCPI.CALCulate(1).SELected.MARKer(1).NOTCH.DATA
```

Related Objects

SCPI.CALCulate(Ch).SELected.MARKer.NOTCh.STATe

SCPI.CALCulate(Ch).SELected.MARKer(Mk).NOTCh.THReshold

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

:CALCulate{[1]-4}[:SELected]:MARKer{[1]-10}:NOTCh:DATA?

Query response

{value 1},{value 2},{value 3},{value 4}<newline><^END>

Example of use

```
10 OUTPUT 717;":CALC1:MARK1:NOTC :DATA?"
20 ENTER 717;A,B,C,D
```

SCPI.CALCulate(*Ch*).SELected.MARKer(*Mk*).NOTCh.THReshold

Object type

Property (**Read-Write**)

Syntax

SCPI.CALCulate(*Ch*).SELected.MARKer(*Mk*).NOTCh.THReshold = *Value*

Value = SCPI.CALCulate(*Ch*).SELected.MARKer(*Mk*).NOTCh.THReshold

Description

This command sets/gets the notch definition value of marker 1 to 9 (*Mk*) and reference marker (*Mk*:10), for the active trace of the selected channel (*Ch*), specified with the :CALC{[1]-4}:PAR{[1]-4}:SEL command.

Variable

Parameter	*Value*
Description	Notch definition value
Range	-5E8 to 5E8
Preset Value	-3
Unit	Varies depending on the data format as follows: • Amplitude (MLOG):dB (decibel) • Phase (PHAS), Expanded phase (UPH),Positive phase (PPH): ° (degree) • Group delay (GDEL): s (second) • Others: No unit

NOTE 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.

Examples

```
Dim NotchVal As Double
SCPI.CALCulate(1).PARameter(1).SELect
SCPI.CALCulate(1).SELected.MARKer(1).NOTCH.THReshold = -6
NotchVal = SCPI.CALCulate(1).SELected.MARKer(1).NOTCH.THReshold
```

Related Objects

SCPI.CALCulate(Ch).SELected.MARKer.NOTCh.STATe

SCPI.CALCulate(Ch).SELected.MARKer(Mk).NOTCh.DATA

Equivalent key

Marker Search > **Notch Value**

Equivalent SCPI command

Syntax

:CALCulate{[1]-4}[:SELected]:MARKer{[1]-10}:NOTCh:THReshold <value>

:CALCulate{[1]-4}[:SELected]:MARKer{[1]-10}:NOTCh:THReshold?

Query response

{value}<newline><^END>

Example of use

```
10 OUTPUT 717;":CALC1:MARK1:NOTC :THR 5"
20 OUTPUT 717;":CALC1:MARK1:NOTC :THR?"
30 ENTER 717;A
```

SCPI.CALCulate(*Ch*).SELected.MARKer(*Mk*).SET

Object type

Property (**Write Only**)

Syntax

SCPI.CALCulate(*Ch*).SELected.MARKer(*Mk*).SET = *Param*

Description

This command sets the value at the position of marker 1 to 9 (*Mk*) and reference marker (*Mk*:10) to the value of the instrument setting item (*Param*), for the active trace of the selected channel (*Ch*).

Variable

Parameter	*Param*
Description	Instrument setting item
Data type	Character string type (String)
Range	Select from the following: • "STARt": Sets the sweep start value to the stimulus value at the marker position. • "STOP": Sets the sweep stop value to the stimulus value at the marker position. • "CENTer": Sets the sweep center value to the stimulus value at the marker position. • "RLEVel": Sets the reference line value to the response value at the marker position. • "DELay": Sets the electrical delay time value to the value of the group delay at the marker position (a value smoothed with the aperture of 20%).

Examples

```
Dim MkrTo As String
SCPI.CALCulate(1).PARameter(1).SELect
SCPI.CALCulate(1).SELected.MARKer(1).SET = "cent"
```

Related objects

SCPI.CALCulate(Ch).PARameter(Tr).SELect

SCPI.CALCulate(Ch).SELected.MARKer.REFerence.STATe

Equivalent key

Marker Fctn > Marker -> Start|Marker -> Stop|Marker -> Center|Marker -> Reference|Marker -> Delay

Equivalent SCPI command

Syntax

:CALCulate{[1]-4}[:SELected]:MARKer{[1]-10}:SET {STARt|STOP|CENTer|RLEVel|DELay}

Example of use

```
10 OUTPUT 717;":CALC1:MARK1:SET CENT"
```

SCPI.CALCulate(*Ch*).SELected.MARKer(*Mk*).STATe

Object type

Property (**Read-Write**)

Syntax

SCPI.CALCulate(*Ch*).SELected.MARKer(*Mk*).STATe = *Status*

Status = SCPI.CALCulate(*Ch*).SELected.MARKer(*Mk*).STATe

Description

This command turns ON/OFF the display of marker 1 to 9 (*Mk*) and reference marker (*Mk*:10), for the active trace of the selected channel (*Ch*).

Variable

Parameter	*Status*
Description	ON/OFF of display of markers 1 to 9 and reference marker
Data type	Boolean type (Boolean)
Range	Select from the following: • True or ON: Turns ON the display of the marker. • False or OFF: Turns OFF the display of the marker.
Preset value	False or OFF

Examples

```
Dim Mkr As Boolean
SCPI.CALCulate(1).PARameter(2).SELect
SCPI.CALCulate(1).SELected.MARKer(10).STATe = True
Mkr = SCPI.CALCulate(1).SELected.MARKer(10).STATe
```

Related objects

SCPI.CALCulate(Ch).PARameter(Tr).SELect

Equivalent key

When turning ON the display of the marker

Marker > **Marker 1|Marker 2|Marker 3|Marker 4|Ref Marker**

Marker > **More Markers** > **Marker 5|Marker 6|Marker 7|Marker 8|Marker 9**

When performing the operation from the front panel, a marker set to ON is automatically set to the *active marker* .

When turning OFF the display of the marker

Marker > **Clear Marker Menu** > **Marker 1|Marker 2|Marker 3|Marker 4|Marker 5|Marker 6|Marker 7|Marker 8|Marker 9|Ref Marker**

Equivalent SCPI command

Syntax

:CALCulate{[1]-4}[:SELected]:MARKer{[1]-10}[:STATe] {ON|OFF|1|0}

:CALCulate{[1]-4}[:SELected]:MARKer{[1]-10}[:STATe]?

Query response

{1|0}<newline><^END>

Example of use

```
10 OUTPUT 717;":CALC1:MARK1 ON"
20 OUTPUT 717;":CALC1:MARK1?"
30 ENTER 717;A
```

SCPI.CALCulate(*Ch*).SELected.MARKer(*Mk*).X

Object type

Property (**Read-Write**)

Syntax

SCPI.CALCulate(*Ch*).SELected.MARKer(*Mk*).X = *Value*

Value = SCPI.CALCulate(*Ch*).SELected.MARKer(*Mk*).X

Description

This command set the stimulus value for marker 1 to 9 (*Mk*) and reference marker (*Ch*:10), for the active trace of the selected channel (*Ch*).

Variable

Parameter	*Value*
Description	Stimulus value of the marker
Data type	Double precision floating point type (Double)
Range	Sweep start value to sweep stop value
Preset value	Sweep start value
Unit	Hz (hertz), dBm or s (second)
Note	If the specified variable 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.

Examples

```
Dim MkrX As Double
SCPI.CALCulate(1).PARameter(1).SELect
SCPI.CALCulate(1).SELected.MARKer(1).X = 1E9
MkrX = SCPI.CALCulate(1).SELected.MARKer(1).X
```

Related objects

SCPI.CALCulate(Ch).PARameter(Tr).SELect

SCPI.CALCulate(Ch).SELected.MARKer.REFerence.STATe

SCPI.CALCulate(Ch).SELected.MARKer(Mk).Y

Equivalent key

Marker > **Marker 1|Marker 2|Marker 3|Marker 4|Ref Marker**

Marker > **More Markers** > **Marker 5|Marker 6|Marker 7|Marker 8|Marker 9**

When performing the operation from the front panel, you turn ON the marker and set the stimulus value at the same time.

Equivalent SCPI command

Syntax

```
:CALCulate{[1]-4}[:SELected]:MARKer{[1]-10}:X <numeric>
:CALCulate{[1]-4}[:SELected]:MARKer{[1]-10}:X?
```

Query response

```
{numeric}<newline><^END>
```

Example of use

```
10 OUTPUT 717;":CALC1:MARK1:X 1E9"
20 OUTPUT 717;":CALC1:MARK1:X?"
30 ENTER 717;A
```

SCPI.CALCulate(*Ch*).SELected.MARKer(*Mk*).Y

Object type

Property (**Read Only**)

Syntax

Data = SCPI.CALCulate(*Ch*).SELected.MARKer(*Mk*).Y

Description

This command reads the response value of marker 1 to 9 (*Mk*) and reference marker (*Mk*:10), for the active trace of selected channel (*Ch*).

When the reference marker mode is ON ("True" is specified with the SCPI.CALCulate(Ch).SELected.MARKer.REFerence.STATe object), the readout value is the value relative to the reference marker.

Variable

Parameter	*Data*
Description	Indicates 2-element array data (response value of marker). • *Data(0)* :Response value (primary value) at the marker position. • *Data(1)* :Response value (secondary value) at the marker position. Always 0 when the data format is not the Smith chart format or the polar format. The index of the array starts from 0.
Data type	Variant type (Variant)

Examples

```
Dim MkrY As Variant
SCPI.CALCulate(1).PARameter(1).SELect
MkrY = SCPI.CALCulate(1).SELected.MARKer(1).Y
```

Related objects

SCPI.CALCulate(Ch).PARameter(Tr).SELect

SCPI.CALCulate(Ch).SELected.MARKer.REFerence.STATe

SCPI.CALCulate(Ch).SELected.MARKer(Mk).X

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

:CALCulate{[1]-4}[:SELected]:MARKer{[1]-10}:Y?

Query response

{numeric 1},{numeric 2}<newline><^END>

	Description
{numeric 1}	Response value (primary value) at the marker position.
{numeric 2}	Response value (secondary value) at the marker position. Always 0 when the data format is not the Smith chart format or the polar format.

Example of use

```
10 OUTPUT 717;":CALC1:MARK1:Y?"
30 ENTER 717;A,B
```

SCPI.CALCulate(*Ch*).SELected.MARKer.AOFF

Object Type

Method (**Write Only**)

Syntax

SCPI.CALCulate(*Ch*).SELected.MARKer.AOFF

Description

This command turns OFF all markers for the selected channel (*Ch*).

Examples

SCPI.CALCulate(4).SELected.MARKer.AOFF

Related Objects

SCPI.CALCulate(Ch).SELected.MARKer(Mk).STATe

Equivalent Key

Marker > **Clear Marker Menu** > **All OFF**

Equivalent SCPI Command

Syntax

:CALCulate{[1]-4}[:SELected]:MARKer:AOFF

Example of use

10 OUTPUT 717;":CALC1:MARK:AOFF"

SCPI.CALCulate(*Ch*).SELected.MARKer.BWIDth.STATe

Object type

Property (**Read-Write**)

Syntax

SCPI.CALCulate(*Ch*).SELected.MARKer.BWIDth.STATe = *Status*

Status = SCPI.CALCulate(*Ch*).SELected.MARKer.BWIDth.STATe

Description

This command turns ON/OFF the bandwidth search result display, for the active trace of the selected channel (*Ch*).

Variable

Parameter	*Status*
Description	ON/OFF of the bandwidth search result display
Data type	Boolean type (Boolean)
Range	Select from the following: • True or ON: Turns ON the bandwidth search result display. • False or OFF: Turns OFF the bandwidth search result display.
Preset value	False or OFF

Examples

```
Dim BandSrch As Boolean
SCPI.CALCulate(1).PARameter(1).SELect
SCPI.CALCulate(1).SELected.MARKer.BWIDth.STATe = True
BandSrch = SCPI.CALCulate(1).SELected.MARKer.BWIDth.STATe
```

Related objects

SCPI.CALCulate(Ch).PARameter(Tr).SELect

SCPI.CALCulate(Ch).SELected.MARKer(Mk).BWIDth.DATA

SCPI.CALCulate(Ch).SELected.MARKer(Mk).BWIDth.THReshold

Equivalent key

Marker Search > **Bandwidth**

Equivalent SCPI command

Syntax

```
:CALCulate{[1]-4}[:SELected]:MARKer:BWIDth[:STATe] {ON|OFF|1|0}
:CALCulate{[1]-4}[:SELected]:MARKer:BWIDth[:STATe]?
```

Query response

```
{1|0}<newline><^END>
```

Example of use

```
10 OUTPUT 717;":CALC1:MARK:BWID ON"
20 OUTPUT 717;":CALC1:MARK:BWID?"
30 ENTER 717;A
```

SCPI.CALCulate(*Ch*).SELected.MARKer.COUPle

Object type

Property (**Read-Write**)

Syntax

SCPI.CALCulate(*Ch*).SELected.MARKer.COUPle = *Status*

Status = SCPI.CALCulate(*Ch*).SELected.MARKer.COUPle

Description

This command turns ON/OFF the marker coupling between traces, for the selected channel (*Ch*).

Variable

Parameter	*Status*
Description	ON/OFF of the marker coupling between traces
Data type	Boolean type (Boolean)
Range	Select from the following: • True or ON: Turns ON the marker coupling. • False or OFF: Turns OFF the marker coupling.
Preset value	True or ON

Examples

```
Dim MkrCpl As Boolean
SCPI.CALCulate(1).SELected.MARKer.COUPle = False
MkrCpl = SCPI.CALCulate(1).SELected.MARKer.COUPle
```

Equivalent key

Marker Fctn > **Couple**

Equivalent SCPI command

Syntax

:CALCulate{[1]-4}[:SELected]:MARKer:COUPle {ON|OFF|1|0}

:CALCulate{[1]-4}[:SELected]:MARKer:COUPle?

Query response

{1|0}<newline><^END>

Example of use

```
10 OUTPUT 717;":CALC1:MARK:COUP OFF"
20 OUTPUT 717;":CALC1:MARK:COUP?"
30 ENTER 717;A
```

SCPI.CALCulate(*Ch*).SELected.MARKer.DISCrete

Object type

Property (**Read-Write**)

Syntax

SCPI.CALCulate(*Ch*).SELected.MARKer.DISCrete = *Status*

Status = SCPI.CALCulate(*Ch*).SELected.MARKer.DISCrete

Description

This command turns ON/OFF the discrete mode (mode in which the marker moves only at the measurement points) with marker 1 to 9 (*Mk*) and reference marker (*Mk*:10), for the active trace of selected channel (*Ch*).

Variable

Parameter	*Status*
Description	ON/OFF of the marker discrete mode
Data type	Boolean type (Boolean)
Range	Select from the following: • True or ON: Turns ON the discrete mode. • False or OFF: Turns OFF the discrete mode.
Preset value	False or OFF

Examples

```
Dim MkrDsc As Boolean
SCPI.CALCulate(1).PARameter(1).SELect
SCPI.CALCulate(1).SELected.MARKer(1).DISCrete = True
MkrDsc = SCPI.CALCulate(1).SELected.MARKer(1).DISCrete
```

Related objects

SCPI.CALCulate(Ch).PARameter(Tr).SELect

Equivalent key

Marker Fctn > **Discrete**

Equivalent SCPI command

Syntax

:CALCulate{[1]-4}[:SELected]:MARKer:DISCrete {ON|OFF|1|0}

:CALCulate{[1]-4}[:SELected]:MARKer:DISCrete?

Query response

{1|0}<newline><^END>

Example of use

```
10 OUTPUT 717;":CALC1:MARK:DISC OFF"
20 OUTPUT 717;":CALC1:MARK:DISC?"
30 ENTER 717;A
```

SCPI.CALCulate(*Ch*).SELected.MARKer.FUNCtion.DOMain.COUPle

Object type

Property (**Read-Write**)

Syntax

SCPI.CALCulate(*Ch*).SELected.MARKer.FUNCtion.DOMain.COUPle = *Status*

Status = SCPI.CALCulate(*Ch*).SELected.MARKer.FUNCtion.DOMain.COUPle

Description

This command sets/gets the coupling of the marker search range for all traces, for the selected channel (*Ch*).

Variable

Parameter	*Status*
Description	ON/OFF the trace coupling of the marker search range.
Data type	Boolean type (Boolean)
Range	Select from the following: • True or ON: Specifies the search range with the trace coupling. • False or OFF :Specifies the search range for each trace.
Preset value	True or ON

Examples

```
Dim TrCpl As Boolean
SCPI.CALCulate(1).SELected.MARKer.FUNCtion.DOMain.COUPle = False
TrCpl = SCPI.CALCulate(1).SELected.MARKer.FUNCtion.DOMain.COUPle
```

Related objects

SCPI.CALCulate(Ch).SELected.MARKer(Mk).FUNCtion.EXECute

Equivalent key

Marker Search > **Search Range** > **Couple**

Equivalent SCPI command

Syntax

:CALCulate{[1]-4}[:SELected]:MARKer:FUNCtion:DOMain:COUPle {ON|OFF|1|0}

:CALCulate{[1]-4}[:SELected]:MARKer:FUNCtion:DOMain:COUPle?

Query response

{1|0}<newline><^END>

Example of use

```
10 OUTPUT 717;":CALC1:MARK:FUNC:DOM:COUP OFF"
20 OUTPUT 717;":CALC1:MARK:FUNC:DOM:COUP?"
30 ENTER 717;A
```

SCPI.CALCulate(*Ch*).SELected.MARKer.FUNCtion.DOMain.STARt

Object type

Property (**Read-Write**)

Syntax

SCPI.CALCulate(*Ch*).SELected.MARKer.FUNCtion.DOMain.STARt = *Value*

Value = SCPI.CALCulate(*Ch*).SELected.MARKer.FUNCtion.DOMain.STARt

Description

This command sets/gets the start value of the marker search range, for the selected channel (*Ch*).

NOTE When the trace coupling is OFF, the active trace is the target to be set.

Variable

Parameter	*Value*
Description	The start value of the search range
Data type	Double precision floating point type (Double)
Preset value	0
Unit	Hz (hertz), dBm or s (second)

Examples

```
Dim SchStar As Double
SCPI.CALCulate(1).SELected.MARKer.FUNCtion.DOMain.STARt = 1.7E9
SchStar = SCPI.CALCulate(1).SELected.MARKer.FUNCtion.DOMain.STARt
```

Related objects

SCPI.CALCulate(Ch).SELected.MARKer.FUNCtion.DOMain.STOP

SCPI.CALCulate(Ch).SELected.MARKer.FUNCtion.DOMain.STATe

SCPI.CALCulate(Ch).SELected.MARKer(Mk).FUNCtion.EXECute

Equivalent key

Marker Search > **Search Range** > **Start**

Equivalent SCPI command

Syntax

:CALCulate{[1]-4}[:SELected]:MARKer:FUNCtion:DOMain:STARt <numeric>

:CALCulate{[1]-4}[:SELected]:MARKer:FUNCtion:DOMain:STARt?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":CALC1:MARK:FUNC:DOM:STAR 1.7E9"
20 OUTPUT 717;":CALC1:MARK:FUNC:DOM:STAR?"
30 ENTER 717;A
```

SCPI.CALCulate(*Ch*).SELected.MARKer.FUNCtion.DOMain.STATe

Object type

Property (**Read-Write**)

Syntax

SCPI.CALCulate(*Ch*).SELected.MARKer.FUNCtion.DOMain.STATe = *Status*

Status = SCPI.CALCulate(*Ch*).SELected.MARKer.FUNCtion.DOMain.STATe

Description

This command sets/gets whether to use an arbitrary range when executing the marker search, for the selected channel (*Ch*).

NOTE When the trace coupling is OFF, the active trace is the target to be set.

Variable

Parameter	*Status*
Description	Selects the search range.
Data type	Boolean type (Boolean)
Range	Select from the following: • True or ON: Specifies an arbitrary range. • False or OFF: Specifies the entire sweep range.
Preset value	False or OFF

Examples

```
Dim SchRnge As Boolean
SCPI.CALCulate(1).SELected.MARKer.FUNCtion.DOMain.STARt = 1.5E9
SCPI.CALCulate(1).SELected.MARKer.FUNCtion.DOMain.STOP = 1.8E9
SCPI.CALCulate(1).SELected.MARKer.FUNCtion.DOMain.STATe = True
SchRnge = SCPI.CALCulate(1).SELected.MARKer.FUNCtion.DOMain.STATe
```

Related objects

SCPI.CALCulate(Ch).SELected.MARKer.FUNCtion.DOMain.STARt

SCPI.CALCulate(Ch).SELected.MARKer.FUNCtion.DOMain.STOP

SCPI.CALCulate(Ch).SELected.MARKer(Mk).FUNCtion.EXECute

Equivalent key

Marker Search > Search Range > Search Range [ON/OFF]

Equivalent SCPI command

Syntax

:CALCulate{[1]-4}[:SELected]:MARKer:FUNCtion:DOMain[:STATe] {ON|OFF|1|0}

:CALCulate{[1]-4}[:SELected]:MARKer:FUNCtion:DOMain[:STATe]?

Query response

{1|0}<newline><^END>

Example of use

```
10 OUTPUT 717;":CALC1:MARK:FUNC:DOM ON"
20 OUTPUT 717;":CALC1:MARK:FUNC:DOM?"
30 ENTER 717;A
```

SCPI.CALCulate(*Ch*).SELected.MARKer.FUNCtion.DOMain.STOP

Object type

Property (**Read-Write**)

Syntax

SCPI.CALCulate(*Ch*).SELected.MARKer.FUNCtion.DOMain.STOP = *Value*

Value = SCPI.CALCulate(*Ch*).SELected.MARKer.FUNCtion.DOMain.STOP

Description

This command sets/gets the stop value of the marker search range, for the selected channel (*Ch*).

NOTE When the trace coupling is OFF, the active trace is the target to be set.

Variable

Parameter	*Value*
Description	Stop value of the search range
Data type	Double precision floating point type (Double)
Preset value	0
Unit	Hz (hertz), dBm or s (second)

Examples

```
Dim SchStop As Double
SCPI.CALCulate(1).SELected.MARKer.FUNCtion.DOMain.STOP = 1.8E9
SchStop = SCPI.CALCulate(1).SELected.MARKer.FUNCtion.DOMain.STOP
```

Related objects

SCPI.CALCulate(Ch).SELected.MARKer.FUNCtion.DOMain.STARt

SCPI.CALCulate(Ch).SELected.MARKer.FUNCtion.DOMain.STATe

SCPI.CALCulate(Ch).SELected.MARKer(Mk).FUNCtion.EXECute

Equivalent key

Marker Search > **Search Range** > **Stop**

Equivalent SCPI command

Syntax

:CALCulate{[1]-4}[:SELected]:MARKer:FUNCtion:DOMain:STOP <numeric>

:CALCulate{[1]-4}[:SELected]:MARKer:FUNCtion:DOMain:STOP?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":CALC1:MARK:FUNC:DOM:STOP 1.8E9"
20 OUTPUT 717;":CALC1:MARK:FUNC:DOM:STOP?"
30 ENTER 717;A
```

SCPI.CALCulate(*Ch*).SELected.MARKer.FUNCtion.MULTi.PEXCursion

Object Type

Property (**Read-Write**)

Syntax

SCPI.CALCulate(*Ch*).SELected.MARKer.FUNCtion.MULTi.PEXCursion = *Value*

Value = SCPI.CALCulate(*Ch*).SELected.MARKer.FUNCtion.MULTi.PEXCursion

Description

This command sets/gets the lower limit of peak excursion value for the selected channel (*Ch*) when executing the multi peak search.

Variable

Parameter	*Value*
Description	Lower limit of peak excursion value
Data Type	Double precision floating point type (Double)
Range	0 to 5E8
Preset Value	3
Unit	Varies depending on the data format. • Log magnitude (MLOG): dB (decibel) • Phase (PHAS), Expanded phase (UPH) or Positive phase (PPH): × (degree) • Group delay (GDEL): s (second) • Others: No unit
Note	If the specified variable 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.

Examples

```
Dim PeakExc as Double

SCPI.CALCulate4.SELected.MARKer.FUNCtion.MULTi.PEXCursion = 0.2
PeakExc = SCPI.CALCulate4.SELected.MARKer.FUNCtion.MULTi.PEXCursion
```

Related Objects
Equivalent Key

Marker Search > **Multi Peak** > **Peak Excursion**

Equivalent SCPI Command

Syntax

:CALCulate{[1]-4}[:SELected]:MARKer:FUNCtion:MULTi:PEXCursion <numeric>

:CALCulate{[1]-4}[:SELected]:MARKer:FUNCtion:MULTi:PEXCursion?

Query Response

{numeric} <newline><^END>

Example of use

```
10 OUTPUT 717;":CALC1:MARK:FUNC:MULT:PEXC 0"
20 OUTPUT 717;":CALC1:MARK:FUNC:MULT:PEXC?"
30 ENTER 717;A
```

SCPI.CALCulate(*Ch*).SELected.MARKer.FUNCtion.MULTi.PPOLarity

Object Type

Property (**Read-Write**)

Syntax

SCPI.CALCulate(*Ch*).SELected.MARKer.FUNCtion.MULTi.PPOLarity = *Param*

Param = SCPI.CALCulate(*Ch*).SELected.MARKer.FUNCtion.MULTi.PPOLarity

Description

This command sets/gets the polarity of the multi peak search, for the selected channel (*Ch*).

Variable

Parameter	*Param*
Description	Polarity for multi peak search
Data Type	Character string type (String)
Range	Select from the following: • "POSitive": Specifies the positive peak. • "NEGative": Specifies the negative peak. • "BOTH": Specifies both the positive peak and the negative peak.
Preset Value	"POSitive"

Examples

```
Dim PeakPol as String
SCPI.CALCulate(4).SELected.MARKer.FUNCtion.MULTi.PPOLarity = "both"
PeakPol = SCPI.CALCulate4.SELected.MARKer.FUNCtion.MULTi.PPOLarity
```

Related Objects

Equivalent Key

Marker Search > **Multi Peak** > **Peak Polarity** > **Positive|Negative|Both**

Equivalent SCPI Command

Syntax

:CALCulate{[1]-4}[:SELected]:MARKer:FUNCtion:MULTi:PPOLarity {POSitive|NEGative|BOTH}

:CALCulate{[1]-4}[:SELected]:MARKer:FUNCtion:MULTi:PPOLarity?

Query Response

{POS|NEG|BOTH} <newline><^END>

Example of use

```
10 OUTPUT 717;":CALC1:MARK:FUNC:MULT:PPOL NEG"
20 OUTPUT 717;":CALC1:MARK:FUNC:MULT:PPOL?"
30 ENTER 717;A$
```

SCPI.CALCulate(*Ch*).SELected.MARKer.FUNCtion.MULTi.TARGet

Object Type

Property (**Read-Write**)

Syntax

SCPI.CALCulate(*Ch*).SELected.MARKer.FUNCtion.MULTi.TARGet = *Value*

Value = SCPI.CALCulate(*Ch*).SELected.MARKer.FUNCtion.MULTi.TARGet

Description

This command sets/gets the the target value to be searched with the multi target search function, for the selected channel (*Ch*).

Variable

Parameter	*Value*
Description	Target value for multi target search
Data Type	Double precision floating point type (Double)
Range	-5E8 to 5E8
Preset Value	0
Unit	Varies depending on the data format. • Log magnitude (MLOG): dB (decibel) • Phase (PHAS), Expanded phase (UPH) or Positive phase (PPH): ° (degree) • Group delay (GDEL): s (second) • Others: No unit
Note	If the specified variable 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.

Examples

```
Dim TargVal as Double
SCPI.CALCulate(4).SELected.MARKer.FUNCtion.MULTi.TARGet = 12.5
TargVal = SCPI.CALCulate(4).SELected.MARKer.FUNCtion.MULTi.TARGet
```

Related Objects

Equivalent Key

Marker Search > **Multi Target** > **Target Value**

Equivalent SCPI Command

Syntax

:CALCulate{[1]-4}[:SELected]:MARKer:FUNCtion:MULTi:TARGet <numeric>

:CALCulate{[1]-4}[:SELected]:MARKer:FUNCtion:MULTi:TARGet?

Query Response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":CALC1:MARK:FUNC:MULT:TARG 5"
20 OUTPUT 717;":CALC1:MARK:FUNC:MULT:TARG?"
30 ENTER 717;A
```

SCPI.CALCulate(*Ch*).SELected.MARKer.FUNCtion.MULTi.TRACking

Object Type

Property (**Read-Write**)

Syntax

SCPI.CALCulate(*Ch*).SELected.MARKer.FUNCtion.MULTi.TRACking = *Status*

Status = SCPI.CALCulate(*Ch*).SELected.MARKer.FUNCtion.MULTi.TRACking

Description

This command turns ON/OFF the search tracking (function to repeat search for each sweep) of the multi search, for the selected channel (*Ch*).

Variable

Parameter	*Status*
Description	ON/OFF of the marker search tracking
Data Type	Boolean type (Boolean)
Range	Select from either of the following: • True or ON: Turns ON the notch search result display. • False or OFF: Turns OFF the notch search result display.
Preset Value	False or OFF

Examples

```
Dim StsTrac as Boolean
SCPI.CALCulate(4).SELected.MARKer.FUNCtion.MULTi.TRACking = True
StsTrac = SCPI.CALCulate(4).SELected.MARKer.FUNCtion.MULTi.TRACking
```

Related Objects

Equivalent Key

Marker Search > **Tracking**

Equivalent SCPI Command

Syntax

:CALCulate{[1]-4}[:SELected]:MARKer:FUNCtion:MULTi:TRACking {ON|OFF|1|0}

:CALCulate{[1]-4}[:SELected]:MARKer:FUNCtion:MULTi:TRACking?

Query Response

{1|0} <newline><^END>

Example of use

```
10 OUTPUT 717;":CALC1:MARK:FUNC:MULT:TRAC ON"
20 OUTPUT 717;":CALC1:MARK:FUNC:MULT:TRAC?"
30 ENTER 717;A
```

SCPI.CALCulate(*Ch*).SELected.MARKer.FUNCtion.MULTi.TTRansition

Object Type

Property (**Read-Write**)

Syntax

SCPI.CALCulate(*Ch*).SELected.MARKer.FUNCtion.MULTi.TTRansition = *Param*

Param = SCPI.CALCulate(*Ch*).SELected.MARKer.FUNCtion.MULTi.TTRansition

Description

This command sets/gets the transition type of the multi target search, for the selected channel (*Ch*).

Variable

Parameter	*Param*
Description	Transition type of multi target search
Data Type	Character string type (String)
Range	Select from the following: • "POSitive": Specifies the positive transition. • "NEGative": Specifies the negative transition. • "BOTH": Specifies both the positive transition and the negative transition.
Preset Value	"BOTH"

Examples

```
Dim TargTran as String
SCPI.CALCulate(4).SELected.MARKer.FUNCtion.MULTi.TTRansition = "neg"
TargTran = SCPI.CALCulate(4).SELected.MARKer.FUNCtion.MULTi.TTRansition
```

Related Objects

Equivalent Key

Marker Search > **Multi Target** > **Target Transition** > **Positive|Negative|Both**

Equivalent SCPI Command

Syntax

:CALCulate{[1]-4}[:SELected]:MARKer:FUNCtion:MULTi:TTRansition {POSitive|NEGative|BOTH}

:CALCulate{[1]-4}[:SELected]:MARKer:FUNCtion:MULTi:TTRansition?

Query Response

{POS|NEG|BOTH} <newline><^END>

Example of use

```
10 OUTPUT 717;":CALC1:MARK:FUNC:MULT:TTR POS"
20 OUTPUT 717;":CALC1:MARK:FUNC:MULT:TTR?"
30 ENTER 717;A$
```

SCPI.CALCulate(*Ch*).SELected.MARKer.FUNCtion.MULTi.TYPE

Object Type

Property (**Read-Write**)

Syntax

SCPI.CALCulate(*Ch*).SELected.MARKer.FUNCtion.MULTi.TYPE = *Param*

Param = SCPI.CALCulate(*Ch*).SELected.MARKer.FUNCtion.MULTi.TYPE

Description

This command sets/gets the search type of the multi search, for the selected channel (*Ch*).

Variable

Parameter	*Param*
Description	Search type of the multi search
Data Type	Character string type (String)
Range	Select from the following: • "OFF": Turn OFF the multi search function. • "PEAK": Sets the search type to the multi peak search. • "TARGet": Sets the search type to the multi target search.
Preset Value	"OFF"

Examples

```
Dim SrchType as String
SCPI.CALCulate(4).SELected.MARKer.FUNCtion.MULTi.TYPE = "targ"
SrchType = SCPI.CALCulate(4).SELected.MARKer.FUNCtion.MULTi.TYPE
```

Related Objects

Equivalent Key

Marker Search > **Multi Peak** > **Search Multi Peak**

Marker Search > **Multi Target** > **Search Multi Target**

NOTE When performing the operation from the front panel, you select the search type and execute the search at the same time.

Equivalent SCPI Command

Syntax

:CALCulate{[1]-4}[:SELected]:MARKer:FUNCtion:MULTi:TYPE {OFF|PEAK|TARGet}

:CALCulate{[1]-4}[:SELected]:MARKer:FUNCtion:MULTi:TYPE?

Query Response

{OFF|PEAK|TARG} <newline><^END>

Example of use

```
10 OUTPUT 717;":CALC1:MARK:FUNC:MULT:TYPE TARG"
20 OUTPUT 717;":CALC1:MARK:FUNC:MULT:TYPE?"
30 ENTER 717;A$
```

SCPI.CALCulate(*Ch*).SELected.MARKer.MATH.FLATness.DATA

Object Type

Property (**Read Only**)

Syntax

Data = SCPI.CALCulate(*Ch*).SELected.MARKer.MATH.FLATness.DATA

Description

This command gets the marker flatness values of the active trace, for the selected channel (*Ch*).

Variable

Parameter	*Data*
Description	Indicates 4-elements array data (flatness value). • *Data(0)* :Span • *Data(1)* :Gain • *Data(2)* :Slope • *Data(3)* :Flatness The index of the array starts from 0.
Data Type	Variant type (Variant)

Examples

```
Dim FlatData as Variant
FlatData = SCPI.CALCulate(4).SELected.MARKer.MATH.FLATness.DATA
```

Related Objects

SCPI.CALCulate(Ch).SELected.MARKer.MATH.FLATness.STATe

Equivalent Key

No equivalent key is available on the front panel.

Equivalent SCPI Command

Syntax

:CALCulate{[1]-4}[:SELected]:MARKer:MATH:FLATness:DATA?

Query Response

{numeric 1},{numeric 2},{numeric 3},{numeric 4}<newline><^END>

Parameter	**Description**

{numeric 1}	Span value
{numeric 2}	Gain value
{numeric 3}	Slope value
{numeric 4}	Flatness value

Example of use

```
10 OUTPUT 717;":CALC1:MARK:MATH:FLAT:DATA?"
30 ENTER 717;A$
```

SCPI.CALCulate(*Ch*).SELected.MARKer.MATH.FLATness.STATe

Object Type

Property (**Read-Write**)

Syntax

SCPI.CALCulate(*Ch*).SELected.MARKer.MATH.FLATness.STATe = *Status*

Status = SCPI.CALCulate(*Ch*).SELected.MARKer.MATH.FLATness.STATe

Description

This command turns ON/OFF the marker flatness values display, for the selected channel (*Ch*).

Variable

Parameter	*Status*
Description	ON/OFF of the marker flatness value display
Data Type	Boolean type (Boolean)
Range	Select from the following: • True or ON: Turns ON the marker flatness value display. • False or OFF: Turns OFF the marker flatness value display.
Preset Value	False or OFF

Examples

```
Dim FlatMode as Boolean
SCPI.CALCulate(4).SELected.MARKer.MATH.FLATness.STATe = True
FlatMode = SCPI.CALCulate4.SELected.MARKer.MATH.FLATness.STATe
```

Related Objects

SCPI.CALCulate(Ch).SELected.MARKer.MATH.FLATness.DATA

Equivalent Key

Marker Fctn > **Flatness**

Equivalent SCPI Command

Syntax

:CALCulate{[1]-4}[:SELected]:MARKer:MATH:FLATness[:STATe] {ON|OFF|1|0}

:CALCulate{[1]-4}[:SELected]:MARKer:MATH:FLATness[:STATe]?

Query Response

{1|0} <newline><^END>

Example of use

```
10 OUTPUT 717;":CALC1:MARK1:MATH:FLAT ON"
20 OUTPUT 717;":CALC1:MARK1:MATH:FLAT?"
30 ENTER 717;A
```

SCPI.CALCulate(*Ch*).SELected.MARKer.MATH.FSTatistics.DATA

Object Type

Property (**Read Only**)

Syntax

Data = SCPI.CALCulate(*Ch*).SELected.MARKer.MATH.FSTatistics.DATA

Description

This command gets the RF filter statistics values of the active trace, for the selected channel (*Ch*).

Variable

Parameter	*Data*
Description	Indicates 3-element array data (RF filter statistics value). • *Data(0)* :Loss • *Data(1)* :Ripple (peak to peak) • *Data(2)* :Attenuation The index of the array starts from 0.
Data Type	Variant type (Variant)

Examples

```
Dim FstData as Variant
FstData = SCPI.CALCulate(4).SELected.MARKer.MATH.FSTatistics.DATA
```

Related Objects

SCPI.CALCulate(Ch).SELected.MARKer.MATH.FSTatistics.STATe

Equivalent Key

No equivalent key is available on the front panel.

Equivalent SCPI Command

Syntax

:CALCulate{[1]-4}[:SELected]:MARKer:MATH:FSTatistics:DATA?

Query Response

{numeric 1},{numeric 2},{numeric 3}<newline><^END>

Parameter	**Description**
{numeric 1}	Loss value

{numeric 2}	Ripple value (peak to peak)
{numeric 3}	Attenuation value

Example of use

```
10 OUTPUT 717;":CALC1:MARK:MATH:FST:DATA ?"
30 ENTER 717;A
```

SCPI.CALCulate(*Ch*).SELected.MARKer.MATH.FSTatistics.STATe

Object Type

Property (**Read-Write**)

Syntax

SCPI.CALCulate(*Ch*).SELected.MARKer.MATH.FSTatistics.STATe = *Status*

Status = SCPI.CALCulate(*Ch*).SELected.MARKer.MATH.FSTatistics.STATe

Description

This command turns ON/OFF the RF filter statistics values display, for the selected channel (*Ch*).

Variable

Parameter	*Status*
Description	ON/OFF of the RF filter statistics values display
Data Type	Boolean type (Boolean)
Range	Select from the following: • True or ON: Turns ON the RF filter statistics value display. • False or OFF: Turns OFF the RF filter statistics value display.
Preset Value	False or OFF

Examples

```
Dim FstMode as Boolean
SCPI.CALCulate(4).SELected.MARKer.MATH.FSTatistics.STATe = True
FstMode = SCPI.CALCulate(4).SELected.MARKer.MATH.FSTatistics.STATe
```

Related Objects

SCPI.CALCulate(Ch).SELected.MARKer.MATH.FSTatistics.DATA

Equivalent Key

Marker Fctn > **RF Filter Stats**

Equivalent SCPI Command

Syntax

:CALCulate{[1]-4}[:SELected]:MARKer:MATH:FSTatistics[:STATe] {ON|OFF|1|0}

:CALCulate{[1]-4}[:SELected]:MARKer:MATH:FSTatistics[:STATe]?

Query Response

{1|0} <newline><^END>

Example of use

```
10 OUTPUT 717;":CALC1:MARK:MATH:FST ON"
20 OUTPUT 717;":CALC1:MARK:MATH:FST?"
30 ENTER 717;A
```

SCPI.CALCulate(*Ch*).SELected.MARKer.MATH.STATistics.DATA

Object Type

Property (**Read Only**)

Syntax

Data = SCPI.CALCulate(*Ch*).SELected.MARKer.MATH.STATistics.DATA

Description

This command gets the statistics values of the active trace, for the selected channel (*Ch*).

Variable

Parameter	*Data*
Description	Indicates 4-element array data (statistics value). • *Data(0)* :Span value • *Data(1)* :Mean value • *Data(2)* :Standard deviation • *Data(3)* :Difference between the maximum value and the minimum value (Peak to Peak) The index of the array starts from 0.
Data Type	Variant type (Variant)

Examples

```
Dim StatData as Variant
StatData = SCPI.CALCulate(4).SELected.MARKer.MATH.STATistics.DATA
```

Related Objects

SCPI.CALCulate(Ch).SELected.MARKer.MATH.STATistics.STATe

Equivalent Key

No equivalent key is available on the front panel.

Equivalent SCPI Command

Syntax

:CALCulate{[1]-4}[:SELected]:MARKer:MATH:STATistics:DATA?

Query Response

{numeric 1},{numeric 2},{numeric 3},{numeric 4}<newline><^END>

Parameter	Description

{numeric 1}	Span value
{numeric 2}	Mean value
{numeric 3}	Standard deviation value
{numeric 4}	Peak to Peak Value

Example of use

```
10 OUTPUT 717;":CALC1:MARK:MATH:STAT:DATA ?"
30 ENTER 717;A
```

SCPI.CALCulate(*Ch*).SELected.MARKer.MATH.STATistics.STATe

Object Type

Property (**Read-Write**)

Syntax

SCPI.CALCulate(*Ch*).SELected.MARKer.MATH.STATistics.STATe = *Status*

Status = SCPI.CALCulate(*Ch*).SELected.MARKer.MATH.STATistics.STATe

Description

This command sets/gets turns ON/OFF the statistics values display, for the selected channel (*Ch*).

Variable

Parameter	*Status*
Description	ON/OFF of the statistics values display
Data Type	Boolean type (Boolean)
Range	Select from the following: • True or ON: Turns ON the statistics value display. • False or OFF: Turns OFF the statistics value display.
Preset Value	False or OFF

Examples

```
Dim StatMode as Boolean
SCPI.CALCulate(4).SELected.MARKer.MATH.STATistics.STATe = True
StatMode = SCPI.CALCulate(4).SELected.MARKer.MATH.STATistics.STATe
```

Related Objects

SCPI.CALCulate(Ch).SELected.MARKer.MATH.STATistics.DATA

Equivalent Key

Marker Fctn > **Statistics**

Equivalent SCPI Command

Syntax

:CALCulate{[1]-4}[:SELected]:MARKer:MATH:STATistics[:STATe] {ON|OFF|1|0}

:CALCulate{[1]-4}[:SELected]:MARKer:MATH:STATistics[:STATe]?

Query Response

{1|0} <newline><^END>

Example of use

```
10 OUTPUT 717;":CALC1:MARK:MATH:STAT ON"
20 OUTPUT 717;":CALC1:MARK:MATH:STAT?"
30 ENTER 717;A
```

SCPI.CALCulate(*Ch*).SELected.MARKer.NOTCh.STATe

Object type

Property (**Read-Write**)

Syntax

SCPI.CALCulate(*Ch*).SELected.MARKer.NOTCh.STATe = *Status*

Status = SCPI.CALCulate(*Ch*).SELected.MARKer.NOTCh.STATe

Description

This command turns ON/OFF the notch search result display, for the active trace of the selected channel (*Ch*).

Variable

Parameter	*Status*
Description	ON/OFF of the notch search result display
Data type	Boolean type (Boolean)
Range	Select from either of the following: • True or ON: Turns ON the notch search result display. • False or OFF: Turns OFF the notch search result display.
Preset value	False or OFF

Examples

```
Dim NotchSrch As Boolean
SCPI.CALCulate(1).PARameter(1).SELect
SCPI.CALCulate(1).SELected.MARKer.NOTCh.STATe = True
NotchSrch = SCPI.CALCulate(1).SELected.MARKer.NOTCh.STATe
```

Related Objects

SCPI.CALCulate(Ch).PARameter(Tr).SELect

SCPI.CALCulate(Ch).SELected.MARKer(Mk).NOTCh.DATA

SCPI.CALCulate(Ch).SELected.MARKer(Mk).NOTCh.THReshold

Equivalent key

Marker Search > **Notch**

Equivalent SCPI command

Syntax

```
:CALCulate{[1]-4}[:SELected]:MARKer:NOTCh[:STATe] {ON|OFF|1|0}
:CALCulate{[1]-4}[:SELected]:MARKer:NOTCh[:STATe]?
```

Query response

```
{1|0}<newline><^END>
```

Example of use

```
10 OUTPUT 717;":CALC1:MARK:NOTC ON"
20 OUTPUT 717;":CALC1:MARK:NOTC?"
30 ENTER 717;A
```

SCPI.CALCulate(*Ch*).SELected.MARKer.REFerence.STATe

Object type

Property (**Read-Write**)

Syntax

SCPI.CALCulate(*Ch*).SELected.MARKer.REFerence.STATe = *Status*

Status = SCPI.CALCulate(*Ch*).SELected.MARKer.REFerence.STATe

Description

Turns ON/OFF the reference marker mode for the active trace of the selected channel (*Ch*).

Variable

Parameter	*Status*
Description	ON/OFF of the reference marker mode
Data type	Boolean type (Boolean)
Range	Select from the following: • True or ON: Turns ON the reference marker mode. • False or OFF: Turns OFF the reference marker mode.
Preset value	False or OFF

Examples

```
Dim RefMode As Boolean
SCPI.CALCulate(1).PARameter(1).SELect
SCPI.CALCulate(1).SELected.MARKer.REFerence.STATe = True
RefMode = SCPI.CALCulate(1).SELected.MARKer.REFerence.STATe
```

Related objects

SCPI.CALCulate(Ch).PARameter(Tr).SELect

Equivalent key

Marker > **Ref Marker Mode**

Equivalent SCPI command

Syntax

:CALCulate{[1]-4}[:SELected]:MARKer:REFerence[:STATe] {ON|OFF|1|0}

:CALCulate{[1]-4}[:SELected]:MARKer:REFerence[:STATe]?

Query response

{1|0}<newline><^END>

Example of use

```
10 OUTPUT 717;":CALC1:MARK:REF ON"
20 OUTPUT 717;":CALC1:MARK:REF?"
30 ENTER 717;A
```

SCPI.CALCulate(*Ch*).SELected.MATH.FUNCtion

Object type

Property (**Read-Write**)

Syntax

SCPI.CALCulate(*Ch*).SELected.MATH.FUNCtion = *Param*

Param = SCPI.CALCulate(*Ch*).SELected.MATH.FUNCtion

Description

This command sets/gets the data trace display method (math method between measurement data and memory trace data), for the active trace of the selected channel (*Ch*).

NOTE The math result according to this setting is displayed on the data trace.

Variable

Parameter	*Param*
Description	Math method between measurement data and memory trace data
Data type	Character string type (String)
Range	Select from the following: • "NORMal": Specifies *Data* (no math). • "DIVide": Specifies *Data / Mem*. • "MULTiply": Specifies *Data* × *Mem*. • "SUBTract": Specifies *Data - Mem*. • "ADD": Specifies *Data* + *Mem*. Where *Data* is the measurement data (corrected data array) and *Mem* is the data stored in the memory trace (corrected memory array).
Preset value	"NORMal"

Examples

```
Dim MathFunc As String
SCPI.CALCulate(1).PARameter(1).SELect
```

```
SCPI.CALCulate(1).SELected.MATH.FUNCtion = "div"
MathFunc = SCPI.CALCulate(1).SELected.MATH.FUNCtion
```

Related objects

SCPI.CALCulate(Ch).PARameter(Tr).SELect

Equivalent key

Display > **Data Math** > **OFF** | **Data / Mem** | **Data * Mem** |**Data - Mem** | **Data + Mem**

Equivalent SCPI command

Syntax

:CALCulate{[1]-4}[:SELected]:MATH:FUNCtion {NORMal|SUBTract|DIVide|ADD|MULTiply}

:CALCulate{[1]-4}[:SELected]:MATH:FUNCtion?

Query response

{NORM|DIV|MULT|SUBT|ADD}<newline><^END>

Example of use

```
10 OUTPUT 717;":CALC1:MATH:FUNC DIV"
20 OUTPUT 717;":CALC1:MATH:FUNC?"
30 ENTER 717;A$
```

SCPI.CALCulate(*Ch*).SELected.MATH.MEMorize

Object type

Method (**Write Only**)

Syntax

SCPI.CALCulate(*Ch*).SELected.MATH.MEMorize

Description

This command copies the measurement data to the memory trace, for the active trace of selected channel (*Ch*).

Examples

```
SCPI.CALCulate(1).PARameter(1).SELect
SCPI.CALCulate(1).SELected.MATH.MEMorize
```

Related objects

SCPI.CALCulate(Ch).PARameter(Tr).SELect

Equivalent key

Display > **Data -> Mem**

Equivalent SCPI command

Syntax

:CALCulate{[1]-4}[:SELected]:MATH:MEMorize

Example of use

```
10 OUTPUT 717;":CALC1:MATH:MEM"
```

SCPI.CALCulate(*Ch*).SELected.MSTatistics.DATA

Object type

Property (**Read Only**)

Syntax

Data = SCPI.CALCulate(*Ch*).SELected.MSTatistics.DATA

Description

This command reads the statistics values of the active trace of t selected channel (*Ch*). The statistical values contain: mean value, standard deviation and the difference between the maximum value and the minimum value.

Variable

Parameter	*Data*
Description	Indicates 3-element array data (statistics value). • *Data(0)* :Mean value • *Data(1)* :Standard deviation • *Data(2)* :Difference between the maximum value and the minimum value (Peak to Peak) The index of the array starts from 0.
Data type	Variant type (Variant)

Examples

```
Dim MstData As Variant
SCPI.CALCulate(1).PARameter(1).SELect
MstData = SCPI.CALCulate(1).SELected.MSTatistics.DATA
```

Related objects

SCPI.CALCulate(Ch).PARameter(Tr).SELect

SCPI.CALCulate(Ch).SELected.MSTatistics.STATe

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

:CALCulate{[1]-4}[:SELected]:MSTatistics:DATA?

Query response

{numeric 1},{numeric 2},{numeric 3}<newline><^END>

Parameter	Description
{numeric 1}	Mean value
{numeric 2}	Standard deviation
{numeric 3}	Difference between the maximum value and the minimum value (Peak to Peak)

Example of use

```
10 OUTPUT 717;":CALC1:MST:DATA?"
20 ENTER 717;A,B,C
```

SCPI.CALCulate(*Ch*).SELected.MSTatistics.STATe

Object type

Property (**Read-Write**)

Syntax

SCPI.CALCulate(*Ch*).SELected.MSTatistics.STATe = *Status*

Status = SCPI.CALCulate(*Ch*).SELected.MSTatistics.STATe

Description

This command turns ON/OFF the statistics values display, for the active trace of selected channel (*Ch*). The statistical values contain: mean value, standard deviation and the difference between the maximum value and the minimum value.

Variable

Parameter	*Status*
Description	ON/OFF of the statistics value display
Data type	Boolean type (Boolean)
Range	Select from the following: • True or ON: Turns ON the statistics value display. • False or OFF: Turns OFF the statistics value display.
Preset value	False or OFF

Examples

```
Dim Mst As Boolean
SCPI.CALCulate(1).PARameter(1).SELect
SCPI.CALCulate(1).SELected.MSTatistics.STATe = True
Mst = SCPI.CALCulate(1).SELected.MSTatistics.STATe
```

Related objects

SCPI.CALCulate(Ch).PARameter(Tr).SELect

SCPI.CALCulate(Ch).SELected.MSTatistics.DATA

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

:CALCulate{[1]-4}[:SELected]:MSTatistics[:STATe] {ON|OFF|1|0}

:CALCulate{[1]-4}[:SELected]:MSTatistics[:STATe]?

Query response

{1|0}<newline><^END>

Example of use

```
10 OUTPUT 717;":CALC1:MST ON"
20 OUTPUT 717;":CALC1:MST?"
30 ENTER 717;A
```

SCPI.CALCulate(*Ch*).SELected.RLIMit.DATA

Object type

Property (**Read-Write**)

Syntax

SCPI.CALCulate(*Ch*).SELected.RLIMit.DATA = *Data*

Data = SCPI.CALCulate(*Ch*).SELected.RLIMit.DATA

Description

This command sets/gets the ripple limit table for the active trace (specified with the SCPI.CALCulate(Ch).PARameter(Tr).SELect command) of the selected channel (*Ch*).

The data transfer format when this command is executed depends on the setting with the SCPI.FORMat.DATA command.

Variable

Parameter	*Data*
Description	Indicates the array data (for ripple line) of 1 + Num (number of limit lines)×4. Where n is an integer between 1 and Num. • *Data(0)* :The number of limit lines you want to set. Specify an integer ranging from 0 to 12. When the number of limit lines is set to 0 (clears the limit table), the variable Data is only required with *Data(0)*. • *Data(n×4-3)* :The type of the n-th line. Specify an integer 0 to 1 as follows. 0: OFF 1: ON • *Data(n×4-2)* :The value on the horizontal axis (frequency/power/time) of the start point of the n-th line. • *Data(n×4-1)* :The value on the horizontal axis (frequency/power/time) of the end point of the n-th line. • *Data(n×4)* :The ripple line value (dB) of the n-th line. The index of the array starts from 0.
Data type	Variant type (Variant)

Note	If there is no array data of 1+Num (number of set lines))×4 when setting a formatted memory array, an error occurs when executed and the object is ignored. For *Data(n×4-3)* in the array data, if you specify an integer other than 0 or 1, an error occurs when executed. For *Data(n×4-2)* and *Data(n×4-1)* in the array data, if the specified value 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.

Examples (1)

```
Dim RLimData As Variant
SCPI.CALCulate(1).PARameter(1).SELect
SCPI.CALCulate(1).SELected.RLIMit.DATA = Array(1,1,1E6,1E9,0)
RLimData = SCPI.CALCulate(1).SELected.RLIMit.DATA

SCPI.CALCulate(1).PARameter(1).SELect
SCPI.CALCulate(1).SELected.RLIMit.DATA = Array(0) ''' Clear Ripple Limit Table
```

Examples (2)

```
Dim RlimData(5) As Variant
Dim Ref As Variant
RLimData(0) = 1
RLimData(1) = 1
RLimData(2) = 1e6
RLimData(3) = 1e9
RLimData(4) = 0
SCPI.CALCulate(1).PARameter(1).SELect
SCPI.CALCulate(1).SELected.RLIMit.DATA = RLimData
Ref = SCPI.CALCulate(1).SELected.RLIMit.DATA

Dim RLimData(0) as Variant
RLimData(0) = 0
SCPI.CALCulate(1).PARameter(1).SELect
SCPI.CALCulate(1).SELected.RLIMit.DATA = RLimData ''' Clear Ripple Limit Table
```

Related objects

SCPI.CALCulate(Ch).PARameter(Tr).SELect

SCPI.CALCulate(Ch).SELected.RLIMit.STATe

Equivalent key

Analysis > **Ripple Limit** > **Edit Ripple Limit** > **Add** |**Clear** |**Delete**

Equivalent SCPI command

Syntax

:CALCulate{[1]-4}[:SELected]:RLIMit:DATA <numeric 1>,… ,<numeric 1+(N×4)>

:CALCulate{[1]-4}[:SELected]:RLIMit:DATA?

Query response

{numeric 1},… ,{numeric 1+(N×4)}<newline><^END>

Example of use

```
10 DIM B(1:2,1:4)
20 OUTPUT 717;":CALC1:RLIM:DATA 2,1,1E9,3E9,3,1,5E9,7E9,3"
30 OUTPUT 717;":CALC1:RLIM:DATA?"
40 ENTER 717;A,B(*)
```

```
10 OUTPUT 717;":CALC1:RLIM:DATA 0" ! Clear Ripple Limit Table
```

SCPI.CALCulate(*Ch*).SELected.RLIMit.DISPlay.LINE

Object type

Property (**Read-Write**)

Syntax

SCPI.CALCulate(*Ch*).SELected.RLIMit.DISPlay.LINE = *Status*

Status = SCPI.CALCulate(*Ch*).SELected.RLIMit.DISPlay.LINE

Description

This command turns ON/OFF the ripple limit line display , for the active trace (specified with the SCPI.CALCulate(Ch).PARameter(Tr).SELect command) of the selected channel (*Ch*).

Variable

Parameter	*Status*
Description	ON/OFF the ripple limit line display.
Data type	Boolean type (Boolean)
Range	Select from the following: • True or ON: Turns ON the ripple limit line display. • False or OFF: Turns OFF the ripple limit line display.
Preset value	False or OFF

Examples

```
Dim RLimDisp As Boolean
SCPI.CALCulate(1).SELected.RLIMit.DISPlay.LINE = True
RLimDisp = SCPI.CALCulate(1).SELected.RLIMit.DISPLay.LINE
```

Related objects

SCPI.CALCulate(Ch).PARameter(Tr).SELect

SCPI.CALCulate(Ch).SELected.RLIMit.STATe

SCPI.CALCulate(Ch).SELected.RLIMit.DISPlay.SELect

SCPI.CALCulate(Ch).SELected.RLIMit.DISPlay.VALue

Equivalent key

Analysis > **Ripple Limit** > **Ripple Limit**

Equivalent SCPI command

Syntax

```
:CALCulate{[1]-4}[:SELected]:RLIMit:DISPlay:LINE {ON|OFF|1|0}
:CALCulate{[1]-4}[:SELected]:RLIMit:DISPlay:LINE?
```

Query response

```
{1|0}<newline><^END>
```

Example of use

```
10 OUTPUT 717;":CALC1:RLIM:DISP:LINE ON"
20 OUTPUT 717;":CALC1:RLIM:DISP:LINE?"
30 ENTER 717;A
```

SCPI.CALCulate(*Ch*).SELected.RLIMit.DISPlay.SELect

Object type

Property (**Read-Write**)

Syntax

SCPI.CALCulate(*Ch*).SELected.RLIMit.DISPlay.SELect = *Value*

Value = SCPI.CALCulate(*Ch*).SELected.RLIMit.DISPlay.SELect

Description

This command sets/gets the ripple limit band for ripple value display for the selected channel (*Ch*).

Variable

Parameter	*Value*
Description	The ripple limit band.
Data type	Long integer type (Long)
Range	1 to 12
Preset value	1

Examples

```
Dim RBand As Long
SCPI.CALCulate(1).SELected.RLIMit.DISPlay.SELect = 2
RBand = SCPI.CALCulate(1).SELected.RLIMit.DISPlay.SELect
```

Related objects

SCPI.CALCulate(Ch).PARameter(Tr).SELect

SCPI.CALCulate(Ch).SELected.RLIMit.STATe

SCPI.CALCulate(Ch).SELected.RLIMit.DISPlay.LINE

SCPI.CALCulate(Ch).SELected.RLIMit.DISPlay.VALue

Equivalent key

Analysis > **Ripple Limit** > **Ripple Band** > **1** to **12**

Equivalent SCPI command

Syntax

:CALCulate{[1]-4}[:SELected]:RLIMit:DISPlay:SELect <numeric>

:CALCulate{[1]-4}[:SELected]:RLIMit:DISPlay:SELect?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":CALC1:RLIM:DISP:SEL 5"
20 OUTPUT 717;":CALC1:RLIM:DISP:SEL?"
30 ENTER 717;A
```

SCPI.CALCulate(*Ch*).SELected.RLIMit.DISPlay.VALue

Object type

Property (**Read-Write**)

Syntax

SCPI.CALCulate(*Ch*).SELected.RLIMit.DISPlay.VALue = *Param*

Param = SCPI.CALCulate(*Ch*).SELected.RLIMit.DISPlay.VALue

Description

This command sets/gets the display type of ripple value for the active trace (specified with the SCPI.CALCulate(Ch).PARameter(Tr).SELect command) of the selected channel (*Ch*).

Variable

Parameter	*Param*
Description	The displaying type of ripple value.
Data type	Character string type (String)
Range	Select from the following: • "OFF": Specifies the display OFF. • "ABSolute": Specifies the absolute value for display type. • "MARgin": Specifies the margin for display type.
Preset value	"OFF"

Examples

```
Dim RDisp As String
SCPI.CALCulate(1).SELected.RLIMit.DISPlay.VALue = "ABSolute"
RDisp = SCPI.CALCulate(1).SELected.RLIMit.DISPlay.VALue
```

Related objects

SCPI.CALCulate(Ch).PARameter(Tr).SELect

SCPI.CALCulate(Ch).SELected.RLIMit.STATe

SCPI.CALCulate(Ch).SELected.RLIMit.DISPlay.LINE

SCPI.CALCulate(Ch).SELected.RLIMit.DISPlay.SELect

Equivalent key

Analysis > Ripple Limit > Ripple Value > OFF|Absolute|Margin

Equivalent SCPI command

Syntax

```
:CALCulate{[1]-4}[:SELected]:RLIMit:DISPlay:VALue
{OFF|ABSolute|MARgin}
:CALCulate{[1]-4}[:SELected]:RLIMit:DISPlay:VALue?
```

Query response

```
{OFF|ABS|MAR}<newline><^END>
```

Example of use

```
10 OUTPUT 717;":CALC1:RLIM:DISP:VAL ABS"
20 OUTPUT 717;":CALC1:RLIM:DISP:VAL?"
30 ENTER 717;A$
```

SCPI.CALCulate(*Ch*).SELected.RLIMit.FAIL

Object type

Property (**Read Only**)

Syntax

Status = SCPI.CALCulate(*Ch*).SELected.RLIMit.FAIL

Description

This command reads the ripple test result for the active trace (specified with the SCPI.CALCulate(Ch).PARameter(Tr).SELect command) of the selected channel (*Ch*).

Variable

Parameter	*Status*
Description	The ripple test result
Data type	Boolean type (Boolean)
Range	Select from the following: • True or ON: Turns ON the ripple test result is FAIL. • False or OFF: Turns OFF the ripple test result is FAIL.
Note	When the ripple test if set to OFF, False or OFF is always read out.

Examples

```
Dim Result As Boolean
Result = SCPI.CALCulate(1).SELected.RLIMit.FAIL
```

Related objects

SCPI.CALCulate(Ch).PARameter(Tr).SELect

SCPI.CALCulate(Ch).SELected.RLIMit.STATe

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

:CALCulate{[1]-4}[:SELected]:RLIMit:FAIL?

Example of use

```
10 OUTPUT 717;":CALC1:RLIM:FAIL?"
20 ENTER 717;A
```

SCPI.CALCulate(*Ch*).SELected.RLIMit.REPort.DATA

Object type

Property (**Read Only**)

Syntax

Data = SCPI.CALCulate(*Ch*).SELected.RLIMit.REPort.DATA

Description

This command reads the ripple value of the ripple test for the active trace (specified with the SCPI.CALCulate(Ch).PARameter(Tr).SELect command) of the selected channel (*Ch*).

The data transfer format when this command is executed depends on the setting with the SCPI.FORMat.DATA command.(Read only)

Variable

Parameter	*Data*
Description	Indicates the array data (for ripple line) of 1 + Num (number of limit lines)×3. Where n is an integer between 1 and 12. • *Data(0)* :Number of ripple limit line. • *Data(n×3-2)* :Number of ripple limit bands. • *Data(n×3-1)* :Ripple value. • *Data(n×3)* :Results of ripple test. Select from the following. 0:PASS 1:FAIL. The index of the array starts from 0.
Data type	Variant type (Variant)

Examples

```
Dim RData As Variant
SCPI.CALCulate(1).PARameter(1).SELect
RData = SCPI.CALCulate(1).SELected.RLIMit.REPort.DATA
```

Related objects

SCPI.CALCulate(Ch).PARameter(Tr).SELect

SCPI.CALCulate(Ch).SELected.RLIMit.STATe

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

:CALCulate{[1]-4}[:SELected]:RLIMit:REPort[:DATA]?

Query response

{numeric 1},… ,{numeric 1+N×3}<newline><^END>

Type	Description
<numeric 1>	Number of ripple limit line (1 to 12)
<numeric 1+(n×3)-2>	Number of ripple limit bands
<numeric 1+(n×3)-1>	Ripple value
<numeric 1+(n×3)>	Results of ripple test 0: Pass 1: Fail

Where N is the number of lines (specified with <numeric 1>) and n is an integer between 1 and 12.

Example of use

```
10 DIM B(1:2,1:3)
20 OUTPUT 717;":CALC1:RLIM:REP?"
30 ENTER 717;A,B(*)
```

SCPI.CALCulate(*Ch*).SELected.RLIMit.STATe

Object type

Property (**Read-Write**)

Syntax

SCPI.CALCulate(*Ch*).SELected.RLIMit.STATe = *Status*

Status = SCPI.CALCulate(*Ch*).SELected.RLIMit.STATe

Description

This command turns ON/OFF the ripple test function for the active trace (specified with the SCPI.CALCulate(Ch).PARameter(Tr).SELect command) of the selected channel (*Ch*).

Variable

Parameter	*Status*
Description	ON/OFF the ripple test function
Data type	Boolean type (Boolean)
Range	Select from the following: • True or ON: Turns ON the ripple test function. • False or OFF: Turns OFF the ripple test function.
Preset value	False or OFF

Examples

```
Dim RLimTest As Boolean
SCPI.CALCulate(1).PARameter(1).SELect
SCPI.CALCulate(1).SELected.RLIMit.STATe = True
RLimTest = SCPI.CALCulate(1).SELected.RLIMit.STATe
```

Related Objects

SCPI.CALCulate(Ch).PARameter(Tr).SELect

SCPI.CALCulate(Ch).SELected.RLIMit.DATA

SCPI.CALCulate(Ch).SELected.RLIMit.DISPlay.LINE

SCPI.CALCulate(Ch).SELected.RLIMit.DISPlay.SELect

SCPI.CALCulate(Ch).SELected.RLIMit.DISPlay.VALue

SCPI.CALCulate(Ch).SELected.RLIMit.FAIL

SCPI.CALCulate(Ch).SELected.RLIMit.REPort.DATA

Equivalent key

Analysis > **Ripple Limit** > **Ripple Limit Test**

Equivalent SCPI command

Syntax

:CALCulate{[1]-4}[:SELected]:RLIMit[:STATe] {ON|OFF|1|0}

:CALCulate{[1]-4}[:SELected]:RLIMit[:STATe]?

Query response

{1|0}<newline><^END>

Example of use

```
10 OUTPUT 717;":CALC1:RLIM ON"
20 OUTPUT 717;":CALC1:RLIM?"
30 ENTER 717;A
```

SCPI.CALCulate(*Ch*).SELected.SMOothing.APERture

Object type

Property (**Read-Write**)

Syntax

SCPI.CALCulate(*Ch*).SELected.SMOothing.APERture = *Value*

Value = SCPI.CALCulate(*Ch*).SELected.SMOothing.APERture

Description

This command sets/gets the smoothing aperture (percentage to the sweep span value) of the active trace of the selected channel (*Ch*).

Variable

Parameter	*Value*
Description	Smoothing aperture
Data type	Double precision floating point type (Double)
Range	0.05 to 25
Preset value	1.5
Unit	% (percent)
Note	If the specified variable 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.

Examples

```
Dim SmoAper As Double
SCPI.CALCulate(1).PARameter(1).SELect
SCPI.CALCulate(1).SELected.SMOothing.APERture = 2.5
SmoAper = SCPI.CALCulate(1).SELected.SMOothing.APERture
```

Related objects

SCPI.CALCulate(Ch).PARameter(Tr).SELect

SCPI.CALCulate(Ch).SELected.SMOothing.STATe

Equivalent key

Avg > **Smo Aperture**

Equivalent SCPI command

Syntax

:CALCulate{[1]-4}[:SELected]:SMOothing:APERture <numeric>

:CALCulate{[1]-4}[:SELected]:SMOothing:APERture?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":CALC1:SMO:APER 2.5"
20 OUTPUT 717;":CALC1:SMO:APER?"
30 ENTER 717;A
```

SCPI.CALCulate(*Ch*).SELected.SMOothing.STATe

Object type

Property (**Read-Write**)

Syntax

SCPI.CALCulate(*Ch*).SELected.SMOothing.STATe = *Status*

Status = SCPI.CALCulate(*Ch*).SELected.SMOothing.STATe

Description

This command turns ON/OFF the smoothing, for the active trace of the selected channel (*Ch*).

Variable

Parameter	*Status*
Description	ON/OFF of the smoothing
Data type	Boolean type (Boolean)
Range	Select from the following: • True or ON: Turns ON the smoothing. • False or OFF: Turns OFF the smoothing.
Preset value	False or OFF

Examples

```
Dim Smo As Boolean
SCPI.CALCulate(1).PARameter(1).SELect
SCPI.CALCulate(1).SELected.SMOothing.STATe = True
Smo = SCPI.CALCulate(1).SELected.SMOothing.STATe
```

Related objects

SCPI.CALCulate(Ch).PARameter(Tr).SELect

SCPI.CALCulate(Ch).SELected.SMOothing.APERture

Equivalent key

Avg > **Smoothing**

Equivalent SCPI command

Syntax

:CALCulate{[1]-4}[:SELected]:SMOothing[:STATe] {ON|OFF|1|0}

:CALCulate{[1]-4}[:SELected]:SMOothing[:STATe]?

Query response

{1|0}<newline><^END>

Example of use

```
10 OUTPUT 717;":CALC1:SMO ON"
20 OUTPUT 717;":CALC1:SMO?"
30 ENTER 717;A
```

SCPI.CALCulate(Ch).SELected.TRANsform.DISTance.CENTer

Object Type

Property (**Read-Write**)

Syntax

SCPI.CALCulate(Ch).SELected.TRANsform.DISTance.CENTer = *Value*

Value = SCPI.CALCulate(Ch).SELected.TRANsform.DISTance.CENTer

Description

This command sets/gets the center value used for the transformation function of the fault location display, for the active trace of the selected channel (*Ch*).

Variable

Parameter	*Value*
Description	Center value
Data Type	Double precision floating point type (Double)
Range	Varies depending on the frequency span and the number of points.
Preset Value	0
Unit	ft (feet) or m (meters)
Note	If the specified variable 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.

Examples

```
Dim Cent as Double
SCPI.CALCulate(4).SELected.TRANsform.DISTance.CENTer = 1E-8
Cent = SCPI.CALCulate(4).SELected.TRANsform.DISTance.CENTer
```

Related Objects

SCPI.CALCulate(Ch).SELected.TRANsform.DISTance.SPAN

SCPI.CALCulate(Ch).SELected.TRANsform.DISTance.STARt

SCPI.CALCulate(Ch).SELected.TRANsform.DISTance.STOP

SCPI.CALCulate(Ch).SELected.TRANsform.DISTance.UNIT

Equivalent Key

Analysis > Fault Location > Center

Equivalent SCPI Command

Syntax

:CALCulate{[1]-4}[:SELected]:TRANsform:DISTance:CENTer <numeric>

:CALCulate{[1]-4}[:SELected]:TRANsform:DISTance:CENTer?

Query Response

{numeric} <newline><^END>

Example of use

```
10 OUTPUT 717;":CALC1:TRAN:DIST:CENT 0"
20 OUTPUT 717;":CALC1:TRAN:DIST:CENT?"
30 ENTER 717;A
```

SCPI.CALCulate(*Ch*).SELected.TRANsform.DISTance.CLOSs

Object Type

Property (**Read-Write**)

Syntax

SCPI.CALCulate(*Ch*).SELected.TRANsform.DISTance.CLOSs = *Value*

Value = SCPI.CALCulate(*Ch*).SELected.TRANsform.DISTance.CLOSs

Description

This command sets/gets the cable loss value used for the transformation function of the fault location display, for the active trace of the selected channel (*Ch*).

Variable

Parameter	*Value*
Description	Cable Loss value
Data Type	Double precision floating point type (Double)
Range	Varies depending on the distance unit.
Preset Value	0
Unit	dB/100m or dB/100ft
Note	If the specified variable 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.

Examples

```
Dim Closs as Double
SCPI.CALCulate(4).SELected.TRANsform.DISTance.CLOSs = 10
Closs = SCPI.CALCulate(4).SELected.TRANsform.DISTance.CLOSs
```

Related Objects

SCPI.CALCulate(Ch).SELected.TRANsform.DISTance.UNIT

Equivalent Key

Analysis > **Fault Location** > **Cable Loss**

Equivalent SCPI Command

Syntax

:CALCulate{[1]-4}[:SELected]:TRANsform:DISTance:CLOSs <numeric>

:CALCulate{[1]-4}[:SELected]:TRANsform:DISTance:CLOSs?

Query Response

{numeric} <newline><^END>

Example of use

```
10 OUTPUT 717;":CALC1:TRAN:DIST:CLOS 0"
20 OUTPUT 717;":CALC1:TRAN:DIST:CLOS?"
30 ENTER 717;A
```

SCPI.CALCulate(*Ch*).SELected.TRANsform.DISTance.IMPulse.WIDTh

Object Type

Property (**Read-Write**)

Syntax

SCPI.CALCulate(*Ch*).SELected.TRANsform.DISTance.IMPulse.WIDTh = *Value*

Value = SCPI.CALCulate(*Ch*).SELected.TRANsform.DISTance.IMPulse.WIDTh

Description

This command sets/gets the shape of the Kayser Bessel window using the impulse width used for the transformation function of the fault location display, for the active trace of selected channel (*Ch*).

Variable

Parameter	*Value*
Description	Impulse width
Data Type	Double precision floating point type (Double)
Range	Varies depending on the frequency span and transformation type.
Preset Value	650.395679856p sec
Unit	s (second)
Note	If the specified variable 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.

Examples

```
Dim ImpWid as Double
SCPI.CALCulate(4).SELected.TRANsform.DISTance.IMPulse.WIDTh = 1E-10
ImpWid = SCPI.CALCulate(4).SELected.TRANsform.DISTance.IMPulse.WIDTh
```

Related Objects

SCPI.CALCulate(Ch).SELected.TRANsform.DISTance.KBESsel

SCPI.CALCulate(Ch).SELected.TRANsform.DISTance.STEP.RTIMe

Equivalent Key

Analysis > **Fault Location** > **Window** > **Impulse Width**

Equivalent SCPI Command

Syntax

:CALCulate{[1]-4}[:SELected]:TRANsform:DISTance:IMPulse:WIDTh <numeric>

:CALCulate{[1]-4}[:SELected]:TRANsform:DISTance:IMPulse:WIDTh?

Query Response

{numeric} <newline><^END>

Example of use

```
10 OUTPUT 717;":CALC1:TRAN:DIST:IMP:WIDT 7E-10"
20 OUTPUT 717;":CALC1:TRAN:DIST:IMP:WIDT?"
30 ENTER 717;A
```

SCPI.CALCulate(*Ch*).SELected.TRANsform.DISTance.KBESsel

Object Type

Property (**Read-Write**)

Syntax

SCPI.CALCulate(*Ch*).SELected.TRANsform.DISTance.KBESsel = *Value*

Value = SCPI.CALCulate(*Ch*).SELected.TRANsform.DISTance.KBESsel

Description

This command sets/gets the shape of the Kayser Bessel window using β used for the transformation function of the fault location display, for the active trace of the selected channel (*Ch*).

Variable

Parameter	*Value*
Description	The value of β
Data Type	Double precision floating point type (Double)
Range	0 to 13
Preset Value	6
Note	If the specified variable 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.

Examples

```
Dim KBeta as Double
SCPI.CALCulate(4).SELected.TRANsform.DISTance.KBESsel = 3
KBeta = SCPI.CALCulate(4).SELected.TRANsform.DISTance.KBESsel
```

Related Objects

SCPI.CALCulate(Ch).SELected.TRANsform.DISTance.IMPulse.WIDTh

SCPI.CALCulate(Ch).SELected.TRANsform.DISTance.STEP.RTIMe

Equivalent Key

Analysis > **Fault Location** > **Window** > **Kaiser Beta**

Equivalent SCPI Command

Syntax

:CALCulate{[1]-4}[:SELected]:TRANsform:DISTance:KBESsel <numeric>

:CALCulate{[1]-4}[:SELected]:TRANsform:DISTance:KBESsel?

Query Response

{numeric} <newline><^END>

Example of use

```
10 OUTPUT 717;":CALC1:TRAN:DIST:KBES 0"
20 OUTPUT 717;":CALC1:TRAN:DIST:KBES?"
30 ENTER 717;A
```

SCPI.CALCulate(*Ch*).SELected.TRANsform.DISTance.LPFRequency

Object Type

Method (**Write Only**)

Syntax

SCPI.CALCulate(*Ch*).SELected.TRANsform.DISTance.LPFRequency

Description

This command changes the frequency range to match with the low-pass type transformation of the transformation function of the fault location display, for the active trace of the selected channel (*Ch*).

Examples

SCPI.CALCulate(4).SELected.TRANsform.DISTance.LPFRequency

Related Objects

SCPI.CALCulate(Ch).SELected.TRANsform.DISTance.STATe

SCPI.CALCulate(Ch).SELected.TRANsform.DISTance.TYPE

Equivalent Key

Analysis > **Fault Location** > **Set Freq Low Pass**

Equivalent SCPI Command

Syntax

:CALCulate{[1]-4}[:SELected]:TRANsform:DISTance:LPFRequency

Example of use

10 OUTPUT 717;":CALC1:TRAN:DIST:LPFR"

SCPI.CALCulate(*Ch*).SELected.TRANsform.DISTance.REFLection.TYPE

Object Type

Property (**Read-Write**)

Syntax

SCPI.CALCulate(*Ch*).SELected.TRANsform.DISTance.REFLection.TYPE = *Param*

Param = SCPI.CALCulate(*Ch*).SELected.TRANsform.DISTance.REFLection.TYPE

Description

This command sets/gets the reflection distance either one way or round trip, for the active trace of the selected channel (*Ch*).

Variable

Parameter	*Param*
Description	The reflection distance either one way or round trip
Data Type	Character string type (String)
Range	Select from the following: • "OWAY": Specifies the One Way. • "RTRip": Specifies the Round Trip.
Preset Value	"RTRip"

Examples

```
Dim StimType as String
SCPI.CALCulate(4).SELected.TRANsform.DISTance.REFLection.TYPE = "OWAY"
StimType = SCPI.CALCulate(4).SELected.TRANsform.DISTance.REFLection.TYPE
```

Related Objects

Equivalent Key

Analysis > **Fault Location** > **Reflection Type**

Equivalent SCPI Command

Syntax

:CALCulate{[1]-4}[:SELected]:TRANsform:DISTance:REFLection:TYPE {OWAY|RTRip}

:CALCulate{[1]-4}[:SELected]:TRANsform:DISTance:REFLection:TYPE?

Query Response

{OWAY|RTR} <newline><^END>

Example of use

```
10 OUTPUT 717;":CALC1:TRAN:DIST:REFL:TYPE RTR"
20 OUTPUT 717;":CALC1:TRAN:DIST:REFL:TYPE?"
30 ENTER 717;A$
```

SCPI.CALCulate(*Ch*).SELected.TRANsform.DISTance.SPAN

Object Type

Property (**Read-Write**)

Syntax

SCPI.CALCulate(*Ch*).SELected.TRANsform.DISTance.SPAN = *Value*

Value = SCPI.CALCulate(*Ch*).SELected.TRANsform.DISTance.SPAN

Description

This command sets/gets the span value used for the transformation function of the fault location display, for the active trace of the selected channel (*Ch*).

Variable

Parameter	*Value*
Description	Span value
Data Type	Double precision floating point type (Double)
Range	Varies depending on the frequency span and the number of points.
Preset Value	19.6714211286 ft or 5.99584916 m
Unit	ft (feet) or m (meters)
Note	If the specified variable 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.

Examples

```
Dim Span as Double
SCPI.CALCulate(4).SELected.TRANsform.DISTance.SPAN = 0.42E+1
Span = SCPI.CALCulate(4).SELected.TRANsform.DISTance.SPAN
```

Related Objects

SCPI.CALCulate(Ch).SELected.TRANsform.DISTance.CENTer

SCPI.CALCulate(Ch).SELected.TRANsform.DISTance.STARt

SCPI.CALCulate(Ch).SELected.TRANsform.DISTance.STOP

SCPI.CALCulate(Ch).SELected.TRANsform.DISTance.UNIT

Equivalent Key

Analysis > Fault Location > Span

Equivalent SCPI Command

Syntax

:CALCulate{[1]-4}[:SELected]:TRANsform:DISTance:SPAN <numeric>

:CALCulate{[1]-4}[:SELected]:TRANsform:DISTance:SPAN?

Query Response

{numeric} <newline><^END>

Example of use

```
10 OUTPUT 717;":CALC1:TRAN:DIST:SPAN 4.42"
20 OUTPUT 717;":CALC1:TRAN:DIST:SPAN?"
30 ENTER 717;A
```

SCPI.CALCulate(*Ch*).SELected.TRANsform.DISTance.STARt

Object Type

Property (**Read-Write**)

Syntax

SCPI.CALCulate(*Ch*).SELected.TRANsform.DISTance.STARt = *Value*

Value = SCPI.CALCulate(*Ch*).SELected.TRANsform.DISTance.STARt

Description

This command sets/gets the start value used for the transformation function of the fault location display, for the active trace of the selected channel (*Ch*).

Variable

Parameter	*Value*
Description	Start value
Data Type	Double precision floating point type (Double)
Range	Varies depending on the frequency span and the number of points.
Preset Value	-9.8357105643 ft or -2.99792458 m
Unit	ft (feet) or m (meters)
Note	If the specified variable 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.

Examples

```
Dim Star as Double
SCPI.CALCulate(4).SELected.TRANsform.DISTance.STARt = 0
Star = SCPI.CALCulate(4).SELected.TRANsform.DISTance.STARt
```

Related Objects

SCPI.CALCulate(Ch).SELected.TRANsform.DISTance.CENTer

SCPI.CALCulate(Ch).SELected.TRANsform.DISTance.SPAN

SCPI.CALCulate(Ch).SELected.TRANsform.DISTance.STOP

SCPI.CALCulate(Ch).SELected.TRANsform.DISTance.UNIT

Equivalent Key

Analysis > **Fault Location** > **Start**

Equivalent SCPI Command

Syntax

:CALCulate{[1]-4}[:SELected]:TRANsform:DISTance:STARt <numeric>

:CALCulate{[1]-4}[:SELected]:TRANsform:DISTance:STARt?

Query Response

{numeric} <newline><^END>

Example of use

```
10 OUTPUT 717;":CALC1:TRAN:DIST:STAR 0"
20 OUTPUT 717;":CALC1:TRAN:DIST:STAR?"
30 ENTER 717;A
```

SCPI.CALCulate(*Ch*).SELected.TRANsform.DISTance.STATe

Object Type

Property (**Read-Write**)

Syntax

SCPI.CALCulate(*Ch*).SELected.TRANsform.DISTance.STATe = *Value*

Value = SCPI.CALCulate(*Ch*).SELected.TRANsform.DISTance.STATe

Description

This command turns ON/OFF the transformation function of the fault location display, for the active trace of the selected channel (*Ch*).

You can enable the transformation function only when the sweep type is the linear sweep and the number of points is three or more. If you execute this object to try to enable the transformation function when the sweep type is other than the linear sweep or the number of points is less than three, an error occurs and the object is ignored.

NOTE When the sweep type is the power sweep, you cannot turn ON the transformation function. If you execute this object trying to turn ON the transformation function during the power sweep, an error occurs and the object is ignored.

Variable

Parameter	*Status*
Description	ON/OFF of the transformation function
Data type	Boolean type (Boolean)
Range	Select from the following: • True or ON: Turns ON the transformation function. • False or OFF: Turns OFF the transformation function.
Preset value	False or OFF

Examples

```
Dim Trans as Boolean
SCPI.CALCulate(4).SELected.TRANsform.DISTance.STATe = True
Trans = SCPI.CALCulate(4).SELected.TRANsform.DISTance.STATe
```

Related Objects

E5061B

Equivalent Key

Analysis > **Fault Location** > **Fault Location**

Equivalent SCPI Command

Syntax

:CALCulate{[1]-4}[:SELected]:TRANsform:DISTance:STATe {ON|OFF|1|0}

:CALCulate{[1]-4}[:SELected]:TRANsform:DISTance:STATe?

Query Response

{1|0} <newline><^END>

Example of use

```
10 OUTPUT 717;":CALC1:TRAN:DIST:STAT ON"
20 OUTPUT 717;":CALC1:TRAN:DIST:STAT?"
30 ENTER 717;A
```

SCPI.CALCulate(*Ch*).SELected.TRANsform.DISTance.STEP.RTIMe

Object Type

Property (**Read-Write**)

Syntax

SCPI.CALCulate(*Ch*).SELected.TRANsform.DISTance.STEP.RTIMe = *Value*

Value = SCPI.CALCulate(*Ch*).SELected.TRANsform.DISTance.STEP.RTIMe

Description

This command sets/gets the shape of the Kayser Bessel window using the rise time of step signal used for the transformation function of the fault location display, for the active trace of the selected channel (*Ch*).

Variable

Parameter	*Value*
Description	The rise time of step signal
Data Type	Double precision floating point type (Double)
Range	Varies depending on the frequency span and the number of points.
Preset Value	328.677622587p sec
Unit	s (second)
Note	If the specified variable 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.

Examples

```
Dim Rtime as Double
SCPI.CALCulate(4).SELected.TRANsform.DISTance.STEP.RTIMe = 4.2E-10
Rtime = SCPI.CALCulate(4).SELected.TRANsform.DISTance.STEP.RTIMe
```

Related Objects

SCPI.CALCulate(Ch).SELected.TRANsform.DISTance.IMPulse.WIDTh

SCPI.CALCulate(Ch).SELected.TRANsform.DISTance.KBESsel

Equivalent Key

Analysis > **Fault Location** > **Window** > **Step Rise**

Equivalent SCPI Command

Syntax

:CALCulate{[1]-4}[:SELected]:TRANsform:DISTance:STEP:RTIMe <numeric>

:CALCulate{[1]-4}[:SELected]:TRANsform:DISTance:STEP:RTIMe?

Query Response

{numeric} <newline><^END>

Example of use

```
10 OUTPUT 717;":CALC1:TRAN:DIST:STEP:RTIM 4.2E-10"
20 OUTPUT 717;":CALC1:TRAN:DIST:STEP:RTIM?"
30 ENTER 717;A
```

SCPI.CALCulate(*Ch*).SELected.TRANsform.DISTance.STIMulus

Object Type

Property (**Read-Write**)

Syntax

SCPI.CALCulate(*Ch*).SELected.TRANsform.DISTance.STIMulus = *Param*

Param = SCPI.CALCulate(*Ch*).SELected.TRANsform.DISTance.STIMulus

Description

This command sets/gets the stimulus type used for the transformation function of the fault location display, for the active trace of the selected channel (*Ch*).

Variable

Parameter	*Param*
Description	The stimulus type
Data Type	Character string type (String)
Range	Select from the following: • "IMPulse": Specifies the impulse • "STEP": Specifies the step
Preset Value	"IMPulse"

Examples

```
Dim StimType as String
SCPI.CALCulate(4).SELected.TRANsform.DISTance.STIMulus = "IMPulse"
StimType = SCPI.CALCulate(4).SELected.TRANsform.DISTance.STIMulus
```

Related Objects

SCPI.CALCulate(Ch).SELected.TRANsform.DISTance.STATe

SCPI.CALCulate(Ch).SELected.TRANsform.DISTance.TYPE

Equivalent Key

Analysis > **Fault Location** > **Type** > **Bandpass|Lowpass Step|Lowpass Imp**

NOTE When performing this operation from the front panel, you select the transformation type at the same time.

Equivalent SCPI Command

Syntax

:CALCulate{[1]-4}[:SELected]:TRANsform:DISTance:STIMulus {IMPulse|STEP}

:CALCulate{[1]-4}[:SELected]:TRANsform:DISTance:STIMulus?

Query Response

{IMP|STEP} <newline><^END>

Example of use

```
10 OUTPUT 717;":CALC1:TRAN:DIST LPAS"
20 OUTPUT 717;":CALC1:TRAN:DIST:STIM STEP"
30 OUTPUT 717;":CALC1:TRAN:DIST:STIM?"
40 ENTER 717;A$
```

SCPI.CALCulate(*Ch*).SELected.TRANsform.DISTance.STOP

Object Type

Property (**Read-Write**)

Syntax

SCPI.CALCulate(*Ch*).SELected.TRANsform.DISTance.STOP = *Value*

Value = SCPI.CALCulate(*Ch*).SELected.TRANsform.DISTance.STOP

Description

This command sets/gets the stop value used for the transformation function of the fault location display, for the active trace of the selected channel (*Ch*).

Variable

Parameter	*Value*
Description	Stop value
Data Type	Double precision floating point type (Double)
Range	Varies depending on the frequency span and the number of points.
Preset Value	9.8357105643 ft or 2.99792458 m
Unit	ft (feet) or m (meters)
Note	If the specified variable 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.

Examples

```
Dim DistStop as Double
SCPI.CALCulate(4).SELected.TRANsform.DISTance.STOP = -2.5
DistStop = SCPI.CALCulate(4).SELected.TRANsform.DISTance.STOP
```

Related Objects

SCPI.CALCulate(Ch).SELected.TRANsform.DISTance.CENTer

SCPI.CALCulate(Ch).SELected.TRANsform.DISTance.SPAN

SCPI.CALCulate(Ch).SELected.TRANsform.DISTance.STARt

SCPI.CALCulate(Ch).SELected.TRANsform.DISTance.UNIT

Equivalent Key

Analysis > Fault Location > Stop

Equivalent SCPI Command

Syntax

:CALCulate{[1]-4}[:SELected]:TRANsform:DISTance:STOP <numeric>

:CALCulate{[1]-4}[:SELected]:TRANsform:DISTance:STOP?

Query Response

{numeric} <newline><^END>

Example of use

```
10 OUTPUT 717;":CALC1:TRAN:DIST:STOP -2.2"
20 OUTPUT 717;":CALC1:TRAN:DIST:STOP?"
30 ENTER 717;A
```

SCPI.CALCulate(*Ch*).SELected.TRANsform.DISTance.TYPE

Object Type

Property (**Read-Write**)

Syntax

SCPI.CALCulate(*Ch*).SELected.TRANsform.DISTance.TYPE = *Param*

Param = SCPI.CALCulate(*Ch*).SELected.TRANsform.DISTance.TYPE

Description

This command sets/gets the transformation type used for the transformation function of the fault location, for the active trace of the selected channel (*Ch*).

Variable

Parameter	*Param*
Description	The transformation type
Data type	Character string type (String)
Range	Select from the following: • "BPASs": Specifies the band-pass • "LPASs": Specifies the low-pass
Preset value	"BPASs"

Examples

```
Dim Typ as String
SCPI.CALCulate(4).SELected.TRANsform.DISTance.TYPE = "LPASs"
Typ = SCPI.CALCulate(4).SELected.TRANsform.DISTance.TYPE
```

Related Objects

SCPI.CALCulate(Ch).SELected.TRANsform.DISTance.STATe

SCPI.CALCulate(Ch).SELected.TRANsform.DISTance.STIMulus

Equivalent Key

Analysis > **Fault Location** > **Type** > **Bandpass|Lowpass Step|Lowpass Imp**

NOTE When performing this operation from the front panel, you select the stimulus type at the same time.

Equivalent SCPI Command

Syntax

:CALCulate{[1]-4}[:SELected]:TRANsform:DISTance[:TYPE] {BPASs|LPASs}

:CALCulate{[1]-4}[:SELected]:TRANsform:DISTance[:TYPE]?

Query Response

{BPAS|LPAS} <newline><^END>

Example of use

```
10 OUTPUT 717;":CALC1:TRAN:DIST LPAS"
20 OUTPUT 717;":CALC1:TRAN:DIST?"
30 ENTER 717;A$
```

SCPI.CALCulate(*Ch*).SELected.TRANsform.DISTance.UNIT

Object Type

Property (**Read-Write**)

Syntax

SCPI.CALCulate(*Ch*).SELected.TRANsform.DISTance.UNIT = *Param*

Param = SCPI.CALCulate(*Ch*).SELected.TRANsform.DISTance.UNIT

Description

This command sets/gets the distance unit used for the transform function of the fault location display, for the active trace of the selected channel (*Ch*).

Variable

Parameter	*Param*
Description	The distance unit used for the transform function
Data Type	Character string type (String)
Range	Select from the following. • "METers": Specifies the meter. • "FEET": Specifies the feet.
Preset Value	"METers"

Examples

```
Dim DistUnit as String
SCPI.CALCulate(4).SELected.TRANsform.DISTance.UNIT = "METers"
DistUnit = SCPI.CALCulate(4).SELected.TRANsform.DISTance.UNIT
```

Related Objects

SCPI.CALCulate(Ch).SELected.TRANsform.DISTance.CENTer

SCPI.CALCulate(Ch).SELected.TRANsform.DISTance.SPAN

SCPI.CALCulate(Ch).SELected.TRANsform.DISTance.STARt

SCPI.CALCulate(Ch).SELected.TRANsform.DISTance.STOP

Equivalent Key

Analysis > **Fault Location** > **Unit** > **Meters|Feet|Seconds**

NOTE When performing this operation from the front panel, you select the horizontal axis of the transform function either time or distance at the same time.

Equivalent SCPI Command

Syntax

:CALCulate{[1]-4}[:SELected]:TRANsform:DISTance:UNIT {METers|FEET}

:CALCulate{[1]-4}[:SELected]:TRANsform:DISTance:UNIT?

Query Response

{MET|FEET} <newline><^END>

Example of use

```
10 OUTPUT 717;":CALC1:TRAN:DIST:UNIT FEET"
20 OUTPUT 717;":CALC1:TRAN:DIST:UNIT?"
30 ENTER 717;A$
```

SCPI.CALCulate(*Ch*).SELected.TRANsform.METHod

Object Type

Property (**Read-Write**)

Syntax

SCPI.CALCulate(*Ch*).SELected.TRANsform.METHod = *Param*

Param = SCPI.CALCulate(*Ch*).SELected.TRANsform.METHod

Description

This command sets/gets the horizontal axis of the transform function either as time or distance, for the active trace of the selected channel (*Ch*).

Variable

Parameter	*Param*
Description	The horizontal axis of the transform function
Data Type	Character string type (String)
Range	Select from the following. • "TIME": Specifies the time. • "DISTance": Specifies the distance.
Preset Value	"TIME"

Examples

```
Dim TranMeth as String
SCPI.CALCulate(4).SELected.TRANsform.METHod = "TIME"
TranMeth = SCPI.CALCulate(4).SELected.TRANsform.METHod
```

Related Objects

SCPI.CALCulate(Ch).SELected.TRANsform.DISTance.UNIT

Equivalent Key

Analysis > **Fault Location** > **Unit** > **Meters|Feet|Seconds**

Equivalent SCPI Command

Syntax

:CALCulate{[1]-4}[:SELected]:TRANsform:METHod {TIME|DISTance}

:CALCulate{[1]-4}[:SELected]:TRANsform:METHod?

Query Response

{TIME|DIST} <newline><^END>

Example of use

```
10 OUTPUT 717;":CALC1:TRAN:METH DIST"
20 OUTPUT 717;":CALC1:TRAN:METH?"
30 ENTER 717;A$
```

SCPI.CALCulate(*Ch*).SELected.TRANsform.TIME.CENTer

Object type

Property (**Read-Write**)

Syntax

SCPI.CALCulate(*Ch*).SELected.TRANsform.TIME.CENTer = *Value*

Value = SCPI.CALCulate(*Ch*).SELected.TRANsform.TIME.CENTer

Description

This command sets/gets the center value used for the transformation function of the time domain function, for the active trace of the selected channel (*Ch*).

Variable

Parameter	*Value*
Description	Center value
Data type	Double precision floating point type (Double)
Range	Varies depending on the frequency span and the number of points.
Preset value	0
Unit	s (second)
Note	If the specified variable 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.

Examples

```
Dim Cent As Double
SCPI.CALCulate(1).PARameter(1).SELect
SCPI.CALCulate(1).SELected.TRANsform.TIME.CENTer = 1E-8
Cent = SCPI.CALCulate(1).SELected.TRANsform.TIME.CENTer
```

Related objects

SCPI.CALCulate(Ch).SELected.TRANsform.TIME.SPAN

SCPI.CALCulate(Ch).SELected.TRANsform.TIME.STATe

SCPI.CALCulate(Ch).PARameter(Tr).SELect

Equivalent key

Analysis > Fault Location > Center

Equivalent SCPI command

Syntax

:CALCulate{[1]-4}[:SELected]:TRANsform:TIME:CENTer <numeric>

:CALCulate{[1]-4}[:SELected]:TRANsform:TIME:CENTer?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":CALC1:TRAN:TIME:CENT 1E-8"
20 OUTPUT 717;":CALC1:TRAN:TIME:CENT?"
30 ENTER 717;A
```

SCPI.CALCulate(*Ch*).SELected.TRANsform.TIME.CLOSs

Object Type

Property (**Read-Write**)

Syntax

SCPI.CALCulate(*Ch*).SELected.TRANsform.TIME.CLOSs = *Value*

Value = SCPI.CALCulate(*Ch*).SELected.TRANsform.TIME.CLOSs

Description

This command sets/gets the cable loss value used for the transformation function of the time domain function, for the active trace of the selected channel (*Ch*).

Variable

Parameter	*Value*
Description	Cable loss value
Data Type	Double precision floating point type (Double)
Range	0 to 1000
Preset Value	0
Unit	dB/μsec
Note	If the specified variable 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.

Examples

```
Dim Closs as Double
SCPI.CALCulate(4).SELected.TRANsform.TIME.CLOSs = 3.4
Closs = SCPI.CALCulate(4).SELected.TRANsform.TIME.CLOSs
```

Related Objects

Equivalent Key

Analysis > **Fault Location** > **Cable Loss**

Equivalent SCPI Command

Syntax

:CALCulate{[1]-4}[:SELected]:TRANsform:TIME:CLOSs <numeric>

:CALCulate{[1]-4}[:SELected]:TRANsform:TIME:CLOSs?

Query Response

{numeric} <newline><^END>

Example of use

```
10 OUTPUT 717;":CALC1:TRAN:TIME:CLOS 0"
20 OUTPUT 717;":CALC1:TRAN:TIME:CLOS?"
30 ENTER 717;A
```

SCPI.CALCulate(*Ch*).SELected.TRANsform.TIME.IMPulse.WIDTh

Object type

Property (**Read-Write**)

Syntax

SCPI.CALCulate(*Ch*).SELected.TRANsform.TIME.IMPulse.WIDTh = *Value*

Value = SCPI.CALCulate(*Ch*).SELected.TRANsform.TIME.IMPulse.WIDTh

Description

This command sets/gets the shape of the Kayser Bessel window using the impulse width used for the transformation function of the time domain function, for the active trace of the selected channel (*Ch*).

Variable

Parameter	*Value*
Description	Impulse width
Data type	Double precision floating point type (Double)
Range	Varies depending on the frequency span and transformation type.
Preset value	Varies depending on the frequency span and transformation type.
Unit	s (second)
Note	If the specified variable 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.

Examples

```
Dim ImpWid As Double
SCPI.CALCulate(1).PARameter(1).SELect
SCPI.CALCulate(1).SELected.TRANsform.TIME.IMPulse.WIDTh = 1E-10
ImpWid = SCPI.CALCulate(1).SELected.TRANsform.TIME.IMPulse.WIDTh
```

Related objects

SCPI.CALCulate(Ch).SELected.TRANsform.TIME. KBESsel

SCPI.CALCulate(Ch).SELected.TRANsform.TIME.STEP.RTIMe

SCPI.CALCulate(Ch).SELected.TRANsform.TIME.STATe

SCPI.CALCulate(Ch).PARameter(Tr).SELect

Equivalent key

Analysis > **Fault Location** > **Window** > **Impulse Width**

Equivalent SCPI command

Syntax

:CALCulate{[1]-4}[:SELected]:TRANsform:TIME:IMPulse:WIDTh <numeric>

:CALCulate{[1]-4}[:SELected]:TRANsform:TIME:IMPulse:WIDTh?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":CALC1:TRAN:TIME:IMP:WIDT 1E-10"
20 OUTPUT 717;":CALC1:TRAN:TIME:IMP:WIDT?"
30 ENTER 717;A
```

SCPI.CALCulate(*Ch*).SELected.TRANsform.TIME.KBESsel

Object Type

Property (**Read-Write**)

Syntax

SCPI.CALCulate(*Ch*).SELected.TRANsform.TIME.KBESsel = *Value*

Value = SCPI.CALCulate(*Ch*).SELected.TRANsform.TIME.KBESsel

Description

This command sets/gets the shape of the Kayser Bessel window used for the transformation function of the time domain function, for the active trace of the selected channel (*Ch*).

Variable

Parameter	*Value*
Description	The value of b
Data type	Double precision floating point type (Double)
Range	0 to 13
Preset value	6
Note	If the specified variable 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.

Examples

```
Dim Beta As Double
SCPI.CALCulate(1).PARameter(1).SELect
SCPI.CALCulate(1).SELected.TRANsform.TIME.KBESsel = 3
Beta = SCPI.CALCulate(1).SELected.TRANsform.TIME.KBESsel
```

Related objects

SCPI.CALCulate(Ch).SELected.TRANsform.TIME. IMPulse.WIDTh

SCPI.CALCulate(Ch).SELected.TRANsform.TIME.STEP.RTIMe

SCPI.CALCulate(Ch).SELected.TRANsform.TIME.STATe

SCPI.CALCulate(Ch).PARameter(Tr).SELect

Equivalent key

Analysis > Fault Location > Window > Kaiser Beta

Equivalent SCPI command

Syntax

:CALCulate{[1]-4}[:SELected]:TRANsform:TIME:KBESsel <numeric>

:CALCulate{[1]-4}[:SELected]:TRANsform:TIME:KBESsel?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":CALC1:TRAN:TIME:KBES 3"
20 OUTPUT 717;":CALC1:TRAN:TIME:KBES?"
30 ENTER 717;A
```

SCPI.CALCulate(*Ch*).SELected.TRANsform.TIME.LPFRequency

Object type

Property (**Write Only**)

Syntax

SCPI.CALCulate(*Ch*).SELected.TRANsform.TIME.LPFRequency = *Value*

Description

This command changes the frequency range to match with the low-pass type transformation of the transformation function of the time domain function, for the active trace of the selected channel (*Ch*).

Examples

```
SCPI.CALCulate(1).PARameter(1).SELect
SCPI.CALCulate(1).SELected.TRANsform.TIME.LPFRequency = 1E5
```

Related objects

SCPI.CALCulate(Ch).SELected.TRANsform.TIME.TYPE

SCPI.CALCulate(Ch).SELected.TRANsform.TIME.STATe

SCPI.CALCulate(Ch).PARameter(Tr).SELect

Equivalent key

Analysis > **Fault Location** > **Set Freq Low pass**

Equivalent SCPI command

Syntax

:CALCulate{[1]-4}[:SELected]:TRANsform:TIME:LPFRequency

Example of use

```
10 OUTPUT 717;":CALC1:TRAN:TIME:LPFR 1E5"
```

SCPI.CALCulate(*Ch*).SELected.TRANsform.TIME.REFLection.TYPE

Object Type

Property (**Read-Write**)

Syntax

SCPI.CALCulate(*Ch*).SELected.TRANsform.TIME.REFLection.TYPE = *Param*

Param = SCPI.CALCulate(*Ch*).SELected.TRANsform.TIME.REFLection.TYPE

Description

This command sets/gets the reflection time either one way or round trip, for the active trace of the selected channel (*Ch*).

Variable

Parameter	*Param*
Description	The reflection time either one way or round trip
Data Type	Character string type (String)
Range	Select from the following: • "OWAY": Specifies the One Way. • "RTRip": Specifies the Round Trip.
Preset Value	"RTRip"

Examples

```
Dim RefType as String
SCPI.CALCulate(4).SELected.TRANsform.TIME.REFLection.TYPE = "RTRip"
RefType = SCPI.CALCulate(4).SELected.TRANsform.TIME.REFLection.TYPE
```

Related Objects

Equivalent Key

Analysis > **Fault Location** > **Reflection Type**

Equivalent SCPI Command

Syntax

:CALCulate{[1]-4}[:SELected]:TRANsform:TIME:REFLection:TYPE {OWAY|RTRip}

:CALCulate{[1]-4}[:SELected]:TRANsform:TIME:REFLection:TYPE?

Query Response

{OWAY|RTR} <newline><^END>

Example of use

```
10 OUTPUT 717;":CALC1:TRAN:TIME:REFL:TYPE OWAY"
20 OUTPUT 717;":CALC1:TRAN:TIME:REFL:TYPE?"
30 ENTER 717;A$
```

SCPI.CALCulate(*Ch*).SELected.TRANsform.TIME.SPAN

Object type

Property (**Read-Write**)

Syntax

SCPI.CALCulate(*Ch*).SELected.TRANsform.TIME.SPAN = *Value*

Value = SCPI.CALCulate(*Ch*).SELected.TRANsform.TIME.SPAN

Description

This command selects the span value used for the transformation function of the time domain function, for the active trace of the selected channel (*Ch*).

Variable

Parameter	*Value*
Description	Span value
Data type	Double precision floating point type (Double)
Range	Varies depending on the frequency span and the number of points.
Preset value	2E-8
Unit	s (second)
Note	If the specified variable 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.

Examples

```
Dim Span As Double
SCPI.CALCulate(1).PARameter(1).SELect
SCPI.CALCulate(1).SELected.TRANsform.TIME.SPAN = 1E-8
Cent = SCPI.CALCulate(1).SELected.TRANsform.TIME.SPAN
```

Related objects

SCPI.CALCulate(Ch).SELected.TRANsform.TIME.CENTer

SCPI.CALCulate(Ch).SELected.TRANsform.TIME.STATe

SCPI.CALCulate(Ch).PARameter(Tr).SELect

Equivalent key

Analysis > Fault Location > Center

Equivalent SCPI command

Syntax

:CALCulate{[1]-4}[:SELected]:TRANsform:TIME:SPAN <numeric>

:CALCulate{[1]-4}[:SELected]:TRANsform:TIME:SPAN?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":CALC1:TRAN:TIME:SPAN 1E-8"
20 OUTPUT 717;":CALC1:TRAN:TIME:SPAN?"
30 ENTER 717;A
```

SCPI.CALCulate(*Ch*).SELected.TRANsform.TIME.STARt

Object type

Property (**Read-Write**)

Syntax

SCPI.CALCulate(*Ch*).SELected.TRANsform.TIME.STARt = *Value*

Value = SCPI.CALCulate(*Ch*).SELected.TRANsform.TIME.STARt

Description

This command sets/gets the start value used for the transformation function of the time domain function, for the active trace of the selected channel (*Ch*).

Variable

Parameter	*Value*
Description	Start value
Data type	Double precision floating point type (Double)
Range	Varies depending on the frequency span and the number of points.
Preset value	-1E-8
Unit	s (second)
Note	If the specified variable 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.

Examples

```
Dim Star As Double
SCPI.CALCulate(1).PARameter(1).SELect
SCPI.CALCulate(1).SELected.TRANsform.TIME.STARt = 0
Star = SCPI.CALCulate(1).SELected.TRANsform.TIME.STARt
```

Related objects

SCPI.CALCulate(Ch).SELected.TRANsform.TIME.STOP

SCPI.CALCulate(Ch).SELected.TRANsform.TIME.STATe

SCPI.CALCulate(Ch).PARameter(Tr).SELect

Equivalent key

Analysis > Fault Location > Start

Equivalent SCPI command

Syntax

:CALCulate{[1]-4}[:SELected]:TRANsform:TIME:STARt <numeric>

:CALCulate{[1]-4}[:SELected]:TRANsform:TIME:STARt?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":CALC1:TRAN:TIME:STAR 0"
20 OUTPUT 717;":CALC1:TRAN:TIME:STAR?"
30 ENTER 717;A
```

SCPI.CALCulate(*Ch*).SELected.TRANsform.TIME.STATe

Object type

Property (**Read-Write**)

Syntax

SCPI.CALCulate(*Ch*).SELected.TRANsform.TIME.STATe = *Status*

Status = SCPI.CALCulate(*Ch*).SELected.TRANsform.TIME.STATe

Description

This command turns ON/OFF the transformation function of the time domain function, for the active trace of the selected channel (*Ch*).

You can enable the transformation function only when the sweep type is the linear sweep and the number of points is three or more. If you execute this object to enable the transformation function when the sweep type is other than the linear sweep or the number of points is less than three, an error occurs and the object is ignored.

> NOTE When the sweep type is the power sweep, you cannot turn ON the transformation function. If you execute this object to turn ON the transformation function during the power sweep, an error occurs and the object is ignored.

Variable

Parameter	*Status*
Description	ON/OFF of the gating function
Data type	Boolean type (Boolean)
Range	Select from the following: • True or ON: Turns ON the transformation function. • False or OFF: Turns OFF the transformation function.
Preset value	False or OFF

Examples

```
Dim Trans As Boolean
SCPI.CALCulate(1).PARameter(1).SELect
SCPI.CALCulate(1).SELected.TRANsform.TIME.STATe = True
Trans = SCPI.CALCulate(1).SELected.TRANsform.TIME.STATe
```

Related objects

SCPI.CALCulate(Ch).PARameter(Tr).SELect

SCPI.SENSe(Ch).SWEep.TYPE

SCPI.SENSe(Ch).SWEep.POINts

Equivalent key

Analysis > **Fault Location** > **Transform**

Equivalent SCPI command

Syntax

:CALCulate{[1]-4}[:SELected]:TRANsform:TIME:STATe {ON|OFF|1|0}

:CALCulate{[1]-4}[:SELected]:TRANsform:TIME:STATe?

Query response

{1|0}<newline><^END>

Example of use

```
10 OUTPUT 717;":CALC1:TRAN:TIME:STAT ON"
20 OUTPUT 717;":CALC1:TRAN:TIME:STAT?"
30 ENTER 717;A
```

SCPI.CALCulate(*Ch*).SELected.TRANsform.TIME.STEP.RTIMe

Object type

Property (**Read-Write**)

Syntax

SCPI.CALCulate(*Ch*).SELected.TRANsform.TIME.STEP.RTIMe = *Value*

Value = SCPI.CALCulate(*Ch*).SELected.TRANsform.TIME.STEP.RTIMe

Description

This command sets/gets the shape of the Kayser Bessel window using the rise time of step signal used for the transformation function of the time domain function, for the active trace of the selected channel (*Ch*).

Variable

Parameter	*Value*
Description	The rise time of step signal
Data type	Double precision floating point type (Double)
Range	Varies depending on the frequency span.
Preset value	Varies depending on the frequency span.
Unit	s (second)
Note	If the specified variable 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.

Examples

```
Dim RTime As Double
SCPI.CALCulate(1).PARameter(1).SELect
SCPI.CALCulate(1).SELected.TRANsform.TIME.STEP.RTIMe = 1E-10
RTime = SCPI.CALCulate(1).SELected.TRANsform.TIME.STEP.RTIMe
```

Related objects

SCPI.CALCulate(Ch).SELected.TRANsform.TIME.IMPulse.WIDTh

SCPI.CALCulate(Ch).SELected.TRANsform.TIME.KBESsel

SCPI.CALCulate(Ch).SELected.TRANsform.TIME.STATe

SCPI.CALCulate(Ch).PARameter(Tr).SELect

Equivalent key

Analysis > **Fault Location** > **Center**

Equivalent SCPI command

Syntax

:CALCulate{[1]-4}[:SELected]:TRANsform:TIME:STEP:RTIMe <numeric>

:CALCulate{[1]-4}[:SELected]:TRANsform:TIME:STEP:RTIMe?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":CALC1:TRAN:TIME:STEP:RTIM 1E-10"
20 OUTPUT 717;":CALC1:TRAN:TIME:STEP:RTIM?"
30 ENTER 717;A
```

SCPI.CALCulate(*Ch*).SELected.TRANsform.TIME.STIMulus

Object Type

Property (**Read-Write**)

Syntax

SCPI.CALCulate(*Ch*).SELected.TRANsform.TIME.STIMulus = *Param*

Param = SCPI.CALCulate(*Ch*).SELected.TRANsform.TIME.STIMulus

Description

This command sets/gets the stimulus type used for the transformation function of the time domain function, for the active trace of the selected channel (*Ch*).

Variable

Parameter	*Param*
Description	The stimulus type
Data type	Character string type (String)
Range	Select from the following: • "IMPulse": Specifies the impulse • "STEP": Specifies the step
Preset value	"IMPulse"

Examples

```
Dim StimType As String
SCPI.CALCulate(1).PARameter(1).SELect
SCPI.CALCulate(1).SELected.TRANsform.TIME.STIMulus = "step"
StimType = SCPI.CALCulate(1).SELected.TRANsform.TIME.STIMulus
```

Related objects

SCPI.CALCulate(Ch).SELected.TRANsform.TIME.TYPE

SCPI.CALCulate(Ch).SELected.TRANsform.TIME.STATe

SCPI.CALCulate(Ch).PARameter(Tr).SELect

Equivalent key

Analysis > **Fault Location** > **Type** > **Bandpass|Lowpass Step|Lowpass Imp.**

NOTE When performing this operation from the front panel, you select the transformation type at the same time.

Equivalent SCPI command

Syntax

:CALCulate{[1]-4}[:SELected]:TRANsform:TIME:STIMulus {IMPulse|STEP}

:CALCulate{[1]-4}[:SELected]:TRANsform:TIME:STIMulus?

Query response

{IMP|STEP}<newline><^END>

Example of use

```
10 OUTPUT 717;":CALC1:TRAN:TIME LPAS"
20 OUTPUT 717;":CALC1:TRAN:TIME:STIM STEP"
30 OUTPUT 717;":CALC1:TRAN:TIME:STIM?"
40 ENTER 717;A$
```

SCPI.CALCulate(*Ch*).SELected.TRANsform.TIME.STOP

Object type

Property (**Read-Write**)

Syntax

SCPI.CALCulate(*Ch*).SELected.TRANsform.TIME.STOP = *Value*

Value = SCPI.CALCulate(*Ch*).SELected.TRANsform.TIME.STOP

Description

This command sets/gets the stop value used for the transformation function of the time domain function, for the active trace of the selected channel (*Ch*).

Variable

Parameter	*Value*
Description	Stop value
Data type	Double precision floating point type (Double)
Range	Varies depending on the frequency span and the number of points.
Preset value	1E-8
Unit	s (second)
Note	If the specified variable 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.

Examples

```
Dim Span As Double
SCPI.CALCulate(1).PARameter(1).SELect
SCPI.CALCulate(1).SELected.TRANsform.TIME.STOP = 2E-8
Cent = SCPI.CALCulate(1).SELected.TRANsform.TIME.STOP
```

Related objects

SCPI.CALCulate(Ch).SELected.TRANsform.TIME.STARt

SCPI.CALCulate(Ch).SELected.TRANsform.TIME.STATe

SCPI.CALCulate(Ch).PARameter(Tr).SELect

Equivalent key

Analysis > Fault Location > Stop

Equivalent SCPI command

Syntax

```
:CALCulate{[1]-4}[:SELected]:TRANsform:TIME:STOP <numeric>
:CALCulate{[1]-4}[:SELected]:TRANsform:TIME:STOP?
```

Query response

```
{numeric}<newline><^END>
```

Example of use

```
10 OUTPUT 717;":CALC1:TRAN:TIME:STOP 2E-8"
20 OUTPUT 717;":CALC1:TRAN:TIME:STOP?"
30 ENTER 717;A
```

SCPI.CALCulate(*Ch*).SELected.TRANsform.TIME.TYPE

Object type

Property (**Read-Write**)

Syntax

SCPI.CALCulate(*Ch*).SELected.TRANsform.TIME.TYPE = *Param*

Param = SCPI.CALCulate(*Ch*).SELected.TRANsform.TIME.TYPE

Description

This command sets/gets the transformation type used for the transformation function of the time domain function, for the active trace of the selected channel (*Ch*).

Variable

Parameter	*Param*
Description	The transformation type
Data type	Character string type (String)
Range	Select from the following: • "BPASs": Specifies the band-pass • "LPASs": Specifies the low-pass
Preset value	"BPASs"

Examples

```
Dim Typ As String
SCPI.CALCulate(1).PARameter(1).SELect
SCPI.CALCulate(1).SELected.TRANsform.TIME.SHAPe = "lpas"
Typ = SCPI.CALCulate(1).SELected.TRANsform.TIME.SHAPe
```

Related objects

SCPI.CALCulate(Ch).SELected.TRANsform.TIME.STIMulus

SCPI.CALCulate(Ch).SELected.TRANsform.TIME.STATe

SCPI.CALCulate(Ch).PARameter(Tr).SELect

Equivalent key

Analysis > **Fault Location** > **Type** > **Bandpass|Lowpass Step|Lowpass Imp.**

When performing this operation from the front panel, you select the stimulus type at the same time.

Equivalent SCPI command

Syntax

:CALCulate{[1]-4}[:SELected]:TRANsform:TIME[:TYPE] {BPASs|LPASs}

:CALCulate{[1]-4}[:SELected]:TRANsform:TIME[:TYPE]?

Query response

{BPAS|LPAS}<newline><^END>

Example of use

```
10 OUTPUT 717;":CALC1:TRAN:TIME LPAS"
20 OUTPUT 717;":CALC1:TRAN:TIME?"
30 ENTER 717;A$
```

SCPI.CALCulate(*Ch*).SELected.ZPARameter.DEFine

Object Type

Property (**Read-Write**)

Syntax

SCPI.CALCulate(*Ch*).SELected.ZPARameter.DEFine = *Param*

Param = SCPI.CALCulate(*Ch*).SELected.ZPARameter.DEFine

Description

This command sets/gets the measurement parameter of the selected trace (*Tr*), for the selected channel (*Ch*). When you define the parameters, follows the sequence shown in the examples below.

Variable

Parameter	*Param*
Description	Measurement parameter
Data Type	Character string type (String)
Range	Select one of the following parameters: Z|Y|Cp|Cs|Lp|Ls|Rp|Rs|D|Q
Preset Value	Z
Unit	-
Resolution	-

Examples

S-Parameter measurement port

For |Z|,|Y|

```
SCPI.CALCulate(1).PARameter.DEFine = "S11"
SCPI.SENSe.Z.METHod = "P1Reflection" (or "P2Reflection", "TSERies", "TSHunt")
SCPI.CALCulate(1).PARameter.DEFine = "Z"
SCPI.CALCulate(1).SELected.ZPARameter.DEFine = "Z" (or "Y")
SCPI.CALCulate(1).SELected.FORMat = "MLINear"
```

For Cp, Cs, Lp, Ls, Rp, Rs, D, Q

```
SCPI.CALCulate(1).PARameter.DEFine = "S11"
SCPI.SENSe.Z.METHod = "P1Reflection" (or "P2Reflection", "TSERies", "TSHunt")
SCPI.CALCulate(1).PARameter.DEFine = "Z"
```

```
SCPI.CALCulate(1).SELected.ZPARameter.DEFine = "CP" (or "CS", "LP", "LS", "RP, "RS" ,"D", "Q")
SCPI.CALCulate(1).SELected.FORMat = "REAL"
```

For θZ

```
SCPI.CALCulate(1).PARameter.DEFine = "S11"
SCPI.SENSe.Z.METHod = "P1Reflection" (or "P2Reflection", "TSERies", "TSHunt")
SCPI.CALCulate(1).PARameter.DEFine = "Z"
SCPI.CALCulate(1).SELected.ZPARameter.DEFine = "Z"
SCPI.CALCulate(1).SELected.FORMat = "Phase"
```

For |Y|

```
SCPI.CALCulate(1).PARameter.DEFine = "S11"
SCPI.SENSe.Z.METHod = "P1Reflection" (or "P2Reflection", "TSERies", "TSHunt")
SCPI.CALCulate(1).PARameter.DEFine = "Z"
SCPI.CALCulate(1).SELected.ZPARameter.DEFine = "Y"
SCPI.CALCulate(1).SELected.FORMat = "MLINear"
```

For θy

```
SCPI.CALCulate(1).PARameter.DEFine = "S11"
SCPI.SENSe.Z.METHod = "P1Reflection" (or "P2Reflection", "TSERies", "TSHunt")
SCPI.CALCulate(1).PARameter.DEFine = "Z"
SCPI.CALCulate(1).SELected.ZPARameter.DEFine = "Y"
SCPI.CALCulate(1).SELected.FORMat = "Phase"
```

For R

```
SCPI.CALCulate(1).PARameter.DEFine = "S11"
SCPI.SENSe.Z.METHod = "P1Reflection" (or "P2Reflection", "TSERies", "TSHunt")
SCPI.CALCulate(1).PARameter.DEFine = "Z"
SCPI.CALCulate(1).SELected.ZPARameter.DEFine =  "Z"
SCPI.CALCulate(1).SELected.FORMat = "Real"
```

For X

```
SCPI.CALCulate(1).PARameter.DEFine = "S11"
SCPI.SENSe.Z.METHod = "P1Reflection" (or "P2Reflection", "TSERies", "TSHunt")
SCPI.CALCulate(1).PARameter.DEFine = "Z"
SCPI.CALCulate(1).SELected.ZPARameter.DEFine =  "Z"
SCPI.CALCulate(1).SELected.FORMat = "Imag"
```

For G

```
SCPI.CALCulate(1).PARameter.DEFine = "S11"
SCPI.SENSe.Z.METHod = "P1Reflection" (or "P2Reflection", "TSERies", "TSHunt")
SCPI.CALCulate(1).PARameter.DEFine = "Z"
SCPI.CALCulate(1).SELected.ZPARameter.DEFine = "Y"
SCPI.CALCulate(1).SELected.FORMat = "Real"
```

For B

```
SCPI.CALCulate(1).PARameter.DEFine = "S11"
SCPI.SENSe.Z.METHod = "P1Reflection" (or "P2Reflection", "TSERies", "TSHunt")
SCPI.CALCulate(1).PARameter.DEFine = "Z"
SCPI.CALCulate(1).SELected.ZPARameter.DEFine = "Y"
SCPI.CALCulate(1).SELected.FORMat = "Imag"
```

Gain-Phase measurement port

For |Z|,|Y|

SCPI.CALCulate(1).PARameter.DEFine = "TR"
SCPI.SENSe.Z.METHod = "GSERies" (or "GSHunt")
SCPI.CALCulate(1).PARameter.DEFine = "Z"
SCPI.CALCulate(1).SELected.ZPARameter.DEFine = "Z" (or "Y")
SCPI.CALCulate(1).SELected.FORMat = "MLINear"

For Cp, Cs, Lp, Ls, Rp, Rs, D, Q

SCPI.CALCulate(1).PARameter.DEFine = "TR"
SCPI.SENSe.Z.METHod = "GSERies" (or "GSHunt")
SCPI.CALCulate(1).PARameter.DEFine = "Z"
SCPI.CALCulate(1).SELected.ZPARameter.DEFine = "CP" (or "CS", "LP", "LS", "RP, "RS" ,"D", "Q")
SCPI.CALCulate(1).SELected.FORMat = "REAL"

For θZ

SCPI.CALCulate(1).PARameter.DEFine = "TR"
SCPI.SENSe.Z.METHod = "GSERies" (or "GSHunt")
SCPI.CALCulate(1).PARameter.DEFine = "Z"
SCPI.CALCulate(1).SELected.ZPARameter.DEFine = "Z"
SCPI.CALCulate(1).SELected.FORMat = "Phase"

For |Y|

SCPI.CALCulate(1).PARameter.DEFine = "TR"
SCPI.SENSe.Z.METHod = "GSERies" (or "GSHunt")
SCPI.CALCulate(1).PARameter.DEFine = "Z"
SCPI.CALCulate(1).SELected.ZPARameter.DEFine = "Y"
SCPI.CALCulate(1).SELected.FORMat = "MLINear"

For θy

SCPI.CALCulate(1).PARameter.DEFine = "TR"
SCPI.SENSe.Z.METHod = "GSERies" (or "GSHunt")
SCPI.CALCulate(1).PARameter.DEFine = "Z"
SCPI.CALCulate(1).SELected.ZPARameter.DEFine = "Y"
SCPI.CALCulate(1).SELected.FORMat = "Phase"

For R

SCPI.CALCulate(1).PARameter.DEFine = "TR"
SCPI.SENSe.Z.METHod = "GSERies" (or "GSHunt")
SCPI.CALCulate(1).PARameter.DEFine = "Z"
SCPI.CALCulate(1).SELected.ZPARameter.DEFine = "Z"
SCPI.CALCulate(1).SELected.FORMat = "Real"

For X

SCPI.CALCulate(1).PARameter.DEFine = "TR"
SCPI.SENSe.Z.METHod = "GSERies" (or "GSHunt")
SCPI.CALCulate(1).PARameter.DEFine = "Z"

SCPI.CALCulate(1).SELected.ZPARameter.DEFine = "Z"
SCPI.CALCulate(1).SELected.FORMat = "Imag"

For G

SCPI.CALCulate(1).PARameter.DEFine = "TR"
SCPI.SENSe.Z.METHod = "GSERies" (or "GSHunt")
SCPI.CALCulate(1).PARameter.DEFine = "Z"
SCPI.CALCulate(1).SELected.ZPARameter.DEFine = "Y"
SCPI.CALCulate(1).SELected.FORMat = "Real"

For B

SCPI.CALCulate(1).PARameter.DEFine = "TR"
SCPI.SENSe.Z.METHod = "GSERies" (or "GSHunt")
SCPI.CALCulate(1).PARameter.DEFine = "Z"
SCPI.CALCulate(1).SELected.ZPARameter.DEFine = "Y"
SCPI.CALCulate(1).SELected.FORMat = "Imag"

Related Objects

SCPI.CALCulate(Ch).PARameter(Tr).DEFine

SCPI.CALCulate(Ch).SELected.FORMat

Equivalent Key

Meas > **Impedance Analysis Menu** > **|Z|** or

Meas > **Impedance Analysis Menu** > **|Y|** or

Meas > **Impedance Analysis Menu** > **Cp** or

Meas > **Impedance Analysis Menu** > **Cs** or

Meas > **Impedance Analysis Menu** > **Lp** or

Meas > **Impedance Analysis Menu** > **Ls** or

Meas > **Impedance Analysis Menu** > **Rp** or

Meas > **Impedance Analysis Menu** > **Rs** or

Meas > **Impedance Analysis Menu** > **D** or

Meas > **Impedance Analysis Menu** > **Q**

Equivalent SCPI Command

Syntax

:CALCulate{[1]-4}[:SELected]:ZPARameter:DEFine <string>

:CALCulate{[1]-4}[:SELected]:ZPARameter:DEFine?

Query Response

<string><newline><^END>

Example of use

See the examples in COM

SCPI.CALCulate(*Ch*).SRL.CONNector(*Pt*).CAPacitance

Object Type

Property (**Read-Write**)

Syntax

SCPI.CALCulate(*Ch*).SRL.CONNector(*Pt*).CAPacitance = *Value*

Value = SCPI.CALCulate(*Ch*).SRL.CONNector(*Pt*).CAPacitance

Description

This command sets/gets the connector capacitance value of the specified port (*Pt*) for the connector mismatch compensation, for the active trace of the selected channel (*Ch*).

Variable

Parameter	*Value*
Description	Connector capacitance for compensation
Data Type	Double precision floating point type (Double)
Range	-2E-12 to 2E-12
Preset Value	0
Unit	F (farad)
Note	If the specified variable 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.

Examples

```
Dim ConnCap as Double
SCPI.CALCulate(4).SRL.CONNector(2).CAPacitance = 1E-3
ConnCap = SCPI.CALCulate(4).SRL.CONNector(2).CAPacitance
```

Related Objects

SCPI.CALCulate(Ch).SRL.CONNector(Pt).IMMediate

SCPI.CALCulate(Ch).SRL.CONNector(Pt).LENGth

Equivalent Key

Analysis > **SRL** > **Port1 Connector|Port2 Connector** > **Capacitance**

Equivalent SCPI Command

Syntax

:CALCulate{[1]-4}:SRL:CONNector{[1]|2}:CAPacitance <numeric>

:CALCulate{[1]-4}:SRL:CONNector{[1]|2}:CAPacitance?

Query Response

{numeric} <newline><^END>

Example of use

```
10 OUTPUT 717;":CALC1:SRL:CONN2:CAP 1.2E-2"
20 OUTPUT 717;":CALC1:SRL:CONN2:CAP?"
30 ENTER 717;A
```

SCPI.CALCulate(*Ch*).SRL.CONNector(*Pt*).IMMediate

Object Type

Method (**Write Only**)

Syntax

SCPI.CALCulate(*Ch*).SRL.CONNector(*Pt*).IMMediate

Description

This command measures the terminated cable connected to the specified port (*Pt*) and automatically sets the connector length and capacitance values for connector mismatch compensation, for the active trace of the selected channel (*Ch*).

Examples

SCPI.CALCulate(2).SRL.CONNector(1).IMMediate

Related Objects

SCPI.CALCulate(Ch).SRL.CONNector(Pt).CAPacitance

SCPI.CALCulate(Ch).SRL.CONNector(Pt).LENGth

Equivalent Key

Analysis > **SRL** > **Port1 Connector|Port2 Connector** > **Measure Connector**

Equivalent SCPI Command

Syntax

:CALCulate{[1]-4}:SRL:CONNector{[1]|2}:IMMediate

Example of use

```
10 OUTPUT 717;":CALC1:SRL:CONN2:IMM"
```

SCPI.CALCulate(*Ch*).SRL.CONNector(*Pt*).IMPedance

Object Type

Property (**Read Only**)

Syntax

Value = SCPI.CALCulate(*Ch*).SRL.CONNector(*Pt*).IMPedance

Description

This command gets the average cable impedance of specified port (*Pt*) for the SRL calculation, for the active trace of the selected channel (*Ch*).

Variable

Parameter	*Value*
Description	Impedance value
Data Type	Double precision floating point type (Double)
Unit	Ω

Examples

```
Dim PortImp as Double
PortImp = SCPI.CALCulate(2).SRL.CONNector(2).IMPedance
```

Related Objects

SCPI.CALCulate(Ch).SRL.CONNector(Pt).IMMediate

Equivalent Key

No equivalent key is available on the front panel.

Equivalent SCPI Command

Syntax

:CALCulate{[1]-4}:SRL:CONNector{[1]|2}:IMPedance?

Query Response

{numeric} <newline><^END>

Example of use

```
10 OUTPUT 717;":CALC1:SRL:CONN2:IMP?"
20 ENTER 717;A
```

SCPI.CALCulate(*Ch*).SRL.CONNector(*Pt*).LENGth

Object Type

Property (**Read-Write**)

Syntax

SCPI.CALCulate(*Ch*).SRL.CONNector(*Pt*).LENGth = *Value*

Value = SCPI.CALCulate(*Ch*).SRL.CONNector(*Pt*).LENGth

Description

This command sets/gets the connector length value of specified port (*Pt*) for the connector mismatch compensation, for the active trace of the selected channel (*Ch*).

Variable

Parameter	*Value*
Description	Connector length value for compensation
Data Type	Double precision floating point type (Double)
Range	-0.02 to 0.2
Preset Value	0
Unit	m (meters)

Examples

```
Dim Leng as Double
SCPI.CALCulate(2).SRL.CONNector(1).LENGth = -0.01
Leng = SCPI.CALCulate(2).SRL.CONNector(1).LENGth
```

Related Objects

SCPI.CALCulate(Ch).SRL.CONNector(Pt).CAPacitance

SCPI.CALCulate(Ch).SRL.CONNector(Pt).IMMediate

Equivalent Key

Analysis > **SRL** > **Port1 Connector|Port2 Connector** > **Length**

Equivalent SCPI Command

Syntax

:CALCulate{[1]-4}:SRL:CONNector{[1]|2}:LENGth <numeric>

:CALCulate{[1]-4}:SRL:CONNector{[1]|2}:LENGth?

Query Response

{numeric} <newline><^END>

Example of use

```
10 OUTPUT 717;":CALC1:SRL:CONN2:LENG -0.02"
20 OUTPUT 717;":CALC1:SRL:CONN2:LENG?"
30 ENTER 717;A
```

SCPI.CALCulate(*Ch*).SRL.IMPedance.AUTO.CUToff

Object Type

Property (**Read-Write**)

Syntax

SCPI.CALCulate(*Ch*).SRL.IMPedance.AUTO.CUToff = *Value*

Value = SCPI.CALCulate(*Ch*).SRL.IMPedance.AUTO.CUToff

Description

This command sets/gets the cutoff frequency of the auto calculation for the average cable impedance, for the active trace of the selected channel (*Ch*).

Variable

Parameter	*Value*
Description	The cutoff frequency of the auto calculation
Data Type	Double precision floating point type (Double)
Range	5 to 3E+9
Preset Value	2.1E+8
Unit	Hz
Note	If the specified variable 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.

Examples

```
Dim CutFreq as Double
SCPI.CALCulate(2).SRL.IMPedance.AUTO.CUToff = 2.1E9
CutFreq = SCPI.CALCulate(2).SRL.IMPedance.AUTO.CUToff
```

Related Objects

SCPI.CALCulate(Ch).SRL.IMPedance.AUTO.STATe

SCPI.CALCulate(Ch).SRL.IMPedance.MANual

Equivalent Key

Analysis > **SRL** > **Z Cutoff Freq**

Equivalent SCPI Command

Syntax

:CALCulate{[1]-4}:SRL:IMPedance:AUTO:CUToff <numeric>

:CALCulate{[1]-4}:SRL:IMPedance:AUTO:CUToff?

Query Response

{numeric} <newline><^END>

Example of use

```
10 OUTPUT 717;":CALC1:SRL:IMP:AUTO:CUT 0"
20 OUTPUT 717;":CALC1:SRL:IMP:AUTO:CUT?"
30 ENTER 717;A
```

SCPI.CALCulate(*Ch*).SRL.IMPedance.AUTO.STATe

Object Type

Property (**Read-Write**)

Syntax

SCPI.CALCulate(*Ch*).SRL.IMPedance.AUTO.STATe = *Status*

Status = SCPI.CALCulate(*Ch*).SRL.IMPedance.AUTO.STATe

Description

This command turns ON/OFF the auto impedance calculation function of the SRL measurement, for the active trace of the selected channel (*Ch*).

Variable

Parameter	*Status*
Description	Auto impedance calculation status
Data Type	Boolean type (Boolean)
Range	Select from the following: • True or ON: Turns ON the auto impedance calculation function. • False or OFF: Turns OFF the auto impedance calculation function.
Preset Value	True or ON

Examples

```
Dim Stat as Boolean
SCPI.CALCulate(4).SRL.IMPedance.AUTO.STATe = True
Stat = SCPI.CALCulate(4).SRL.IMPedance.AUTO.STATe
```

Related Objects

SCPI.CALCulate(Ch).SRL.IMPedance.AUTO.CUToff

SCPI.CALCulate(Ch).SRL.IMPedance.MANual

Equivalent Key

Analysis > **SRL** > **Auto Z**

Equivalent SCPI Command

Syntax

:CALCulate{[1]-4}:SRL:IMPedance:AUTO[:STATe] {ON|OFF|1|0}

:CALCulate{[1]-4}:SRL:IMPedance:AUTO[:STATe]?

Query Response

{1|0} <newline><^END>

Example of use

```
10 OUTPUT 717;":CALC2:SRL:IMP:AUTO ON"
20 OUTPUT 717;":CALC2:SRL:IMP:AUTO?"
30 ENTER 717;A
```

SCPI.CALCulate(*Ch*).SRL.IMPedance.MANual

Object Type

Property (**Read-Write**)

Syntax

SCPI.CALCulate(*Ch*).SRL.IMPedance.MANual = *Value*

Value = SCPI.CALCulate(*Ch*).SRL.IMPedance.MANual

Description

This command sets/gets the average cable impedance to be used in the SRL calculation, for the active trace of the selected channel (*Ch*).

Variable

Parameter	*Value*
Description	The average cable impedance value
Data Type	Double precision floating point type (Double)
Range	10 to 1000
Preset Value	50
Unit	Ω

Examples

```
Dim Imp as Double
SCPI.CALCulate(1).SRL.IMPedance.MANual = 75
Imp = SCPI.CALCulate(1).SRL.IMPedance.MANual
```

Related Objects

SCPI.CALCulate(Ch).SRL.IMPedance.AUTO.CUToff

SCPI.CALCulate(Ch).SRL.IMPedance.AUTO.STATe

Equivalent Key

Analysis > **SRL** > **Manual Z**

Equivalent SCPI Command

Syntax

:CALCulate{[1]-4}:SRL:IMPedance:MANual <numeric>

:CALCulate{[1]-4}:SRL:IMPedance:MANual?

Query Response

{numeric} <newline><^END>

Example of use

```
10 OUTPUT 717;":CALC1:SRL:IMP:MAN 0"
20 OUTPUT 717;":CALC1:SRL:IMP:MAN?"
30 ENTER 717;A
```

SCPI.CALCulate(*Ch*).SRL.STATe

Object Type

Property (**Read-Write**)

Syntax

SCPI.CALCulate(*Ch*).SRL.STATe = *Status*

Status = SCPI.CALCulate(*Ch*).SRL.STATe

Description

This command turns ON/OFF the SRL measurement function, for the active trace of the selected channel (*Ch*).

Variable

Parameter	*Status*
Description	SRL measurement function status
Data Type	Boolean type (Boolean)
Range	Select from the following: • True or ON: Turns ON the SRL measurement function. • False or OFF: Turns OFF the SRL measurement function.
Preset Value	False or OFF

Examples

```
Dim SRLStat as Boolean
SCPI.CALCulate(2).SRL.STATe = True
SRLStat = SCPI.CALCulate(2).SRL.STATe
```

Related Objects

SCPI.CALCulate(Ch).SRL.CONNector(Pt).CAPacitance

SCPI.CALCulate(Ch).SRL.CONNector(Pt).LENGth

SCPI.CALCulate(Ch).SRL.IMPedance.MANual

Equivalent Key

Analysis > **SRL** > **SRL**

Equivalent SCPI Command

Syntax

```
:CALCulate{[1]-4}:SRL[:STATe] {ON|OFF|1|0}
:CALCulate{[1]-4}:SRL[:STATe]?
```

Query Response

```
{1|0} <newline><^END>
```

Example of use

```
10 OUTPUT 717;":CALC1:SRL ON"
20 OUTPUT 717;":CALC1:SRL?"
30 ENTER 717;A
```

CONTROL

SCPI.CONTrol.HANDler.A.DATA

Object type

Method (**Write Only**)

Syntax

SCPI.CONTrol.HANDler.A.DATA = *Value*

Description

This command sets/gets information of output port A (A0 to A7) of the handler I/O. Port information is output as 8-bit binary data using A0 as LSB and A7 as MSB.

Variable

Parameter	*Value*
Description	Port information (output)
Data type	Long integer type (Long)
Range	0 to 255
Note	If the specified variable 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.

Example of use

SCPI.CONTrol.HANDler.A.DATA = 15

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

:CONTrol:HANDler:A[:DATA] <numeric>

Example of use

```
10 OUTPUT 717;":CONT:HAND:A 15"
20 OUTPUT 717;":CONT:HAND:A:DATA 15"
```

SCPI.CONTrol.HANDler.B.DATA

Object type

Method (**Write Only**)

Syntax

SCPI.CONTrol.HANDler.B.DATA = *Value*

Description

This command sets/gets information of output port B (B0 to B7) of the handler I/O. Port information is output as 8-bit binary data using B0 as LSB and B7 as MSB.

NOTE The bit 6 of the data outputted by this project is ignored when outputting the INDEX signal is turned ON (specifying True with the SCPI.CONTrol.HANDler.EXTension.INDex.STATe object).

NOTE The bit 7 of the data outputted by this project is ignored when outputting the READY FOR TRIGGER signal is turned ON (specifying True with the SCPI.CONTrol.HANDler.EXTension.RTRigger.STATe object).

Variable

Parameter	*Value*
Description	Port information (output)
Data type	Long integer type (Long)
Range	0 to 255
Note	If the specified variable 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.

Examples

```
SCPI.CONTrol.HANDler.B.DATA = 15
```

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

:CONTrol:HANDler:B[:DATA] <numeric>

Example of use

```
10 OUTPUT 717;":CONT:HAND:B 15"
20 OUTPUT 717;":CONT:HAND:B:DATA 15"
```

SCPI.CONTrol.HANDler.C.DATA

Object type

Property (**Read-Write**)

Syntax

SCPI.CONTrol.HANDler.C.DATA = *Value* (for output port)

Value = SCPI.CONTrol.HANDler.C.DATA (for input port)

Description

When input/output port C of the handler I/O is set to the output port, it outputs port information to the output port C (C0 to C3).

When input/output port C of the handler I/O is set to the input port, it reads out port information inputted to port C (C0 to C3).

Port information is input/output as 4-bit binary data, using C0 as LSB and C3 as MSB.

Variable

Parameter	*Value*
Description	Port information (output/input)
Data type	Long integer type (Long)
Range	0 to 15
Note	If the specified variable 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.

Examples

```
SCPI.CONTrol.HANDler.C.MODE = "outp"
SCPI.CONTrol.HANDler.C.DATA = 8
```

```
Dim HdlCinp As Long
SCPI.CONTrol.HANDler.C.MODE = "inp"
HdlCinp = SCPI.CONTrol.HANDler.C.DATA
```

Related objects

SCPI.CONTrol.HANDler.C.MODE

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

:CONTrol:HANDler:C[:DATA] <numeric>

:CONTrol:HANDler:C[:DATA]?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":CONT:HAND:C:MODE OUTP"
20 OUTPUT 717;":CONT:HAND:C 15"
```

```
10 OUTPUT 717;":CONT:HAND:C:MODE INP"
20 OUTPUT 717;":CONT:HAND:C?"
30 ENTER 717;A
```

SCPI.CONTrol.HANDler.C.MODE

Object type

Property (**Read-Write**)

Syntax

SCPI.CONTrol.HANDler.C.MODE = *Param*

Param = SCPI.CONTrol.HANDler.C.MODE

Description

This command sets/gets the input/output direction of port C of the handler I/O.

Variable

Parameter	*Param*
Description	Input/output direction of port C
Data type	Character string type (String)
Range	Select from the following: "INPut":Sets the port C to input. "OUTPut":Sets the port C to output.
Preset value	"INPut"

Examples

```
Dim HdlCmode As String
SCPI.CONTrol.HANDler.C.MODE = "OUTP"
HdlCmode = SCPI.CONTrol.HANDler.C.MODE
```

Related objects

SCPI.CONTrol.HANDler.C.DATA

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

:CONTrol:HANDler:C:MODE {INPut|OUTPut}

:CONTrol:HANDler:C:MODE?

Query response

{INP|OUTP}<newline><^END>

Example of use

```
10 OUTPUT 717;":CONT:HAND:C:MODE OUTP"
20 OUTPUT 717;":CONT:HAND:C:MODE?"
30 ENTER 717;A$
```

SCPI.CONTrol.HANDler.D.DATA

Object type

Property (**Read-Write**)

Syntax

SCPI.CONTrol.HANDler.D.DATA = *Value* (for output port)

Value = SCPI.CONTrol.HANDler.D.DATA (for input port)

Description

When input/output port D of the handler I/O is set to the output port, it outputs port information to output port D (D0 to D3).

When input/output port D of the handler I/O is set to the input port, it reads out port information to input to port D (D0 to D3).

Port information is output as 4-bit binary data using D0 as LSB and D3 as MSB.

Variable

Parameter	*Value*
Description	Port information (output/input)
Data type	Long integer type (Long)
Range	0 to 15
Note	If the specified variable 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.

Examples

```
SCPI.CONTrol.HANDler.D.MODE = "outp"
SCPI.CONTrol.HANDler.D.DATA = 8

Dim HdlDinp As Long
SCPI.CONTrol.HANDler.D.MODE = "inp"
HdlDinp = SCPI.CONTrol.HANDler.D.DATA
```

Related objects

SCPI.CONTrol.HANDler.D.MODE

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

:CONTrol:HANDler:D[:DATA] <numeric>

:CONTrol:HANDler:D[:DATA]?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":CONT:HAND:D:MODE OUTP"
20 OUTPUT 717;":CONT:HAND:D 15"
```

```
10 OUTPUT 717;":CONT:HAND:D:MODE INP"
20 OUTPUT 717;":CONT:HAND:D?"
30 ENTER 717;A
```

SCPI.CONTrol.HANDler.D.MODE

Object type

Property (**Read-Write**)

Syntax

SCPI.CONTrol.HANDler.D.MODE = *Param*

Param = SCPI.CONTrol.HANDler.D.MODE

Description

This command sets/gets the input/output direction of port D of the handler I/O.

Variable

Parameter	*Param*
Description	Input/output direction of port D
Data type	Character string type (String)
Range	Select from the following: "INPut": Sets the port D to input. "OUTPut":Sets the port D to output.
Preset value	"INPut"

Examples

```
Dim HdlDmode As String
SCPI.CONTrol.HANDler.D.MODE = "OUTP"
HdlDmode = SCPI.CONTrol.HANDler.D.MODE
```

Related objects

SCPI.CONTrol.HANDler.D.DATA

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

:CONTrol:HANDler:D:MODE {INPut|OUTPut}

```
:CONTrol:HANDler:D:MODE?
```

Query response

```
{INP|OUTP}<newline><^END>
```

Example of use

```
10 OUTPUT 717;":CONT:HAND:D:MODE OUTP"
20 OUTPUT 717;":CONT:HAND:D:MODE?"
30 ENTER 717;A$
```

SCPI.CONTrol.HANDler.E.DATA

Object type

Property (**Read-Write**)

Syntax

SCPI.CONTrol.HANDler.E.DATA = *Value* (for output)

Value = SCPI.CONTrol.HANDler.E.DATA (for input port)

Description

When input/output port E (port C + port D) of the handler I/O is set to the output port, it outputs port information to output port E (C0 to D3).

When input/output port E of the handler I/O is set to the input port, it reads out port information inputted to port E (C0 to D3).

Port information is output as 8-bit binary data using C0 as LSB and D3 as MSB.

Variable

Parameter	*Value*
Description	Port information (output/input)
Data type	Long integer type (Long)
Range	0 to 255
Note	If the specified variable 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.

Examples

```
SCPI.CONTrol.HANDler.C.MODE = "outp"
SCPI.CONTrol.HANDler.D.MODE = "outp"
SCPI.CONTrol.HANDler.E.DATA = 128
```

```
Dim HdlEinp As Long
SCPI.CONTrol.HANDler.C.MODE = "inp"
SCPI.CONTrol.HANDler.D.MODE = "inp"
HdlEinp = SCPI.CONTrol.HANDler.E.DATA
```

Related objects

SCPI.CONTrol.HANDler.C.MODE

SCPI.CONTrol.HANDler.D.MODE

SCPI.CONTrol.HANDler.C.DATA

SCPI.CONTrol.HANDler.D.DATA

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

:CONTrol:HANDler:E[:DATA] <numeric>

:CONTrol:HANDler:E[:DATA]?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":CONT:HAND:C:MODE OUTP"
20 OUTPUT 717;":CONT:HAND:D:MODE OUTP"
30 OUTPUT 717;":CONT:HAND:E 128"
```

```
10 OUTPUT 717;":CONT:HAND:C:MODE INP"
20 OUTPUT 717;":CONT:HAND:D:MODE INP"
30 OUTPUT 717;":CONT:HAND:E?"
40 ENTER 717;A
```

SCPI.CONTrol.HANDler.EXTension.INDex.STATe

Object type

Property (**Read-Write**)

Syntax

SCPI.CONTrol.HANDler.EXTension.INDex.STATe = *Status*

Status = SCPI.CONTrol.HANDler.EXTension.INDex.STATe

Description

Turns ON/OFF output of the INDEX signal to B6 of the handler I/O.

NOTE When you use port B6 as the output port, turn OFF the INDEX signal output. When output of the INDEX signal is turned ON, the bit 6 of the data output by the SCPI.CONTrol.HANDler.B.DATA object (the bit 14 of the data outputted by the SCPI.CONTrol.HANDler.F.DATA object) is ignored.

Variable

Parameter	*Status*
Description	ON/OFF of the INDEX signal output
Data type	Boolean type (Boolean)
Range	Select from the following: True or ON:Turns ON the INDEX signal output. False or OFF: Turns OFF the INDEX signal output.
Preset value	False or OFF

Examples

```
Dim Indx As Boolean
SCPI.CONTrol.HANDler.EXTension.INDex.STATe = 1
Indx = SCPI.CONTrol.HANDler.EXTension.INDex.STATe
```

Related objects

SCPI.CONTrol.HANDler.EXTension.RTRigger.STATe

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

```
:CONTrol:HANDler[:EXTension]:INDex:STATe {ON|OFF|1|0}
:CONTrol:HANDler[:EXTension]:INDex:STATe?
```

Query response

```
{1|0}<newline><^END>
```

Example of use

```
10 OUTPUT 717;":CONT:HAND:IND:STAT ON"
20 OUTPUT 717;":CONT:HAND:IND:STAT?"
30 ENTER 717;A
```

SCPI.CONTrol.HANDler.EXTension.RTRigger.STATe

Object type

Property (**Read-Write**)

Syntax

SCPI.CONTrol.HANDler.EXTension.RTRigger.STATe = *Status*

Status = SCPI.CONTrol.HANDler.EXTension.RTRigger.STATe

Description

This command turns ON/OFF the output of READY FOR TRIGGER signal to B7 of the handler I/O.

NOTE When you use port B7 as the output port, turn OFF the READY FOR TRIGGER signal output. When outputting the READY FOR TRIGGER signal is turned ON, the bit 7of the data output by the SCPI.CONTrol.HANDler.B.DATA object (the bit 15 of the data output by the SCPI.CONTrol.HANDler.F.DATA object) is ignored.

Variable

Parameter	*Status*
Description	ON/OFF of the READY FOR TRIGGER signal output
Data type	Boolean type (Boolean)
Range	Select from the following: True or ON:Turns ON the READY FOR TRIGGER signal output. False or OFF:Turns OFF the READY FOR TRIGGER signal output.
Preset value	False or OFF

Examples

```
Dim RdyTrig As Boolean
SCPI.CONTrol.HANDler.EXTension.RTRigger.STATe = 0
RdyTrig = SCPI.CONTrol.HANDler.EXTension.RTRigger.STATe
```

Related objects

SCPI.CONTrol.HANDler.EXTension.INDex.STATe

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

:CONTrol:HANDler[:EXTension]:RTRigger:STATe {ON|OFF|1|0}

:CONTrol:HANDler[:EXTension]:RTRigger:STATe?

Query response

{1|0}<newline><^END>

Example of use

```
10 OUTPUT 717;":CONT:HAND:RTR:STAT ON"
20 OUTPUT 717;":CONT:HAND:RTR:STAT?"
30 ENTER 717;A
```

SCPI.CONTrol.HANDler.F.DATA

Object type

Method (**Write Only**)

Syntax

SCPI.CONTrol.HANDler.F.DATA = *Value*

Description

Outputs port information to output port F (port A + port B) of the handler I/O. Port information is output as 16-bit binary using A0 as LSB and B7 as MSB.

NOTE The bit 14 of the data output by this project is ignored when outputting the INDEX signal is turned ON (specifying True with the SCPI.CONTrol.HANDler.EXTension.INDex.STATe object).

NOTE The bit 15 of the data output by this project is ignored when outputting the READY FOR TRIGGER signal is turned ON (specifying True with the SCPI.CONTrol.HANDler.EXTension.RTRigger.STATe object).

Variable

Parameter	*Value*
Description	Port information (output)
Data type	Long integer type (Long)
Range	0 to 65535
Note	If the specified variable 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.

Examples

```
SCPI.CONTrol.HANDler.F.DATA = 511
```

Related objects

SCPI.CONTrol.HANDler.A.DATA

SCPI.CONTrol.HANDler.B.DATA

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

```
:CONTrol:HANDler:F[:DATA] <numeric>
```

Example of use

```
10 OUTPUT 717;":CONT:HAND:F 511"
```

SCPI.CONTrol.HANDler.OUTPut(*Num*).DATA

Object type

Property (**Read-Write**)

Syntax

SCPI.CONTrol.HANDler.OUTPut(*Num*).DATA = *Value*

Value = SCPI.CONTrol.HANDler.OUTPut(*Num*).DATA

Description

This command sets/gets data to OUTPUT1 or OUTPUT2 of the handler I/O.

Variable

Parameter	*Num*
Description	Number of the OUTPUT terminal
Data type	Long integer type (Long)
Range	1 or 2
Preset value	1
Note	If the specified variable is out of the allowable setup range, an error occurs when executed.

Parameter	*Value*
Description	Polarity (High/Low)
Data type	Long integer type (Long)
Range	Select from the following: • 1: Specifies LOW. • 0: Specifies HIGH.

Examples

```
Dim HdlPol As Long
SCPI.CONTrol.HANDler.OUTPut(1).DATA = 1
HdlPol = SCPI.CONTrol.HANDler.OUTPut(1).DATA
```

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

:CONTrol:HANDler:OUTPut{[1]|2}[:DATA] {1|0}

:CONTrol:HANDler:OUTPut{[1]|2}[:DATA]?

Query response

{1|0}<newline><^END>

Example of use

```
10 OUTPUT 717;":CONT:HAND:OUTP1 1"
20 OUTPUT 717;":CONT:HAND:OUTP1?"
30 ENTER 717;A
```

DISPLAY

SCPI.DISPlay.ANNotation.FREQuency.STATe

Object type

Property (**Read-Write**)

Syntax

SCPI.DISPlay.ANNotation.FREQuency.STATe = *Status*

Status = SCPI.DISPlay.ANNotation.FREQuency.STATe

Description

This command turns ON/OFF the frequency display on the LCD display.

Variable

Parameter	*Status*
Description	Sets/Gets ON/OFF state of the frequency display
Data type	Boolean type (Boolean)
Range	Select either of the following: • True or ON: Turns ON the frequency display. • False or OFF: Turns OFF the frequency display.
Preset value	True or ON

Examples

```
Dim DispFreq As Boolean
SCPI.DISPlay.ANNotation.FREQuency.STATe = 0
DispFreq = SCPI.DISPlay.ANNotation.FREQuency.STATe
```

Equivalent key

Display > **Frequency**

Equivalent SCPI command

Syntax

:DISPlay:ANNotation:FREQuency[:STATe] {ON|OFF|1|0}

:DISPlay:ANNotation:FREQuency[:STATe]?

Query response

{1|0}<newline><^END>

Example of use

```
10 OUTPUT 717;":DISP:ANN:FREQ OFF"
20 OUTPUT 717;":DISP:ANN:FREQ?"
30 ENTER 717;A
```

SCPI.DISPlay.CCLear

Object type

Method (**Write Only**)

Syntax

SCPI.DISPlay.CCLear

Description

This command clears the error message displayed in the status bar (at the bottom of the LCD display).

Examples

```
SCPI.DISPlay.CCLear
```

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

:DISPlay:CCLear

Example of use

```
10 OUTPUT 717;":DISP:CCL"
```

SCPI.DISPlay.CLOCk

Object type

Property (**Read-Write**)

Syntax

SCPI.DISPlay.CLOCk = *Status*

Status = SCPI.DISPlay.CLOCk

Description

This command turns ON/OFF the clock display in the instrument status bar (at the right bottom of the LCD display).

Variable

Parameter	*Status*
Description	ON/OFF of the clock display
Data type	Boolean type (Boolean)
Range	Select either of the following: • True or ON: Turns ON the clock display. • False or OFF: Turns OFF the clock display.
Preset value	True or ON

Examples

```
Dim DispTime As Boolean
SCPI.DISPlay.CLOCk = ON
DispTime = SCPI.DISPlay.CLOCk
```

Equivalent key

System > **Misc Setup** > **Clock Setup** > **Show Clock**

Equivalent SCPI command

Syntax

:DISPlay:CLOCk {ON|OFF|1|0}

:DISPlay:CLOCk?

Query response

```
{1|0}<newline><^END>
```

Example of use

```
10 OUTPUT 717;":DISP:CLOC OFF"
20 OUTPUT 717;":DISP:CLOC?"
30 ENTER 717;A
```

SCPI.DISPlay.COLor(*Dnum*).BACK

Object type

Property (**Read-Write**)

Syntax

SCPI.DISPlay.COLor(*Dnum*).BACK = *Data*

Data = SCPI.DISPlay.COLor(*Dnum*).BACK

Description

This command sets/gets the background color for normal display (*Dnum*:1) and inverted display (Dnum:2).

Variable

Parameter	*Dnum*
Description	Select either of the following: 1: Normal display 2: Inverted display
Data type	Long integer type (Long)
Range	1 or 2
Preset value	1
Note	If the specified variable is out of the allowable setup range, an error occurs when the command is executed.

Parameter	*Data*
Description	Indicates 3-element array data. • *Data(0) :* Sets amount of red. • *Data(1) :* Sets amount of green. • *Data(2) :* Sets amount of blue. The index of the array starts from 0.
Data type	Variant type (Variant)

Range	• *Data(0)* 0 to 5 • *Data(1)* 0 to 5 • *Data(2)* 0 to 5
Resolution	1
Note	If the specified variable 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.

Examples

```
Dim BackColor As Variant
SCPI.DISPlay.COLor(1).BACK = Array(1,2,3)
BackColor = SCPI.DISPlay.COLor(1).BACK
```

Related objects

SCPI.DISPlay.COLor(Dnum).RESet

Equivalent key

System > **Misc Setup** > **Display Setup** > **Color Setup** > **Normal|Invert** > **Background**

Equivalent SCPI command

Syntax

:DISPlay:COLor{[1]|2}:BACK <numeric 1>,<numeric 2>,<numeric 3>

:DISPlay:COLor{[1]|2}:BACK?

Query response

{numeric 1},{numeric 2},{numeric 3}<newline><^END>

Example of use

```
10 OUTPUT 717;":DISP:COL:BACK 1,2,3"
20 OUTPUT 717;":DISP:COL:BACK?"
30 ENTER 717;A,B,C
```

SCPI.DISPlay.COLor(*Dnum*).GRATicule(*Gnum*)

Object type

Property (**Read-Write**)

Syntax

SCPI.DISPlay.COLor(*Dnum*).GRATicule(*Gnum*) = *Data*

Data = SCPI.DISPlay.COLor(*Dnum*).GRATicule(*Gnum*)

Description

This command sets/gets:

1. Color of the graticule label.
2. Outer frame line of the graph (*Gnum*:1).
3. Color of the grid line of the graph (*Gnum*:2).

for the normal display (*Dnum*:1) and inverted display (*Dnum*:2).

Variable

Parameter	*Gnum*
Description	The number of items: 1: The outer frame line of the graph 2: The color of the grid line of the graph
Data type	Long integer type (Long)
Range	1 to 2
Preset value	1
Note	If the specified variable is out of the allowable setup range, an error occurs when executed.

Parameter	*Data*
Description	Indicates 3-element array data. • *Data(0) :* Sets amount of red. • *Data(1) :* Sets amount of green.

	• *Data(2) :* Sets amount of blue. The index of the array starts from 0.
Data type	Variant type (Variant)
Range	• *Data(0) :* 0 to 5 • *Data(1) :* 0 to 5 • *Data(2) :* 0 to 5
Resolution	1
Note	If the specified variable 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.

Examples

```
Dim GritColor As Variant
SCPI.DISPlay.COLor(1).GRATicule(1) = Array(1,2,3)
GritColor = SCPI.DISPlay.COLor(1).GRATicule(1)
```

Related objects

SCPI.DISPlay.COLor(Dnum).RESet

Equivalent key

System > **Misc Setup** > **Display Setup** > **Color Setup** > **Normal|Invert** > **Graticule Main|Graticule Sub**

Equivalent SCPI command

Syntax

:DISPlay:COLor{[1]|2}:GRATicule{[1]|2} <numeric 1>,<numeric 2>,<numeric 3>

:DISPlay:COLor{[1]|2}:GRATicule{[1]|2}?

Query response

{numeric 1},{numeric 2},{numeric 3}<newline><^END>

Example of use

```
10 OUTPUT 717;":DISP:COL1:GRAT1 1,2,3"
20 OUTPUT 717;":DISP:COL1:GRAT1?"
30 ENTER 717;A,B,C
```

SCPI.DISPlay.COLor(*Dnum*).LIMit(*Lnum*)

Object type

Property (**Read-Write**)

Syntax

SCPI.DISPlay.COLor(*Dnum*).LIMit(*Lnum*) = *Data*

Data = SCPI.DISPlay.COLor(*Dnum*).LIMit(*Lnum*)

Description

This command sets/gets:

- Fail display color used for the limit test result, Bandwidth test result and Ripple test result (*Lnum*:1)
- Color of the limit line (*Lnum*:2)
  for normal display (*Dnum*:1) and inverted display (*Dnum*:2).

Variable

Parameter	*Lnum*
Description	The number of item 1: The limit test result (Fail/Pass) 2: The limit line
Data type	Long integer type (Long)
Range	1 to 2
Preset value	1
Note	If the specified variable is out of the allowable setup range, an error occurs when executed.

Parameter	*Data*
Description	Indicates 3-element array data. • *Data(0) :* Sets amount of red. • *Data(1) :* Sets amount of green.

	• *Data(2) :* Sets amount of blue. The index of the array starts from 0.
Data type	Variant type (Variant)
Range	• *Data(0) :* 0 to 5 • *Data(1) :*0 to 5 • *Data(2) :*0 to 5
Resolution	1
Note	If the specified variable 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.

Examples

```
Dim LimColor As Variant
SCPI.DISPlay.COLor(1).LIMit(1) = Array(1,2,3)
LimColor = SCPI.DISPlay.COLor(1).LIMit(1)
```

Related objects

SCPI.DISPlay.COLor(Dnum).RESet

Equivalent key

System > **Misc Setup** > **Color Setup** > **Normal|Invert** > **Limit Fail|Limit Line**

Equivalent SCPI command

Syntax

:DISPlay:COLor{[1]|2}:LIMit{[1]|2} <numeric 1>,<numeric 2>,<numeric 3>

:DISPlay:COLor{[1]|2}:LIMit{[1]|2}?

Query response

{numeric 1},{numeric 2},{numeric 3}<newline><^END>

Example of use

```
10 OUTPUT 717;":DISP:COL1:LIM1 1,2,3"
20 OUTPUT 717;":DISP:COL1:LIM1?"
30 ENTER 717;A,B,C
```

SCPI.DISPlay.COLor(*Dnum*).RESet

Object type

Method (**Write-only**)

Syntax

SCPI.DISPlay.COLor(*Dnum*).RESet

Description

This command resets the display color settings for all the items to the factory preset state, for normal display (*Dnum*:1) and inverted display (*Dnum*:2).

Examples

SCPI.DISPlay.COLor(1).RESet

Related objects

SCPI.DISPlay.COLor(Dnum).BACK

SCPI.DISPlay.COLor(Dnum).GRATicule(Gnum)

SCPI.DISPlay.COLor(Dnum).LIMit(Lnum)

SCPI.DISPlay.COLor(Dnum).TRACe(Tr).DATA

SCPI.DISPlay.COLor(Dnum).TRACe(Tr).MEMory

Equivalent key

System > **Misc Setup** > **Color Setup** > **Normal|Invert** > **Reset Color** > **OK**

Equivalent SCPI command

Syntax

:DISPlay:COLor{[1]|2}:RESet

Example of use

```
10 OUTPUT 717;":DISP:COL1:RES"
```

SCPI.DISPlay.COLor(*Dnum*).TRACe(*Tr*).DATA

Object type

Property (**Read-Write**)

Syntax

SCPI.DISPlay.COLor(*Dnum*).TRACe(*Tr*).DATA = *Data*

Data = SCPI.DISPlay.COLor(*Dnum*).TRACe(*Tr*).DATA

Description

This command sets/gets the color of selected trace (*Tr*), for normal display (*Dnum*:1) and inverted display (*Dnum*:2).

Variable

Parameter	*Data*
Description	Indicates 3-element array data. • *Data(0) :*Sets amount of red. • *Data(1) :*Sets amount of green. • *Data(2) :*Sets amount of blue. The index of the array starts from 0.
Data type	Variant type (Variant)
Range	• *Data(0) :* 0 to 5 • *Data(1) :* 0 to 5 • *Data(2) :* 0 to 5
Resolution	1
Note	If the specified variable 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.

Examples

```
Dim TrColor As Variant
SCPI.DISPlay.COLor(1).TRACe(1).DATA = Array(1,2,3)
TrColor = SCPI.DISPlay.COLor(1).TRACe(1).DATA
```

Related objects

SCPI.DISPlay.COLor(Dnum).RESet

Equivalent key

System > **Misc Setup** > **Color Setup** > **Normal|Invert** > **Data Trace X** (X=1 to 4)

Equivalent SCPI command

Syntax

:DISPlay:COLor{[1]|2}:TRACe{[1]-4}:DATA <numeric 1>,<numeric 2>,<numeric 3>

:DISPlay:COLor{[1]|2}:TRACe{[1]-4}:DATA?

Query response

{numeric 1},{numeric 2},{numeric 3}<newline><^END>

Example of use

```
10 OUTPUT 717;":DISP:COL1:TRAC1:DATA 1,2,3"
20 OUTPUT 717;":DISP:COL1:TRAC1:DATA?"
30 ENTER 717;A,B,C
```

SCPI.DISPlay.COLor(*Dnum*).TRACe(*Tr*).MEMory

Object type

Property (**Read-Write**)

Syntax

SCPI.DISPlay.COLor(*Dnum*).TRACe(*Tr*).MEMory = *Data*

Data = SCPI.DISPlay.COLor(*Dnum*).TRACe(*Tr*).MEMory

Description

This command sets/gets the color of the memory trace for the selected (*Tr*), for normal display (*Dnum*:1) and inverted display (*Dnum*:2).

Variable

Parameter	*Data*
Description	Indicates 3-element array data. • *Data(0) :*Sets amount of red. • *Data(1) :*Sets amount of green. • *Data(2) :*Sets amount of blue. The index of the array starts from 0.
Data type	Variant type (Variant)
Range	• *Data(0) :*0 to 5 • *Data(1) :*0 to 5 • *Data(2) :*0 to 5
Resolution	1
Note	If the specified variable 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.

Examples

```
Dim TrColor As Variant
SCPI.DISPlay.COLor(1).TRACe(1).MEMory = Array(1,2,3)
TrColor = SCPI.DISPlay.COLor(1).TRACe(1).MEMory
```

Related objects

SCPI.DISPlay.COLor(Dnum).RESet

Equivalent key

System > **Misc Setup** > **Color Setup** > **Normal|Invert** > **Mem Trace X** (X=1 to 4)

Equivalent SCPI command

Syntax

:DISPlay:COLor{[1]|2}:TRACe{[1]-4}:MEMory <numeric 1>,<numeric 2>,<numeric 3>

:DISPlay:COLor{[1]|2}:TRACe{[1]-4}:MEMory?

Query response

{numeric 1},{numeric 2},{numeric 3}<newline><^END>

Example of use

```
10 OUTPUT 717;":DISP:COL1:TRAC1:MEM 1,2,3"
20 OUTPUT 717;":DISP:COL1:TRAC1:MEM?"
30 ENTER 717;A,B,C
```

SCPI.DISPlay.ECHO.CLEar

Object type

Method (**Write Only**)

Syntax

SCPI.DISPlay.ECHO.CLEar

Description

This command clears all character strings displayed in the echo window.

Examples

```
SCPI.DISPlay.ECHO.CLEar
```

Related objects

SCPI.DISPlay.ECHO.DATA

Equivalent key

Macro Setup > **Clear Echo**

Equivalent SCPI command

Syntax

:DISPlay:ECHO:CLEar

Example of use

```
10 OUTPUT 717;":DISP:ECHO:CLE"
```

SCPI.DISPlay.ECHO.DATA

Object type

Property (**Write Only**)

Syntax

SCPI.DISPlay.ECHO.DATA = *Cont*

Description

This command displays a character string in the echo window. This command is different from ECHO command as it displays a single character string.

Variable

Parameter	*Cont*
Description	String you want to display in the echo window.
Data type	Character string type (String)
Range	254 characters or less

Examples

```
SCPI.DISPlay.ECHO.DATA = "Test Result"
SCPI.DISPlay.TABLe.TYPE = "echo"
SCPI.DISPlay.TABLe.STATe = True
```

Related objects

ECHO

SCPI.DISPlay.TABLe.TYPE

SCPI.DISPlay.TABLe.STATe

SCPI.DISPlay.ECHO.CLEar

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

:DISPlay:ECHO[:DATA] <string>

Example of use

```
10 OUTPUT 717;":DISP:ECHO ""TEST RESULT"""
```

SCPI.DISPlay.ENABle

Object type

Property (**Read-Write**)

Syntax

SCPI.DISPlay.ENABle = *Status*

Status = SCPI.DISPlay.ENABle

Description

This command turns ON/OFF the display update on the E5061B measurement screen.

Variable

Parameter	*Status*
Description	ON/OFF of the display update of the E5061B measurement screen
Data type	Boolean type (Boolean)
Range	Select from the following: • True or ON: Turns ON the display update. • False or OFF : Turns OFF the display update.
Preset value	True or ON

Examples

```
Dim DispUpdt As Boolean
SCPI.DISPlay.ENABle = False
DispUpdt = SCPI.DISPlay.ENABle
```

Equivalent key

Display > **Update**

Equivalent SCPI command

Syntax

:DISPlay:ENABle {ON|OFF|1|0}

:DISPlay:ENABle?

Query response

{1|0}<newline><^END>

Example of use

```
10 OUTPUT 717;":DISP:ENAB OFF"
20 OUTPUT 717;":DISP:ENAB?"
30 ENTER 717;A
```

SCPI.DISPlay.FSIGn

Object type

Property (**Read-Write**)

Syntax

SCPI.DISPlay.FSIGn = *Status*

Status = SCPI.DISPlay.FSIGn

Description

This command turns ON/OFF the "Fail" display on the LCD screen when the limit test, bandwidth test and ripple test fails.

Variable

Parameter	*Status*
Description	ON/OFF of the "Fail" display when the limit test fails
Data type	Boolean type (Boolean)
Range	Select from the following: • True or ON: Turns ON the "Fail" display. • False or OFF : Turns OFF the "Fail" display.
Preset value	True or ON

ON/OFF of the Fail display cannot be set at each test. When the Fail display of either of test is turned ON, the Fail display of other tests turns ON, too.

Examples

```
Dim DispFail As Boolean
SCPI.DISPlay.FSIGn = False
DispFail = SCPI.DISPlay.FSIGn
```

Related objects

SCPI.CALCulate(Ch).SELected.LIMit.STATe

SCPI.CALCulate(Ch).SELected.RLIMit.STATe

SCPI.CALCulate(Ch).SELected.BLIMit.STATe

Equivalent key

Analysis > **Limit Test > Fail Sign**

Analysis > **Ripple Limit > Fail Sign**

Analysis > **Bandwidth limit > Fail Sign**

Equivalent SCPI command

Syntax

:DISPlay:FSIGn {ON|OFF|1|0}

:DISPlay:FSIGn?

Query response

{1|0}<newline><^END>

Example of use

```
10 OUTPUT 717;":DISP:FSIG OFF"
20 OUTPUT 717;":DISP:FSIG?"
30 ENTER 717;A
```

SCPI.DISPlay.IMAGe

Object type

Property (**Read-Write**)

Syntax

SCPI.DISPlay.IMAGe = *Param*

Param = SCPI.DISPlay.IMAGe

Description

This command sets/gets the display type of the LCD display.

Variable

Parameter	*Param*
Description	Display type of the LCD display
Data type	Character string type (String)
Range	Select from the following: • "NORMal" : Specifies the normal display (background color: black). • "INVert": Specifies the display in which the color of the normal display is inversed (background color: white).
Preset value	"NORMal"

Examples

```
Dim DispImg As String
SCPI.DISPlay.IMAGe = "inv"
DispImg = SCPI.DISPlay.IMAGe
```

Equivalent key

Display > **Invert Color**

Equivalent SCPI command

Syntax

:DISPlay:IMAGe {NORMal|INVert}

:DISPlay:IMAGe?

Query response

{NORM|INV}<newline><^END>

Example of use

```
10 OUTPUT 717;":DISP:IMAG INV"
20 OUTPUT 717;":DISP:IMAG?"
30 ENTER 717;A$
```

SCPI.DISPlay.MAXimize

Object type

Property (**Read-Write**)

Syntax

SCPI.DISPlay.MAXimize = *Status*

Status = SCPI.DISPlay.MAXimize

Description

This command turns ON/OFF the window maximization of the active channel.

If the maximization is set to ON, only the window of the active channel is maximized on the LCD display and the windows of the other channels are not displayed.

Variable

Parameter	*Status*
Description	ON/OFF of the window maximization
Data type	Boolean type (Boolean)
Range	Select from the following: • True or ON: Turns ON the window maximization. • False or OFF: Turns OFF the window maximization.
Preset value	False or OFF

Examples

```
Dim ChMax As Boolean
SCPI.DISPlay.SPLit = "d1_2"
SCPI.DISPlay.WINDow(2).ACTivate
SCPI.DISPlay.MAXimize = True
ChMax = SCPI.DISPlay.MAXimize
```

Related objects

SCPI.DISPlay.WINDow(Ch).ACTivate

Equivalent key

Channel Max

Equivalent SCPI command

Syntax

:DISPlay:MAXimize {ON|OFF|1|0}

:DISPlay:MAXimize?

Query response

{1|0}<newline><^END>

Example of use

```
10 OUTPUT 717;":DISP:MAX ON"
20 OUTPUT 717;":DISP:MAX?"
30 ENTER 717;A
```

SCPI.DISPlay.SKEY.STATe

Object type

Property (**Read-Write**)

Syntax

SCPI.DISPlay.SKEY.STATe = *Status*

Status = SCPI.DISPlay.SKEY.STATe

Description

This command turns ON/OFF the display of the softkey menu bar.

Variable

Parameter	*Status*
Description	ON/OFF of the softkey menu bar display
Data type	Boolean type (Boolean)
Range	Select from the following: • True or ON: Turns ON the softkey menu bar display. • False or OFF: Turns OFF the softkey menu bar display.
Preset value	True or ON

Examples

```
Dim DispSkey As Boolean
SCPI.DISPlay.SKEY.STATe = False
DispSkey = SCPI.DISPlay.SKEY.STATe
```

Equivalent key

Entry Off

Equivalent SCPI command

Syntax

:DISPlay:SKEY[:STATe] {ON|OFF|1|0}

:DISPlay:SKEY[:STATe]?

Query response

{1|0}<newline><^END>

Example of use

```
10 OUTPUT 717;":DISP:SKEY ON"
20 OUTPUT 717;":DISP:SKEY?"
30 ENTER 717;A
```

SCPI.DISPlay.SPLit

Object type

Property (**Read-Write**)

Syntax

SCPI.DISPlay.SPLit = *Param*

Param = SCPI.DISPlay.SPLit

Description

Sets the layout of the channel windows on the LCD display.

Variable

Parameter	*Param*
Description	Layout of channel windows
Data type	Character string type (String)
Range	Select from the following: • "D1" • "D12" • "D1_2" • "D112" • "D1_1_2" • "D123" • "D1_2_3" • "D12_33" • "D11_23" • "D13_23" • "D12_13" • "D1234" • "D1_2_3_4" • "D12_34" Refer to link " Window Graph layouts and command parameters" for the confirmation of the window layout.

Preset value	"D1"

Examples

```
Dim ChanAloc As String
SCPI.DISPlay.SPLit = "d12_34"
ChanAloc = SCPI.DISPlay.SPLit
```

Related objects

SCPI.DISPlay.WINDow(Ch).SPLit

Equivalent key

Display > **Allocate Channels**

Equivalent SCPI command

Syntax

:DISPlay:SPLit

{D1|D12|D1_2|D112|D1_1_2|D123|D1_2_3|D12_33|D11_23|D13_23|D12_13| D1234|D1_2_3_4|D12_34}

:DISPlay:SPLit?

Query response

{D1|D12|D1_2|D112|D1_1_2|D123|D1_2_3|D12_33|D11_23|D13_23|D12_13| D1234|D1_2_3_4|D12_34}<newline><^END>

Example of use

```
10 OUTPUT 717;":DISP:SPL D1_2"
20 OUTPUT 717;":DISP:SPL?"
30 ENTER 717;A$
```

SCPI.DISPlay.TABLe.POSition.RECTangle

Object type

Property (**Read Only**)

Syntax

Param = SCPI.DISPlay.TABLe.POSition.RECTangle

Description

This command reads the display coordinates position of Table area (the top left of the display is [0, 0]). If SCPI.DISPlay.TABLe.STATe is OFF, 0, 0, 0, 0 are returned. .

Variable

Parameter	*Param*
Description	Indicates the coordinates position of Table Area. • *Param(0)* : coordinates X position of top left of Table Area. • *Param(1)* : coordinates Y position of top left of Table Area. • *Param(2)* : coordinates X position of bottom right of Table Area. • *Param(3)* : coordinates Y position of bottom right of Table Area. The index of the array starts from 0.
Data type	Variant type (Variant)
Resolution	1

Examples

```
Dim TablePos() As Variant
SCPI.DISPlay.TABLe.STATe = True
TablePos = SCPI.DISPlay.TABLe.POSition.RECTangle
```

Related objects

SCPI.DISPlay.TABLe.STATe

Equivalent key

None

Equivalent SCPI command

Syntax

:DISPlay:TABLe:POSition[:RECTangle]?

Query response

{numeric1},{numeric2},{numeric3},{numeric4},<newline><^END>

Example of use

```
10 OUTPUT 717;":DISP:TABL:STAT ON"
20 OUTPUT 717;":DISP:TABL:POS?"
20 ENTER 717;A, B, C, D
```

SCPI.DISPlay.TABLe.STATe

Object type

Property (**Read-Write**)

Syntax

SCPI.DISPlay.TABLe.STATe = *Status*

Status = SCPI.DISPlay.TABLe.STATe

Description

This command turns ON/OFF the display of the window that appears in the lower part of the LCD display (specified by SCPI.DISPlay.TABLe.TYPE object).

Variable

Parameter	*Status*
Description	ON/OFF of the display of the window that appears in the lower part of the LCD display
Data type	Boolean type (Boolean)
Range	Select from the following: • True or ON: Turns ON the display. • False or OFF: Turns OFF the display.
Preset value	False or OFF

Examples

```
Dim DispTbl As Boolean
SCPI.DISPlay.TABLe.TYPE = "echo"
SCPI.DISPlay.TABLe.STATe = True
DispTbl = SCPI.DISPlay.TABLe.STATe
```

Related objects

SCPI.DISPlay.TABLe.TYPE

Equivalent key

Sweep Setup > **Edit Segment Table**

Marker Fctn > **Marker Table**

Analysis > **Limit Test** > **Edit Limit Line**

Analysis > Ripple Limit > Edit Ripple Line

Macro Setup > Echo Window

Cal > Power Calibration > Loss Compen

Cal > Power Calibration > Sensor A Settings | Sensor B Settings

When performing the operation from the front panel, you select the type of the window that appears in the lower part of the LCD display and turn ON/OFF the display at the same time.

Equivalent SCPI command

Syntax

:DISPlay:TABLe[:STATe] {ON|OFF|1|0}

:DISPlay:TABLe[:STATe]?

Query response

{1|0}<newline><^END>

Example of use

```
10 OUTPUT 717;":DISP:TABL ON"
20 OUTPUT 717;":DISP:TABL?"
30 ENTER 717;A
```

SCPI.DISPlay.TABLe.TYPE

Object type

Property (**Read-Write**)

Syntax

SCPI.DISPlay.TABLe.TYPE = *Param*

Param = SCPI.DISPlay.TABLe.TYPE

Description

This command selects the type of the window that appears in the lower part of the LCD display.

Variable

Parameter	*Param*
Description	Window type
Data type	Character string type (String)
Range	Select from the following: • "MARKer": Specifies the marker table window. • "LIMit": Specifies the limit test table window. • "SEGMent": Specifies the segment table window. • "ECHO": Specifies the echo window. • "RLIMit": Specifies the ripple test table window.
Preset value	"MARKer"

Examples

```
Dim TblType As String
SCPI.DISPlay.TABLe.TYPE = "echo"
SCPI.DISPlay.TABLe.STATe = True
TblType = SCPI.DISPlay.TABLe.TYPE
```

Related objects

SCPI.DISPlay.TABLe.STATe

Equivalent key

Sweep Setup > **Edit Segment Table**

Marker Fctn > **Marker Table**

Analysis > **Limit Test** > **Edit Limit Line**

Analysis > **Ripple Limit** > **Edit Ripple Line**

Macro Setup > **Echo Window**

When performing the operation from the front panel, you select the type of the window that appears in the lower part of the LCD display and turn ON/OFF the display at the same time.

Equivalent SCPI command

Syntax

:DISPlay:TABLe:TYPE {MARKer|LIMit|SEGMent|ECHO|RLIMit}

:DISPlay:TABLe:TYPE?

Query response

{MARK|LIM|SEGM|ECHO|RLIM}<newline><^END>

Example of use

```
10 OUTPUT 717;":DISP:TABL:TYPE SEGM"
20 OUTPUT 717;":DISP:TABL:TYPE?"
30 ENTER 717;A$
```

SCPI.DISPlay.UPDate.IMMediate

Object type

Method (**Write-only**)

Syntax

SCPI.DISPlay.UPDate.IMMediate

Description

This command executes the display update once when the display update of the LCD screen is set to OFF (specifying False with the SCPI.DISPlay.ENABle object).

Examples

```
SCPI.DISPlay.ENABle = False
SCPI.DISPlay.UPDate.IMMediate
```

Related objects

SCPI.DISPlay.ENABle

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

:DISPlay:UPDate[:IMMediate]

Example of use

```
10 OUTPUT 717;":DISP:UPD"
```

SCPI.DISPlay.WINDow(*Ch*).ACTivate

Object type

Method (**Write-only**)

Syntax

SCPI.DISPlay.WINDow(*Ch*).ACTivate

Description

This command specifies selected channel (*Ch*) as the active channel.

NOTE You can set only a channel displayed to the active channel. If this object is used to set a channel not displayed to the active channel, an error occurs when executed and the object is ignored.

Examples

```
SCPI.DISPlay.SPLit = "d1_2"
SCPI.DISPlay.WINDow(2).ACTivate
```

Related objects

SCPI.CALCulate(Ch).PARameter(Tr).SELect

Equivalent key

Channel Prev / **Channel Next**

Equivalent SCPI command

Syntax

:DISPlay:WINDow{[1]-4}:ACTivate

Example of use

```
10 OUTPUT 717;":DISP:WIND1:ACT"
```

SCPI.DISPlay.WINDow(*Ch*).ANNotation.MARKer.ALIGn.STATe

Object type

Property (**Read-Write**)

Syntax

SCPI.DISPlay.WINDow(*Ch*).ANNotation.MARKer.ALIGn.STATe = *Status*

Status = SCPI.DISPlay.WINDow(*Ch*).ANNotation.MARKer.ALIGn.STATe

Description

This command turn ON/OFF the mode that align the marker display position of each trace based on trace 1, for the selected channel (*Ch*).

Variable

Parameter	*Status*
Description	ON/OFF the mode that align the marker display position of each trace based on trace 1
Data type	Boolean type (Boolean)
Range	Select from the following: • True or ON:Turns ON the mode that allign marker display position based on trace 1. • False or OFF:Turns OFF the alignment.
Preset value	True or ON

Examples

```
Dim AnnMarkAlig As Boolean
SCPI.DISPlay.WINDow(1).ANNotation.MARKer.ALIGn.STATe = False
AnnMarkAlig = SCPI.DISPlay.WINDow(1).ANNotation.MARKer.ALIGn.STATe
```

Related objects

SCPI.DISPlay.WINDow(Ch).ANNotation.MARKer.SINGle.STATe

SCPI.DISPlay.WINDow(Ch).TRACe(Tr).ANNotation.MARKer.POSition.X

SCPI.DISPlay.WINDow(Ch).TRACe(Tr).ANNotation.MARKer.POSition.Y

Equivalent key

Marker Fctn > **Annotation Options** > **Align**

Equivalent SCPI command

Syntax

:DISPlay:WINDow{[1]-4}:ANNotation:MARKer:ALIGn[:STATe] {ON|OFF|1|0}

:DISPlay:WINDow{[1]-4}:ANNotation:MARKer:ALIGn[:STATe]?

Query response

{1|0}<newline><^END>

Example of use

```
10 OUTPUT 717;":DISP:WIND1:ANN:MARK:ALIG OFF"
20 OUTPUT 717;":DISP:WIND1:ANN:MARK:ALIG?"
30 ENTER 717;A
```

SCPI.DISPlay.WINDow(*Ch*).ANNotation.MARKer.SINGle.STATe

Object type

Property (**Read-Write**)

Syntax

SCPI.DISPlay.WINDow(*Ch*).ANNotation.MARKer.SINGle.STATe = *Status*

Status = SCPI.DISPlay.WINDow(*Ch*).ANNotation.MARKer.SINGle.STATe

Description

This command turns ON/OFF the display of the marker value of only active traces, for the selected channel (*Ch*).

NOTE If the function is turned OFF, marker values of all traces (markers) are displayed.

Variable

Parameter	*Status*
Description	ON/OFF the display of the marker value of only active
Data type	Boolean type (Boolean)
Range	Select from the following: • True or ON: Displays the marker values of only the active traces.(ON) • False or OFF: Displays the marker values of all the traces. (OFF)
Preset value	True or ON

Examples

```
Dim AnnMarkAlig As Boolean
SCPI.DISPlay.WINDow(1).ANNotation.MARKer.SINGle.STATe = False
AnnMarkAlig = SCPI.DISPlay.WINDow(1).ANNotation.MARKer.SINGle.STATe
```

Related objects

SCPI.DISPlay.WINDow(Ch).ANNotation.MARKer.ALIGn.STATe

SCPI.DISPlay.WINDow(Ch).TRACe(Tr).ANNotation.MARKer.POSition.X

SCPI.DISPlay.WINDow(Ch).TRACe(Tr).ANNotation.MARKer.POSition.Y

Equivalent key

Marker Fctn > **Annotation Options** > **Active Only**

Equivalent SCPI command

Syntax

:DISPlay:WINDow{[1]-4}:ANNotation:MARKer:SINGle[:STATe] {ON|OFF|1|0}

:DISPlay:WINDow{[1]-4}:ANNotation:MARKer:SINGle[:STATe]?

Query response

{1|0}<newline><^END>

Example of use

```
10 OUTPUT 717;":DISP:WIND1:ANN:MARK:SING OFF"
20 OUTPUT 717;":DISP:WIND1:ANN:MARK:SING?"
30 ENTER 717;A
```

SCPI.DISPlay.WINDow(*Ch*).LABel

Object type

Property (**Read-Write**)

Syntax

SCPI.DISPlay.WINDow(*Ch*).LABel = *Status*

Status = SCPI.DISPlay.WINDow(*Ch*).LABel

Description

This command turns ON/OFF the graticule label display of the graph of the selected channel (*Ch*).

Variable

Parameter	*Status*
Description	ON/OFF of the graticule label display of the graph
Data type	Boolean type (Boolean)
Range	Select from the following: • True or ON: Turns ON the graticule label display. • False or OFF: Turns OFF the graticule label display.
Preset value	True or ON

Examples

```
Dim DispGrat As Boolean
SCPI.DISPlay.WINDow(1).LABel = False
DispGrat = SCPI.DISPlay.WINDow(1).LABel
```

Equivalent key

Display > **Graticule Label**

Equivalent SCPI command

Syntax

:DISPlay:WINDow{[1]-4}:LABel {ON|OFF|1|0}

:DISPlay:WINDow{[1]-4}:LABel?

Query response

{1|0}<newline><^END>

Example of use

```
10 OUTPUT 717;":DISP:WIND1:LAB ON"
20 OUTPUT 717;":DISP:WIND1:LAB?"
30 ENTER 717;A
```

SCPI.DISPlay.WINDow(*Ch*).MAXimize

Object type

Property (**Read-Write**)

Syntax

SCPI.DISPlay.WINDow(*Ch*).MAXimize = *Status*

Status = SCPI.DISPlay.WINDow(*Ch*).MAXimize

Description

This command turns ON/OFF the maximization of the active trace of selected channel (*Ch*).

NOTE If you turned ON the maximization, only the maximized active trace is displayed in the window and the other traces are not displayed.

Variable

Parameter	*Status*
Description	ON/OFF of the maximization of the active trace
Data type	Boolean type (Boolean)
Range	Select from the following: • True or ON: Turns ON the maxim display. • False or OFF: Turns OFF the maxim display.
Preset value	False or OFF

Examples

```
Dim TracMax As Boolean
SCPI.CALCulate(1).PARameter(2).SELect
SCPI.DISPlay.WINDow(1).MAXimize = True
TracMax = SCPI.DISPlay.WINDow(1).MAXimize
```

Related objects

SCPI.CALCulate(Ch).PARameter(Tr).SELect

SCPI.DISPlay.MAXimize

Equivalent key

Trace Max

Equivalent SCPI command

Syntax

:DISPlay:WINDow{[1]-4}:MAXimize {ON|OFF|1|0}

:DISPlay:WINDow{[1]-4}:MAXimize?

Query response

{1|0}<newline><^END>

Example of use

```
10 OUTPUT 717;":DISP:WIND1:MAX ON"
20 OUTPUT 717;":DISP:WIND1:MAX?"
30 ENTER 717;A
```

SCPI.DISPlay.WINDow(*Ch*).SPLit

Object type

Property (**Read-Write**)

Syntax

SCPI.DISPlay.WINDow(*Ch*).SPLit = *Param*

Param = SCPI.DISPlay.WINDow(*Ch*).SPLit

Description

This command sets/gets the graph layout of the selected channel (*Ch*).

Variable

Parameter	*Param*
Description	Graph layout
Data type	Character string type (String)
Range	Select from the following: • "D1" • "D12" • "D1_2" • "D112" • "D1_1_2" • "D123" • "D1_2_3" • "D12_33" • "D11_23" • "D13_23" • "D12_13" • "D1234" • "D1_2_3_4" • "D12_34" Refer to link " Window Graph layouts and command parameters" for the confirmation of the channel window layout.

Preset value	"D1"

Examples

```
Dim TracAloc As String
SCPI.DISPlay.WINDow(1).SPLit = "d1_2"
TracAloc = SCPI.DISPlay.WINDow(1).SPLit
```

Related objects

SCPI.DISPlay.SPLit

Equivalent key

Display > **Allocate Traces**

Equivalent SCPI command

Syntax

:DISPlay:WINDow{[1]-4}:SPLit {D1|D12|D1_2|D112|D1_1_2|
D123|D1_2_3|D12_33|D11_23|D13_23|D12_13|D1234|D1_2_3_4|D12_3
4|
D1X1}

:DISPlay:WINDow{[1]-4}:SPLit?

Query response

{D1|D12|D1_2|D112|D1_1_2|D123|D1_2_3|D12_33|D11_23|D13_23|
D12_13|D1234|D1_2_3_4|D12_34}<newline><^END>

Example of use

```
10 OUTPUT 717;":DISP:WIND:SPL D1_2"
20 OUTPUT 717;":DISP:WIND:SPL?"
30 ENTER 717;A$
```

SCPI.DISPlay.WINDow(*Ch*).TITLe.DATA

Object type

Property (**Read-Write**)

Syntax

SCPI.DISPlay.WINDow(*Ch*).TITLe.DATA = *Lbl*

Lbl = SCPI.DISPlay.WINDow(*Ch*).TITLe.DATA

Description

This command sets/gets the title label displayed in the title area of the selected channel (*Ch*).

Variable

Parameter	*Lbl*
Description	Title label
Data type	Character string type (String)
Range	254 characters or less
Preset value	""

Examples

```
Dim TtlLbl As String
SCPI.DISPlay.WINDow(1).TITLe.DATA = "Filter"
SCPI.DISPlay.WINDow(1).TITLe.STATe = True
TtlLbl = SCPI.DISPlay.WINDow(1).TITLe.DATA
```

Related objects

SCPI.DISPlay.WINDow(Ch).TITLe.STATe

Equivalent key

Display > **Edit Title Label**

Equivalent SCPI command

Syntax

:DISPlay:WINDow{[1]-4}:TITLe:DATA <string>

:DISPlay:WINDow{[1]-4}:TITLe:DATA?

Query response

{string}<newline><^END>

Example of use

```
10 OUTPUT 717;":DISP:WIND1:TITL:DATA ""Title"""
20 OUTPUT 717;":DISP:WIND1:TITL?"
30 ENTER 717;A$
```

SCPI.DISPlay.WINDow(*Ch*).TITLe.STATe

Object type

Property (**Read-Write**)

Syntax

SCPI.DISPlay.WINDow(*Ch*).TITLe.STATe = *Status*

Status = SCPI.DISPlay.WINDow(*Ch*).TITLe.STATe

Description

This command turns ON/OFF the title label display in the title area of channels 1 to 16 (*Ch*).

Variable

Parameter	*Status*
Description	ON/OFF of the title label display
Data type	Boolean type (Boolean)
Range	Select from the following: • True or ON: Turns ON the title label display. • False or OFF: Turns ON the title label display.
Preset value	False or OFF

Examples

```
Dim DispTtl As Boolean
SCPI.DISPlay.WINDow(1).TITLe.DATA = "Filter"
SCPI.DISPlay.WINDow(1).TITLe.STATe = True
DispTtl = SCPI.DISPlay.WINDow(1).TITLe.STATe
```

Related objects

SCPI.DISPlay.WINDow(Ch).TITLe.DATA

Equivalent key

Display > **Title Label**

Equivalent SCPI command

Syntax

:DISPlay:WINDow{[1]-4}:TITLe[:STATe] {ON|OFF|1|0}

:DISPlay:WINDow{[1]-4}:TITLe[:STATe]?

Query response

{1|0}<newline><^END>

Example of use

```
10 OUTPUT 717;":DISP:WIND1:TITL ON"
20 OUTPUT 717;":DISP:WIND1:TITL?"
30 ENTER 717;A
```

SCPI.DISPlay.WINDow(*Ch*).TRACe(*Tr*).ANNotation.MARKer.POSition.X

Object type

Property (**Read-Write**)

Syntax

SCPI.DISPlay.WINDow(*Ch*).TRACe(*Tr*).ANNotation.MARKer.POSition.X = *Value*

Value = SCPI.DISPlay.WINDow(*Ch*).TRACe(*Tr*).ANNotation.MARKer.POSition.X

Description

This command sets/gets the display position of the marker value on the X-axis by a percentage of a width of the display span, for the selected trace (*Tr*) of the selected channel (*Ch*).

Variable

Parameter	*Value*
Description	Display position of the marker value on the X-axis.
Data type	Long integer type (Long)
Range	-15 to 100
Preset value	1
Unit	% (percent)
Note	If the specified variable 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.

Examples

```
Dim AnnMPosX As Long
SCPI.DISPlay.WINDow(1).TRACe(1).ANNotation.MARKer.POSition.X = 15
AnnMPosX = SCPI.DISPlay.WINDow(1).TRACe(1).ANNotation.MARKer.POSition.X
```

Related objects

SCPI.DISPlay.WINDow(Ch).ANNotation.MARKer.ALIGn.STATe

SCPI.DISPlay.WINDow(Ch).ANNotation.MARKer.SINGle.STATe

SCPI.DISPlay.WINDow(Ch).TRACe(Tr).ANNotation.MARKer.POSition.Y

Equivalent key

Marker Fctn > **Annotation Options** > **Marker Info X Pos**

Equivalent SCPI command

Syntax

:DISPlay:WINDow{[1]-4}:TRACe{[1]-4}:ANNotation:MARKer:POSition:X <numeric>

:DISPlay:WINDow{[1]-4}:TRACe{[1]-4}:ANNotation:MARKer:POSition:X?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":DISP:WIND1:TRAC1:ANN:MARK:POS:X 33"
20 OUTPUT 717;":DISP:WIND1:TRAC1:ANN:MARK:POS:X?"
30 ENTER 717;A
```

SCPI.DISPlay.WINDow(*Ch*).TRACe(*Tr*).ANNotation.MARKer.POSition.Y

Object type

Property (**Read-Write**)

Syntax

SCPI.DISPlay.WINDow(*Ch*).TRACe(*Tr*).ANNotation.MARKer.POSition.Y = *Value*

Value = SCPI.DISPlay.WINDow(*Ch*).TRACe(*Tr*).ANNotation.MARKer.POSition.X

Description

This command sets/gets the display position of the marker value on the X-axis by a percentage of a width of the display span, for the selected trace (*Tr*) of the selected channel (*Ch*), and the marker value on Y axis by a percentage of a height of the display span.

Variable

Parameter	*Value*
Description	Display position of the marker value on the Y-axis.
Data type	Long integer type (Long)
Range	-15 to 100
Preset value	1
Unit	% (percent)
Note	If the specified variable 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.

Examples

```
Dim AnnMPosY As Long
SCPI.DISPlay.WINDow(1).TRACe(1).ANNotation.MARKer.POSition.Y = 23
AnnMPosY = SCPI.DISPlay.WINDow(1).TRACe(1).ANNotation.MARKer.POSition.Y
```

Related objects

SCPI.DISPlay.WINDow(Ch).ANNotation.MARKer.ALIGn.STATe

SCPI.DISPlay.WINDow(Ch).ANNotation.MARKer.SINGle.STATe

SCPI.DISPlay.WINDow(Ch).TRACe(Tr).ANNotation.MARKer.POSition.X

Equivalent key

Marker Fctn > **Annotation Options** > **Marker Info Y Pos**

Equivalent SCPI command

Syntax

:DISPlay:WINDow{[1]-4}:TRACe{[1]-4}:ANNotation:MARKer:POSition:Y <numeric>

:DISPlay:WINDow{[1]-4}:TRACe{[1]-4}:ANNotation:MARKer:POSition:Y?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":DISP:WIND1:TRAC1:ANN:MARK:POS:Y 33"
20 OUTPUT 717;":DISP:WIND1:TRAC1:ANN:MARK:POS:Y?"
30 ENTER 717;A
```

SCPI.DISPlay.WINDow(*Ch*).TRACe(*Tr*).ANNotation.YAXis.MODE

Object Type

Property (**Read-Write**)

Syntax

SCPI.DISPlay.WINDow(*Ch*).TRACe(*Tr*).ANNotation.YAXis.MODE = *Param*

Param = SCPI.DISPlay.WINDow(*Ch*).TRACe(*Tr*).ANNotation.YAXis.MODE

Description

This command sets/gets the color mode for the Y-axis labels, for the selected trace (*Tr*) of the selected channel (*Ch*).

Variable

Parameter	*Param*
Description	The color mode for the Y-axis labels
Data Type	Character string type (String)
Range	Select from the following: • "AUTO" : Specifies the same color as the Graticule Main color (default color is R:3 G:3 B:3. • "RELative": Specifies the same color as the Data Trace color (The default color is different at each trace).
Preset Value	"AUTO"

Examples

```
Dim Ylbl_Color as String
SCPI.DISPlay.WINDow(1).TRACe(4).ANNotation.YAXis.MODE = "REL"
Ylbl_Color = SCPI.DISPlay.WINDow(1).TRACe(4).ANNotation.YAXis.MODE
```

Related Objects

Equivalent Key

No equivalent key is available on the front panel.

Equivalent SCPI Command

Syntax

:DISPlay:WINDow{[1]-4}:TRACe{[1]-4}:ANNotation:YAXis:MODE {AUTO|RELative}

:DISPlay:WINDow{[1]-4}:TRACe{[1]-4}:ANNotation:YAXis:MODE?

Query Response

{AUTO|REL} <newline><^END>

Example of use

```
10 OUTPUT 717;":DISP:WIND2:TRAC1:ANN:YAX:MODE AUTO"
20 OUTPUT 717;":DISP:WIND2:TRAC1:ANN:YAX:MODE?"
30 ENTER 717;A$
```

SCPI.DISPlay.WINDow(*Ch*).TRACe(*Tr*).MEMory. STATe

Object type

Property (**Read-Write**)

Syntax

SCPI.DISPlay.WINDow(*Ch*).TRACe(*Tr*).MEMory.STATe = *Status*

Status = SCPI.DISPlay.WINDow(*Ch*).TRACe(*Tr*).MEMory.STATe

Description

This command turns ON/OFF the memory trace display, for the selected trace (*Tr*) of the selected channel (*Ch*).

Variable

Parameter	*Status*
Description	ON/OFF of the memory trace display
Data type	Boolean type (Boolean)
Range	Select from the following: • True or ON: Turns ON the memory trace display. • False or OFF: Turns OFF the memory trace display.
Preset value	False or OFF

Examples

```
Dim DispMem As Boolean
SCPI.DISPlay.WINDow(1).TRACe(2).MEMory.STATe = True
DispMem = SCPI.DISPlay.WINDow(1).TRACe(2).MEMory.STATe
```

Related objects

SCPI.CALCulate(Ch).SELected.MATH.MEMorize

SCPI.DISPlay.WINDow(Ch).TRACe(Tr).STATe

Equivalent key

Display > **Display** > **Mem** (when the data trace display is OFF)

Display > **Display** > **Data & Mem** (when the data trace display is ON)

Equivalent SCPI command

Syntax

:DISPlay:WINDow{[1]-4}:TRACe{[1]-4}:MEMory[:STATe] {ON|OFF|1|0}

:DISPlay:WINDow{[1]-4}:TRACe{[1]-4}:MEMory[:STATe]?

Query response

{1|0}<newline><^END>

Example of use

```
10 OUTPUT 717;":DISP:WIND1:TRAC1:MEM ON"
20 OUTPUT 717;":DISP:WIND1:TRAC1:MEM?"
30 ENTER 717;A
```

SCPI.DISPlay.WINDow(*Ch*).TRACe(*Tr*).STATe

Object type

Property (**Read-Write**)

Syntax

SCPI.DISPlay.WINDow(*Ch*).TRACe(*Tr*).STATe = *Status*

Status = SCPI.DISPlay.WINDow(*Ch*).TRACe(*Tr*).STATe

Description

This command turns ON/OFF the data trace display, for the selected trace (*Tr*) of the selected channel (*Ch*).

Variable

Parameter	*Status*
Description	ON/OFF of the data trace display
Data type	Boolean type (Boolean)
Range	Select from the following: • True or ON: Turns ON the data trace display. • False or OFF: Turns OFF the data trace display.
Preset value	True or ON

Examples

```
Dim DispTrac As Boolean
SCPI.DISPlay.WINDow(1).TRACe(2).STATe = False
DispTrac = SCPI.DISPlay.WINDow(1).TRACe(2).STATe
```

Related objects

SCPI.DISPlay.WINDow(Ch).TRACe(Tr).MEMory.STATe

Equivalent key

Display > **Display** > **Data** (when the memory trace display is OFF)

Display > **Display** > **Data & Mem** (when the memory trace display is ON)

Equivalent SCPI command

Syntax

:DISPlay:WINDow{[1]-4}:TRACe{[1]-4} :STATe {ON|OFF|1|0}

:DISPlay:WINDow{[1]-4}:TRACe{[1]-4} :STATe?

Query response

{1|0}<newline><^END>

Example of use

```
10 OUTPUT 717;":DISP:WIND1:TRAC1:STAT ON"
20 OUTPUT 717;":DISP:WIND1:TRAC1:STAT?"
30 ENTER 717;A
```

SCPI.DISPlay.WINDow(*Ch*).TRACe(Tr).Y.SCALe.ABSolute.UNIT

Object Type

Property (**Read-Write**)

Syntax

SCPI.DISPlay.WINDow(*Ch*).TRACe(Tr).Y.SCALe.ABSolute.UNIT = *Value*

Value = SCPI.DISPlay.WINDow(*Ch*).TRACe(Tr).Y.SCALe.ABSolute.UNIT

Description

This command defines or gets Y-Axis unit in absolute format (LIN-MAG, REAL or IMAG). This is applicable for impedance measurement as well as network measurement. By default, the unit is "Units".Use this command to define other units to display "Ω", "F", "H" etc. in impedance measurement.

Variable

Parameter	*Value*
Description	Y-Axis
Data Type	Character string type (String)
Range	254 chars
Preset Value	"Units"
Unit	-
Resolution	-

Examples

```
Dim OriginalUnit as String
Dim NewUnit as String
Dim TestUnit as String
OriginalUnit = SCPI.DISPlay.WINDow(1).TRACe(1).Y.SCALe.ABSolute.UNIT
'*Return value = Units

NewUnit ="abc"
SCPI.DISPlay.WINDow(1).TRACe(1).Y.SCALe.ABSolute.UNIT = NewUnit
TestUnit = SCPI.DISPlay.WINDow(1).TRACe(1).Y.SCALe.ABSolute.UNIT
'*Return value = abc
```

Related Objects

SCPI.CALCulate(Ch).SELected.ZPARameter.DEFine

E5061B

Equivalent Key

No equivalent key is available on the front panel.

Equivalent SCPI Command

Syntax

:DISPlay:WINDow{[1]-4}:TRACe{[1]-4}:Y[:SCALe]:ABSolute:UNIT {String}

:DISPlay:WINDow{[1]-4}:TRACe{[1]-4}:Y[:SCALe]:ABSolute:UNIT?

Query Response

{String}<newline><^END>

Example of use

```
10 OUTPUT 717;":DISP:WIND1:TRAC1:Y:ABS:UNIT Sample_Text"
20 OUTPUT 717;":DISP:WIND1:TRAC1:Y:ABS:UNIT?"
30 ENTER 717;A$
```

SCPI.DISPlay.WINDow(*Ch*).TRACe(*Tr*).Y.SCALe.AUTO

Object type

Method (**Write Only**)

Syntax

SCPI.DISPlay.WINDow(*Ch*).TRACe(*Tr*).Y.SCALe.AUTO

Description

This command executes the auto scale function, for the selected trace (*Tr*) of the selected channel (*Ch*). The Auto Scale function automatically adjusts the value of the reference division line and the scale per division to display the trace appropriately.

Examples

```
SCPI.DISPlay.WINDow(1).TRACe(2).Y.SCALe.AUTO
```

Related objects

SCPI.DISPlay.WINDow(Ch).TRACe(Tr).Y.SCALe.PDIVision

SCPI.DISPlay.WINDow(Ch).TRACe(Tr).Y.SCALe.RLEVel

Equivalent key

Scale > **Auto Scale**

Equivalent SCPI command

Syntax

:DISPlay:WINDow{[1]-4}:TRACe{[1]-4}:Y[:SCALe]:AUTO

Example of use

```
10 OUTPUT 717;":DISP:WIND1:TRAC1:Y:AUTO"
```

SCPI.DISPlay.WINDow(*Ch*).TRACe(Tr).Y.SCALe.BOTTom

Object Type

Property (**Read-Write**)

Syntax

SCPI.DISPlay.WINDow(*Ch*).TRACe(Tr).Y.SCALe.BOTTom = *Value*

Value = SCPI.DISPlay.WINDow(*Ch*).TRACe(Tr).Y.SCALe.BOTTom

Description

This command sets or gets the minimum scale value for the Log-Y Axis.

Variable

Parameter	*Value*
Description	Minimum scale value for Log-Y-Axis
Data Type	Double precision floating point type (Double)
Range	1a ~ 500P
Preset Value	1m
Unit	Depending on Measurement format
Resolution	-

Examples

See SCPI.DISPlay.WINDow(Ch).TRACe(Tr).Y.SPACing

Related Objects

SCPI.DISPlay.WINDow(Ch).TRACe(Tr).Y.SPACing

SCPI.DISPlay.WINDow(Ch).TRACe(Tr).Y.SCALe.TOP

Equivalent Key

Scale > **Log Y-Axis Top/Bottom** > **Bottom Value**

Using either the keyboard or ENTRY keys on the front panel, enter the bottom value.

Equivalent SCPI Command

Syntax

:DISPlay:WINDow{[1]-4}:TRACe{[1]-4}:Y[:SCALe]:BOTTom <numeric>

:DISPlay:WINDow{[1]-4}:TRACe{[1]-4}:Y[:SCALe]:BOTTom?

Query Response

<numeric><newline><^END>

Example of use

```
10 OUTPUT 717;":DISP:WIND1:TRAC1:Y:BOTT 0.000000000001"
20 OUTPUT 717;":DISP:WIND1:TRAC1:Y:BOTT?"
30 ENTER 717;A
```

SCPI.DISPlay.WINDow(*Ch*).TRACe(*Tr*).Y.SCALe.PDIVision

Object type

Property (**Read-Write**)

Syntax

SCPI.DISPlay.WINDow(*Ch*).TRACe(*Tr*).Y.SCALe.PDIVision = *Value*

Value = SCPI.DISPlay.WINDow(*Ch*).TRACe(*Tr*).Y.SCALe.PDIVision

Description

For the selected trace (*Tr*) of selected channel (*Ch*), when the data format is not the Smith chart format or the polar format, sets the scale per division. When the data format is the Smith chart format or the polar format, sets the full scale value (the value of the outermost circumference).

Variable

Parameter	*Value*
Description	Scale value
Data type	Double precision floating point type (Double)
Range	1E-18 to 1E8
Preset value	Varies depending the data format. • Log magnitude: 10 • Phase, Expanded phase or Positive phase: 90 • Group delay: 1E-8 • Smith chart or Polar or SWR: 1 • Linear magnitude: 0.1 • Real or Imaginary: 0.2
Unit	Varies depending on the data format. • Log magnitude: dB (decibel) • Phase, Expanded phase or Positive phase: ° (degree) • Group delay: s (second) • Others: No unit

Note	If the specified variable 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.

Examples

```
Dim Pdiv As Double
SCPI.CALCulate(1).PARameter(2).SELect
SCPI.CALCulate(1).SELected.FORMat = "gdel"
SCPI.DISPlay.WINDow(1).TRACe(2).Y.SCALe.PDIVision = 1E-9
Pdiv = SCPI.DISPlay.WINDow(1).TRACe(2).Y.SCALe.PDIVision
```

Related objects

SCPI.CALCulate(Ch).SELected.FORMat

SCPI.DISPlay.WINDow(Ch).Y.SCALe.DIVisions

SCPI.DISPlay.WINDow(Ch).TRACe(Tr).Y.SCALe.RLEVel

SCPI.DISPlay.WINDow(Ch).TRACe(Tr).Y.SCALe.RPOSition

Equivalent key

Scale > **Scale/Div**

Equivalent SCPI command

Syntax

:DISPlay:WINDow{[1]-4}:TRACe{[1]-4} :Y[:SCALe]:PDIVision <numeric>

:DISPlay:WINDow{[1]-4}:TRACe{[1]-4} :Y[:SCALe]:PDIVision?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":DISP:WIND1:TRAC1:Y:PDIV 2.5"
20 OUTPUT 717;":DISP:WIND1:TRAC1:Y:PDIV?"
30 ENTER 717;A
```

SCPI.DISPlay.WINDow(*Ch*).TRACe(*Tr*).Y.SCALe.RLEVel

Object type

Property (**Read-Write**)

Syntax

SCPI.DISPlay.WINDow(*Ch*).TRACe(*Tr*).Y.SCALe.RLEVel = *Value*

Value = SCPI.DISPlay.WINDow(*Ch*).TRACe(*Tr*).Y.SCALe.RLEVel

Description

This command sets/gets the value of the reference division line, for the selected trace (*Tr*) of the selected channel (*Ch*).

Variable

Parameter	*Value*
Description	Value of reference division line
Data type	Double precision floating point type (Double)
Range	-5E8 to 5E8
Preset value	0
Unit	Varies depending on the data format. • Log magnitude (MLOG): dB (decibel) • Phase (PHAS), Expanded phase (UPH) or Positive phase (PPH): ° (degree) • Group delay (GDEL): s (second) • Others: No unit
Note	If the specified variable 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.

Examples

```
Dim RefLvl As Double
SCPI.CALCulate(1).PARameter(2).SELect
SCPI.CALCulate(1).SELected.FORMat = "phas"
```

```
SCPI.DISPlay.WINDow(1).TRACe(2).Y.SCALe.RLEVel = 90
Pdiv = SCPI.DISPlay.WINDow(1).TRACe(2).Y.SCALe.RLEVel
```

Related objects

SCPI.CALCulate(Ch).SELected.FORMat

SCPI.DISPlay.WINDow(Ch).Y.SCALe.DIVisions

SCPI.DISPlay.WINDow(Ch).TRACe(Tr).Y.SCALe.PDIVision

SCPI.DISPlay.WINDow(Ch).TRACe(Tr).Y.SCALe.RPOSition

Equivalent key

Scale > **Reference Value**

Equivalent SCPI command

Syntax

:DISPlay:WINDow{[1]-4}:TRACe{[1]-4} :Y[:SCALe]:RLEVel <numeric>

:DISPlay:WINDow{[1]-4}:TRACe{[1]-4} :Y[:SCALe]:RLEVel?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":DISP:WIND1:TRAC1:Y:RLEV 1E2"
20 OUTPUT 717;":DISP:WIND1:TRAC1:Y:RLEV?"
30 ENTER 717;A
```

SCPI.DISPlay.WINDow(*Ch*).TRACe(*Tr*).Y.SCALe.RPOSition

Object type

Property (**Read-Write**)

Syntax

SCPI.DISPlay.WINDow(*Ch*).TRACe(*T36r*).Y.SCALe.RPOSition = *Value*

Value = SCPI.DISPlay.WINDow(*Ch*).TRACe(*Tr*).Y.SCALe.RPOSition

Description

This command specifies the position of a reference division line with its number (an integer assigned starting from 0 from the lowest division), for the selected trace (*Tr*) of selected channel (*Ch*).

Variable

Parameter	*Value*
Description	Position of reference division line
Data type	Long integer type (Long)
Range	0 to the number of divisions
Preset value	5
Note	If the specified variable 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.

Examples

```
Dim RefPos As Long
SCPI.DISPlay.WINDow(1).TRACe(2).Y.SCALe.RPOSition = 6
RefPos = SCPI.DISPlay.WINDow(1).TRACe(2).Y.SCALe.RPOSition
```

Related objects

SCPI.CALCulate(Ch).SELected.FORMat

SCPI.DISPlay.WINDow(Ch).Y.SCALe.DIVisions

SCPI.DISPlay.WINDow(Ch).TRACe(Tr).Y.SCALe.PDIVision

SCPI.DISPlay.WINDow(Ch).TRACe(Tr).Y.SCALe.RLEVel

Equivalent key

Scale > **Reference Position**

Equivalent SCPI command

Syntax

:DISPlay:WINDow{[1]-4}:TRACe{[1]-4} :Y[:SCALe]:RPOSition <numeric>

:DISPlay:WINDow{[1]-4}:TRACe{[1]-4} :Y[:SCALe]:RPOSition?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":DISP:WIND1:TRAC1:Y:RPOS 6"
20 OUTPUT 717;":DISP:WIND1:TRAC1:Y:RPOS?"
30 ENTER 717;A
```

SCPI.DISPlay.WINDow(*Ch*).TRACe(Tr).Y.SCALe.TOP

Object Type

Property (**Read-Write**)

Syntax

SCPI.DISPlay.WINDow(*Ch*).TRACe(Tr).Y.SCALe.TOP = *Value*

Value = SCPI.DISPlay.WINDow(*Ch*).TRACe(Tr).Y.SCALe.TOP

Description

This command sets or gets the maximum scale value for the Log-Y Axis.

Variable

Parameter	*Value*
Description	Maximum scale value for Log-Y-Axis
Data Type	Double precision floating point type (Double)
Range	2a ~ 1E
Preset Value	1k
Unit	Depending on Measurement format
Resolution	-

Examples

See SCPI.DISPlay.WINDow(Ch).TRACe(Tr).Y.SPACing

Related Objects

SCPI.DISPlay.WINDow(Ch).TRACe(Tr).Y.SPACing

SCPI.DISPlay.WINDow(Ch).TRACe(Tr).Y.SCALe.BOTTom

Equivalent Key

Scale > **Log Y-Axis Top/Bottom** > **Top Value**

Using either the keyboard or ENTRY keys on the front panel, enter the top value.

Equivalent SCPI Command

Sntax

:DISPlay:WINDow{[1]-4}:TRACe{[1]-4}:Y[:SCALe]:TOP <numeric>

:DISPlay:WINDow{[1]-4}:TRACe{[1]-4}:Y[:SCALe]:TOP?

Query Response

<numeric><newline><^END>

Example of use

```
10 OUTPUT 717;":DISP:WIND1:TRAC1:Y:TOP 0.000000000002"
20 OUTPUT 717;":DISP:WIND1:TRAC1:Y:TOP?"
30 ENTER 717;A
```

SCPI.DISPlay.WINDow(*Ch*).TRACe(Tr).Y.SPACing

Object Type

Property (**Read-Write**)

Syntax

SCPI.DISPlay.WINDow(*Ch*).TRACe(Tr).Y.SPACing = *Value*

Value = SCPI.DISPlay.WINDow(*Ch*).TRACe(Tr).Y.SPACing

Description

This command sets or gets Y-Axis format in Log or Linear format.

Variable

Parameter	*Param*
Description	Vertical axis display type of the graph (in Log or Linear)
Data Type	Character string type (String)
Range	LINear|LOGarithmic Select from the following: "LINear": Sets Y-Axis to Linear scale. "LOGarithmic" : Sets Y-Axis to Log scale.
Preset Value	LINear
Unit	-
Resolution	-

Examples

```
Dim Bottom as Double, Top As Double


SCPI.DISPlay.WINDow(1).TRACe(1).Y.SPACing = "LOG"

SCPI.DISPlay.WINDow(1).TRACe(1).Y.SCALe.BOTTom = 0.00000001
SCPI.DISPlay.WINDow(1).TRACe(1).Y.SCALe.TOP =1000

Bottom = SCPI.DISPlay.WINDow(1).TRACe(1).Y.SCALe.BOTTom
Top = SCPI.DISPlay.WINDow(1).TRACe(1).Y.SCALe.TOP
```

Related Objects

SCPI.DISPlay.WINDow(Ch).TRACe(Tr).Y.SCALe.BOTTom

SCPI.DISPlay.WINDow(Ch).TRACe(Tr).Y.SCALe.TOP

Equivalent Key

Scale > **Y-Axis**

Equivalent SCPI Command

Syntax

:DISPlay:WINDow{[1]-4}:TRACe{[1]-4}:Y:SPACing {LIN|LOG}

:DISPlay:WINDow{[1]-4}:TRACe{[1]-4}:Y:SPACing?

Query Response

{LIN|LOG}<newline><^END>

Example of use

```
10 OUTPUT 717;":DISP:WIND1:TRAC1:Y:SPAC LIN"
20 OUTPUT 717;":DISP:WIND1:TRAC1:Y:SPAC?"
30 ENTER 717;A$
```

SCPI.DISPlay.WINDow(*Ch*).TRACe(*Tr*).Y.TRACk.FREQuency

Object Type

Property (**Read-Write**)

Syntax

SCPI.DISPlay.WINDow(*Ch*).TRACe(*Tr*).Y.TRACk.FREQuency = *Value*

Value = SCPI.DISPlay.WINDow(*Ch*).TRACe(*Tr*).Y.TRACk.FREQuency

Description

This command sets/gets a frequency when you want to specify a frequency on the trace data as the reference value, for the selected trace (*Tr*) of the selected channel (*Ch*).

NOTE Tracking is not performed when the specified frequency lies outside the preset range. When a frequency that does not match any measurement point is specified, interpolation is performed using the preceding and following measurement points, and the resulting value is used as the reference value for tracking.

Variable

Parameter	*Value*
Description	Frequency for tracking
Data Type	Double precision floating point type (Double)
Range	-1E+12 to 1E+12
Preset Value	0
Unit	Hz (hertz)

Examples

```
Dim RefFreq as Double
SCPI.DISPlay.WINDow(1).TRACe(4).Y.TRACk.FREQuency = 945E8
RefFreq = SCPI.DISPlay.WINDow(1).TRACe(4).Y.TRACk.FREQuency
```

Related Objects

SCPI.DISPlay.WINDow(Ch).TRACe(Tr).Y.TRACk.MODE

Equivalent Key

Scale > **Reference Tracking** > **Track Frequency**

Equivalent SCPI Command

Syntax

:DISPlay:WINDow{[1]-4}:TRACe{[1]-4}:Y:TRACk:FREQuency <numeric>

:DISPlay:WINDow{[1]-4}:TRACe{[1]-4}:Y:TRACk:FREQuency?

Query Response

{numeric} <newline><^END>

Example of use

```
10 OUTPUT 717;":DISP:WIND1:TRAC2:Y:TRAC:FREQ 1.3E9"
20 OUTPUT 717;":DISP:WIND1:TRAC2:Y:TRAC:FREQ?"
30 ENTER 717;A
```

SCPI.DISPlay.WINDow(*Ch*).TRACe(*Tr*).Y.TRACk.MODE

Object Type

Property (**Read-Write**)

Syntax

SCPI.DISPlay.WINDow(*Ch*).TRACe(*Tr*).Y.TRACk.MODE = *Param*

Param = SCPI.DISPlay.WINDow(*Ch*).TRACe(*Tr*).Y.TRACk.MODE

Description

This command sets/gets the tracking method to offset the trace data after sweep, for the selected trace (*Tr*) of the selected channel (*Ch*).

Variable

Parameter	*Param*
Description	Tracking method to offset the trace data
Data Type	Character string type (String)
Range	Select from the following: • "OFF" : Does not perform tracking for trace data. • "PEAK": Specifies the peak value as the reference value. • "FREQuency": Specifies the specified frequency as the reference value.
Preset Value	"OFF"

Examples

```
Dim TracMode as String
SCPI.DISPlay.WINDow(2).TRACe(1).Y.TRACk.MODE = "FREQ"
TracMode = SCPI.DISPlay.WINDow(2).TRACe(1).Y.TRACk.MODE
```

Related Objects

SCPI.DISPlay.WINDow(Ch).TRACe(Tr).Y.TRACk.FREQuency

Equivalent Key

Scale > **Reference Tracking** > **Tracking** > **Off** | **Track Peak** | **Track Freq**

Equivalent SCPI Command

Syntax

:DISPlay:WINDow{[1]-4}:TRACe{[1]-4}:Y:TRACk:MODE {OFF|PEAK|FREQuency}

:DISPlay:WINDow{[1]-4}:TRACe{[1]-4}:Y:TRACk:MODE?

Query Response

{OFF|PEAK|FREQ} <newline><^END>

Example of use

```
10 OUTPUT 717;":DISP:WIND1:TRAC4:Y:TRAC:MODE FREQ"
20 OUTPUT 717;":DISP:WIND1:TRAC4:Y:TRAC:MODE?"
30 ENTER 717;A$
```

SCPI.DISPlay.WINDow(*Ch*).X.SPACing

Object type

Property (**Read-Write**)

Syntax

SCPI.DISPlay.WINDow(*Ch*).X.SPACing = *Param*

Param = SCPI.DISPlay.WINDow(*Ch*).X.SPACing

Description

This command selects the display type of the graph horizontal axis of the selected channel (*Ch*) for segment sweep.

Variable

Parameter	*Param*
Description	Horizontal axis display type of the graph for segment sweep
Data type	Character string type (String)
Range	Select from the following: • "LINear": Specifies the frequency base (linear frequency axis with the minimum frequency at the left edge and the maximum frequency at the right edge). • "OBASe" : Specifies the order base (axis in which the measurement point numbers are positioned evenly in the order of measurement).
Preset value	"OBASe"

Examples

```
Dim DispSegm As String
SCPI.SENSe(1).SWEep.TYPE = "segm"
SCPI.DISPlay.WINDow(1).X.SPACing = "obas"
DispSegm = SCPI.DISPlay.WINDow(1).X.SPACing
```

Related objects

SCPI.SENSe(Ch).SWEep.TYPE

Equivalent key

Sweep Setup > **Segment Display** > **Freq Base|Order Base**

Equivalent SCPI command

Syntax

```
:DISPlay:WINDow{[1]-4}:X:SPACing {LINear|OBASe}
:DISPlay:WINDow{[1]-4}:X:SPACing?
```

Query response

```
{LIN|OBAS}<newline><^END>
```

Example of use

```
10 OUTPUT 717;":DISP:WIND1:X:SPAC OBAS"
20 OUTPUT 717;":DISP:WIND1:X:SPAC?"
30 ENTER 717;A$
```

SCPI.DISPlay.WINDow(*Ch*).Y.SCALe.DIVisions

Object type

Property (**Read-Write**)

Syntax

SCPI.DISPlay.WINDow(*Ch*).Y.SCALe.DIVisions = *Value*

Value = SCPI.DISPlay.WINDow(*Ch*).Y.SCALe.DIVisions

Description

This command sets/gets the number of divisions in all the graphs, for the selected channel (*Ch*).

Variable

Parameter	*Value*
Description	Number of divisions of graph
Data type	Long integer type (Long)
Range	4 to 30
Preset value	10
Resolution	2
Note	If the specified variable 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.

Examples

```
Dim Divs As Long
SCPI.DISPlay.WINDow(1).Y.SCALe.DIVisions = 12
Divs = SCPI.DISPlay.WINDow(1).Y.SCALe.DIVisions
```

Related objects

SCPI.DISPlay.WINDow(Ch).TRACe(Tr).Y.SCALe.PDIVision

SCPI.DISPlay.WINDow(Ch).TRACe(Tr).Y.SCALe.RLEVel

SCPI.DISPlay.WINDow(Ch).TRACe(Tr).Y.SCALe.RPOSition

Equivalent key

Scale > **Divisions**

Equivalent SCPI command

Syntax

```
:DISPlay:WINDow{[1]-4}:Y[:SCALe]:DIVisions <numeric>
:DISPlay:WINDow{[1]-4}:Y[:SCALe]:DIVisions?
```

Query response

```
{numeric}<newline><^END>
```

Example of use

```
10 OUTPUT 717;":DISP:WIND1:Y:DIV 12"
20 OUTPUT 717;":DISP:WIND1:Y:DIV?"
30 ENTER 717;A
```

FORMAT

SCPI.FORMat.BORDer

Object type

Property (**Read-Write**)

Syntax

SCPI.FORMat.BORDer = *Param*

Param = SCPI.FORMat.BORDer

Description

This command sets/gets the transfer order of each byte in the output data (byte order), when the data transfer format is set to binary mode (by specifying "REAL" with SCPI.FORMat.DATA object).

NOTE This object is NOT used when controlling the E5061B using COM objects through E5061B VBA.

Variable

Parameter	*Param*
Description	Byte order
Data type	Character string type (String)
Range	Select from the following: "NORMal": Specifies the byte order in which transfer starts from the byte including MSB (Most Significant Bit). "SWAPped": Specifies the byte order in which transfer starts from the byte including LSB (Least Significant Bit).
Preset value	"NORMal"

Examples

```
Dim BitOrd As String
SCPI.FORMat.BORDer = "swap"
BitOrd = SCPI.FORMat.BORDer
```

Related objects

SCPI.FORMat.DATA

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

:FORMat:BORDer {NORMal|SWAPped}

:FORMat:BORDer?

Query response

{NORM|SWAP}<newline><^END>

Example of use

```
10 OUTPUT 717;":FORM:BORD SWAP"
20 OUTPUT 717;":FORM:BORD?"
30 ENTER 717;A$
```

SCPI.FORMat.DATA

Object type

Property (**Read-Write**)

Syntax

SCPI.FORMat.DATA = *Param*

Param = SCPI.FORMat.DATA

Description

This command can be used to set/get format data using the following SCPI commands:

SCPI.CALCulate(Ch).SELected.DATA.FDATa

SCPI.CALCulate(Ch).SELected.DATA.FMEMory

SCPI.CALCulate(Ch).SELected.DATA.SDATa

SCPI.CALCulate(Ch).SELected.DATA.SMEMory

SCPI.CALCulate(Ch).SELected.FUNCtion.DATA

SCPI.CALCulate(Ch).SELected.LIMit.DATA

SCPI.CALCulate(Ch).SELected.LIMit.REPort.ALL

SCPI.CALCulate(Ch).SELected.LIMit.REPort.DATA

SCPI.CALCulate(Ch).SELected.BLIMit.REPort.DATA

SCPI.CALCulate(Ch).SELected.RLIMit.DATA

SCPI.CALCulate(Ch).SELected.RLIMit.REPort.DATA

SCPI.SENSe(Ch).CORRection.COEFficient.DATA

SCPI.SENSe(Ch).CORRection.COEFficient.GPData

SCPI.SENSe(Ch).FREQuency.DATA

SCPI.SENSe(Ch).SEGMent.DATA

NOTE ASCII transfer format must be specified when controlling the E5061B using SCPI commands with the Parse object in the E5061B VBA.

Variable

Parameter	*Param*
Description	Data transfer format
Data type	Character string type (String)

Range	Select from the following: • "ASCii": Specifies the ASCII transfer format. • "REAL": Specifies the IEEE 64-bit floating point binary transfer format. • "REAL32": Specifies the IEEE 32-bit floating point binary transfer format.
Preset value	"ASCii"

Examples

```
Dim Fmt As String
SCPI.FORMat.DATA = "ASC"
Fmt = SCPI.FORMat.DATA
```

Related objects

SCPI.FORMat.BORDer

Parse

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

:FORMat:DATA {ASCii|REAL|REAL32}

:FORMat:DATA?

Query response

{ASC|REAL|REAL32}<newline><^END>

Example of use

```
10 OUTPUT 717;":FORM:DATA REAL"
20 OUTPUT 717;":FORM:DATA?"
30 ENTER 717;A$
```

SCPI.FORMat.REAL.ASCii.LENGth

Object Type

Property (**Read-Write**)

Syntax

SCPI.FORMat.REAL.ASCii.LENGth = *Value*

Value = SCPI.FORMat.REAL.ASCii.LENGth

Description

This command sets and gets the number of significant digits of a floating point number format in Ascii format. See ASCII Transfer Format.

For example, 1000 is expressed as :

- When 12 is set, "+1.00000000000E+003."
- When 14 is set, "+1.0000000000000E+003."

Variable

Parameter	*Value*
Description	Number of significant digits
Data Type	Long integer type (Long)
Range	12|14
Preset Value	12
Unit	-
Resolution	-

Examples

```
Dim Var as Long
SCPI.FORMat.REAL.ASCii.LENGth = 14
Var = SCPI.FORMat.REAL.ASCii.LENGth
```

Related Objects

Equivalent Key

No equivalent key is available on the front panel.

Equivalent SCPI Command

Syntax

:FORMat:REAL:ASCii:LENGth {12|14}

:FORMat:REAL:ASCii:LENGth?

Query Response

{12|14}<newline><^END>

Example of use

```
10 OUTPUT 717;":FORM:REAL:ASC:LENG 14"
20 OUTPUT 717;":FORM:REAL:ASC:LENG?"
30 ENTER 717;A
```

HCOPY

SCPI.HCOPy.ABORt

Object type

Method (**Write Only**)

Syntax

SCPI.HCOPy.ABORt

Description

This command aborts the print output.

Examples

SCPI.HCOPy.ABORt

Related objects

SCPI.HCOPy.IMMediate

Equivalent key

System > **Abort Printing**

Equivalent SCPI command

Syntax

:HCOPy:ABORt

Example of use

```
10 OUTPUT 717;":HCOP:ABOR"
```

SCPI.HCOPy.SDUMp.DATA.FORMat

Object Type

Property (**Read-Write**)

Syntax

SCPI.HCOPy.SDUMp.DATA.FORMat = *Value*

Value = SCPI.HCOPy.SDUMp.DATA.FORMat

Description

This command sets or gets the image format (print screen image).

Variable

Parameter	*Value*
Description	Image format
Data Type	Character string type (String)
Range	PNG|BMP
Preset Value	PNG
Unit	-
Resolution	-

Examples

```
Dim PixType as String
PixType = "PNG"
SCPI.HCOPy.SDUMp.DATA.FORMat = PixType

ReadPixType = SCPI.HCOPy.SDUMp.DATA.FORMat
```

Related Objects

SCPI.HCOPy.SDUMp.DATA.IMMediate

Equivalent Key

No equivalent key is available on the front panel.

Equivalent SCPI Command

Syntax

:HCOPy:SDUMp:DATA:FORMat {PNG|BMP}

:HCOPy:SDUMp:DATA:FORMat?

Query Response

{PNG|BMP}<newline><^END>

Example of use

```
10 OUTPUT 717;":HCOP:SDUM:DATA:FORM PNG"
20 OUTPUT 717;":HCOP:SDUM:DATA:FORM?"
30 ENTER 717;A$
```

SCPI.HCOPy.SDUMp.DATA.IMMediate

Object Type

Property (**Read Only**)

Syntax

Value = SCPI.HCOPy.SDUMp.DATA.IMMediate

Description

This command gets print screen image.

Variable

Parameter	*Value*
Description	print screen
Data Type	Binary (Byte)
Range	-
Preset Value	-
Unit	-
Resolution	-

Examples

```
Dim TmpData() as Byte
Open "C:\Screen.PNG" For Binary As #1
TmpData() = SCPI.HCOPy.SDUMp.DATA.IMMediate
Put #1, , TmpData()
Close #1
```

Related Objects

SCPI.HCOPy.SDUMp.DATA.FORMat

Equivalent Key

No equivalent key is available on the front panel.

Equivalent SCPI Command

Syntax

:HCOPy:SDUMp:DATA[:IMMediate]?

Query Response

<Binary><newline><^END>

Example of use (VISA-COM)

```
'*** Save the screen image to the file
Age506x.WriteString ":HCOP:SDUM:DATA:FORM PNG", True
Age506x.WriteString ":HCOP:SDUM:DATA?", True
PNGdata = Age506x.ReadIEEEBlock(BinaryType_UI1, False, True)

Open "C:\TEST.png" For Binary As #1
   Put #1, , PNGdata()
Close
```

SCPI.HCOPy.IMAGe

Object type

Property (**Read-Write**)

Syntax

SCPI.HCOPy.IMAGe = *Param*

Param = SCPI.HCOPy.IMAGe

Description

This command sets/gets the print color for output (to the printer).

Variable

Parameter	*Param*
Description	Print color for output to the printer.
Data type	Character string type (String)
Range	Select from the following: • "NORMal": Specifies printing in close color to the display color. • "INVert": Specifies printing in the inverted color of the display color.
Preset value	"INVert"

Examples

```
Dim Img As String
SCPI.HCOPy.IMAGe = "norm"
Img = SCPI.HCOPy.IMAGe
```

Related objects

SCPI.HCOPy.IMMediate

Equivalent key

System > **Invert Image**

Equivalent SCPI command

Syntax

:HCOPy:IMAGe {NORMal|INVert}

:HCOPy:IMAGe?

Query response

{NORM|INV}<newline><^END>

Example of use

```
10 OUTPUT 717;":HCOP:IMAG NORM"
20 OUTPUT 717;":HCOP:IMAG?"
30 ENTER 717;A$
```

SCPI.HCOPy.IMMediate

Object type

Method (**Write Only**)

Syntax

SCPI.HCOPy.IMMediate

Description

This command outputs the display image on the LCD display to the printer connected to the E5061B.

NOTE When printing the E5061B measurement screen, execute the VBA program with the Visual Basic application closed. For the method, see Running a Program from the E5061B Measurement Screen.

Examples

```
SCPI.HCOPy.IMMediate
```

Related objects

SCPI.HCOPy.ABORt

SCPI.HCOPy.IMAGe

Equivalent key

System > **Print**

When performing the operation from the front panel, the image on the LCD display memorized in the volatile memory (clipboard) (the image on the LCD display when the **Capture** (**System**) key is pressed) is printed. If no image is memorized in the clipboard, in the same way as the SCPI.HCOPy.IMMediate object, the image on the LCD display at the execution is memorized in the clipboard and then it is printed.

Equivalent SCPI command

Syntax

:HCOPy[:IMMediate]

Example of use

```
10 OUTPUT 717;":HCOP"
```

IEEE

SCPI.IEEE4882.CLS

Object type

Method (**Write Only**)

Syntax

SCPI.IEEE4882.CLS

Description

This command clears the following:

- Error Queue
- Status Byte Register
- Standard Event Status Register
- Operation Status Event Register
- Questionable Status Event Register
- Questionable Limit Status Event Register
- Questionable Limit Channel Status Event Register
- Questionable Bandwidth Limit Status Event Register
- Questionable Bandwidth Limit Channel Status Event Register
- Questionable Ripple Limit Status Event Register
- Questionable Ripple Limit Channel Status Event Register

Examples

```
SCPI.IEEE4882.CLS
```

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

*CLS

Example of use

```
10 OUTPUT 717;"*CLS"
```

SCPI.IEEE4882.ESE

Object type

Property (**Read-Write**)

Syntax

SCPI.IEEE4882.ESE = *Value*

Value = SCPI.IEEE4882.ESE

Description

This command sets/gets the value of the Standard Event Status Enable Register.

Variable

Parameter	*Value*
Description	Value of the Standard Event Status Enable Register
Data type	Long integer type (Long)
Range	0 to 255
Preset value	0
Note	If the specified variable is out of the allowable setup range, the result of bitwise AND with 255 (0xff) is set.

Examples

```
Dim Stat As Long
SCPI.IEEE4882.ESE = 16
Stat = SCPI.IEEE4882.ESE
```

Related objects

SCPI.IEEE4882.SRE

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

*ESE <numeric>

*ESE?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;"*ESE 16"
20 OUTPUT 717;"*ESE?"
30 ENTER 717;A
```

SCPI.IEEE4882.ESR

Object type

Property (**Read Only**)

Syntax

Value = SCPI.IEEE4882.ESR

Description

This command reads the value of the Standard Event Status Register. Execution of this command clears the register value.

Variable

Parameter	*Value*
Description	Value of the Standard Event Status Register
Data type	Long integer type (Long)
Preset value	128

Examples

```
Dim Stat As Long
Stat = SCPI.IEEE4882.ESR
```

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

*ESR?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;"*ESR?"
20 ENTER 717;A
```

SCPI.IEEE4882.IDN

Object type

Property (**Read Only**)

Syntax

Cont = SCPI.IEEE4882.IDN

Description

This command reads the product information (manufacturer, model number, serial number, and firmware revision number) of the E5061B.

Variable

Parameter	*Cont*
Description	Product information ("{string 1},{string 2},{string 3},{string 4}") • {string 1}: Manufacturer. Agilent Technologies is always read out. • {string 2}: Model number. E5061B is always read out. • {string 3}: Serial number (example: MY123400101). • {string 4}: Firmware revision number (example: A.07.00).
Data type	Character string type (String)

Examples

```
Dim Who As String
Who = SCPI.IEEE4882.IDN
```

Equivalent key

System > **Firmware Revision**

Equivalent SCPI command

Syntax

*IDN?

Query response

{string 1},{string 2},{string 3},{string 4}<newline><^END>

Example of use

```
10 OUTPUT 717;"*IDN?"
20 ENTER 717;A$
```

SCPI.IEEE4882.LRN

Object Type

Property (**Read Only**)

Syntax

Value = SCPI.IEEE4882.LRN

Description

This command gets the device setup query. This command is defined as "Learn Device Setup Query" in IEEE 488.2. The command returns instrument settings by binary block data (same as Save/Recall state file contents) with "SYSTem:SET " prefix.

The returned data is the same as the contents of state file which can be saved by the SCPI.MMEMory.STORe.STATe. Therefore, the returned data contents is changed according to the setting of SCPI.MMEMory.STORe.STYPe.

Variable

Parameter	*Value*
Description	Setting Data
Data Type	Binary Type (Byte)
Range	-
Preset Value	-
Unit	-
Resolution	-

Examples

```
'To write LRN data
Dim TempWrite() As Byte
Open "C:\LRN.dat" For Binary As #1
TempWrite() = SCPI.IEEE4882.LRN
Put #1, , TempWrite()
Close #1
```

To save/get the E5061B state for COM, use SCPI.MMEMory.STORe.STATe and SCPI.MMEMory.LOAD.STATe.

Related objects

SCPI.SYSTem.SET

SCPI.MMEMory.LOAD.STATe

SCPI.MMEMory.STORe.STATe

SCPI.MMEMory.STORe.STYPe

Equivalent Key

No equivalent key is available on the front panel.

Equivalent SCPI Command

Syntax

*LRN?

Query Response

<binary><newline><^END>

Example of use (VISA-COM)

```
Dim LRNData() As Byte, SETData() As Byte, NoofByte As Double
 '*** Get the LRN data as Binaray data
 Age506x.WriteString "*LRN?", True
 LRNData = Age506x.ReadIEEEBlock(BinaryType_UI1, False, True)

 '*** Save the LRN data in the file
Open "C:\LRN.dat" For Binary As #1
          Put #1, , LRNData()
Close

MsgBox "Get Data"

'*** Recall the LRN data from the file
Open "C:\LRN.dat" For Binary As #1
          NoofByte = LOF(1)
         ReDim SETData(NoofByte)
         Get #1, , SETData()
Close
```

```
'*** Send the LRN data to E5061B
Age506x.IO.Write SETData, NoofByte
```

SCPI.IEEE4882.OPC

Object type

Property (**Read-Write**)

Syntax

(1)SCPI.IEEE4882.OPC = *Dummy*

(2)*Value* = SCPI.IEEE4882.OPC

Description

Case (1):

This command sets/gets 1 the OPC bit (bit 0) of the Standard Event Status Register when all of pending operations complete. For information on the structure of the status register, see Status Reporting System.

Variable

Case (2):

Parameter	*Value*
Description	1 is returned when all pending operations are completed
Data type	Long integer type (Long)

Examples

```
Dim Opcbit as Long
SCPI.IEEE4882.OPC = 1
Opcbit = SCPI.IEEE4882.OPC
```

Related objects

SCPI.SENSe(Ch).CORRection.COLLect.ACQuire.ISOLation

SCPI.SENSe(Ch).CORRection.COLLect.ACQuire.LOAD

SCPI.SENSe(Ch).CORRection.COLLect.ACQuire.OPEN

SCPI.SENSe(Ch).CORRection.COLLect.ACQuire.SHORt

SCPI.SENSe(Ch).CORRection.COLLect.ACQuire.THRU

SCPI.TRIGger.SEQuence.SINGle

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

*OPC

Example of use

```
10 OUTPUT 717;"*OPC"
```

SCPI.IEEE4882.OPT

Object type

Property (**Read Only**)

Syntax

Cont = SCPI.IEEE4882.OPT

Description

This command reads the identification numbers of options installed in the E5061B.

Variable

Parameter	*Cont*
Description	Identification numbers of installed options
Data type	Character string type (String)
Note	If there is no installed option, 0 is read out.

Examples

```
Dim OptNum As String
OptNum = SCPI.IEEE4882.OPT
```

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

*OPT?

Query response

{numeric}<newline><^END>

Example of use

```
Call viVPrintf(vi, "*OPT?" + vbLf, 0)
Call viVScanf(vi, "%t", Result)
```

SCPI.IEEE4882.RST

Object type

Method (**Write Only**)

Syntax

SCPI.IEEE4882.RST

Description

This command presets the E5061B to its default settings and is different from setting state preset with the SCPI.SYSTem.PRESet object as the continuous initiation mode (see SCPI.INITiate(Ch).CONTinuous) of channel 1 is set to OFF.

Examples

```
SCPI.IEEE4882.RST
```

Related objects

SCPI.INITiate(Ch).CONTinuous

SCPI.SYSTem.PRESet

SCPI.SYSTem.UPReset

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

*RST

Example of use

```
10 OUTPUT 717;"*RST"
```

SCPI.IEEE4882.SRE

Object type

Property (**Read-Write**)

Syntax

SCPI.IEEE4882.SRE = *Value*

Value = SCPI.IEEE4882.SRE

Description

This command sets/gets the value of Service Request Enable Register.

Variable

Parameter	*Value*
Description	Value of the Service Request Enable Register
Data type	Long integer type (Long)
Range	0 to 255
Preset value	0
Note	If the specified variable is out of the allowable setup range, the result of bitwise AND with 255 (0xff) is set. Note that bit 6 cannot be set to 1.

Examples

```
Dim Stat As Long
SCPI.IEEE4882.SRE = 8
Stat = SCPI.IEEE4882.SRE
```

Related objects

SCPI.IEEE4882.ESE

SCPI.STATus.OPERation.ENABle

SCPI.STATus.QUEStionable.ENABle

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

*SRE <numeric>

*SRE?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;"*SRE 128"
20 OUTPUT 717;"*SRE?"
30 ENTER 717;A
```

SCPI.IEEE4882.STB

Object type

Property (**Read Only**)

Syntax

Value = SCPI.IEEE4882.STB

Description

This command reads the value of Status Byte Register.

Variable

Parameter	*Value*
Description	Value of the Status Byte Register
Data type	Long integer type (Long)
Preset Value	0

Examples

```
Dim Stat As Long
Stat = SCPI.IEEE4882.STB
```

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

*STB?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;"*STB?"
20 ENTER 717;A
```

SCPI.IEEE4882.TRG

Object type

Method (**Write Only**)

Syntax

SCPI.IEEE4882.TRG

Description

This command triggers the E5061B if the trigger source is set to GPIB/LAN (set to BUS with the SCPI.TRIGger.SEQuence.SOURce).

Examples

```
SCPI.TRIGger.SEQuence.SOURce = "bus"
SCPI.IEEE4882.TRG
```

Related objects

SCPI.TRIGger.SEQuence.SOURce

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

*TRG

Example of use

```
10 OUTPUT 717;"*TRG"
```

SCPI.IEEE4882.TST

Object Type

Property (**Read Only**)

Syntax

Value = SCPI.IEEE4882.TST

Description

Does nothing. The self-test is not executed by this command in the case of the E5061B. Always returns 0.

Variable

Parameter	*Value*
Description	Self-test query
Data Type	Long integer type (Long)
Range	0 (Always returns 0)
Preset Value	0

Examples

```
Dim Dmy as Long
Dmy = SCPI.IEEE4882.TST
```

Related Objects

Equivalent Key

No equivalent key is available on the front panel.

Equivalent SCPI Command

Syntax

*TST?

Query Response

{numeric} <newline><^END>

NOTE Always returns 0.

Example of use

```
10 OUTPUT 717;"*TST?"
20 ENTER 717;A
```

SCPI.IEEE4882.WAI

Object type

Method (**Write Only**)

Syntax

SCPI.IEEE4882.WAI

Description

This command waits for the execution of all objects sent before this command is completed.

Examples

```
SCPI.TRIGger.SEQuence.SOURce = "bus"
SCPI.TRIGger.SEQuence.SINGle
SCPI.IEEE4882.WAI
MsgBox "Done"
```

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

*WAI

Example of use

```
10 OUTPUT 717;"*WAI"
```

INIT

SCPI.INITiate(*Ch*).CONTinuous

Object type

Property (**Read-Write**)

Syntax

SCPI.INITiate(*Ch*).CONTinuous = *Status*

Status = SCPI.INITiate(*Ch*).CONTinuous

Description

This command turns ON/OFF the continuous initiation mode (setting by which the trigger system initiates continuously) of the selected channel (*Ch*) in the trigger system. For more information on the trigger system, see Section Trigger System.

Variable

Parameter	*Status*
Description	ON/OFF of the continuous initiation mode
Data type	Boolean type (Boolean)
Range	Select from the following: • True or ON: Turns ON the continuous initiation mode. • False or OFF: Turns OFF the continuous initiation mode.
Preset value	Varies depending on [variable (*Ch*)]

Examples

```
Dim ContMode As Boolean
SCPI.INITiate(2).CONTinuous = True
ContMode = SCPI.INITiate(2).CONTinuous
```

Related objects

SCPI.INITiate(Ch).IMMediate

Equivalent key

Trigger > **Continuous** (ON)

Trigger > **Hold** (OFF)

Trigger > **Hold All Channels** (OFF for all channels)

Trigger > **Continuous Disp Channels** (ON for displayed channels)

Equivalent SCPI command

Syntax

:INITiate{[1]-4}:CONTinuous {ON|OFF|1|0}

:INITiate{[1]-4}:CONTinuous?

Query response

{1|0}<newline><^END>

Example of use

```
10 OUTPUT 717;":INIT1:CONT OFF"
20 OUTPUT 717;":INIT1:CONT?"
30 ENTER 717;A
```

SCPI.INITiate(*Ch*).IMMediate

Object type

Method (**Write Only**)

Syntax

SCPI.INITiate(*Ch*).IMMediate

Description

This command changes the state of each channel of the selected channel (*Ch*) to the initiation state in the trigger system.

When this object is executed for a channel in the idle state in the trigger system, it goes into the initiation state immediately. Then, after measurement is executed once, it goes back to the idle state.

If this object is executed for a channel that is not in the idle state or a channel for which the continuous initiation mode is set to ON (setting by which the trigger system initiates continuously) in the trigger system, an error occurs when executed and the object is ignored. For more information on the trigger system, see Trigger System.

Examples

```
SCPI.INITiate(1).CONTinuous = False
SCPI.INITiate(1).IMMediate
```

Related objects

SCPI.INITiate(Ch).CONTinuous

Equivalent key

Trigger > **Single**

Equivalent SCPI command

Syntax

:INITiate{[1]-4}[:IMMediate]

Example of use

```
10 OUTPUT 717;":INIT1"
```

INPUT

SCPI.INPut.ATTenuation.GPPort.R

Object Type

Property (**Read-Write**)

Syntax

SCPI.INPut.ATTenuation.GPPort.R = *Value*

Value = SCPI.INPut.ATTenuation.GPPort.R

Description

This command sets/gets the input attenuation of port R for gain phase measurement.

Variable

Parameter	*Value*
Description	input attenuation of port R
Data Type	Double precision floating point type (Double)
Range	0 or 20
Preset Value	20
Unit	dB

Examples

```
Dim AttPortR as Double
SCPI.INPut.ATTenuation.GPPort.R = 20
AttPortR = SCPI.INPut.ATTenuation.GPPort.R
```

Related Objects

SCPI.INPut.ATTenuation.GPPort.T

SCPI.INPut.IMPedance.GPPort.R

SCPI.INPut.IMPedance.GPPort.T

Equivalent Key

Meas > **Gain-Phase** > **R Attenuator** > **0dB** | **20dB**

System > **Overload Recovery** > **R Attenuator** > **0dB** | **20dB**

Equivalent SCPI Command

Syntax

:INPut:ATTenuation:GPPort:R <numeric>

```
:INPut:ATTenuation:GPPort:R?
```

Query Response

```
{numeric} <newline><^END>
```

Example of use

```
10 OUTPUT 717;":INP:ATT:GPP:R 20"
20 OUTPUT 717;":INP:ATT:GPP:R ?"
30 ENTER 717;A
```

SCPI.INPut.ATTenuation.GPPort.T

Object Type

Property (**Read-Write**)

Syntax

SCPI.INPut.ATTenuation.GPPort.T = *Value*

Value = SCPI.INPut.ATTenuation.GPPort.T

Description

This command sets/gets the input attenuation of port T for gain phase measurement.

Variable

Parameter	*Value*
Description	input attenuation of port T
Data Type	Double precision floating point type (Double)
Range	0 or 20
Preset Value	20
Unit	dB

Examples

```
Dim AttPortT as Double
SCPI.INPut.ATTenuation.GPPort.T = 20
AttPortT = SCPI.INPut.ATTenuation.GPPort.T
```

Related Objects

SCPI.INPut.ATTenuation.GPPort.R

SCPI.INPut.IMPedance.GPPort.R

SCPI.INPut.IMPedance.GPPort.T

Equivalent Key

Meas > **Gain-Phase** > **T Attenuator** > **0dB** | **20dB**

System > **Overload Recovery** > **T Attenuator** > **0dB** | **20dB**

Equivalent SCPI Command

Syntax

:INPut:ATTenuation:GPPort:T <numeric>

:INPut:ATTenuation:GPPort:T?

Query Response

{numeric} <newline><^END>

Example of use

```
10 OUTPUT 717;":INP:ATT:GPP:T 20"
20 OUTPUT 717;":INP:ATT:GPP:T?"
30 ENTER 717;A
```

SCPI.INPut.IMPedance.GPPort.R

Object Type

Property (**Read-Write**)

Syntax

SCPI.INPut.IMPedance.GPPort.R = *Value*

Value = SCPI.INPut.IMPedance.GPPort.R

Description

This command sets/gets input impedance of port R for gain phase measurement.

Variable

Parameter	*Value*
Description	input impedance of port R
Data Type	Double precision floating point type (Double)
Range	50 or 1E+6
Preset Value	1E+6
Unit	Ω

Examples

```
Dim ImpPortR as Double
SCPI.INPut.IMPedance.GPPort.R = 50
ImpPortR = SCPI.INPut.IMPedance.GPPort.R
```

Related Objects

SCPI.INPut.ATTenuation.GPPort.R

SCPI.INPut.ATTenuation.GPPort.T

SCPI.INPut.IMPedance.GPPort.T

Equivalent Key

Meas > **Gain-Phase** > **R Input Z** > **50Ω | 1MΩ**

System > **Overload Recovery** > **R Input Z** > **50Ω | 1MΩ**

Equivalent SCPI Command

Syntax

:INPut:IMPedance:GPPort:R <numeric>

:INPut:IMPedance:GPPort:R?

Query Response

{numeric} <newline><^END>

Example of use

```
10 OUTPUT 717;":INP:IMP:GPP:R 1E+6"
20 OUTPUT 717;":INP:IMP:GPP:R?"
30 ENTER 717;A
```

SCPI.INPut.IMPedance.GPPort.T

Object Type

Property (**Read-Write**)

Syntax

SCPI.INPut.IMPedance.GPPort.T = *Value*

Value = SCPI.INPut.IMPedance.GPPort.T

Description

This command sets/gets input impedance of port T for gain phase measurement.

Variable

Parameter	*Value*
Description	input impedance of port T
Data Type	Double precision floating point type (Double)
Range	50 or 1E+6
Preset Value	1E+6
Unit	Ω

Examples

```
Dim ImpPortT as Double
SCPI.INPut.IMPedance.GPPort.T = 1E+6
ImpPortT = SCPI.INPut.IMPedance.GPPort.T
```

Related Objects

SCPI.INPut.ATTenuation.GPPort.R

SCPI.INPut.ATTenuation.GPPort.T

SCPI.INPut.IMPedance.GPPort.R

Equivalent Key

Meas > **Gain-Phase** > **T Input Z** > **50Ω | 1MΩ**

System > **Overload Recovery** > **T Input Z** > **50Ω | 1MΩ**

Equivalent SCPI Command

Syntax

:INPut:IMPedance:GPPort:T <numeric>

:INPut:IMPedance:GPPort:T?

Query Response

{numeric} <newline><^END>

Example of use

```
10 OUTPUT 717;":INP:IMP:GPP:T 50"
20 OUTPUT 717;":INP:IMP:GPP:T?"
30 ENTER 717;A
```

LXI

SCPI.LXI.IDENtify.STATe

Object Type

Property (**Read-Write**)

Syntax

SCPI.LXI.IDENtify.STATe = *Status*

Status = SCPI.LXI.IDENtify.STATe

Description

This command sets or gets the LXI Status Indicator state.

Variable

Parameter	*Status*
Description	ON/OFF LXI Control Identification
Data Type	Boolean type (Boolean)
Range	Select from the following: True or On: Sets the LXI Status Indicator to the 'IDENTIFY' state. False or Off: Set the LXI Status Indicator to the 'NORMAL' state.
Preset Value	False
Unit	-
Resolution	-

Examples

```
Dim Var as Boolean
SCPI.LXI.IDENtify.STATe = True
Var = SCPI.LXI.IDENtify.STATe
```

Equivalent Key

There is no equivalent key is available on the front panel. However, the similar key is:

System > **Misc Setup** > **Network Setup** > **LAN Dialog...**

Equivalent SCPI Command

Syntax

:LXI:IDENtify[:STATe] {ON|OFF|1|0}

:LXI:IDENtify[:STATe]?

Query Response

{1|0}<newline><^END>

Example of use

```
10 OUTPUT 717;":LXI:IDEN ON"
20 OUTPUT 717;":LXI:IDEN?"
30 ENTER 717;A
```

MEMORY

:MMEMory:LOAD:PROGram

No equivalent COM Commands

Syntax

:MMEMory:LOAD:PROGram <string>

Description

This command loads (or imports) a VBA project (a file with the .vba extension), a module (a file with the .bas extension), a user form (a file with the .frm extension) or a class module (a file with the .cls extension). If the specified file does not exist, an error occurs and the command is ignored.

Variables

Parameter	*String*
Description	File name in which you want to load the VBA project
Range	254 characters or less
Preset value	""

Example of use

```
10 OUTPUT 717;":MMEM:LOAD:PROG ""Test1/Test1_01.vba"""
10 OUTPUT 717;":MMEM:LOAD:PROG ""A:Test1_01.vba"""
```

Related commands

:MMEM:STOR:PROG

Equivalent key

Macro Setup > **Load VBA Project**

:MMEMory:STORe:PROGram

No equivalent COM Commands

Syntax

:MMEMory:STORe:PROGram <string>

Description

This command saves a VBA project opened on the VBA editor into a file. The file name needs to have a .vba extension. If a file with the specified file name exists, its contents are overwritten.

Variables

Parameter	*String*
Description	File name in which you want to save the VBA project
Range	254 characters or less
Preset value	""

Example of use

10 OUTPUT 717;":MMEM:STOR:PROG ""Test1/Test1_01.vba"""

10 OUTPUT 717;":MMEM:STOR:PROG ""D:Test1_01.vba"""

Related commands

:MMEM:LOAD:PROG

Equivalent key

Macro Setup > **Save VBA Project**

:MMEMory:TRANsfer

No equivalent COM Commands

Syntax

:MMEMory:TRANsfer <string>,<block>

:MMEMory:TRANsfer? <string>

Description

This command sets/gets data to/from a file on the built-in storage device of the E5061B. By reading out data with this command and writing it to a file on the external controller, file transfer from the E5061B to the external controller can be realized.

On the other hand, by reading out data from the external controller and writing it to a file on the E5061B with this command, file transfer from the external controller to the E5061B can be realized.

When you use directory names and file names, separate them with "/" (slash) or "\" (backslash). If a file with the specified file name already exists for writing or if the specified file does not exist for reading out (Query), an error occurs and the command is ignored.

Variables

Parameter	*String*
Description	File name on the built-in storage device
Range	254 characters or less

Parameter	*block*
Description	Data written on/read out from the file
Range	GPIB: 20 Mbytes or less LAN: 100 Kbytes or less

Query response

{block}<newline><^END>

Example of use

```
10 OUTPUT 717;":MMEM:TRAN ""Trace01.csv"",#6012345";Dat$
```

```
10 OUTPUT 717;":MMEM:TRAN? ""Trace01.csv"""
20 ENTER 717 USING "#,A";A$
30 ENTER 717 USING "#,A";Digit$
40 Img$="#,"&Digit$&"A"
50 ENTER 717 USING Img$;Byte$
60 Img$=Byte$&"A"
70 ALLOCATE Dat$[VAL(Byte$)]
80 ENTER 717 USING Img$;Dat$
```

Equivalent key

No equivalent key is available on the front panel.

SCPI.MMEMory.CATalog(*Dir*)

Object type

Property (**Read Only**)

Syntax

Cont = SCPI.MMEMory.CATalog(*Dir*)

Description

This command reads the following information on the built-in storage device of the E5061B:

- Space in use
- Available space
- Name and size of all files (including directories) in the specified directory

NOTE To read out the information in the root directory (folder), specify "\ " (backslash). Separate between directory names (file name) with "\ " (back slash), or "/" (slash).

Variable

Parameter	*Cont*
Description	Directory information ("{A},{B},{Name 1},,{Size 1},{Name 2},,{Size 2}, ... ,{Name N},,{Size N}") Where N is the number of all files in the specified directory and n is an integer between 1 and N. {A}: Space in use of the built-in storage device (byte). {B}: Available space of the built-in storage device (byte). {Name n}: Name of the n-th file (directory). {Size n}: Size (byte) of the n-th file (directory). Always 0 for directories.
Data type	Character string type (String)

Parameter	*Dir*

Description	Directory name of which the information you want to read out
Data type	Character string type (String)
Range	254 characters or less

Examples

```
Dim DirCont As String
DirCont = SCPI.MMEMory.CATalog("d:\")
```

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

:MMEMory:CATalog? <string 1>

Query response

{string 2}<newline><^END>

The format of the readout character string is as follows:

"{used_size},{free_size},{name 1},,{size 1}, ... ,{name N},,{size N}"

Where:

N is the number of all the files in the specified directory and n is an integer between 1 and N.

{used_size}:

Space in use of the built-in storage device (byte).

{free_size}:

Available space of the built-in storage device (byte).

{name n}:

Name of the n-th file (directory).

{size n}:

Size (byte) of the n-th file (directory). Always 0 for directories.

Example of use

```
10 DIM A$[1000]
20 OUTPUT 717;":MMEM:CAT? ""\"""
30 ENTER 717;A$
```

SCPI.MMEMory.COPY

Object type

Method (**Write Only**)

Syntax

SCPI.MMEMory.COPY = *File*

Description

This command copies a file.

NOTE Specify the file name with the extension. When you use directory names (folder names) and file name, separate them with "\" (back slash), or "/" (slash).

Variable

Parameter	*File*
Description	Indicates 2 file names (copy source and copy destination). *File(0):* Copy source file name *File(1):* Copy destination file name The index of the array starts from 0.
Data type	Variant type (Variant)
Range	254 characters or less
Note	If the specified copy source file does not exist, an error occurs when executed and the object is ignored. Notice that, if a file with the same name as the specified copy destination file name exists, its contents are overwritten.

Examples

```
SCPI.MMEMory.COPY = Array("test/state01.sta","d:test01.sta")
```

```
Dim File(1) As Variant
File(0) = "test/state01.sta"
File(1) = "d:test01.sta"
SCPI.MMEMory.COPY = File
```

Equivalent key

Save/Recall > **Save State** > **File Dialog...**

Equivalent SCPI command

Syntax

:MMEMory:COPY <string 1>,<string 2>

Example of use

```
10 OUTPUT 717;":MMEM:COPY ""Test1/State01.sta"",""d:Test1_01.sta"""
```

SCPI.MMEMory.DELete

Object type

Method (**Write Only**)

Syntax

SCPI.MMEMory.DELete = *File*

Description

This command deletes an existing file or directory (folder).

NOTE When you delete a directory, all the files and directories in it are deleted.

Specify the file name with the extension.

When you specify a file (directory) under an existing directory, separate them with "\" (back slash), or "/" (slash).

To delete all the files in the directory (folder), specify "\" (backslash).

Variable

Parameter	*File*
Description	File name or directory name you want to delete
Data type	Character string type (String)
Range	254 characters or less
Note	If the specified file or directory does not exist, an error occurs when executed and the object is ignored.

Examples

```
SCPI.MMEMory.DELete = "d:\"
SCPI.MMEMory.DELete = "test/state01.sta"
```

Equivalent key

Save/Recall > **Save State** > **File Dialog...**

Equivalent SCPI command

Syntax

:MMEMory:DELete <string>

Example of use

```
10 OUTPUT 717;":MMEM:DEL ""Test1/State01.sta"""
10 OUTPUT 717;":MMEM:DEL ""D:State01.sta"""
```

SCPI.MMEMory.LOAD.CHANnel.COEFficient

Object Type

Method (**Write Only**)

Syntax

SCPI.MMEMory.LOAD.CHANnel.COEFficient = *Param*

Description

This command recalls the channel coefficient data of the active channel from the specified register.

Variable

Parameter	*Param*
Description	Register
Data Type	Character string type (String)
Range	Select from the following: • "A": Specifies register A. • "B": Specifies register B. • "C" :Specifies register C. • "D" :Specifies register D.
Note	If no channel state has been saved in the specified register, an error occurs and the object is ignored.

Examples

```
Dim Rgstr as String
Rgstr = "A"
SCPI.MMEMory.LOAD.CHANnel.COEFficient = Rgstr
```

Related Objects

SCPI.MMEMory.LOAD.CHANnel.STATe

SCPI.MMEMory.STORe.CHANnel.COEFficient

SCPI.MMEMory.STORe.CHANnel.STATe

Equivalent Key

No equivalent key is available on the front panel.

Equivalent SCPI Command

Syntax

:MMEMory:LOAD:CHANnel:COEFficient {A|B|C|D}

Query Response

{A|B|C|D} <newline><^END>

Example of use

```
10 OUTPUT 717;":MMEM:LOAD:CHAN:COEF B"
```

SCPI.MMEMory.LOAD.CHANnel.STATe

Object type

Method (**Write Only**)

Syntax

SCPI.MMEMory.LOAD.CHANnel.STATe = *Register*

Description

This command recalls the instrument state for an individual channel (saved with the SCPI.MMEMory.STORe.CHANnel.STATe object) from the specified register as the setting of the active channel.

Variable

Parameter	*Register*
Description	Register
Data type	Character string type (String)
Range	Select from the following: "A": Specifies register A. "B": Specifies register B. "C" :Specifies register C. "D" :Specifies register D.
Note	If no instrument state has been saved in the specified register, an error occurs and the object is ignored.

Examples

```
SCPI.MMEMory.LOAD.CHANnel.STATe = "a"
```

Related objects

SCPI.MMEMory.STORe.CHANnel.STATe

SCPI.DISPlay.WINDow(Ch).ACTivate

Equivalent key

Save/Recall > **Recall Channel** > **A|B|C|D**

Equivalent SCPI command

Syntax

:MMEMory:LOAD:CHANnel[:STATe] {A|B|C|D}

Example of use

```
10 OUTPUT 717;":MMEM:LOAD:CHAN A"
```

SCPI.MMEMory.LOAD.LIMit

Object type

Method (**Write Only**)

Syntax

SCPI.MMEMory.LOAD.LIMit = *File*

Description

This command recalls the specified limit table file (file with the .csv extension saved with SCPI.MMEMory.STORe.LIMit), from the limit table for the active trace of the active channel.

NOTE Specify the file name with the extension. When you use directory names and file name, separate them with "\" (back slash), or "/" (slash).

Variable

Parameter	*File*
Description	File name of limit table (extension ".csv")
Data type	Character string type (String)
Range	254 characters or less
Note	If the specified file does not exist, an error occurs when executed and the object is ignored.

Examples

```
SCPI.DISPlay.WINDow(1).ACTivate
SCPI.CALCulate(1).PARameter(1).SELect
SCPI.MMEMory.LOAD.LIMit = "d:\limit01.csv"
```

```
SCPI.DISPlay.WINDow(1).ACTivate
SCPI.CALCulate(1).PARameter(1).SELect
SCPI.MMEMory.LOAD.LIMit = "test/limit01.csv"
```

Related objects

SCPI.DISPlay.WINDow(Ch).ACTivate

SCPI.CALCulate(Ch).PARameter(Tr).SELect

SCPI.MMEMory.STORe.LIMit

Equivalent key

Analysis > **Limit Test** > **Edit Limit Line** > **Import from CSV File**

Equivalent SCPI command

Syntax

:MMEMory:LOAD:LIMit <string>

Example of use

```
10 OUTPUT 717;":MMEM:LOAD:LIM ""Test1/Limit01.csv"""
10 OUTPUT 717;":MMEM:LOAD:LIM ""D:Limit01.csv"""
```

SCPI.MMEMory.LOAD.RLIMit

Object type

Method (**Write Only**)

Syntax

SCPI.MMEMory.LOAD.RLIMit = File

Description

This command recalls the specified ripple limit table file (file with the .csv extension saved with SCPI.MMEMory.STORe.RLIMit) of the active channel (specified with the SCPI.DISPlay.WINDow(Ch).ACTivate command), as ripple limit table for the active trace (specified with the SCPI.CALCulate(Ch).PARameter(Tr).SELect).

NOTE Specify the file name with the extension. When you write directory names and file name, separate them with "/" (slash) or "\" (backslash).
If the specified file does not exist, an error occurs and the command is ignored.

Variable

Parameter	*File*
Description	File name of the ripple limit table (extension ".csv")
Data type	Character string type (String)
Range	254 characters or less
Note	If the specified file does not exist, an error occurs when executed and the object is ignored.

Examples

```
SCPI.DISPlay.WINDow(1).ACTive
SCPI.CALCulate(1).PARameter(1).SELect
SCPI.MMEMory.LOAD.RIMit = "D:\Rlimit01.csv"
```

```
SCPI.DISPlay.WINDow(1).ACTive
SCPI.CALCulate(1).PARameter(1).SELect
SCPI.MMEMory.LOAD.RLIMit = "test/Rlimit01.csv"
```

Related objects

SCPI.DISPlay.WINDow(Ch).ACTivate

SCPI.CALCulate(Ch).PARameter(Tr).SELect

SCPI.MMEMory.STORe.RLIMit

Equivalent key

Analysis > **Ripple Limit** > **Edit Ripple Line** > **Import from CSV File**

Equivalent SCPI command

Syntax

:MMEMory:LOAD:RLIMit <string>

Example of use

```
10 OUTPUT 717;":MMEM:LOAD:RLIM ""RTest1/Rlim01.csv"""
10 OUTPUT 717;":MMEM:LOAD:RLIM ""D:Rlim01.csv"""
```

SCPI.MMEMory.LOAD.SEGMent

Object type

Method (**Write Only**)

Syntax

SCPI.MMEMory.LOAD.SEGMent = *File*

Description

This command recalls the specified segment sweep table file (file with a .csv extension saved with the SCPI.MMEMory.STORe.SEGMent), as the segment sweep table of the active channel.

NOTE Specify the file name with the extension. When you use directory names and file name, separate them with "\" (back slash), or "/" (slash).

Variable

Parameter	*File*
Description	File name of segment sweep table (extension ".csv")
Data type	Character string type (String)
Range	254 characters or less
Note	If the specified file does not exist, an error occurs when executed and the object is ignored.

Examples

```
SCPI.DISPlay.WINDow(1).ACTivate
SCPI.MMEMory.LOAD.SEGMent = "d:\segm01.csv"
```

```
SCPI.DISPlay.WINDow(1).ACTivate
SCPI.MMEMory.LOAD.SEGMent = "test/segm01.csv"
```

Related objects

SCPI.DISPlay.WINDow(Ch).ACTivate

SCPI.MMEMory.STORe.SEGMent

Equivalent key

Sweep Setup > **Edit Segment Table** > **Import from CSV File**

Equivalent SCPI command

Syntax

:MMEMory:LOAD:SEGMent <string>

Example of use

```
10 OUTPUT 717;":MMEM:LOAD:SEGM ""Test1/Segm01.csv"""
10 OUTPUT 717;":MMEM:LOAD:SEGM ""D:Segm01.csv"""
```

SCPI.MMEMory.LOAD.STATe

Object type

Method (**Write Only**)

Syntax

SCPI.MMEMory.LOAD.STATe = *File*

Description

This command recalls the specified instrument state file (file with a .sta extension saved with SCPI.MMEMory.STORe.STATe).

NOTE Specify the file name with the extension. When you use directory names and file name, separate them with "\" (back slash), or "/" (slash).

Variable

Parameter	*File*
Description	File name of instrument state (extension ".sta")
Data type	Character string type (String)
Range	254 characters or less
Note	If the specified file does not exist, an error occurs when executed and the object is ignored.

Examples

SCPI.MMEMory.LOAD.STATe = "d:\state01.sta"

SCPI.MMEMory.LOAD.STATe = "test/state01.sta"

Related objects

SCPI.MMEMory.STORe.STATe

Equivalent key

Save/Recall > **Recall State** > **StateX** (X=01 to 08) | **AutoRec** | **File Dialog...** | **UserPres**

Equivalent SCPI command

Syntax

:MMEMory:LOAD[:STATe] <string>

Example of use

```
10 OUTPUT 717;":MMEM:LOAD ""Test1/State01.sta"""
10 OUTPUT 717;":MMEM:LOAD ""d:State01.sta"""
```

SCPI.MMEMory.MDIRectory

Object type

Method (**Write Only**)

Syntax

SCPI.MMEMory.MDIRectory = *File*

Description

This command creates a new directory (folder).

NOTE When you create a directory under an existing directory, separate between the directory names with "\" (back slash), or "/" (slash).

Variable

Parameter	*File*
Description	Directory name you want to create
Data type	Character string type (String)
Range	254 characters or less
Note	If a directory with the same name as the specified directory name exists, an error occurs when executed and the object is ignored.

Examples

```
SCPI.MMEMory.MDIRectory = "d:\test"
SCPI.MMEMory.MDIRectory = "test"
```

Equivalent key

Save/Recall > **Save State** > **File Dialog...**

Equivalent SCPI command

Syntax

:MMEMory:MDIRectory <string>

Example of use

```
10 OUTPUT 717;":MMEM:MDIR ""Test1"""
10 OUTPUT 717;":MMEM:MDIR ""d:Test1"""
```

SCPI.MMEMory.STORe.CHANnel.CLEar

Object type

Method (**Write Only**)

Syntax

SCPI.MMEMory.STORe.CHANnel.CLEar

Description

This command deletes the instrument state for each channel (saved with the SCPI.MMEMory.STORe.CHANnel.STATe object) in all the registers.

Examples

SCPI.MMEMory.STORe.CHANnel.CLEar

Related objects

SCPI.MMEMory.STORe.CHANnel.STATe

Equivalent key

Save/Recall > **Save Channel** > **Clear States** > **OK**

Equivalent SCPI command

Syntax

:MMEMory:STORe:CHANnel:CLEar

Example of use

10 OUTPUT 717;":MMEM:STOR:CHAN:CLE"

SCPI.MMEMory.STORe.CHANnel.COEFficient

Object Type

Method (**Write Only**)

Syntax

SCPI.MMEMory.STORe.CHANnel.COEFficient = *Param*

Description

This command saves the channel coefficient data of the active channel specific to that channel only into the specified register (volatile memory).

Variable

Parameter	*Param*
Description	Register
Data Type	Character string type (String)
Range	Select from the following: • "A": Specifies register A. • "B": Specifies register B. • "C" :Specifies register C. • "D" :Specifies register D.
Note	If an instrument state has been already saved in the specified register, its contents are overwritten.

Examples

```
Dim Rgstr as String
Rgstr = "B"
SCPI.MMEMory.STORe.CHANnel.COEFficient = Rgstr
```

Related Objects

SCPI.MMEMory.LOAD.CHANnel.COEFficient

SCPI.MMEMory.LOAD.CHANnel.STATe

SCPI.MMEMory.STORe.CHANnel.STATe

Equivalent Key

Save/Recall > **Save Channel** > **Cal Only A** , **Cal Only B** , **Cal Only C** , **Cal Only D**

Equivalent SCPI Command

Syntax

:MMEMory:STORe:CHANnel:COEFficient {A|B|C|D}

Query Response

{A|B|C|D} <newline><^END>

Example of use

```
10 OUTPUT 717;":MMEM:STOR:CHAN:COEF C"
```

SCPI.MMEMory.STORe.CHANnel.STATe

Object type

Method (**Write Only**)

Syntax

SCPI.MMEMory.STORe.CHANnel.STATe = *Register*

Description

This command saves the instrument state of the items set for the active channel specific to that channel only into the specified register (volatile memory).

Variable

Parameter	*Register*
Description	Register
Data type	Character string type (String)
Range	Select from the following: "A": Specifies register A. "B": Specifies register B. "C": Specifies register C. "D": Specifies register D.
Note	If an instrument state has been already saved in the specified register, its contents are overwritten.

Examples

SCPI.MMEMory.STORe.CHANnel.STATe = "a"

Related objects

SCPI.MMEMory.LOAD.CHANnel.STATe

SCPI.DISPlay.WINDow(Ch).ACTivate

Equivalent key

Save/Recall > **Save Channel** > **A|B|C|D**

Equivalent SCPI command

Syntax

:MMEMory:STORe:CHANnel[:STATe] {A|B|C|D}

Example of use

```
10 OUTPUT 717;":MMEM:STOR:CHAN A"
```

SCPI.MMEMory.STORe.EPARameters

Object Type

Method (**Write Only**)

Syntax

SCPI.MMEMory.STORe.EPARameters = *Value*

Description

This command saves the equivalent circuit parameters of an active channel into a CSV file at defined location.

Variable

Parameter	*Value*
Description	Filename and destination
Data Type	Character string type (String)
Range	254 chars
Preset Value	-
Unit	-
Resolution	-

Examples

```
Dim Var as String
Var = "C:\sample.csv"
SCPI.MMEMory.STORe.EPARameters = Var
```

Equivalent Key

There is no equivalent key available on the front panel. However, similar key which exports to TXT file is shown below:

Analysis > **Equivalent Circuit** > **Export to TXT File...**

Equivalent SCPI Command

Syntax

:MMEMory:STORe:EPARameters

Example of use

```
10 OUTPUT 717;":MMEM:STOR:EPAR Sample_Text"
```

SCPI.MMEMory.STORe.FDATa

Object type

Method (**Write Only**)

Syntax

SCPI.MMEMory.STORe.FDATa = *File*

Description

This command saves the formatted data array into a file in the CSV format (extension ".csv"), for the active trace of the active channel.

NOTE Specify the file name with the extension. When you use directory names and file name, separate them with "\" (back slash), or "/" (slash).

Variable

Parameter	*File*
Description	File name in which you want to save the formatted data array (extension ".csv")
Data type	Character string type (String)
Range	254 characters or less
Note	If a file with the same name as the specified file name exists, its contents are overwritten.

Examples

```
SCPI.DISPlay.WINDow(1).ACTivate
SCPI.CALCulate(1).PARameter(1).SELect
SCPI.MMEMory.STORe.FDATa = "d:\trace01.csv"
```

```
SCPI.DISPlay.WINDow(1).ACTivate
SCPI.CALCulate(1).PARameter(1).SELect
SCPI.MMEMory.STORe.FDATa = "test/trace01.csv"
```

Related objects

SCPI.DISPlay.WINDow(Ch).ACTivate

SCPI.CALCulate(Ch).PARameter(Tr).SELect

Equivalent key

Save/Recall > **Save Trace Data**

Equivalent SCPI command

Syntax

:MMEMory:STORe:FDATa <string>

Example of use

```
10 OUTPUT 717;":MMEM:STOR:FDAT ""Result/Trace01.csv"""
10 OUTPUT 717;":MMEM:STOR:FDAT ""D:Trace01.csv"""
```

SCPI.MMEMory.STORe.IMAGe

Object type

Method (**Write Only**)

Syntax

SCPI.MMEMory.STORe.IMAGe = *File*

Description

This command saves the display image on the LCD display at the execution of the object into a file in the bitmap (extension ".bmp") or portable network graphics (extension ".png") format.

NOTE When saving the E5061B measurement screen, execute the VBA program with the Visual Basic editor closed. For more information, see Running a Program from the E5061B Measurement Screen.

Specify the file name with the extension. When you use directory names and file name, separate them with "\" (back slash), or "/" (slash).

Variable

Parameter	*File*
Description	File name in which you want to save the display image on the LCD display (extension ".bmp" or ".png")
Data type	Character string type (String)
Range	254 characters or less
Note	If a file with the same name as the specified file name exists, its contents are overwritten.

Examples

```
SCPI.MMEMory.STORe.IMAGe = "d:\image01.bmp"
SCPI.MMEMory.STORe.IMAGe = "test/image01.png"
```

Equivalent key

System > **Dump Screen Image**

When performing the operation from the front panel, the image on the LCD display memorized in the volatile memory (clipboard) (the image on the LCD display when the **Capture** (**System**) key is pressed) is saved.

If no image is memorized in the clipboard, in the same way as the SCPI.MMEMory.STORe.IMAGe object, the image on the LCD display at the execution is memorized in the clipboard and then it is saved.

Equivalent SCPI command

Syntax

:MMEMory:STORe:IMAGe <string>

Example of use

```
10 OUTPUT 717;":MMEM:STOR:IMAG ""Result/Image01.bmp"""
10 OUTPUT 717;":MMEM:STOR:IMAG ""D:Image01.png"""
```

SCPI.MMEMory.STORe.LIMit

Object type

Method (**Write Only**)

Syntax

SCPI.MMEMory.STORe.LIMit = *File*

Description

This command saves the limit table of the active trace of the active channel into a file in the CSV format (extension ".csv").

NOTE Specify the file name with the extension. When you use directory names and file name, separate them with "\ " (back slash), or "/" (slash).

Variable

Parameter	*File*
Description	File name to save the limit table (extension ".csv")
Data type	Character string type (String)
Range	254 characters or less
Note	If a file with the same name as the specified file name exists, its contents are overwritten.

Examples

```
SCPI.DISPlay.WINDow(1).ACTivate
SCPI.CALCulate(1).PARameter(1).SELect
SCPI.MMEMory.STORe.LIMit = "d:\limit01.csv"
```

```
SCPI.DISPlay.WINDow(1).ACTivate
SCPI.CALCulate(1).PARameter(1).SELect
SCPI.MMEMory.STORe.LIMit = "test/limit01.csv"
```

Related objects

SCPI.DISPlay.WINDow(Ch).ACTivate

SCPI.CALCulate(Ch).PARameter(Tr).SELect

SCPI.MMEMory.LOAD.LIMit

Equivalent key

Analysis > **Limit Test** > **Edit Limit Line** > **Export to CSV File**

Equivalent SCPI command

Syntax

```
:MMEMory:STORe:LIMit <string>
```

Example of use

```
10 OUTPUT 717;":MMEM:STOR:LIM ""Test1/Limit01.csv"""
10 OUTPUT 717;":MMEM:STOR:LIM ""D:Limit01.csv"""
```

SCPI.MMEMory.STORe.RLIMit

Object type

Method (**Write Only**)

Syntax

SCPI.MMEMory.STORe.RLIMit = File

Description

This command saves the ripple limit table of the active trace (specified with the SCPI.CALCulate(Ch).PARameter(Tr).SELect command) of the active channel (specified with the SCPI.DISPlay.WINDow(Ch).ACTivate command) into a file in the CSV format.

NOTE Specify the file name with the .sta extension. When you write directory names and file name, separate them with "/" (slash) or "\" (backslash).

If a file with the specified file name already exists, its contents are overwritten.

Variable

Parameter	*File*
Description	File name used to save the ripple limit table (extension ".csv")
Data type	Character string type (String)
Range	254 characters or less
Note	If the specified file does not exist, a runtime error occurs.

Examples (1)

```
SCPI.DISPlay.WINDow(1).ACTivate
SCPI.CALCulate(1).PARameter(1).SELect
SCPI.MMEMory.STORe.RLIMit = "D:\Rlimit01.csv"
```

Examples (2)

```
SCPI.DISPlay.WINDow(1).ACTivate
SCPI.CALCulate(1).PARameter(1).SELect
SCPI.MMEMory.STORe.RLIMit = "test/Rlimit01.csv"
```

Related objects

SCPI.DISPlay.WINDow(Ch).ACTivate

SCPI.CALCulate(Ch).PARameter(Tr).SELect

SCPI.MMEMory.LOAD.RLIMit

Equivalent key

Analysis > **Ripple Limit** > **Edit Ripple Line** > **Export to CSV File**

Equivalent SCPI command

Syntax

:MMEMory:STORe:RLIMit <string>

Example of use

```
10 OUTPUT 717;":MMEM:STOR:RLIM ""RTest1/Rlim01.csv"""
10 OUTPUT 717;":MMEM:STOR:RLIM ""D:Rlim01.csv"""
```

SCPI.MMEMory.STORe.SALL

Object type

Property (**Read-Write**)

Syntax

SCPI.MMEMory.STORe.SALL = *Status*

Status = SCPI.MMEMory.STORe.SALL

Description

This command sets/gets whether to save the settings of all channels/traces or that of the displayed channels/traces only, as the instrument state to be saved.

Variable

Parameter	*Status*
Description	Selecting content to be saved as the instrument state setting.
Data type	Boolean type (Boolean)
Range	Select from the following: • True or ON: Specifies the setting of all channels/traces as the target to be saved. • False or OFF: Specifies the setting of displayed channels/traces only as the target to be saved.
Preset value	False or OFF

Examples

```
Dim Obj As Boolean
SCPI.MMEMory.STORe.SALL = True
Obj = SCPI.MMEMory.STORe.SALL
```

Related objects

SCPI.MMEMory.STORe.STATe

Equivalent key

Save/Recall > **Channel/Trace** > **Disp Only** | **ALL**

Equivalent SCPI command

Syntax

:MMEMory:STORe:SALL {ON|OFF|1|0}

:MMEMory:STORe:SALL?

Query response

{1|0}<newline><^END>

Example of use

```
10 OUTPUT 717;":MMEM:STOR:SALL ON"
20 OUTPUT 717;":MMEM:STOR:SALL?"
30 ENTER 717;A
```

SCPI.MMEMory.STORe.SEGMent

Object type

Method (**Write Only**)

Syntax

SCPI.MMEMory.STORe.SEGMent = *File*

Description

This command saves the segment sweep table of the active channel into a file in the CSV format (extension ".csv").

NOTE Specify the file name with the extension. When you use directory names and file name, separate them with "\" (back slash), or "/" (slash).

Variable

Parameter	*File*
Description	File name to save segment sweep table (extension ".csv")
Data type	Character string type (String)
Range	254 characters or less
Note	If a file with the same name as the specified file name exists, its contents are overwritten.

Examples

```
SCPI.DISPlay.WINDow(1).ACTivate
SCPI.MMEMory.STORe.SEGMent = "d:\segm01.csv"

SCPI.DISPlay.WINDow(1).ACTivate
SCPI.MMEMory.STORe.SEGMent = "test/segm01.csv"
```

Related objects

SCPI.DISPlay.WINDow(Ch).ACTivate

SCPI.MMEMory.LOAD.SEGMent

Equivalent key

Sweep Setup > **Edit Segment Table** > **Export to CSV File**

Equivalent SCPI command

Syntax

:MMEMory:STORe:SEGMent <string>

Example of use

```
10 OUTPUT 717;":MMEM:STOR:SEGM ""Test1/Segm01.csv"""
10 OUTPUT 717;":MMEM:STOR:SEGM ""D:Segm01.csv"""
```

SCPI.MMEMory.STORe.SNP.DATA

Object type

Method (**Write Only**)

Syntax

SCPI.MMEMory.STORe.SNP.DATA = *File*

Description

Saves the measurement data for the active channel (specified with the SCPI.DISPlay.WINDow(Ch).ACTivate command) into a file in the touchstone format.

You need to specify a file format and file type before saving a file. The extension differs depending on file types as shown in the following table:

<file type>	<extension>
When specifying one port	s1p
When specifying two ports	s2p

NOTE When you use directory names and file name, separate them with "/" (slash) or "\" (back slash).

If a file with the specified file name already exists, its contents are overwritten.

Variable

Parameter	*File*
Description	File name you want to use when saving file in the touchstone format
Range	254 characters or less

NOTE When invalid extension is specified, an error message appears and the command is ignored.

Examples

```
Dim SnpPorts(1) As Variant
SCPI.DISPlay.WINDow(1).ACTivate
SCPI.MMEMory.STORe.SNP.FORMat = "RI"
SnpPorts(0) = 1
```

```
SnpPorts(1) = 2
SCPI.MMEMory.STORe.SNP.TYPE.S2P = SnpPorts
SCPI.MMEMory.STORe.SNP.DATA = "SNP01.s2p"
```

Related objects

SCPI.DISPlay.WINDow(Ch).ACTivate

SCPI.MMEMory.STORe.SNP.FORMat

SCPI.MMEMory.STORe.SNP.TYPE.S1P

SCPI.MMEMory.STORe.SNP.TYPE.S2P

Equivalent key

After a file type is specified, a dialog box will appear.

Equivalent SCPI command

Syntax

```
:MMEMory:STORe:SNP[:DATA] <string>
```

Example of use

```
10 OUTPUT 717;"DISP:WIND1:ACT"
20 OUTPUT 717;":MMEM:STOR:SNP:FORM RI"
30 OUTPUT 717;":MMEM:STOR:SNP:TYPE:S2P 1,2"
40 OUTPUT 717;":MMEM:STOR:SNP ""SNP01.s2p"""
```

SCPI.MMEMory.STORe.SNP.FORMat

Object type

Property (**Read-Write**)

Syntax

SCPI.MMEMory.STORe.SNP.FORMat = *Param*

Param = SCPI.MMEMory.STORe.SNP.FORMat

Description

This command sets/gets the data format for saving measurement data for the active channel (specified with SCPI.DISPlay.WINDow(Ch).ACTivate command) into a file in the touchstone format.

Variable

Parameter	*Param*
Description	Touchstone file format
Data type	Character string type (String)
Range	Select from the following: "AUTO": Specifies data format automatically according to the display format of the active trace. "MA": Specifies data format “log magnitude > angle”. "DB": Specifies data format “linear magnitude > angle”. "RI": Specifies data format “real part > imaginary part”.
Preset value	"AUTO"

Examples

```
Dim Fmt As String
SCPI.MMEMory.STORe.SNP.FORMat = "MA"
Fmt = SCPI.MMEMory.STORe.SNP.FORMat
```

Related objects

SCPI.DISPlay.WINDow(Ch).ACTivate

SCPI.MMEMory.STORe.SNP.DATA

Equivalent key

Save/Recall > **Save Snp** > **Snp Format** > **AUTO|LogMag/Angle|LinMag/Angle|Real/Imaginary**

Equivalent SCPI command

Syntax

```
:MMEMory:STORe:SNP:FORMat {AUTO|MA|DB|RI}
:MMEMory:STORe:SNP:FORMat?
```

Query response

```
{AUTO|MA|DB|RI}<newline><^END>
```

Example of use

```
10 OUTPUT 717;":MMEM:STOR:SNP:FORM MA"
20 OUTPUT 717;":MMEM:STOR:SNP:FORM?"
30 ENTER 717;A$
```

SCPI.MMEMory.STORe.SNP.TYPE.S1P

Object type

Property (**Read-Write**)

Syntax

SCPI.MMEMory.STORe.SNP.TYPE.S1P = *Port*

Port = SCPI.MMEMory.STORe.SNP.TYPE.S1P

Description

This command sets/gets the specified port to the file type (1 port) when saving measurement data for the active channel (specified with SCPI.DISPlay.WINDow(Ch).ACTivate command) into a file in the touchstone format.

Variable

Parameter	*Port*
Description	Port number
Range	1 to 2
Resolution	1

Examples

```
10 OUTPUT 717;":MMEM:STOR:SNP:TYPE:S1P 2"
20 OUTPUT 717;":MMEM:STOR:SNP:TYPE:S1P?"
30 ENTER 717;A$
```

Related objects

SCPI.DISPlay.WINDow(Ch).ACTivate

SCPI.MMEMory.STORe.SNP.DATA

SCPI.MMEMory.STORe.SNP.FORMat

Equivalent key

Save/Recall > **Save SnP** > **S1P** > **1|2**

Equivalent SCPI command

Syntax

:MMEMory:STORe:SNP:TYPE:S1P <numeric>

:MMEMory:STORe:SNP:TYPE:S1P?

Example of use

```
10 OUTPUT 717;":MMEM:STOR:SNP:TYPE:S1P 2"
20 OUTPUT 717;":MMEM:STOR:SNP:TYPE:S1P?"
30 ENTER 717;A$
```

SCPI.MMEMory.STORe.SNP.TYPE.S2P

Object type

Property (**Read-Write**)

Syntax

SCPI.MMEMory.STORe.SNP.TYPE.S2P = *Ports*

Ports = SCPI.MMEMory.STORe.SNP.TYPE.S2P

Description

This command sets/gets the specified port to the file type (2 port) when saving measurement data for the active channel (specified with SCPI.DISPlay.WINDow(Ch).ACTivate command) into a file in the touchstone format.

Variable

Parameter	*Ports*
Description	Indicates 2-element array data (port number). • *Ports(0):* Specifies a port for file type. • *Ports(1):* Specifies the other port for file type. The index of the array starts from 0.
Data type	Variant type (Variant)
Range	1 to 2
Resolution	1
Note	If the specified variable is out of the allowable setup range, an error occurs when it is executed. If you specify the same port number to two ports, an error occurs during execution. The order of the two port numbers to be specified is arbitrary.

Examples

```
Dim Ports(1) As Long
Ports(0) = 1
Ports(1) = 2
SCPI.MMEMory.STORe.SNP.TYPE.S2P = Ports
Ports = SCPI.MMEMory.STORe.SNP.TYPE.S2P
```

Related objects

SCPI.DISPlay.WINDow(Ch).ACTivate

SCPI.MMEMory.STORe.SNP.DATA

SCPI.MMEMory.STORe.SNP.FORMat

Equivalent key

Save/Recall > **Save Snp** > **S2p** > **1-2**

Equivalent SCPI command

Syntax

:MMEMory:STORe:SNP:TYPE:S2P <numeric1>, <numeric 2>

:MMEMory:STORe:SNP:TYPE:S2P?

Example of use

```
10 OUTPUT 717;":MMEM:STOR:SNP:TYPE:S2P 1,2"
20 OUTPUT 717;":MMEM:STOR:SNP:TYPE:S2P?"
30 ENTER 717;A$
```

SCPI.MMEMory.STORe.STATe

Object type

Method (**Write Only**)

Syntax

SCPI.MMEMory.STORe.STATe = *File*

Description

This command saves the instrument state (contents to be saved specified with the SCPI.MMEMory.STORe.STYPe object) into a file (file with the .sta extension).

NOTE Specify the file name with the extension. When you use directory names and file name, separate them with "\" (back slash), or "/" (slash).

Variable

Parameter	*File*
Description	File name to save the instrument state (extension ".sta")
Data type	Character string type (String)
Range	254 characters or less
Note	If a file with the same name as the specified file name exists, its contents are overwritten.

Examples

```
Dim StaType As String
SCPI.MMEMory.STORe.STYPe = "cdst"
SCPI.MMEMory.STORe.STATe = "d:\state01.sta"
```

```
Dim StaType As String
SCPI.MMEMory.STORe.STYPe = "cdst"
SCPI.MMEMory.STORe.STATe = "test/state01.sta"
```

Related objects

SCPI.MMEMory.STORe.STYPe

SCPI.MMEMory.LOAD.STATe

Equivalent key

Save/Recall > **Save State** > **StateX** (X=01 to 08) | **AutoRec** | **File Dialog...** | **UserPres**

Equivalent SCPI command

Syntax

```
:MMEMory:STORe[:STATe] <string>
```

Example of use

```
10 OUTPUT 717;":MMEM:STOR ""Test1/State01.sta"""
10 OUTPUT 717;":MMEM:STOR ""D:State01.sta"""
```

SCPI.MMEMory.STORe.STYPe

Object type

Property (**Read-Write**)

Syntax

SCPI.MMEMory.STORe.STYPe = *Param*

Param = SCPI.MMEMory.STORe.STYPe

Description

Selects the contents saved when saving the instrument state into a file with the SCPI.MMEMory.STORe.STATe object.

Variable

Parameter	*Param*
Description	Data of instrument state
Data type	Character string type (String)
Range	Select from the following. • "STATe": Specifies the save of the measurement conditions only. • "CSTate":Specifies the save of the measurement conditions and the calibration state. • "DSTate": Specifies the save of the measurement conditions and the formatted data array. • "CDSTate": Specifies the save of the measurement conditions, the calibration state, and the formatted data array.
Preset value	"CSTate"

Examples

```
Dim StaType As String
SCPI.MMEMory.STORe.STYPe = "cdst"
StaType = SCPI.MMEMory.STORe.STYPe
```

Related objects

SCPI.MMEMory.STORe.STATe

Equivalent key

Save/Recall > Save Type > State Only|State & Cal|State & Trace|All

Equivalent SCPI command

Syntax

:MMEMory:STORe:STYPe {STATe|CSTate|DSTate|CDSTate}

:MMEMory:STORe:STYPe?

Query response

{STAT|CST|DST|CDST}<newline><^END>

Example of use

```
10 OUTPUT 717;":MMEM:STOR:STYP CDST"
20 OUTPUT 717;":MMEM:STOR:STYP?"
30 ENTER 717;A$
```

OUTPUT

SCPI.OUTPut.COUPling

Object Type

Property (**Read-Write**)

Syntax

SCPI.OUTPut.COUPling = *Param*

Param = SCPI.OUTPut.COUPling

Description

This command sets/gets the port coupling type whether the signal is AC or DC to the output port.

Variable

Parameter	*Param*
Description	Port coupling type
Data Type	Character string type (String)
Range	Select from the following: • "AC": Specifies the AC coupling. • "DC": Specifies the DC coupling.
Preset Value	"DC"

Examples

```
Dim PtType as String
SCPI.OUTPut.COUPling = "DC"
PtType = SCPI.OUTPut.COUPling
```

Related Objects

Equivalent Key

System > **S-Param Port Couple** > **AC|DC**

Equivalent SCPI Command

Syntax

:OUTPut:COUPling {AC|DC}

:OUTPut:COUPling?

Query Response

{AC|DC} <newline><^END>

Example of use

```
10 OUTPUT 717;":OUTP:COUP DC"
20 OUTPUT 717;":OUTP:COUP?"
30 ENTER 717;A$
```

SCPI.OUTPut.STATe

Object type

Property (**Read-Write**)

Syntax

SCPI.OUTPut.STATe = *Status*

Status = SCPI.OUTPut.STATe

Description

This command turns ON/OFF of the stimulus signal output. Measurement cannot be made until the stimulus signal output is turned ON.

Variable

Parameter	*Status*
Description	ON/OFF of the stimulus signal output
Data type	Boolean type (Boolean)
Range	Select from the following: • True or ON: Turns on the stimulus signal. • False or OFF: Turns off the stimulus signal.
Preset value	True or ON

Examples

```
Dim Outp As Boolean
SCPI.OUTPut.STATe = True
Outp = SCPI.OUTPut.STATe
```

Equivalent key

Sweep Setup > **Power** > **RF Out**

System > **Overload Recovery** > **RF Out**

Equivalent SCPI command

Syntax

:OUTPut[:STATe] {ON|OFF|1|0}

:OUTPut[:STATe]?

Query response

```
{1|0}<newline><^END>
```

Example of use

```
10 OUTPUT 717;":OUTP ON"
20 OUTPUT 717;":OUTP?"
30 ENTER 717;A
```

PROGRAM

:PROGram:CATalog?

No equivalent COM Commands

Syntax

:PROGram:CATalog?

Description

This command reads the list of all executable VBA macros (procedures defined by Public including the VBA project loaded on the VBA editor).

Query response

{string}<newline><^END>

The character string in the following format, in which each macro is separated by a comma (.), is read out.

"{macro 1},{macro 2}, ... ,{macro N}"

Where N is the total number of VBA macros.

{macro n}: VBA macro name (module name.procedure name)

Example of use

```
10 DIM A$[1000]
20 OUTPUT 717;":PROG:CAT?"
30 ENTER 717;A$
```

:PROGram[:SELected]:NAME

No equivalent COM Commands

Syntax

:PROGram[:SELected]:NAME <string>

:PROGram[:SELected]:NAME?

Description

This command sets/gets the VBA macro controlled with the :PROG:STAT command.

NOTE Selectable VBA macro names can be read with the :PROG:CAT? command.

Variable

Parameter	*String*
Description	VBA macro name (module name.procedure name)
Range	254 characters or less
Preset value	""

Query response

{string}<newline><^END>

Example of use

```
10 OUTPUT 717;":PROG:NAME ""Module1.main"""
20 OUTPUT 717;":PROG:NAME?"
30 ENTER 717;A$
```

Related commands

:PROG:CAT?

:PROG:STAT

Equivalent key

Macro Setup > **Select Macro**

NOTE When performing a similar operation from the front panel, you should select the VBA macro and execute it at the same.

:PROGram[:SELected]:STATe

No equivalent COM Commands

Syntax

:PROGram[:SELected]:STATe {STOP|RUN}

:PROGram[:SELected]:STATe?

Description

This command sets/gets the control/state of the VBA macro selected with the :PROG:STAT command.

Variable

Range	STOP: Stop the program. RUN: Run the Program.
Preset value	STOP

Query response

{STOP|RUN}<newline><^END>

Example of use

```
10 OUTPUT 717;":PROG:STAT RUN"
20 OUTPUT 717;":PROG:STAT?"
30 ENTER 717;A$
```

Related commands

:PROG:NAME

Equivalent key

Macro Break (to stop)

Macro Setup > **Select Macro** (to run)

NOTE When performing the operation from the front panel, you select the VBA macro and execute it at the same time.

SCPI.PROGram.VARiable.ARRay(*Vnum*).DATA

Object type

Property (**Read-Write**)

Syntax

SCPI.PROGram.VARiable.ARRay(*Vnum*).DATA = *Data*

Data = SCPI.PROGram.VARiable.ARRay(*Vnum*).DATA

Description

This command sets/gets an array of Data that can be exchanged between an external PC and E5061B built-in VBA using GPIB/LAN/USB.

Variable

Parameter	*Vnum*
Description	Variable Number
Range	1 to 10
Preset value	1

Parameter	*Data*
Description	"n" is the number obtained from the value specified with the SCPI.PROGram.VARiable.ARRay(Vnum).SIZE object. • Data(0): The first array data • Data(n-1): The n-th array data The index of the array starts from 0.
Data Type	Variant type (Variant)
Note	If the data size is not specified, an error occurs when executed.

Examples

```
Dim Var1(2) As Long
Dim Var2 as Variant
Dim ArraySize as Long
ArraySize=3
```

```
Var1(1) = 2
Var1(2) = 56
SCPI.PROGram.VARiable.ARRay(1).SIZE = ArraySize
SCPI.PROGram.VARiable.ARRay(1).DATA = Var1
Var2= SCPI.PROGram.VARiable.ARRay(1).DATA
```

Related objects

SCPI.PROGram.VARiable.ARRay(Vnum).SIZE

SCPI.PROGram.VARiable.DOUBle(Vnum).DATA

SCPI.PROGram.VARiable.LONG(Vnum).DATA

SCPI.PROGram.VARiable.STRing(Vnum).DATA

Equivalent key

None

Equivalent SCPI command

Syntax

:PROGram:VARiable:ARRay{[1] - 10}[:DATA] <Data Array>

:PROGram:VARiable:ARRay{[1] - 10}[:DATA]?

Example of use

```
10 OUTPUT 717;"PROG:VAR:ARR2:SIZE 4"
20 OUTPUT 717;"PROG:VAR:ARR2 1.0,2.0,3.0,4.0"
30 OUTPUT 717;"PROG:VAR:ARR2?"
40 ENTER 717;A(*)
```

SCPI.PROGram.VARiable.ARRay(*Vnum*).SIZE

Object type

Property (**Read-Write**)

Syntax

SCPI.PROGram.VARiable.ARRay(*Vnum*).SIZE = *Value*

Value = SCPI.PROGram.VARiable.ARRay(*Vnum*).SIZE

Description

This command sets/gets the size of an array of Data, specified by SCPI.PROGram.VARiable.ARRay(1-10).DATA, that can be exchanged between an external PC and E5061B built-in VBA using GPIB/LAN/USB.

Variable

Parameter	*Vnum*
Description	Variable Number
Range	1 to 10
Preset value	1

Parameter	*Value*
Description	The value of data size
Data Type	Long integer type (long)
Range	1 to 40002
Preset value	402
Resolution	1

Examples

```
Dim Var1(2) As Long
Dim Var2 as Variant
Dim ArraySize as Long
```

```
ArraySize=3
Var1(1) = 2
Var1(2) = 56
SCPI.PROGram.VARiable.ARRay(1).SIZE = ArraySize
SCPI.PROGram.VARiable.ARRay(1).DATA = Var1
Var2= SCPI.PROGram.VARiable.ARRay(1).DATA
```

Related objects

SCPI.PROGram.VARiable.ARRay(Vnum).DATA

SCPI.PROGram.VARiable.DOUBle(Vnum).DATA

SCPI.PROGram.VARiable.LONG(Vnum).DATA

SCPI.PROGram.VARiable.STRing(Vnum).DATA

Equivalent key

None

Equivalent SCPI command

Syntax

:PROGram:VARiable:ARRay{[1] - 10}:SIZE <Data Array Size>

:PROGram:VARiable:ARRay{[1] - 10}:SIZE ?

Example of use

```
10 OUTPUT 717;"PROG:VAR:ARR2:SIZE 4"
20 OUTPUT 717;"PROG:VAR:ARR2 1.0,2.0,3.0,4.0"
30 OUTPUT 717;"PROG:VAR:ARR2?"
40 ENTER 717;A(*)
```

SCPI.PROGram.VARiable.DOUBle(*Vnum*).DATA

Object type

Property (**Read-Write**)

Syntax

SCPI.PROGram.VARiable.DOUBle(*Vnum*).DATA = *Value*

Value = SCPI.PROGram.VARiable.DOUBle(*Vnum*).DATA

Description

This command can be used to exchange Double type data between an external PC and E5061B built-in VBA using GPIB/LAN/USB.

Variable

Parameter	*Vnum*
Description	Variable Number
Range	1 to 10
Preset value	1

Parameter	*Value*
Description	The value of the double precision floating point type
Data type	Double precision floating point type (Double)
Range	Compliant with the double precision floating point type
Preset value	0

Examples

```
Dim DblVal As Double
DblVal =55.7890

SCPI.PROGram.VARiable.DOUBle(1).DATA = DblVal
DblVal= SCPI.PROGram.VARiable.DOUBle(1).DATA
```

Related objects

SCPI.PROGram.VARiable.ARRay(Vnum).DATA

SCPI.PROGram.VARiable.ARRay(Vnum).SIZE

SCPI.PROGram.VARiable.LONG(Vnum).DATA

SCPI.PROGram.VARiable.STRing(Vnum).DATA

Equivalent key

None

Equivalent SCPI command

Syntax

:PROGram:VARiable:DOUBle{[1]-10}[:DATA] <Numeric value, Double Type>

:PROGram:VARiable:DOUBle{[1]-10}[:DATA]?

Example of use

```
10 OUTPUT 717;"PROGram:VARiable:DOUBle1 12345.89607"
20 OUTPUT 717;":PROGram:VARiable:DOUBle1?"
30 ENTER 717;A$
```

SCPI.PROGram.VARiable.LONG(*Vnum*).DATA

Object type

Property (**Read-Write**)

Syntax

SCPI.PROGram.VARiable.LONG(*Vnum*).DATA = *Value*

Value = SCPI.PROGram.VARiable.LONG(*Vnum*).DATA

Description

This command can be used to exchange Long type data between an external PC and E5061B built-in VBA using GPIB/LAN/USB.

Variable

Parameter	*Vnum*
Description	Variable Number
Range	1 to 10
Preset value	1

Parameter	*Value*
Description	The value of the long integer type
Data type	Long integer type (Long)
Range	Compliant with the long integer type
Preset value	0

Examples

```
Dim LngVal As Long
LngVal =5512345
SCPI.PROGram.VARiable.LONG(1).DATA = LngVal
LngVal= SCPI.PROGram.VARiable.LONG(1).DATA
```

Related objects

SCPI.PROGram.VARiable.ARRay(Vnum).DATA

SCPI.PROGram.VARiable.ARRay(Vnum).SIZE

SCPI.PROGram.VARiable.DOUBle(Vnum).DATA

SCPI.PROGram.VARiable.STRing(Vnum).DATA

Equivalent key

None

Equivalent SCPI command

Syntax

:PROGram:VARiable:LONG{[1] - 10}:DATA <Numeric value, Long Type>

:PROGram:VARiable:LONG{[1] - 10}:DATA?

Example of use

```
10 OUTPUT 717;":PROGram:VARiable:LONG1:DATA 123459607"
20 OUTPUT 717;":PROGram:VARiable:LONG1:DATA?"
30 ENTER 717;A$
```

SCPI.PROGram.VARiable.STRing(*Vnum*).DATA

Object type

Property (**Read-Write**)

Syntax

SCPI.PROGram.VARiable.STRing(*Vnum*).DATA = *Value*

Value = SCPI.PROGram.VARiable.STRing(*Vnum*).DATA

Description

This command can be used to exchange String type data between an external PC and E5061B built-in VBA using GPIB/LAN/USB.

Variable

Parameter	*Vnum*
Description	Variable Number
Range	1 to 10
Preset value	1

Parameter	*Value*
Description	The value of the character string type
Data type	Character string type (String)
Range	Compliant with the character string type
Preset value	""

Examples

```
Dim StrVal As String
StrVal = "TestS11"

SCPI.PROGram.VARiable.STRing(1).DATA = StrVal
StrVal= SCPI.PROGram.VARiable.String(1).DATA
```

Related objects

SCPI.PROGram.VARiable.ARRay(Vnum).DATA

SCPI.PROGram.VARiable.ARRay(Vnum).SIZE

SCPI.PROGram.VARiable.DOUBle(Vnum).DATA

SCPI.PROGram.VARiable.LONG(Vnum).DATA

Equivalent key

None

Equivalent SCPI command

Syntax

:PROGram:VARiable:STRing{[1] - 10}[:DATA] <Character value, String Type>

:PROGram:VARiable:STRing{[1] - 10}[:DATA]?

Example of use

```
10 OUTPUT 717;" PROG:VAR:STR ""TEST DATA"""
20 OUTPUT 717;" PROG:VAR:STR?"
30 ENTER 717;A$
```

SENSE

SCPI.SENSe(*Ch*).AVERage.CLEar

Object type

Method (**Write Only**)

Syntax

SCPI.SENSe(*Ch*).AVERage.CLEar

Description

This command resets the data count to 0, it is used for averaging the selected channel (*Ch*). Measurement data before the execution of this object is not used for averaging.

Examples

```
SCPI.SENSe(1).AVERage.CLEar
```

Related objects

SCPI.SENSe(Ch).AVERage.COUNt

SCPI.SENSe(Ch).AVERage.STATe

Equivalent key

Avg > Averaging Restart

Equivalent SCPI command

Syntax

:SENSe{[1]-4}:AVERage:CLEar

Example of use

```
10 OUTPUT 717;":SENS1:AVER:CLE"
```

SCPI.SENSe(*Ch*).AVERage.COUNt

Object type

Property (**Read-Write**)

Syntax

SCPI.SENSe(*Ch*).AVERage.COUNt = *Value*

Value = SCPI.SENSe(*Ch*).AVERage.COUNt

Description

This command sets/gets the averaging factor of the selected channel (*Ch*).

Variable

Parameter	*Value*
Description	Averaging factor
Data type	Long integer type (Long)
Range	1 to 999
Preset value	16
Note	If the specified variable 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.

Examples

```
Dim AvgCnt As Long
SCPI.SENSe(1).AVERage.COUNt = 4
AvgCnt = SCPI.SENSe(1).AVERage.COUNt
```

Related objects

SCPI.SENSe(Ch).AVERage.STATe

SCPI.SENSe(Ch).AVERage.CLEar

Equivalent key

Avg > **Avg Factor**

Equivalent SCPI command

Syntax

:SENSe{[1]-4}:AVERage:COUNt <numeric>

```
:SENSe{[1]-4}:AVERage:COUNt?
```

Query response

```
{numeric}<newline><^END>
```

Example of use

```
10 OUTPUT 717;":SENS1:AVER:COUN 4"
20 OUTPUT 717;":SENS1:AVER:COUN?"
30 ENTER 717;A
```

SCPI.SENSe(*Ch*).AVERage.STATe

Object type

Property (**Read-Write**)

Syntax

SCPI.SENSe(*Ch*).AVERage.STATe = *Status*

Status = SCPI.SENSe(*Ch*).AVERage.STATe

Description

This command sets/gets the averaging function of the selected channel (*Ch*).

Variable

Parameter	*Status*
Description	ON/OFF of the averaging function
Data type	Boolean type (Boolean)
Range	Select from the following: • True or ON(1): Turns ON • False or OFF(0): Turns OFF
Preset value	False or OFF

Examples

```
Dim Avg As Boolean
SCPI.SENSe(1).AVERage.STATe = True
Avg = SCPI.SENSe(1).AVERage.STATe
```

Related objects

SCPI.SENSe(Ch).AVERage.COUNt

SCPI.SENSe(Ch).AVERage.CLEar

Equivalent key

Avg > **Averaging**

Equivalent SCPI command

Syntax

:SENSe{[1]-4}:AVERage[:STATe] {ON|OFF|1|0}

:SENSe{[1]-4}:AVERage[:STATe]?

Query response

{1|0}<newline><^END>

Example of use

```
10 OUTPUT 717;":SENS1:AVER ON"
20 OUTPUT 717;":SENS1:AVER?"
30 ENTER 717;A
```

SCPI.SENSe(*Ch*).BANDwidth.RESolution

Object type

Property (**Read-Write**)

Syntax

SCPI.SENSe(*Ch*).BANDwidth.RESolution = *Value*

Value = SCPI.SENSe(*Ch*).BANDwidth.RESolution

Description

This command sets/gets the IF bandwidth of the selected channel (*Ch*).

NOTE This command is similar to SCPI.SENSe(Ch).BWIDth.RESolution.

Variable

Parameter	*Value*
Description	IF bandwidth
Data type	Double precision floating point type (Double)
Range	1 to 300000 (1|1.5|2|3|4|5|7|10|15|20|30|40|50|70|100|150|200|300|400|500|700|1k|1.5k|2k|3k|4k|5k|7k|10k|15k|20k|30k|40k|50k|70k|100k|150k|200k|300k)
Preset value	30000
Unit	Hz (hertz)
Resolution	Refer the Range.
Note	If the specified variable 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.

Examples

```
Dim IfBw As Double
SCPI.SENSe(1).BANDwidth.RESolution = 1.5E3
IfBw = SCPI.SENSe(1).BANDwidth.RESolution
```

Related objects

SCPI.SENSe(Ch).BWIDth.RESolution

Equivalent key

Avg > **IF Bandwidth**

Equivalent SCPI command

Syntax

```
:SENSe{[1]-4}:BANDwidth[:RESolution] <numeric>
:SENSe{[1]-4}:BANDwidth[:RESolution]?
```

Query response

```
{numeric}<newline><^END>
```

Example of use

```
10 OUTPUT 717;":SENS1:BAND 1.5E3"
20 OUTPUT 717;":SENS1:BAND?"
30 ENTER 717;A
```

SCPI.SENSe(*Ch*).BWAuto.LIMit.RESolution

Object Type

Property (**Read-Write**)

Syntax

SCPI.SENSe(*Ch*).BWAuto.LIMit.RESolution = *Value*

Value = SCPI.SENSe(*Ch*).BWAuto.LIMit.RESolution

Description

This command sets/gets maximum IF bandwidth in bandwidth auto mode, for the selected channel (*Ch*).

Variable

Parameter	*Value*
Description	maximum IF bandwidth in bandwidth auto mode
Data Type	Double precision floating point type (Double)
Range	1 to 3E+5 (1|1.5|2|3|4|5|7|10|15|20|30|40|50|70|100|150|200|300|400|500|700|1k|1.5k|2k|3k|4k|5k|7k|10k|15k|20k|30k|40k|50k|70k|100k|150k|200k|300k)
Preset Value	3E+4
Unit	Hz

Examples

```
Dim BWALim as Double
SCPI.SENSe(4).BWAuto.LIMit.RESolution = 3E+4
BWALim = SCPI.SENSe(4).BWAuto.LIMit.RESolution
```

Related Objects

SCPI.SENSe(Ch).BWAuto.STATus

Equivalent Key

Avg > **IFBW Auto Limit**

Equivalent SCPI Command

Syntax

:SENSe{[1]-4}:BWAuto:LIMit[:RESolution] <numeric>

:SENSe{[1]-4}:BWAuto:LIMit[:RESolution]?

Query Response

{numeric} <newline><^END>

Example of use

```
10 OUTPUT 717;":SENS1:BWA:LIM 1E+5"
20 OUTPUT 717;":SENS1:BWA:LIM?"
30 ENTER 717;A
```

SCPI.SENSe(*Ch*).BWAuto.STATus

Object Type

Property (**Read-Write**)

Syntax

SCPI.SENSe(*Ch*).BWAuto.STATus = *Status*

Status = SCPI.SENSe(*Ch*).BWAuto.STATus

Description

This command turns ON/OFF the IF bandwidth auto function, for the selected channel (*Ch*).

Variable

Parameter	*Status*
Description	ON/OFF the IF bandwidth auto function
Data Type	Boolean type (Boolean)
Range	Select from either of the following: • True or ON(1): Turns ON • False or OFF(0): Turns OFF
Preset Value	False or OFF

Examples

```
Dim StatBWA as Boolean
SCPI.SENSe(4).BWAuto.STATus = True
StatBWA = SCPI.SENSe(4).BWAuto.STATus
```

Related Objects

SCPI.SENSe(Ch).BWAuto.LIMit.RESolution

Equivalent Key

Avg > **IFBW Auto**

Equivalent SCPI Command

Syntax

:SENSe{[1]-4}:BWAuto[:STATus] {ON|OFF|1|0}

:SENSe{[1]-4}:BWAuto[:STATus]?

Query Response

{1|0} <newline><^END>

Example of use

```
10 OUTPUT 717;":SENS1:BWA OFF"
20 OUTPUT 717;":SENS1:BWA?"
30 ENTER 717;A
```

SCPI.SENSe(*Ch*).BWIDth.RESolution

Object type

Property (**Read-Write**)

Syntax

SCPI.SENSe(*Ch*).BWIDth.RESolution = *Value*

Value = SCPI.SENSe(*Ch*).BWIDth.RESolution

Description

This command sets/gets the IF bandwidth of the selected channel (*Ch*).

NOTE This command is similar to SCPI.SENSe(Ch).BANDwidth.RESolution.

Variable

Parameter	*Value*
Description	IF bandwidth
Data type	Double precision floating point type (Double)
Range	1 to 300000 (1|1.5|2|3|4|5|7|10|15|20|30|40|50|70|100|150|200|300|400|500|700|1k|1.5k|2k|3k|4k|5k|7k|10k|15k|20k|30k|40k|50k|70k|100k|150k|200k|300k)
Preset value	30000
Unit	Hz (hertz)
Resolution	Refer the Range.
Note	If the specified variable 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.

Examples

```
Dim IfBw As Double
SCPI.SENSe(1).BWIDth.RESolution = 1.5E3
IfBw = SCPI.SENSe(1).BWIDth.RESolution
```

Related objects

SCPI.SENSe(Ch).BANDwidth.RESolution

Equivalent key

Avg > **IF Bandwidth**

Equivalent SCPI command

Syntax

:SENSe{[1]-4}:BWIDth[:RESolution] <numeric>

:SENSe{[1]-4}:BWIDth[:RESolution]?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":SENS1:BWID 1.5E3"
20 OUTPUT 717;":SENS1:BWID?"
30 ENTER 717;A
```

SCPI.SENSe(*Ch*).CORRection.CLEar

Type of object

Method (**Write Only**)

Syntax

SCPI.SENSe(*Ch*).CORRection.CLEar

Description

This command clears the error coefficient for calibration when the frequency offset feature is set to OFF, for the selected channel (*Ch*).

This command also clears the fixture compensation data when option 005 is installed.

Example of use

SCPI.SENSe(1).CORRection.CLEar

Related objects

SCPI.SENSe(Ch).OFFSet.STATe

SCPI.SENSe(Ch).CORRection.OFFSet.CLEar

Equivalent key

Cal > **Clear** > **OK**

Equivalent SCPI command

Syntax

:SENSe{[1]-4}:CORRection:CLEar

Example of use

10 OUTPUT 717;":SENS1:CORR:CLE"

SCPI.SENSe(*Ch*).CORRection.COEFficient.DATA

Object type

Property (**Read-Write**)

Syntax

SCPI.SENSe(*Ch*).CORRection.COEFficient.DATA (*Str*, *Int1*, *Int2*) = *Array*

Array = SCPI.SENSe(*Ch*).CORRection.COEFficient.DATA (*Str*, *Int1*, *Int2*)

Description

This command sets/gets the calibration coefficient data for the specified channel.

> NOTE When the calibration factor is interpolated, the interpolated calibration coefficient array is read. Similarly, when the calibration factor is not interpolated, a non-interpolated calibration coefficient array is read.
>
> After writing the calibration coefficient array, the written value becomes effective only after the (SCPI.SENSe(Ch).CORRection.COEFficient.SAVE) command is executed.

Variable

Parameter	*Array*
Description	Indicates the array data (corrected data array) of NOP (number of measurement points)×2. Where n is an integer between 1 and NOP. *Data(n×2-2)* Real part of data (complex number) at the n-th measurement point. *Data(n×2-1)* Imaginary part of data (complex number) at the n-th measurement point. The index of the array starts from 0.
Data type	Variant type (Variant)

Parameter	*Str*
Description	Calibration type
Data type	Character string type (String)
Range	Select from the following: • "ES": Source match • "ER": Reflection tracking • "ED": Directivity • "EL": Load match • "ET": Transmission tracking • "EX": Isolation

Parameter	*Int1*
Description	Response port
Data type	Integer type (Integer)
Range	1 or 2
Note	If ES, ER, or ED is used, the response port and the stimulus port must be the same, while EL, ET, or EX is used, the response port and the stimulus port must be different.

Parameter	*Int2*

Description	Stimulus port
Data type	Integer type (Integer)
Range	1 or 2
Note	If ES, ER, or ED is used, the response port and the stimulus port must be the same, while EL, ET, or EX is used, the response port and the stimulus port must be different.

For information on the variable (*Ch*), see Ch

Examples

```
DIM CoefArray As Variant
CoefArray = SCPI.SENSe(1).CORRection.COEFficient.DATA("EL", 1, 2)


SCPI.SENSe(2).CORRection.COEFficient.DATA("EL", 1, 2) = CoefArray
SCPI.SENSe(2).CORRection.COEFficient.SAVE
```

Related objects

SCPI.SENSe(Ch).CORRection.COEFficient.METHod.ERESponse

SCPI.SENSe(Ch).CORRection.COEFficient.METHod.RESPonse.OPEN

SCPI.SENSe(Ch).CORRection.COEFficient.METHod.RESPonse.SHORt

SCPI.SENSe(Ch).CORRection.COEFficient.METHod.RESPonse.THRU

SCPI.SENSe(Ch).CORRection.COEFficient.METHod.SOLT1

SCPI.SENSe(Ch).CORRection.COEFficient.METHod.SOLT2

SCPI.SENSe(Ch).CORRection.COEFficient.SAVE

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

:SENSe{[1]-4}:CORRection:COEFficient[:DATA] {ES|ER|ED|EL|ET|EX},<numeric 1>, <numeric 2>, <numeric 3>,..., <numeric 3 n×2>

:SENSe{[1]-4}:CORRection:COEFficient[:DATA]? {ES|ER|ED|EL|ET|EX},<numeric 1>, <numeric 2>

Query response

{numeric 1}, ... ,{numeric NOP×2}<newline><^END>

	Description
{numeric n×2-1}	Real part of data (complex number) at the n-th measurement point.
{numeric n×2}	Imaginary part of data (complex number) at the n-th measurement point.

Because the calibration coefficient array is expressed by a complex number, the real part and the imaginary part of one measurement point are returned and obtained as a value. Here, NOP is the number of measurement points and n is an integer between 1 and NOP.

Example of use

```
10 DIM A(1:201)
20 OUTPUT 717;":SENS1:CORR:COEF? EL,1,2"
30 ENTER 717;A(*)
```

SCPI.SENSe(*Ch*).CORRection.COEFficient.GPData

Object Type

Property (**Read-Write**)

Syntax

SCPI.SENSe(*Ch*).CORRection.COEFficient.GPData(*Str*) = *Array*

Array = SCPI.SENSe(*Ch*).CORRection.COEFficient.GPData(*Str*)

Description

This command sets/gets the calibration coefficient data of Gain-Phase measurement, for the selected channel (*Ch*).

NOTE When the calibration factor is interpolated, the interpolated calibration coefficient array is read. Similarly, when the calibration factor is not interpolated, a non-interpolated calibration coefficient array is read.

After writing the calibration coefficient array, the written value becomes effective only after the (SCPI.SENSe(Ch).CORRection.COEFficient.SAVE) command is executed.

Variable

Parameter	*Array*
Description	Indicates the array data (corrected data array) of NOP (number of measurement points)×2. Where n is an integer between 1 and NOP. *Data(n×2-2)* Real part of data (complex number) at the n-th measurement point. *Data(n×2-1)* Imaginary part of data (complex number) at the n-th measurement point. The index of the array starts from 0.
Data type	Variant type (Variant)

Parameter	*Str*
Description	Calibration coefficient type
Data type	Character string type (String)
Range	Select from the following: • "ES": Source match • "ER": Reflection tracking • "ED": Directivity • "EL": Load match • "ET": Transmission tracking • "EX": Isolation

Examples

```
Dim GpCorr as Variant
GpCorr = SCPI.SENSe(1).CORRection.COEFficient.GPData("ES")

SCPI.SENSe(2).CORRection.COEFficient.GPData("ES") = GpCorr
SCPI.SENSe(2).CORRection.COEFficient.SAVE
```

Related Objects

SCPI.SENSe(Ch).CORRection.COEFficient.METHod.GPResponse.OPEN

SCPI.SENSe(Ch).CORRection.COEFficient.METHod.GPResponse.SHORt

SCPI.SENSe(Ch).CORRection.COEFficient.METHod.GPResponse.THRU

SCPI.SENSe(Ch).CORRection.COEFficient.METHod.GPS1

Equivalent Key

No equivalent key is available on the front panel.

Equivalent SCPI Command

Syntax

:SENSe{[1]-4}:CORRection:COEFficient:GPData {ES|ER|ED|EL|ET|EX}

:SENSe{[1]-4}:CORRection:COEFficient:GPData?

Query Response

{numeric 1}, ... ,{numeric NOP×2}<newline><^END>

	Description
{numeric n×2-1}	Real part of data (complex number) at the n-th measurement point.
{numeric n×2}	Imaginary part of data (complex number) at the n-th measurement point.

Example of use

```
10 DIM A(1:201)
20 OUTPUT 717;":SENS1:CORR:COEF:GPD? EL"
30 ENTER 717;A(*)
```

SCPI.SENSe(*Ch*).CORRection.COEFficient.METHod.ERESponse

Object type

Method (**Write Only**)

Syntax

SCPI.SENSe(*Ch*).CORRection.COEFficient.METHod.ERESponse = *Ports*

Description

This command sets the calibration type to the enhanced response calibration between the two specified ports when the calibration coefficient data array is written with the SCPI.SENSe(Ch).CORRection.COEFficient.DATA command, for the selected channel (*Ch*).

Variable

Parameter	*Ports*
Description	Indicates 2-element array data (port number). • *Ports(0):* Specifies the response port. • *Ports(1):* Specifies the stimulus port. The index of the array starts from 0.
Data type	Variant type (Variant)
Range	1 to 2
Resolution	1
Note	For each parameter, you must specify a different port number. If you specify the same port number for 2 or more parameters, an error occurs and the command is ignored.

Examples

```
Dim ERESPort(1) As Variant
ERESPort(0) = 1
ERESPort(1) = 2
SCPI.SENSe(1).CORRection.COEFficient.METHod.ERESponse = ERESPort
```

Related objects

SCPI.SENSe(Ch).CORRection.COEFficient.DATA

SCPI.SENSe(Ch).CORRection.COEFficient.SAVE

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

:SENSe{[1]-4}:CORRection:COEFficient:METHod:ERESponse <numeric 1>,<numeric 2>

Example of use

```
10 OUTPUT 717;":SENS1:CORR:COEF:METH:ERES 1,2"
```

SCPI.SENSe(*Ch*).CORRection.COEFficient.METHod.GPResponse.OPEN

Object Type

Method (**Write Only**)

Syntax

SCPI.SENSe(*Ch*).CORRection.COEFficient.METHod.GPResponse.OPEN

Description

This command sets the calibration type to the response calibration (open) of the gain phase port when the calibration coefficient data array is written with the SCPI.SENSe(Ch).CORRection.COEFficient.GPData command, for the selected channel (*Ch*).

Examples

SCPI.SENSe(2).CORRection.COEFficient.METHod.GPResponse.OPEN

Related Objects

SCPI.SENSe(Ch).CORRection.COEFficient.GPData

SCPI.SENSe(Ch).CORRection.COEFficient.METHod.GPResponse.SHORt

SCPI.SENSe(Ch).CORRection.COEFficient.METHod.GPResponse.THRU

SCPI.SENSe(Ch).CORRection.COEFficient.METHod.GPS1

Equivalent Key

No equivalent key is available on the front panel.

Equivalent SCPI Command

Syntax

:SENSe{[1]-4}:CORRection:COEFficient:METHod:GPResponse:OPEN

Example of use

10 OUTPUT 717;":SENS1:CORR:COEF:METH:GPR:OPEN"

SCPI.SENSe(*Ch*).CORRection.COEFficient.METHod.GPResponse.SHORt

Object Type

Method (**Write Only**)

Syntax

SCPI.SENSe(*Ch*).CORRection.COEFficient.METHod.GPResponse.SHORt

Description

This command sets the calibration type to the response calibration (short) of the gain phase port when the calibration coefficient data array is written with the SCPI.SENSe(Ch).CORRection.COEFficient.GPData command, for the selected channel (*Ch*).

Examples

SCPI.SENSe(1).CORRection.COEFficient.METHod.GPResponse.SHORt

Related Objects

SCPI.SENSe(Ch).CORRection.COEFficient.GPData

SCPI.SENSe(Ch).CORRection.COEFficient.METHod.GPResponse.OPEN

SCPI.SENSe(Ch).CORRection.COEFficient.METHod.GPResponse.THRU

SCPI.SENSe(Ch).CORRection.COEFficient.METHod.GPS1

Equivalent Key

No equivalent key is available on the front panel.

Equivalent SCPI Command

Syntax

:SENSe{[1]-4}:CORRection:COEFficient:METHod:GPResponse:SHORt

Example of use

10 OUTPUT 717;":SENS1:CORR:COEF:METH:GPR:SHOR"

SCPI.SENSe(*Ch*).CORRection.COEFficient.METHod.GPResponse.THRU

Object Type

Method (**Write Only**)

Syntax

SCPI.SENSe(*Ch*).CORRection.COEFficient.METHod.GPResponse.THRU

Description

This command sets the calibration type to the response calibration (thru) between the T-port and the R-port using gain phase measurement when the calibration coefficient data array is written with the SCPI.SENSe(Ch).CORRection.COEFficient.GPData command, for the selected channel (*Ch*).

Examples

```
SCPI.SENSe(1).CORRection.COEFficient.METHod.GPResponse.THRU
```

Related Objects

SCPI.SENSe(Ch).CORRection.COEFficient.GPData

SCPI.SENSe(Ch).CORRection.COEFficient.METHod.GPResponse.OPEN

SCPI.SENSe(Ch).CORRection.COEFficient.METHod.GPResponse.SHORt

SCPI.SENSe(Ch).CORRection.COEFficient.METHod.GPS1

Equivalent Key

No equivalent key is available on the front panel.

Equivalent SCPI Command

Syntax

:SENSe{[1]-4}:CORRection:COEFficient:METHod:GPResponse:THRU

Example of use

```
10 OUTPUT 717;":SENS1:CORR:COEF:METH:GPR:THRU"
```

SCPI.SENSe(*Ch*).CORRection.COEFficient.METHod.GPS1

Object Type

Method (**Write Only**)

Syntax

SCPI.SENSe(*Ch*).CORRection.COEFficient.METHod.GPS1

Description

This command sets the calibration type to the full 1-port SOLT calibration of the T-port using gain phase measurement, when the calibration coefficient data array is written with the SCPI.SENSe(Ch).CORRection.COEFficient.GPData command, for the selected channel (*Ch*).

Examples

SCPI.SENSe(1).CORRection.COEFficient.METHod.GPS1

Related Objects

SCPI.SENSe(Ch).CORRection.COEFficient.GPData

SCPI.SENSe(Ch).CORRection.COEFficient.METHod.GPResponse.OPEN

SCPI.SENSe(Ch).CORRection.COEFficient.METHod.GPResponse.SHORt

SCPI.SENSe(Ch).CORRection.COEFficient.METHod.GPResponse.THRU

Equivalent Key

No equivalent key is available on the front panel.

Equivalent SCPI Command

Syntax

:SENSe{[1]-4}:CORRection:COEFficient:METHod:GPS1

Example of use

10 OUTPUT 717;":SENS1:CORR:COEF:METH:GPS1"

SCPI.SENSe(*Ch*).CORRection.COEFficient.METHod.RESPonse.OPEN

Object type

Method (**Write Only**)

Syntax

SCPI.SENSe(*Ch*).CORRection.COEFficient.METHod.RESPonse.OPEN = *Port*

Description

This command sets the calibration type to the response calibration (open) of the specified port when the calibration coefficient data array is written with the SCPI.SENSe(Ch).CORRection.COEFficient.DATA command, for the selected channel (*Ch*).

Variable

See *Port*.

Examples

SCPI.SENSe(1).CORRection.COEFficient.METHod.RESPonse.OPEN = 1

Related objects

SCPI.SENSe(Ch).CORRection.COEFficient.DATA

SCPI.SENSe(Ch).CORRection.COEFficient.SAVE

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

:SENSe{[1]-4}:CORRection:COEFficient:METHod[:RESPonse]:OPEN <numeric>

Example of use

10 OUTPUT 717;":SENS1:CORR:COEF:METH:OPEN 1"

SCPI.SENSe(*Ch*).CORRection.COEFficient.METHod.RESPonse.SHORt

Object type

Method (**Write Only**)

Syntax

SCPI.SENSe(*Ch*).CORRection.COEFficient.METHod.RESPonse.SHORt = *Port*

Description

This command sets the calibration type to the response calibration (short) of the specified port for the selected channel, when the calibration coefficient data array is written with the SCPI.SENSe(Ch).CORRection.COEFficient.DATA command.

Variable

See *Port*.

Examples

SCPI.SENSe(1).CORRection.COEFficient.METHod.RESPonse.SHORt = 1

Related objects

SCPI.SENSe(Ch).CORRection.COEFficient.DATA

SCPI.SENSe(Ch).CORRection.COEFficient.SAVE

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

:SENSe{[1]-4}:CORRection:COEFficient:METHod[:RESPonse]:SHORt <numeric>

Example of use

10 OUTPUT 717;":SENS1:CORR:COEF:METH:SHOR 1"

SCPI.SENSe(*Ch*).CORRection.COEFficient.METHod.RESPonse.THRU

Object type

Method (**Write Only**)

Syntax

SCPI.SENSe(*Ch*).CORRection.COEFficient.METHod.RESPonse.THRU = *Ports*

Description

This command sets the calibration type to the response calibration (thru) between the two specified ports when the calibration coefficient data array is written with the SCPI.SENSe(Ch).CORRection.COEFficient.DATA command, for the selected channel (*Ch*)

Variable

Parameter	*Ports*
Description	Indicates 2-element array data (port number). • *Ports(0):* Specifies the response port. • *Ports(1):* Specifies the stimulus port. The index of the array starts from 0.
Data type	Variant type (Variant)
Range	1 to 2
Resolution	1
Note	For each parameter, you must specify a different port number. If you specify the same port number for 2 or more parameters, an error occurs and the command is ignored.

Examples (1)

```
SCPI.SENSe(1).CORRection.COEFficient.METHod.RESPonse.THRU = Array(2, 1)
```

Examples (2)

```
Dim ThruPort(1) As Variant
ThruPort(0) = 2
ThruPort(1) = 1
SCPI.SENSe(1).CORRection.COEFficient.METHod.RESPonse.THRU = ThruPort
```

Related objects

SCPI.SENSe(Ch).CORRection.COEFficient.DATA

SCPI.SENSe(Ch).CORRection.COEFficient.SAVE

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

:SENSe{[1]-4}:CORRection:COEFficient:METHod[:RESPonse]:THRU <numeric 1>,<numeric 2>

Example of use

```
10 OUTPUT 717;":SENS1:CORR:COEF:METH:THRU 2,1"
```

SCPI.SENSe(*Ch*).CORRection.COEFficient.METHod.SOLT1

Object type

Method (**Write Only**)

Syntax

SCPI.SENSe(*Ch*).CORRection.COEFficient.METHod.SOLT1 = *Port*

Description

This command sets the calibration type to the full 1-port calibration of the specified port, when the calibration coefficient data array is written with the SCPI.SENSe(Ch).CORRection.COEFficient.DATA command, for the selected channel (*Ch*).

Variable

See *Port*.

Examples

```
SCPI.SENSe(1).CORRection.COEFficient.METHod.SLOT1 = 1
```

Related objects

SCPI.SENSe(Ch).CORRection.COEFficient.DATA

SCPI.SENSe(Ch).CORRection.COEFficient.SAVE

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

:SENSe{[1]-4}:CORRection:COEFficient:METHod:SOLT1 <numeric>

Example of use

```
10 OUTPUT 717;":SENS1:CORR:COEF:METH:SOLT1 1"
```

SCPI.SENSe(*Ch*).CORRection.COEFficient.METHod.SOLT2

Object type

Method (**Write Only**)

Syntax

SCPI.SENSe(*Ch*).CORRection.COEFficient.METHod.SOLT2 = *Ports*

Description

This command sets the calibration type to full 2-port calibration between the two specified ports, when the calibration coefficient data array is written with the SCPI.SENSe(Ch).CORRection.COEFficient.DATA command for the selected channel (*Ch*).

Variable

Parameter	*Ports*
Description	Indicates 2-element array data (port number). • *Ports(0):* Specifies a port for full 2-port calibration. • *Ports(1):* Specifies the other port for full 2-port calibration. The index of the array starts from 0.
Data type	Variant type (Variant)
Range	1 to 2
Resolution	1
Note	For each parameter, you must specify a different port number. If you specify the same port number for 2 or more parameters, an error occurs and the command is ignored.

Examples (1)

```
SCPI.SENSe(1).CORRection.COEFficient.METHod.SOLT2 = Array(1, 2)
```

Examples (2)

```
Dim CalPort(1) As Variant
CalPort(0) = 1
```

```
CalPort(1) = 2
SCPI.SENSe(1).CORRection.COEFficient.METHod.SOLT2 = CalPort
```

Related objects

SCPI.SENSe(Ch).CORRection.COEFficient.DATA

SCPI.SENSe(Ch).CORRection.COEFficient.SAVE

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

:SENSe{[1]-4}:CORRection:COEFficient:METHod:SOLT2 <numeric 1>,<numeric 2>

Example of use

```
10 OUTPUT 717;":SENS1:CORR:COEF:METH:SOLT2 1,2"
```

SCPI.SENSe(*Ch*).CORRection.COEFficient.SAVE

Object type

Method (**Write Only**)

Syntax

SCPI.SENSe(*Ch*).CORRection.COEFficient.SAVE

Description

This command enables the calibration coefficients depending on the selected calibration type from the writing calibration data for the selected channel (*Ch*).

> **NOTE** Enabling the calibration coefficients clears all the calibration data regardless of whether the data are used for the calculation and also clears the calibration type selections.
>
> If you execute this command before all the calibration data needed for calculating the calibration coefficients are written, an error occurs and the command is ignored.

Examples

```
Dim Dmy As Long
Dim Data(3) as Variant
Data(0) = -1.123
Data(1) = 2.456
Data(2) = -2.249
Data(3) = 2.608
SCPI.SENSe(1).CORRection.COEFficient.METHod.RESPonse.THRU = Array(2, 1)
SCPI.SENSe(1).CORRection.COEFficient("ET", 2, 1) = Data
Dmy = SCPI.IEEE4882.OPC
SCPI.SENSe(1).CORRection.COEFficient.SAVE
```

Related objects

SCPI.SENSe(Ch).CORRection.COEFficient.DATA

SCPI.SENSe(Ch).CORRection.COEFficient.METHod.ERESponse

SCPI.SENSe(Ch).CORRection.COEFficient.METHod.RESPonse.OPEN

SCPI.SENSe(Ch).CORRection.COEFficient.METHod.RESPonse.SHORt

SCPI.SENSe(Ch).CORRection.COEFficient.METHod.RESPonse.THRU

SCPI.SENSe(Ch).CORRection.COEFficient.METHod.SOLT1

SCPI.SENSe(Ch).CORRection.COEFficient.METHod.SOLT2

Equivalent key

Cal > **Calibrate** > **Response (Open)** > **Done**

Cal > **Calibrate** > **Response (Short)** > **Done**

Cal > **Calibrate** > **Response (Thru)** > **Done**

Cal > **Calibrate** > **Enhanced Response** > **Done**

Cal > **Calibrate** > **1-Port Cal** > **Done**

Cal > **Calibrate** > **2-Port Cal** > **Done**

Equivalent SCPI command

Syntax

:SENSe{[1]-4}:CORRection:COEFficient:SAVE

Example of use

```
10 OUTPUT 717;":SENS1:CORR:COEF:SAVE"
```

SCPI.SENSe(*Ch*).CORRection.COLLect.ACQuire.ISOLation

Object type

Method (**Write Only**)

Syntax

SCPI.SENSe(*Ch*).CORRection.COLLect.ACQuire.ISOLation = *Ports*

Description

This command measures the calibration data of the isolation from the specified stimulus port to the specified response port, for the selected channel (*Ch*).

Variable

Parameter	*Ports*
Description	Indicates 2-element array data (port number). • *Ports(0):* Specifies the response port number. • *Ports(1):* Specifies the stimulus port number. The index of the array starts from 0.
Data type	Variant type (Variant)
Range	1 to 2
Resolution	1
Note	If the specified variable is out of the allowable setup range, an error occurs when executed. If you specify the same port number to 2 port numbers, an error occurs when executed.

Examples

```
Dim Dmy As Long
SCPI.SENSe(1).CORRection.COLLect.ACQuire.ISOLation = Array(1,2)
Dmy = SCPI.IEEE4882.OPC


Dim IsPort(1) As Variant
Dim Dmy As Long
IsPort(0) = 1
IsPort(1) = 2
```

```
SCPI.SENSe(1).CORRection.COLLect.ACQuire.ISOLation = IsPort
Dmy = SCPI.IEEE4882.OPC
```

Related objects

SCPI.IEEE4882.OPC

Equivalent key

Cal > **Calibrate** > **Response (Thru)** > **Isolation (Optional)**

Cal > **Calibrate** > **Enhanced Response** > **Isolation (Optional)**

Cal > **Calibrate** > **2-Port Cal** > **Isolation (Optional)** > **Port 1-2 Isol**

Equivalent SCPI command

Syntax

```
:SENSe{[1]-4}:CORRection:COLLect[:ACQuire]:ISOLation <numeric 1>,<numeric 2>
```

Example of use

```
10 OUTPUT 717;":SENS1:CORR:COLL:ISOL 1,2"
20 OUTPUT 717;"*OPC?"
30 ENTER 717;A
```

SCPI.SENSe(*Ch*).CORRection.COLLect.ACQuire.LOAD

Object type

Method (**Write Only**)

Syntax

SCPI.SENSe(*Ch*).CORRection.COLLect.ACQuire.LOAD = *Port*

Description

This command measures the calibration data of the load standard for the specified port, for the selected channel (*Ch*).

Variable

See *Port*.

Examples

```
Dim Dmy As Long
SCPI.SENSe(1).CORRection.COLLect.ACQuire.LOAD = 1
Dmy = SCPI.IEEE4882.OPC
```

Related objects

SCPI.IEEE4882.OPC

Equivalent key

Cal > **Calibrate** > **Response (Open)**|**Response (Short)** > **Load (Optional)**

Cal > **Calibrate** > **Enhanced Response** > **Load (Optional)**

Cal > **Calibrate** > **1-Port Cal** > **Load**

Cal > **Calibrate** > **2-Port Cal** > **Reflection** > **Port 1 Load**|**Port 2 Load**

Equivalent SCPI command

Syntax

:SENSe{[1]-4}:CORRection:COLLect[:ACQuire]:LOAD <numeric>

Example of use

```
10 OUTPUT 717;":SENS1:CORR:COLL:LOAD 1"
20 OUTPUT 717;"*OPC?"
30 ENTER 717;A
```

SCPI.SENSe(*Ch*).CORRection.COLLect.ACQuire.OPEN

Object type

Method (**Write Only**)

Syntax

SCPI.SENSe(*Ch*).CORRection.COLLect.ACQuire.OPEN = *Port*

Description

This command measures the calibration data of the open standard for the specified port, for the selected channel (*Ch*).

Variable

See *Port*.

Examples

```
Dim Dmy As Long
SCPI.SENSe(1).CORRection.COLLect.ACQuire.OPEN = 1
Dmy = SCPI.IEEE4882.OPC
```

Related objects

SCPI.IEEE4882.OPC

Equivalent key

Cal > **Calibrate** > **Response (Open)|Enhanced Response|1-Port Cal** > **Open**

Cal > **Calibrate** > **2-Port Cal** > **Reflection** > **Port 1 Open|Port 2 Open**

Equivalent SCPI command

Syntax

:SENSe{[1]-4}:CORRection:COLLect[:ACQuire]:OPEN <numeric>

Example of use

```
10 OUTPUT 717;":SENS1:CORR:COLL:OPEN 1"
20 OUTPUT 717;"*OPC?"
30 ENTER 717;A
```

SCPI.SENSe(*Ch*).CORRection.COLLect.ACQuire.SHORt

Object type

Method (**Write Only**)

Syntax

SCPI.SENSe(*Ch*).CORRection.COLLect.ACQuire.SHORt = *Port*

Description

This command measures the calibration data of the short standard for the specified port, for the selected channel (*Ch*).

Variable

See *Port*.

Examples

```
Dim Dmy As Long
SCPI.SENSe(1).CORRection.COLLect.ACQuire.SHORt = 1
Dmy = SCPI.IEEE4882.OPC
```

Related objects

SCPI.IEEE4882.OPC

Equivalent key

Cal > **Calibrate** > **Response (Short)|Enhanced Response|1-Port Cal** > **Short**

Cal > **Calibrate** > **2-Port Cal** > **Reflection** > **Port 1 Short|Port 2 Short**

Equivalent SCPI command

Syntax

:SENSe{[1]-4}:CORRection:COLLect[:ACQuire]:SHORt <numeric>

Example of use

```
10 OUTPUT 717;":SENS1:CORR:COLL:SHOR 1"
20 OUTPUT 717;"*OPC?"
30 ENTER 717;A
```

SCPI.SENSe(*Ch*).CORRection.COLLect.ACQuire.THRU

Object type

Method (**Write Only**)

Syntax

SCPI.SENSe(*Ch*).CORRection.COLLect.ACQuire.THRU = *Ports*

Description

This command measures the calibration data of the Thru standard from the specified stimulus port to the specified response port, for the selected channel (*Ch*).

Variable

Parameter	*Ports*
Description	Indicates 2-element array data (port number). • *Ports(0):* Specifies the response port. • *Ports(1):* Specifies the stimulus port. The index of the array starts from 0.
Data type	Variant type (Variant)
Range	1 to 2
Resolution	1
Note	For each parameter, you must specify a different port number. If you specify the same port number for 2 or more parameters, an error occurs and the command is ignored.

Examples

```
Dim Dmy As Long
SCPI.SENSe(1).CORRection.COLLect.ACQuire.THRU = Array(2,1)
Dmy = SCPI.IEEE4882.OPC
```

```
Dim ThruPort(1) As Variant
Dim Dmy As Long
ThruPort(0) = 2
```

```
ThruPort(1) = 1
SCPI.SENSe(1).CORRection.COLLect.ACQuire.THRU = ThruPort
Dmy = SCPI.IEEE4882.OPC
```

Related objects

SCPI.IEEE4882.OPC

Equivalent key

Cal > **Calibrate** > **Response (Thru)|Enhanced Response** > **Thru**

Cal > **Calibrate** > **2-Port Cal** > **Transmission** > **Port 1-2 Thru**

Equivalent SCPI command

Syntax

```
:SENSe{[1]-4}:CORRection:COLLect[:ACQuire]:THRU <numeric 1>,<numeric 2>
```

Example of use

```
10 OUTPUT 717;":SENS1:CORR:COLL:THRU 1,2"
20 OUTPUT 717;"*OPC?"
30 ENTER 717;A
```

SCPI.SENSe(*Ch*).CORRection.COLLect.ADAPter(*Pt*).LENGth

Object type

Property (**Read-Write**)

Syntax

SCPI.SENSe(*Ch*).CORRection.COLLect.ADAPter(*Pt*).LENGth=*length*

length=SCPI.SENSe(*Ch*).CORRection.COLLect.ADAPter(*Pt*).LENGth

Description

This command sets/displays the approximate length of the adaptor, for the selected channel (*Ch*) and for the selected port.

Variable

Parameter	*length*
Description	Adapter Length
Data type	Double precision floating point type (Double)
Range	-10 to +10
Unit	Second
Note	Adapter length is positive for adaptor removal and negative for adaptor insertion.

Examples

SCPI.SENSe(1).CORRection.COLLect.ADAPter(2).LENGth = 0.01

Related objects

SCPI.SENSe(Ch).CORRection.COLLect.METHod.ADAPter.REMoval

Equivalent key

Cal > **Calibrate** > **Adapter Removal** > **Port1|Port2** > **Coaxial Length**

Equivalent SCPI command

Syntax

:SENSe{[1]-4}:CORRection:COLLect:ADAPter{[1]-2}:LENGth <+ or -><value of length>

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":SENS1:CORR:COLL:ADAP2:LENG 0.01"
20 OUTPUT 717;":SENS1:CORR:COLL:ADAP2:LENG?"
30 ENTER 717;A$
```

SCPI.SENSe(*Ch*).CORRection.COLLect.ADAPter(*Pt*).ROTate

Object type

Method (**Write Only**)

Syntax

SCPI.SENSe(*Ch*).CORRection.COLLect.ADAPter(*Pt*).ROTate

Description

This command executes Adapter Removal/Insertion along with moving the phase of adapter (which is removed or inserted) to 180 degrees. This command is useful in cases where auto judgement of phase fails. This command can be executed several times while the calibration remains valid.

NOTE If user cannot execute this command, then "Execution error" is displayed.

Examples

SCPI.SENSe(Ch).CORRection.COLLect.ADAPter(Pt).ROTate

Related Object

SCPI.SENSe(Ch).CORRection.COLLect.ADAPter(Pt).WAVeguide.CUToff

SCPI.SENSe(Ch).CORRection.COLLect.ADAPter(Pt).WAVeguide.LENGth

Equivalent key

Cal > **Calibrate** > **Adapter Removal** > **Port1|Port2** > **Rotate Adapter**

Equivalent SCPI command

Syntax

:SENSe{[1]-4}:CORRection:COLLect:ADAPter{[1]-2}:ROTate

Example of use

```
10 OUTPUT 717;"SENS1:CORR:COLL:ADAP2:ROT"
```

SCPI.SENSe(*Ch*).CORRection.COLLect.CKIT.GPORder.LOAD

Object Type

Property (**Read-Write**)

Syntax

SCPI.SENSe(*Ch*).CORRection.COLLect.CKIT.GPORder.LOAD = *Value*

Value = SCPI.SENSe(*Ch*).CORRection.COLLect.CKIT.GPORder.LOAD

Description

This command sets/gets the standard number used for the load measurement of the gain phase port, for a calibration kit selected for the selected channel (*Ch*).

Variable

Parameter	*Value*
Description	Standard number
Data type	Long integer type (Long)
Range	0 to 21
Note	If the specified variable is out of the allowable setup range, an error occurs when executed.

Examples

```
Dim StanLoad as Long
SCPI.SENSe(1).CORRection.COLLect.CKIT.GPORder.LOAD = 3
StanLoad = SCPI.SENSe(1).CORRection.COLLect.CKIT.GPORder.LOAD
```

Related Objects

SCPI.SENSe(Ch).CORRection.COLLect.CKIT.GPORder.OPEN

SCPI.SENSe(Ch).CORRection.COLLect.CKIT.GPORder.SHORt

SCPI.SENSe(Ch).CORRection.COLLect.CKIT.GPORder.THRU

Equivalent Key

Cal > **Modify Cal Kit** > **Specify CLSs** > **Load** > **GP Port**

Equivalent SCPI Command

Syntax

:SENSe{[1]-4}:CORRection:COLLect:CKIT:GPORder:LOAD <numeric>

:SENSe{[1]-4}:CORRection:COLLect:CKIT:GPORder:LOAD?

Query Response

{numeric} <newline><^END>

Example of use

```
10 OUTPUT 717;":SENS1:CORR:COLL:CKIT:GPOR:LOAD 3"
20 OUTPUT 717;":SENS1:CORR:COLL:CKIT:GPOR:LOAD?"
30 ENTER 717;A
```

SCPI.SENSe(*Ch*).CORRection.COLLect.CKIT.GPORder.OPEN

Object Type

Property (**Read-Write**)

Syntax

SCPI.SENSe(*Ch*).CORRection.COLLect.CKIT.GPORder.OPEN = *Value*

Value = SCPI.SENSe(*Ch*).CORRection.COLLect.CKIT.GPORder.OPEN

Description

This command sets/gets the standard number used for the open measurement of the gain phase port, for a calibration kit selected for the selected channel (*Ch*).

Variable

Parameter	*Value*
Description	Standard number
Data type	Long integer type (Long)
Range	0 to 21
Note	If the specified variable is out of the allowable setup range, an error occurs when executed.

Examples

```
Dim StanOpen as Long
SCPI.SENSe(1).CORRection.COLLect.CKIT.GPORder.OPEN = 2
StanOpen = SCPI.SENSe(1).CORRection.COLLect.CKIT.GPORder.OPEN
```

Related Objects

SCPI.SENSe(Ch).CORRection.COLLect.CKIT.GPORder.LOAD

SCPI.SENSe(Ch).CORRection.COLLect.CKIT.GPORder.SHORt

SCPI.SENSe(Ch).CORRection.COLLect.CKIT.GPORder.THRU

Equivalent Key

Cal > **Modify Cal Kit** > **Specify CLSs** > **Open** > **GP Port**

Equivalent SCPI Command

Syntax

:SENSe{[1]-4}:CORRection:COLLect:CKIT:GPORder:OPEN <numeric>

:SENSe{[1]-4}:CORRection:COLLect:CKIT:GPORder:OPEN?

Query Response

{numeric} <newline><^END>

Example of use

```
10 OUTPUT 717;":SENS1:CORR:COLL:CKIT:GPOR:OPEN 2"
20 OUTPUT 717;":SENS1:CORR:COLL:CKIT:GPOR:OPEN?"
30 ENTER 717;A
```

SCPI.SENSe(*Ch*).CORRection.COLLect.CKIT.GPOrder.SHORt

Object Type

Property (**Read-Write**)

Syntax

SCPI.SENSe(*Ch*).CORRection.COLLect.CKIT.GPOrder.SHORt = *Value*

Value = SCPI.SENSe(*Ch*).CORRection.COLLect.CKIT.GPOrder.SHORt

Description

This command sets/gets the standard number used for the short measurement of the gain phase port, for a calibration kit selected for the selected channel (*Ch*).

Variable

Parameter	*Value*
Description	Standard number
Data type	Long integer type (Long)
Range	0 to 21
Note	If the specified variable is out of the allowable setup range, an error occurs when executed.

Examples

```
Dim StanShort as Long
SCPI.SENSe(1).CORRection.COLLect.CKIT.GPORder.SHORt = 1
StanShort = SCPI.SENSe(1).CORRection.COLLect.CKIT.GPORder.SHORt
```

Related Objects

SCPI.SENSe(Ch).CORRection.COLLect.CKIT.GPORder.LOAD

SCPI.SENSe(Ch).CORRection.COLLect.CKIT.GPORder.OPEN

SCPI.SENSe(Ch).CORRection.COLLect.CKIT.GPORder.THRU

Equivalent Key

Cal > **Modify Cal Kit** > **Specify CLSs** > **Short** > **GP Port**

Equivalent SCPI Command

Syntax

:SENSe{[1]-4}:CORRection:COLLect:CKIT:GPORder:SHORt <numeric>

:SENSe{[1]-4}:CORRection:COLLect:CKIT:GPORder:SHORt?

Query Response

{numeric} <newline><^END>

Example of use

```
10 OUTPUT 717;":SENS1:CORR:COLL:CKIT:GPOR:SHOR 1"
20 OUTPUT 717;":SENS1:CORR:COLL:CKIT:GPOR:SHOR?"
30 ENTER 717;A
```

SCPI.SENSe(*Ch*).CORRection.COLLect.CKIT.GPORder.THRU

Object Type

Property (**Read-Write**)

Syntax

SCPI.SENSe(*Ch*).CORRection.COLLect.CKIT.GPORder.THRU = *Value*

Value = SCPI.SENSe(*Ch*).CORRection.COLLect.CKIT.GPORder.THRU

Description

This command sets/gets the standard number used for the thru measurement of the gain phase ports, for a calibration kit selected for the selected channel (*Ch*).

Variable

Parameter	*Value*
Description	Standard number
Data type	Long integer type (Long)
Range	0 to 21
Note	If the specified variable is out of the allowable setup range, an error occurs when executed.

Examples

```
Dim StanThru as Long
SCPI.SENSe(2).CORRection.COLLect.CKIT.GPORder.THRU = 4
StanThru = SCPI.SENSe(2).CORRection.COLLect.CKIT.GPORder.THRU
```

Related Objects

SCPI.SENSe(Ch).CORRection.COLLect.CKIT.GPORder.LOAD

SCPI.SENSe(Ch).CORRection.COLLect.CKIT.GPORder.OPEN

SCPI.SENSe(Ch).CORRection.COLLect.CKIT.GPORder.SHORt

Equivalent Key

Cal > **Modify Cal Kit** > **Specify CLSs** > **Thru** > **GP Port**

Equivalent SCPI Command

Syntax

:SENSe{[1]-4}:CORRection:COLLect:CKIT:GPORder:THRU <numeric>

:SENSe{[1]-4}:CORRection:COLLect:CKIT:GPORder:THRU?

Query Response

{numeric} <newline><^END>

Example of use

```
10 OUTPUT 717;":SENS1:CORR:COLL:CKIT:GPOR:THRU 4"
20 OUTPUT 717;":SENS1:CORR:COLL:CKIT:GPOR:THRU?"
30 ENTER 717;A
```

SCPI.SENSe(*Ch*).CORRection.COLLect.CKIT.LABel

Object type

Property (**Read-Write**)

Syntax

SCPI.SENSe(*Ch*).CORRection.COLLect.CKIT.LABel = *Lbl*

Lbl = SCPI.SENSe(*Ch*).CORRection.COLLect.CKIT.LABel

Description

This command sets/gets the calibration kit name for the calibration kit selected for the selected channels (*Ch*).

Variable

Parameter	*Lbl*
Description	Calibration kit name
Data type	Character string type (String)
Range	254 characters or less
Preset value	Varies depending on the calibration kit number:

Examples

```
Dim CalLbl As String
SCPI.SENSe(1).CORRection.COLLect.CKIT.LABel = "User 1"
CalLbl = SCPI.SENSe(1).CORRection.COLLect.CKIT.LABel
```

Related objects

SCPI.SENSe(Ch).CORRection.COLLect.CKIT.SELect

Equivalent key

Cal > **Modify Cal Kit** > **Label Kit**

Equivalent SCPI command

Syntax

:SENSe{[1]-4}:CORRection:COLLect:CKIT:LABel <string>

:SENSe{[1]-4}:CORRection:COLLect:CKIT:LABel?

Query response

{string}<newline><^END>

Example of use

```
10 OUTPUT 717;":SENS1:CORR:COLL:CKIT:LAB ""MY_KIT"""
20 OUTPUT 717;":SENS1:CORR:COLL:CKIT:LAB?"
30 ENTER 717;A$
```

SCPI.SENSe(*Ch*).CORRection.COLLect.CKIT.ORDer.LOAD(*Cpt*)

Object type

Property (**Read-Write**)

Syntax

SCPI.SENSe(*Ch*).CORRection.COLLect.CKIT.ORDer.LOAD(*Cpt*) = *Value*

Value = SCPI.SENSe(*Ch*).CORRection.COLLect.CKIT.ORDer.LOAD(*Cpt*)

Description

This command sets/gets the standard used for the load measurement of the specified port (*Cpt*), for a calibration kit selected for the selected channel (*Ch*).

Variable

Parameter	*Value*
Description	Standard number
Data type	Long integer type (Long)
Range	0 to 21
Note	If the specified variable is out of the allowable setup range, an error occurs when executed.

Examples

```
Dim StanLoad As Long
SCPI.SENSe(1).CORRection.COLLect.CKIT.ORDer.LOAD(1) = 10
StanLoad = SCPI.SENSe(1).CORRection.COLLect.CKIT.ORDer.LOAD(1)
```

Related objects

SCPI.SENSe(Ch).CORRection.COLLect.CKIT.SELect

SCPI.SENSe(Ch).CORRection.COLLect.CKIT.ORDer.OPEN(Cpt)

SCPI.SENSe(Ch).CORRection.COLLect.CKIT.ORDer.SHORt(Cpt)

SCPI.SENSe(Ch).CORRection.COLLect.CKIT.ORDer.THRU(Cpt_m,Cpt_n)

Equivalent key

Cal > **Modify Cal Kit** > **Specify CLSs** > **Load** > **Set All|Port 1|Port 2**

Equivalent SCPI command

Syntax

:SENSe{[1]-4}:CORRection:COLLect:CKIT:ORDer:LOAD <numeric 1>,<numeric 2>

```
:SENSe{[1]-4}:CORRection:COLLect:CKIT:ORDer:LOAD? <numeric 1>
```

Query response

```
{numeric 2}<newline><^END>
```

Example of use

```
10 OUTPUT 717;":SENS1:CORR:COLL:CKIT:ORD:LOAD 1,9"
20 OUTPUT 717;":SENS1:CORR:COLL:CKIT:ORD:LOAD? 1"
30 ENTER 717;A
```

SCPI.SENSe(*Ch*).CORRection.COLLect.CKIT.ORDer.OPEN(*Cpt*)

Object type

Property (**Read-Write**)

Syntax

SCPI.SENSe(*Ch*).CORRection.COLLect.CKIT.ORDer.OPEN(*Cpt*) = *Value*

Value = SCPI.SENSe(*Ch*).CORRection.COLLect.CKIT.ORDer.OPEN(*Cpt*)

Description

This command sets/gets the standard used for the open measurement of the specified port (*Cpt*), for a calibration kit selected for the selected channel (*Ch*).

Variable

Parameter	*Value*
Description	Standard number
Data type	Long integer type (Long)
Range	0 to 21
Note	If the specified variable is out of the allowable setup range, an error occurs when executed.

NOTE Since the variable (*Cpt*) has no preset value, you cannot omit it. If you omit the variable (*Cpt*), an error occurs when executed.

Examples

```
Dim StanOpen As Long
SCPI.SENSe(1).CORRection.COLLect.CKIT.ORDer.OPEN(1) = 10
StanOpen = SCPI.SENSe(1).CORRection.COLLect.CKIT.ORDer.OPEN(1)
```

Related objects

SCPI.SENSe(Ch).CORRection.COLLect.CKIT.SELect

SCPI.SENSe(Ch).CORRection.COLLect.CKIT.ORDer.LOAD(Cpt)

SCPI.SENSe(Ch).CORRection.COLLect.CKIT.ORDer.SHORt(Cpt)

SCPI.SENSe(Ch).CORRection.COLLect.CKIT.ORDer.THRU(Cpt_m,Cpt_n)

Equivalent key

Cal > **Modify Cal Kit** > **Specify CLSs** > **Open** > **Set All|Port 1|Port 2**

Equivalent SCPI command

Syntax

:SENSe{[1]-4}:CORRection:COLLect:CKIT:ORDer:OPEN <numeric 1>,<numeric 2>

:SENSe{[1]-4}:CORRection:COLLect:CKIT:ORDer:OPEN? <numeric 1>

Query response

{numeric 2}<newline><^END>

Example of use

```
10 OUTPUT 717;":SENS1:CORR:COLL:CKIT:ORD:OPEN 1,2"
20 OUTPUT 717;":SENS1:CORR:COLL:CKIT:ORD:OPEN? 1"
30 ENTER 717;A
```

SCPI.SENSe(*Ch*).CORRection.COLLect.CKIT.ORDer.SHORt(*Cpt*)

Object type

Property (**Read-Write**)

Syntax

SCPI.SENSe(*Ch*).CORRection.COLLect.CKIT.ORDer.SHORt(*Cpt*) = *Value*

Value = SCPI.SENSe(*Ch*).CORRection.COLLect.CKIT.ORDer.SHORt(*Cpt*)

Description

This command sets/gets the standard used for the short measurement of the specified port (*Cpt*), for the calibration kit selected for the selected channel (*Ch*).

Variable

Parameter	*Value*
Description	Standard number
Data type	Long integer type (Long)
Range	0 to 21
Note	If the specified variable is out of the allowable setup range, an error occurs when executed.

Examples

```
Dim StanShor As Long
SCPI.SENSe(1).CORRection.COLLect.CKIT.ORDer.SHORt(1) = 10
StanShor = SCPI.SENSe(1).CORRection.COLLect.CKIT.ORDer.SHORt(1)
```

Related objects

SCPI.SENSe(Ch).CORRection.COLLect.CKIT.SELect

SCPI.SENSe(Ch).CORRection.COLLect.CKIT.ORDer.LOAD(Cpt)

SCPI.SENSe(Ch).CORRection.COLLect.CKIT.ORDer.OPEN(Cpt)

SCPI.SENSe(Ch).CORRection.COLLect.CKIT.ORDer.THRU(Cpt_m,Cpt_n)

Equivalent key

Cal > **Modify Cal Kit** > **Specify CLSs** > **Short** > **Set All|Port 1|Port 2**

Equivalent SCPI command

Syntax

:SENSe{[1]-4}:CORRection:COLLect:CKIT:ORDer:SHORt <numeric 1>,<numeric 2>

:SENSe{[1]-4}:CORRection:COLLect:CKIT:ORDer:SHORt? <numeric 1>

Query response

{numeric 2}<newline><^END>

Example of use

```
10 OUTPUT 717;":SENS1:CORR:COLL:CKIT:ORD:SHOR 1,1"
20 OUTPUT 717;":SENS1:CORR:COLL:CKIT:ORD:SHOR? 1"
30 ENTER 717;A
```

SCPI.SENSe(*Ch*).CORRection.COLLect.CKIT.ORDer.THRU(*Cpt_m*,*Cpt_n*)

Object type

Property (**Read-Write**)

Syntax

SCPI.SENSe(*Ch*).CORRection.COLLect.CKIT.ORDer.THRU(*Cpt_m*,*Cpt_n*) = *Value*

Value = SCPI.SENSe(*Ch*).CORRection.COLLect.CKIT.ORDer.THRU(*Cpt_m*,*Cpt_n*)

Description

This set/get the standard used for the thru measurement between the specified 2 ports (*Cpt_m* and *Cpt_n*), for the calibration kit selected for the selected channel (*Ch*).

Variable

Parameter	*Value*
Description	Standard number
Data type	Long integer type (Long)
Range	0 to 21
Note	If the specified variable is out of the allowable setup range, an error occurs when executed.

Examples

```
Dim StanThru As Long
SCPI.SENSe(1).CORRection.COLLect.CKIT.ORDer.THRU(1,2) = 10
StanThru = SCPI.SENSe(1).CORRection.COLLect.CKIT.ORDer.THRU(1,2)
```

Related objects

SCPI.SENSe(Ch).CORRection.COLLect.CKIT.SELect

SCPI.SENSe(Ch).CORRection.COLLect.CKIT.ORDer.LOAD(Cpt)

SCPI.SENSe(Ch).CORRection.COLLect.CKIT.ORDer.OPEN(Cpt)

SCPI.SENSe(Ch).CORRection.COLLect.CKIT.ORDer.SHORt(Cpt)

Equivalent key

Cal > **Modify Cal Kit** > **Specify CLSs** > **Thru** > **Set All|Port 1-2**

Equivalent SCPI command

Syntax

:SENSe{[1]-4}:CORRection:COLLect:CKIT:ORDer:THRU <numeric 1>,<numeric 2>, <numeric 3>

:SENSe{[1]-4}:CORRection:COLLect:CKIT:ORDer:THRU? <numeric 1>,<numeric 2>

Query response

<numeric 3><newline><^END>

Example of use

```
10 OUTPUT 717;":SENS1:CORR:COLL:CKIT:ORD:THRU 1,2,11"
20 OUTPUT 717;":SENS1:CORR:COLL:CKIT:ORD:THRU? 1,2"
30 ENTER 717;A
```

SCPI.SENSe(*Ch*).CORRection.COLLect.CKIT.RESet

Object type

Method (**Write Only**)

Syntax

SCPI.SENSe(*Ch*).CORRection.COLLect.CKIT.RESet

Description

This command resets the calibration kit selected for selected channel (*Ch*) to the default factory setting state.

Examples

SCPI.SENSe(1).CORRection.COLLect.CKIT.RESet

Related objects

SCPI.SENSe(Ch).CORRection.COLLect.CKIT.SELect

Equivalent key

Cal > **Modify Cal Kit** > **Restore Cal Kit** > **OK**

Equivalent SCPI command

Syntax

:SENSe{[1]-4}:CORRection:COLLect:CKIT:RESet

Example of use

```
10 OUTPUT 717;":SENS1:CORR:COLL:CKIT:RES"
```

SCPI.SENSe(*Ch*).CORRection.COLLect.CKIT.SELect

Object type

Property (**Read-Write**)

Syntax

SCPI.SENSe(*Ch*).CORRection.COLLect.CKIT.SELect = *Value*

Value = SCPI.SENSe(*Ch*).CORRection.COLLect.CKIT.SELect

Description

This command sets/gets the calibration kit of the selected channel (*Ch*).

Variable

Parameter	*Value*
Description	Number of calibration kit
Data type	Long integer type (Long)
Range	1 to 10
Preset value	1
Note	If the specified variable is out of the allowable setup range, an error occurs when executed.

Examples

```
Dim CalKit As Long
SCPI.SENSe(1).CORRection.COLLect.CKIT.SELect = 3
CalKit = SCPI.SENSe(1).CORRection.COLLect.CKIT.SELect
```

Related objects

SCPI.SENSe(Ch).CORRection.COLLect.CKIT.ORDer.LOAD(Cpt)

SCPI.SENSe(Ch).CORRection.COLLect.CKIT.ORDer.OPEN(Cpt)

SCPI.SENSe(Ch).CORRection.COLLect.CKIT.ORDer.SHORt(Cpt)

SCPI.SENSe(Ch).CORRection.COLLect.CKIT.ORDer.THRU(Cpt_m,Cpt_n)

SCPI.SENSe(Ch).CORRection.COLLect.CKIT.STAN(Std).DELay

Equivalent key

Cal > **Cal Kit**

Equivalent SCPI command

Syntax

:SENSe{[1]-4}:CORRection:COLLect:CKIT[:SELect] <numeric>

:SENSe{[1]-4}:CORRection:COLLect:CKIT[:SELect]?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":SENS1:CORR:COLL:CKIT 3"
20 OUTPUT 717;":SENS1:CORR:COLL:CKIT?"
30 ENTER 717;A
```

SCPI.SENSe(*Ch*).CORRection.COLLect.CKIT.STAN(*Std*).ARBitrary

Object type

Property (**Read-Write**)

Syntax

SCPI.SENSe(*Ch*).CORRection.COLLect.CKIT.STAN(*Std*).ARBitrary = *Value*

Value = SCPI.SENSe(*Ch*).CORRection.COLLect.CKIT.STAN(*Std*).ARBitrary

Description

This command sets/gets the value of the arbitrary impedance of the selected standard (*Std*), for the calibration kit selected for the selected channel (*Ch*). Before defining this, the calibration kit should be selected kit by the SCPI.SENSe(Ch).CORRection.COLLect.CKIT.SELect.

Variable

Parameter	*Value*
Description	Value of arbitrary impedance
Data type	Double precision floating point type (Double)
Range	-1E18 to 1E18
Preset value	Varies depending on the specified calibration kit and standard.
Unit	Ω
Note	If the specified variable 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.

Examples

```
Dim StanArbt As Double
SCPI.SENSe(1).CORRection.COLLect.CKIT.STAN(5).ARBitrary = 50.5
StanArbt = SCPI.SENSe(1).CORRection.COLLect.CKIT.STAN(5).ARBitrary
```

Related objects

SCPI.SENSe(Ch).CORRection.COLLect.CKIT.SELect

Equivalent key

Cal > **Modify Cal Kit** > **Define STDs** > **no. name**> **Arb. Impedance**

Equivalent SCPI command

Syntax

:SENSe{[1]-4}:CORRection:COLLect:CKIT:STAN{[1]-21}:ARBitrary <numeric>

:SENSe{[1]-4}:CORRection:COLLect:CKIT:STAN{[1]-21}:ARBitrary?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":SENS1:CORR:COLL:CKIT:STAN1:ARB 50.5"
20 OUTPUT 717;":SENS1:CORR:COLL:CKIT:STAN1:ARB?"
30 ENTER 717;A
```

SCPI.SENSe(*Ch*).CORRection.COLLect.CKIT.STAN(*Std*).C0

Object type

Property (**Read-Write**)

Syntax

SCPI.SENSe(*Ch*).CORRection.COLLect.CKIT.STAN(*Std*).C0 = *Value*

Value = SCPI.SENSe(*Ch*).CORRection.COLLect.CKIT.STAN(*Std*).C0

Description

This command sets/gets the value of the CO value of the selected standard (*Std*), for the calibration kit selected for the selected channel (*Ch*). Before defining this, the calibration kit should be selected kit by the SCPI.SENSe(Ch).CORRection.COLLect.CKIT.SELect.

Variable

Parameter	*Value*
Description	C0
Data type	Double precision floating point type (Double)
Range	-1E18 to 1E18
Preset value	Varies depending on the specified calibration kit and standard.
Unit	fF (femto farad): 1E-15 F (farad)
Note	If the specified variable 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.

Examples

```
Dim StanC0 As Double
SCPI.SENSe(1).CORRection.COLLect.CKIT.STAN(5).C0 = 12.3
StanC0 = SCPI.SENSe(1).CORRection.COLLect.CKIT.STAN(5).C0
```

Related objects

SCPI.SENSe(Ch).CORRection.COLLect.CKIT.SELect

Equivalent key

Cal > **Modify Cal Kit** > **Define STDs** > **no. name** > **C0**

Equivalent SCPI command

Syntax

:SENSe{[1]-4}:CORRection:COLLect:CKIT:STAN{[1]-21}:C0 <numeric>

:SENSe{[1]-4}:CORRection:COLLect:CKIT:STAN{[1]-21}:C0?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":SENS1:CORR:COLL:CKIT:STAN1:C0 12.3"
20 OUTPUT 717;":SENS1:CORR:COLL:CKIT:STAN1:C0?"
30 ENTER 717;A
```

SCPI.SENSe(*Ch*).CORRection.COLLect.CKIT.STAN(*Std*).C1

Object type

Property (**Read-Write**)

Syntax

SCPI.SENSe(*Ch*).CORRection.COLLect.CKIT.STAN(*Std*).C1 = *Value*

Value = SCPI.SENSe(*Ch*).CORRection.COLLect.CKIT.STAN(*Std*).C1

Description

This command sets/gets the value of the C1 value of the selected standard (*Std*), for the selected calibration kit and channel (*Ch*). Before defining this, the calibration kit should be selected kit by the SCPI.SENSe(Ch).CORRection.COLLect.CKIT.SELect.

Variable

Parameter	*Value*
Description	C1
Data type	Double precision floating point type (Double)
Range	-1E18 to 1E18
Preset value	Varies depending on the specified calibration kit and standard.
Unit	1E-27 F/Hz (1E-27 farad / hertz)
Note	If the specified variable 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.

Examples

```
Dim StanC1 As Double
SCPI.SENSe(1).CORRection.COLLect.CKIT.STAN(5).C1 = 12.3
StanC1 = SCPI.SENSe(1).CORRection.COLLect.CKIT.STAN(5).C1
```

Related objects

SCPI.SENSe(Ch).CORRection.COLLect.CKIT.SELect

Equivalent key

Cal > **Modify Cal Kit** > **Define STDs** > **no. name**> **C1**

Equivalent SCPI command

Syntax

:SENSe{[1]-4}:CORRection:COLLect:CKIT:STAN{[1]-21}:C1 <numeric>

:SENSe{[1]-4}:CORRection:COLLect:CKIT:STAN{[1]-21}:C1?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":SENS1:CORR:COLL:CKIT:STAN1:C1 12.3"
20 OUTPUT 717;":SENS1:CORR:COLL:CKIT:STAN1:C1?"
30 ENTER 717;A
```

SCPI.SENSe(*Ch*).CORRection.COLLect.CKIT.STAN(*Std*).C2

Object type

Property (**Read-Write**)

Syntax

SCPI.SENSe(*Ch*).CORRection.COLLect.CKIT.STAN(*Std*).C2 = *Value*

Value = SCPI.SENSe(*Ch*).CORRection.COLLect.CKIT.STAN(*Std*).C2

Description

This command sets/gets the value of the C2 value of the selected standard (*Std*), for the calibration kit selected for the selected channel (*Ch*). Before defining this, the calibration kit should be selected kit by the SCPI.SENSe(Ch).CORRection.COLLect.CKIT.SELect.

Variable

Parameter	*Value*
Description	C2
Data type	Double precision floating point type (Double)
Range	-1E18 to 1E18
Preset value	Varies depending on the specified calibration kit and standard.
Unit	1E-36 F/Hz^2 (1E-36 farad /hertz^2)
Note	If the specified variable 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.

Examples

```
Dim StanC2 As Double
SCPI.SENSe(1).CORRection.COLLect.CKIT.STAN(5).C2 = 12.3
StanC2 = SCPI.SENSe(1).CORRection.COLLect.CKIT.STAN(5).C2
```

Related objects

SCPI.SENSe(Ch).CORRection.COLLect.CKIT.SELect

Equivalent key

Cal > **Modify Cal Kit** > **Define STDs** > **no. name** > **C2**

Equivalent SCPI command

Syntax

:SENSe{[1]-4}:CORRection:COLLect:CKIT:STAN{[1]-21}:C2 <numeric>

:SENSe{[1]-4}:CORRection:COLLect:CKIT:STAN{[1]-21}:C2?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":SENS1:CORR:COLL:CKIT:STAN1:C2 12.3"
20 OUTPUT 717;":SENS1:CORR:COLL:CKIT:STAN1:C2?"
30 ENTER 717;A
```

SCPI.SENSe(*Ch*).CORRection.COLLect.CKIT.STAN(*Std*).C3

Object type

Property (**Read-Write**)

Syntax

SCPI.SENSe(*Ch*).CORRection.COLLect.CKIT.STAN(*Std*).C3 = *Value*

Value = SCPI.SENSe(*Ch*).CORRection.COLLect.CKIT.STAN(*Std*).C3

Description

This command sets/gets the value of the C3 value of the selected standard (*Std*), for the calibration kit selected for the selected channel (*Ch*). Before defining this, the calibration kit should be selected kit by the SCPI.SENSe(Ch).CORRection.COLLect.CKIT.SELect.

Variable

Parameter	*Value*
Description	C3
Data type	Double precision floating point type (Double)
Range	-1E18 to 1E18
Preset value	Varies depending on the specified calibration kit and standard.
Unit	1E-45 F/Hz^3 (1E-45 farad / hertz^3)
Note	If the specified variable 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.

Examples

```
Dim StanC3 As Double
SCPI.SENSe(1).CORRection.COLLect.CKIT.STAN(5).C3 = 12.3
StanC3 = SCPI.SENSe(1).CORRection.COLLect.CKIT.STAN(5).C3
```

Related objects

SCPI.SENSe(Ch).CORRection.COLLect.CKIT.SELect

Equivalent key

Cal > **Modify Cal Kit** > **Define STDs** > **no. name** > **C3**

Equivalent SCPI command

Syntax

:SENSe{[1]-4}:CORRection:COLLect:CKIT:STAN{[1]-21}:C3 <numeric>

:SENSe{[1]-4}:CORRection:COLLect:CKIT:STAN{[1]-21}:C3?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":SENS1:CORR:COLL:CKIT:STAN1:C3 12.3"
20 OUTPUT 717;":SENS1:CORR:COLL:CKIT:STAN1:C3?"
30 ENTER 717;A
```

SCPI.SENSe(*Ch*).CORRection.COLLect.CKIT.STAN(*Std*).CP

Object Type

Property (**Read-Write**)

Syntax

SCPI.SENSe(*Ch*).CORRection.COLLect.CKIT.STAN(*Std*).CP = *Value*

Value = SCPI.SENSe(*Ch*).CORRection.COLLect.CKIT.STAN(*Std*).CP

Description

This command sets/gets Cp of the selected standard (*Std*) for the selected channel (*Ch*). This value is used when the model of standard is selected at Equivalent Circuit in SCPI.SENSe(Ch).CORRection.COLLect.CKIT.STAN(Std).MODel. Before defining this, the calibration kit should be selected kit by the SCPI.SENSe(Ch).CORRection.COLLect.CKIT.SELect. This command is available when option 005 is installed.

Variable

Parameter	*Value*
Description	Cp
Data Type	Double precision floating point type (Double)
Range	-1E to 1E
Preset Value	-
Unit	F
Resolution	-

Examples

```
Dim Var as Double
Var= 1e-10
SCPI.SENSe(1).CORRection.COLLect.CKIT.STAN(1).CP = Var
Var = SCPI.SENSe(1).CORRection.COLLect.CKIT.STAN(1).CP
```

Related Objects

SCPI.SENSe(Ch).CORRection.COLLect.CKIT.SELect

SCPI.SENSe(Ch).CORRection.COLLect.CKIT.STAN(Std).MODel

SCPI.SENSe(Ch).CORRection.COLLect.CKIT.STAN(Std).LS

SCPI.SENSe(Ch).CORRection.COLLect.CKIT.STAN(Std).RS

Equivalent Key

Calibration > **Modify Cal Kit** > **Define STDs** > **no. name** > **Cp**

Equivalent SCPI Command

Syntax

:SENSe{[1]-4}:CORRection:COLLect:CKIT:STAN{[1]-21}:CP <numeric>

:SENSe{[1]-4}:CORRection:COLLect:CKIT:STAN{[1]-21}:CP?

Query Response

<numeric><newline><^END>

Example of use

```
10 OUTPUT 717;":SENS1:CORR:COLL:CKIT:STAN1:CP 1E-10"
20 OUTPUT 717;":SENS1:CORR:COLL:CKIT:STAN1:CP?"
30 ENTER 717;A
```

SCPI.SENSe(*Ch*).CORRection.COLLect.CKIT.STAN(*Std*).DELay

Object type

Property (**Read-Write**)

Syntax

SCPI.SENSe(*Ch*).CORRection.COLLect.CKIT.STAN(*Std*).DELay = *Value*

Value = SCPI.SENSe(*Ch*).CORRection.COLLect.CKIT.STAN(*Std*).DELay

Description

This command sets/gets the value of the offset delay of the selected standard (*Std*), for the calibration kit selected for the selected channel (*Ch*). Before defining this, the calibration kit should be selected kit by the SCPI.SENSe(Ch).CORRection.COLLect.CKIT.SELect.

Variable

Parameter	*Value*
Description	Offset delay
Data type	Double precision floating point type (Double)
Range	-1E18 to 1E18
Preset value	Varies depending on the specified calibration kit and standard.
Unit	s (second)
Note	If the specified variable 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.

Examples

```
Dim StanDel As Double
SCPI.SENSe(1).CORRection.COLLect.CKIT.STAN(5).DELay = 12.3
StanDel = SCPI.SENSe(1).CORRection.COLLect.CKIT.STAN(5).DELay
```

Related objects

SCPI.SENSe(Ch).CORRection.COLLect.CKIT.SELect

Equivalent key

Cal > **Modify Cal Kit** > **Define STDs** > **no. name** > **Offset Delay**

Equivalent SCPI command

Syntax

:SENSe{[1]-4}:CORRection:COLLect:CKIT:STAN{[1]-21}:DELay <numeric>

:SENSe{[1]-4}:CORRection:COLLect:CKIT:STAN{[1]-21}:DELay?

Query response

<numeric><newline><^END>

Example of use

```
10 OUTPUT 717;":SENS1:CORR:COLL:CKIT:STAN1:DEL 12.3"
20 OUTPUT 717;":SENS1:CORR:COLL:CKIT:STAN1:DEL?"
30 ENTER 717;A
```

SCPI.SENSe(*Ch*).CORRection.COLLect.CKIT.STAN(*Std*).L0

Object type

Property (**Read-Write**)

Syntax

SCPI.SENSe(*Ch*).CORRection.COLLect.CKIT.STAN(*Std*).L0 = *Value*

Value = SCPI.SENSe(*Ch*).CORRection.COLLect.CKIT.STAN(*Std*).L0

Description

This command sets/gets the value of the L0 value of the selected standard (*Std*), for the calibration kit selected for the selected channel (*Ch*). Before defining this, the calibration kit should be selected kit by the SCPI.SENSe(Ch).CORRection.COLLect.CKIT.SELect.

Variable

Parameter	*Value*
Description	L0
Data type	Double precision floating point type (Double)
Range	-1E18 to 1E18
Preset value	Varies depending on the specified calibration kit and standard.
Unit	pH (pico henry)
Note	If the specified variable 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.

Examples

```
Dim StanL0 As Double
SCPI.SENSe(1).CORRection.COLLect.CKIT.STAN(5).L0 = 12.3
StanL0 = SCPI.SENSe(1).CORRection.COLLect.CKIT.STAN(5).L0
```

Related objects

SCPI.SENSe(Ch).CORRection.COLLect.CKIT.SELect

Equivalent key

Cal > **Modify Cal Kit** > **Define STDs** > **no. name** > **L0**

Equivalent SCPI command

Syntax

```
:SENSe{[1]-4}:CORRection:COLLect:CKIT:STAN{[1]-21}:L0 <numeric>
:SENSe{[1]-4}:CORRection:COLLect:CKIT:STAN{[1]-21}:L0?
```

Query response

```
{numeric}<newline><^END>
```

Example of use

```
10 OUTPUT 717;":SENS1:CORR:COLL:CKIT:STAN1:L0 12.3"
20 OUTPUT 717;":SENS1:CORR:COLL:CKIT:STAN1:L0?"
30 ENTER 717;A
```

SCPI.SENSe(*Ch*).CORRection.COLLect.CKIT.STAN(*Std*).L1

Object type

Property (**Read-Write**)

Syntax

SCPI.SENSe(*Ch*).CORRection.COLLect.CKIT.STAN(*Std*).L1 = *Value*

Value = SCPI.SENSe(*Ch*).CORRection.COLLect.CKIT.STAN(*Std*).L1

Description

This command sets/gets the value of the L1 value of the selected standard (*Std*), for the calibration kit selected for the selected channel (*Ch*). Before defining this, the calibration kit should be selected kit by the SCPI.SENSe(Ch).CORRection.COLLect.CKIT.SELect.

Variable

Parameter	*Value*
Description	L1
Data type	Double precision floating point type (Double)
Range	-1E18 to 1E18
Preset value	Varies depending on the specified calibration kit and standard.
Unit	1E-24 H/Hz (1E-24 henry / hertz)
Note	If the specified variable 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.

Examples

```
Dim StanL1 As Double
SCPI.SENSe(1).CORRection.COLLect.CKIT.STAN(5).L1 = 12.3
StanL1 = SCPI.SENSe(1).CORRection.COLLect.CKIT.STAN(5).L1
```

Related objects

SCPI.SENSe(Ch).CORRection.COLLect.CKIT.SELect

Equivalent key

Cal > **Modify Cal Kit** > **Define STDs** > **no. name** > **L1**

Equivalent SCPI command

Syntax

:SENSe{[1]-4}:CORRection:COLLect:CKIT:STAN{[1]-21}:L1 <numeric>

:SENSe{[1]-4}:CORRection:COLLect:CKIT:STAN{[1]-21}:L1?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":SENS1:CORR:COLL:CKIT:STAN1:L1 12.3"
20 OUTPUT 717;":SENS1:CORR:COLL:CKIT:STAN1:L1?"
30 ENTER 717;A
```

SCPI.SENSe(*Ch*).CORRection.COLLect.CKIT.STAN(*Std*).L2

Object type

Property (**Read-Write**)

Syntax

SCPI.SENSe(*Ch*).CORRection.COLLect.CKIT.STAN(*Std*).L2 = *Value*

Value = SCPI.SENSe(*Ch*).CORRection.COLLect.CKIT.STAN(*Std*).L2

Description

This command sets/gets the value of the L2 value of the selected standard (*Std*), for the calibration kit selected for the selected channel (*Ch*). Before defining this, the calibration kit should be selected kit by the SCPI.SENSe(Ch).CORRection.COLLect.CKIT.SELect.

Variable

Parameter	*Value*
Description	L2
Data type	Double precision floating point type (Double)
Range	-1E18 to 1E18
Preset value	Varies depending on the specified calibration kit and standard.
Unit	1E-33 H/Hz^2 (1E-33 henry / $hertz^2$)
Note	If the specified variable 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.

Examples

```
Dim StanL2 As Double
SCPI.SENSe(1).CORRection.COLLect.CKIT.STAN(5).L2 = 12.3
StanL2 = SCPI.SENSe(1).CORRection.COLLect.CKIT.STAN(5).L2
```

Related objects

SCPI.SENSe(Ch).CORRection.COLLect.CKIT.SELect

Equivalent key

Cal > **Modify Cal Kit** > **Define STDs** > **no. name** > **L2**

Equivalent SCPI command

Syntax

:SENSe{[1]-4}:CORRection:COLLect:CKIT:STAN{[1]-21}:L2 <numeric>

:SENSe{[1]-4}:CORRection:COLLect:CKIT:STAN{[1]-21}:L2?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":SENS1:CORR:COLL:CKIT:STAN1:L2 12.3"
20 OUTPUT 717;":SENS1:CORR:COLL:CKIT:STAN1:L2?"
30 ENTER 717;A
```

SCPI.SENSe(*Ch*).CORRection.COLLect.CKIT.STAN(*Std*).L3

Object type

Property (**Read-Write**)

Syntax

SCPI.SENSe(*Ch*).CORRection.COLLect.CKIT.STAN(*Std*).L3 = *Value*

Value = SCPI.SENSe(*Ch*).CORRection.COLLect.CKIT.STAN(*Std*).L3

Description

This command sets/gets the value of the L3 value of the selected standard (*Std*), for the calibration kit selected for the selected channel (*Ch*). Before defining this, the calibration kit should be selected kit by the SCPI.SENSe(Ch).CORRection.COLLect.CKIT.SELect.

Variable

Parameter	*Value*
Description	L3
Data type	Double precision floating point type (Double)
Range	-1E18 to 1E18
Preset value	Varies depending on the specified calibration kit and standard.
Unit	1E-42 H/Hz^3 (1E-42 henry / hertz^3)
Note	If the specified variable 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.

Examples

```
Dim StanL3 As Double
SCPI.SENSe(1).CORRection.COLLect.CKIT.STAN(5).L3 = 12.3
StanL3 = SCPI.SENSe(1).CORRection.COLLect.CKIT.STAN(5).L3
```

Related objects

SCPI.SENSe(Ch).CORRection.COLLect.CKIT.SELect

Equivalent key

Cal > **Modify Cal Kit** > **Define STDs** > **no. name** > **L3**

Equivalent SCPI command

Syntax

:SENSe{[1]-4}:CORRection:COLLect:CKIT:STAN{[1]-21}:L3 <numeric>

:SENSe{[1]-4}:CORRection:COLLect:CKIT:STAN{[1]-21}:L3?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":SENS1:CORR:COLL:CKIT:STAN1:L3 12.3"
20 OUTPUT 717;":SENS1:CORR:COLL:CKIT:STAN1:L3?"
30 ENTER 717;A
```

SCPI.SENSe(*Ch*).CORRection.COLLect.CKIT.STAN(*Std*).LS

Object Type

Property (**Read-Write**)

Syntax

SCPI.SENSe(*Ch*).CORRection.COLLect.CKIT.STAN(*Std*).LS = *Value*

Value = SCPI.SENSe(*Ch*).CORRection.COLLect.CKIT.STAN(*Std*).LS

Description

This command sets/gets Ls of the selected standard (*Std*) for the selected channel (*Ch*). This value is used when the model of standard is selected at Equivalent Circuit in SCPI.SENSe(Ch).CORRection.COLLect.CKIT.STAN(Std).MODel. Before defining this, the calibration kit should be selected kit by the SCPI.SENSe(Ch).CORRection.COLLect.CKIT.SELect. This command is available when option 005 is installed.

Variable

Parameter	*Value*
Description	Ls
Data Type	Double precision floating point type (Double)
Range	-1E to 1E
Preset Value	-
Unit	H
Resolution	-

Examples

```
Dim Var as Double
Var= 1e-10
SCPI.SENSe(1).CORRection.COLLect.CKIT.STAN(1).LS = Var
Var = SCPI.SENSe(1).CORRection.COLLect.CKIT.STAN(1).LS
```

Related Objects

SCPI.SENSe(Ch).CORRection.COLLect.CKIT.SELect

SCPI.SENSe(Ch).CORRection.COLLect.CKIT.STAN(Std).MODel

SCPI.SENSe(Ch).CORRection.COLLect.CKIT.STAN(Std).CP

SCPI.SENSe(Ch).CORRection.COLLect.CKIT.STAN(Std).RS

Equivalent Key

Calibration > Modify Cal Kit > Define STDs > no. name > Ls

Equivalent SCPI Command

Syntax

:SENSe{[1]-4}:CORRection:COLLect:CKIT:STAN{[1]-21}:LS <numeric>

:SENSe{[1]-4}:CORRection:COLLect:CKIT:STAN{[1]-21}:LS?

Query Response

<numeric><newline><^END>

Example of use

```
10 OUTPUT 717;":SENS1:CORR:COLL:CKIT:STAN1:LS 1E-10"
20 OUTPUT 717;":SENS1:CORR:COLL:CKIT:STAN1:LS?"
30 ENTER 717;A
```

SCPI.SENSe(*Ch*).CORRection.COLLect.CKIT.STAN(*Std*).LABel

Object type

Property (**Read-Write**)

Syntax

SCPI.SENSe(*Ch*).CORRection.COLLect.CKIT.STAN(*Std*).LABel = *Lbl*

Lbl = SCPI.SENSe(*Ch*).CORRection.COLLect.CKIT.STAN(*Std*).LABel

Description

This command sets/gets the name of the selected standard (*Std*), for the calibration kit selected for the selected channel (*Ch*). Before defining this, the calibration kit should be selected kit by the SCPI.SENSe(Ch).CORRection.COLLect.CKIT.SELect.

Variable

Parameter	*Lbl*
Description	Standard name
Data type	Character string type (String)
Range	254 characters or less
Preset value	Varies depending on the specified calibration kit and standard.

Examples

```
Dim StanLbl As Double
SCPI.SENSe(1).CORRection.COLLect.CKIT.STAN(5).LABel = "OPEN 3.5mm"
StanLbl = SCPI.SENSe(1).CORRection.COLLect.CKIT.STAN(5).LABel
```

Related objects

SCPI.SENSe(Ch).CORRection.COLLect.CKIT.SELect

Equivalent key

Cal > **Modify Cal Kit** > **Define STDs** > **no. name** > **Label**

Equivalent SCPI command

Syntax

:SENSe{[1]-4}:CORRection:COLLect:CKIT:STAN{[1]-21}:LABel <string>

:SENSe{[1]-4}:CORRection:COLLect:CKIT:STAN{[1]-21}:LABel?

Query response

{string}<newline><^END>

Example of use

```
10 OUTPUT 717;":SENS1:CORR:COLL:CKIT:STAN1:LAB ""OPEN"""
20 OUTPUT 717;":SENS1:CORR:COLL:CKIT:STAN1:LAB?"
30 ENTER 717;A$
```

SCPI.SENSe(*Ch*).CORRection.COLLect.CKIT.STAN(*Std*).LOSS

Object type

Property (**Read-Write**)

Syntax

SCPI.SENSe(*Ch*).CORRection.COLLect.CKIT.STAN(*Std*).LOSS = *Value*

Value = SCPI.SENSe(*Ch*).CORRection.COLLect.CKIT.STAN(*Std*).LOSS

Description

This command sets/gets the value of the offset loss of the selected standard (*Std*), for the calibration kit selected for the selected channel (*Ch*). Before defining this, the calibration kit should be selected kit by the SCPI.SENSe(Ch).CORRection.COLLect.CKIT.SELect.

Variable

Parameter	*Value*
Description	Offset loss
Data type	Double precision floating point type (Double)
Range	-1E18 to 1E18
Preset value	Varies depending on the specified calibration kit and standard.
Unit	Ω/s
Note	If the specified variable 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.

Examples

```
Dim StanLoss As Double
SCPI.SENSe(1).CORRection.COLLect.CKIT.STAN(5).LOSS = 12.3
StanLoss = SCPI.SENSe(1).CORRection.COLLect.CKIT.STAN(5).LOSS
```

Related objects

SCPI.SENSe(Ch).CORRection.COLLect.CKIT.SELect

Equivalent key

Cal > Modify Cal Kit > Define STDs > no. name > Offset Loss

Equivalent SCPI command

Syntax

:SENSe{[1]-4}:CORRection:COLLect:CKIT:STAN{[1]-21}:LOSS <numeric>

:SENSe{[1]-4}:CORRection:COLLect:CKIT:STAN{[1]-21}:LOSS?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":SENS1:CORR:COLL:CKIT:STAN1:LOSS 12.3"
20 OUTPUT 717;":SENS1:CORR:COLL:CKIT:STAN1:LOSS?"
30 ENTER 717;A
```

SCPI.SENSe(*Ch*).CORRection.COLLect.CKIT.STAN(*Std*).MODel

Object Type

Property (**Read-Write**)

Syntax

SCPI.SENSe(*Ch*).CORRection.COLLect.CKIT.STAN(*Std*).MODel = *Model*

Model = SCPI.SENSe(*Ch*).CORRection.COLLect.CKIT.STAN(*Std*).MODel

Description

This command specify the standard model for the selected standard (*Std*) and for the selected channel (*Ch*). (See Defining Calibration Kit for the model) Before defining this, the calibration kit should be selected kit by the SCPI.SENSe(Ch).CORRection.COLLect.CKIT.SELect. This command is available when option 005 is installed.

When you select either equivalent circuit or table, one of either Open, Short, Load and Arbitrary should be selected in SCPI.SENSe(Ch).CORRection.COLLect.CKIT.STAN(Std).TYPE.

Variable

Parameter	*Model*
Description	Standard model of standard kit
Data Type	Character string type (String)
Range	POLYnomial: Polynomial model EQUivalent: Equivalent Circuit model TABLe: Table
Preset Value	-

Examples

```
Dim Model as String
SCPI.SENSe(1).CORRection.COLLect.CKIT.STAN(1).MODel = "POLYnomial"
Model = SCPI.SENSe(1).CORRection.COLLect.CKIT.STAN(1).MODel
```

Related Objects

SCPI.SENSe(Ch).CORRection.COLLect.CKIT.SELect

Equivalent Key

Calibration > **Modify Cal Kit** > **Define STDs** > **<standard>** > **STD Model**

Equivalent SCPI Command

Syntax

:SENSe{[1]-4}:CORRection:COLLect:CKIT:STAN{[1]-21}:MODel {POLYnomial|EQUivalent|TABLe}

:SENSe{[1]-4}:CORRection:COLLect:CKIT:STAN{[1]-21}:MODel?

Query Response

{POLY|EQU|TABL}<newline><^END>

Example of use

```
10 OUTPUT 717;":SENS1:CORR:COLL:CKIT:STAN1:MOD POLY"
20 OUTPUT 717;":SENS1:CORR:COLL:CKIT:STAN1:MOD ?"
30 ENTER 717;A$
```

SCPI.SENSe(*Ch*).CORRection.COLLect.CKIT.STAN(*Std*).RS

Object Type

Property (**Read-Write**)

Syntax

SCPI.SENSe(*Ch*).CORRection.COLLect.CKIT.STAN(*Std*).RS = *Value*

Value = SCPI.SENSe(*Ch*).CORRection.COLLect.CKIT.STAN(*Std*).RS

Description

This command sets/gets Rs of the selected standard (*Std*) for the selected channel (*Ch*). This value is used when the model of standard is selected at Equivalent Circuit in SCPI.SENSe(Ch).CORRection.COLLect.CKIT.STAN(Std).MODel. Before defining this, the calibration kit should be selected kit by the SCPI.SENSe(Ch).CORRection.COLLect.CKIT.SELect. This command is available when option 005 is installed.

Variable

Parameter	*Value*
Description	Rs
Data Type	Double precision floating point type (Double)
Range	-1E to 1E
Preset Value	-
Unit	Ω
Resolution	-

Examples

```
Dim Var as Double
Var= 50
SCPI.SENSe(1).CORRection.COLLect.CKIT.STAN(1).RS = Var
Var = SCPI.SENSe(1).CORRection.COLLect.CKIT.STAN(1).RS
```

Related Objects

SCPI.SENSe(Ch).CORRection.COLLect.CKIT.SELect

SCPI.SENSe(Ch).CORRection.COLLect.CKIT.STAN(Std).MODel

SCPI.SENSe(Ch).CORRection.COLLect.CKIT.STAN(Std).CP

SCPI.SENSe(Ch).CORRection.COLLect.CKIT.STAN(Std).LS

Equivalent Key

Calibration > **Modify Cal Kit** > **Define STDs** > **no. name** > **Rs**

Equivalent SCPI Command

Syntax

:SENSe{[1]-4}:CORRection:COLLect:CKIT:STAN{[1]-21}:RS <numeric>

:SENSe{[1]-4}:CORRection:COLLect:CKIT:STAN{[1]-21}:RS?

Query Response

<numeric><newline><^END>

Example of use

```
10 OUTPUT 717;":SENS1:CORR:COLL:CKIT:STAN1:RS 50"
20 OUTPUT 717;":SENS1:CORR:COLL:CKIT:STAN1:RS?"
30 ENTER 717;A
```

SCPI.SENSe(*Ch*).CORRection.COLLect.CKIT.STAN(Std).TABLe

Object Type

Property (**Read-Write**)

Syntax

SCPI.SENSe(*Ch*).CORRection.COLLect.CKIT.STAN(Std).TABLe = *Value*

Value = SCPI.SENSe(*Ch*).CORRection.COLLect.CKIT.STAN(Std).TABLe

Description

This command sets and gets the the data of table standard model for the selected standard (*Std*) and for the selected channel (*Ch*). Before defining this, the calibration kit should be selected kit by the SCPI.SENSe(Ch).CORRection.COLLect.CKIT.SELect. This command is available when option 005 is installed.

To select the table model, use the SCPI.SENSe(Ch).CORRection.COLLect.CKIT.STAN(Std).MODel and one of either Open, Short, Load and Arbitrary should be selected in SCPI.SENSe(Ch).CORRection.COLLect.CKIT.STAN(Std).TYPE.

Variable

Parameter	*Array*
Description	Indicates the array data of 1 + Num (number of calibration data points) × 3. Where n is an integer between 1 and Num. • *Data(0)* :The number of calibration kit data points you want to set. Specify an integer ranging 1 to 1601. • *Data(n×3-2)* :Frequency at the n-th of standard data. • *Data(n×3-1)* :Real part of data (complex number) at the n-th standard data point. • *Data(n×3)* :Imaginary part of data (complex number) at the n-th standard data point. The index of the array starts from 0.
Data type	Variant type (Variant)

Examples

Dim SetTableAry As Variant, GetTableAray As Variant, NoOfData As Long

```
SetTableAry = Array(3, 1000000#, 50.1, 0, 2000000#, 50.05, 0, 3000000#, 51, 0)
SCPI.SENSe(1).CORRection.COLLect.CKIT.STAN(1).TABLe = SetTableAry

NoOfData = SCPI.SERVice.CHANnel.STAN.TABLe

ReDim GetTableAray(NoOfData - 1)
GetTableAray = SCPI.SENSe(1).CORRection.COLLect.CKIT.STAN(1).TABLe
```

Related Objects

SCPI.SENSe(Ch).CORRection.COLLect.CKIT.STAN(Std).MODel

SCPI.SENSe(Ch).CORRection.COLLect.CKIT.STAN(Std).TYPE

SCPI.SERVice.CHANnel(Ch).STAN(Std).TABLe

Equivalent Key

No equivalent key is available on the front panel.

Equivalent SCPI Command

Syntax

:SENSe{[1]-4}:CORRection:COLLect:CKIT:STAN{[1]-21}:TABLe <numeric 1>, <numeric *n*×3-*2*>, <numeric *n*×3-*1*>, <numeric *n*×3> ,..., <numeric Num×3>

:SENSe{[1]-4}:CORRection:COLLect:CKIT:STAN{[1]-21}:TABLe?

Query response

<numeric 1>, <numeric *n*×3-*2*>, <numeric *n*×3-*1*>, <numeric *n*×3> ,..., <numeric Num×3><newline><^END>

	Description
<numeric 1>	The number of calibration kit data points you want to set. Specify an integer ranging 1 to 4804.
<numeric *n*×3-*2*>	Frequency at the n-th of standard data.
<numeric *n*×3-*1*>	Real part of data (complex number) at the n-th standard data point.
<numeric	Imaginary part of data (complex number) at the n-th

n×3>	standard data point.

Example of use

```
10 Dim A(10)
20 OUTPUT 717;":SENS1:CORR:COLL:CKIT:STAN1:TABL 3, 1000000#, 50.1, 0, 2000000#, 50.05, 0,
3000000#, 51, 0
30 OUTPUT 717;":SENS1:CORR:COLL:CKIT:STAN1:TABL?"
40 ENTER 717;A(*)
```

SCPI.SENSe(*Ch*).CORRection.COLLect.CKIT.STAN(*Std*).TYPE

Object type

Property (**Read-Write**)

Syntax

SCPI.SENSe(*Ch*).CORRection.COLLect.CKIT.STAN(*Std*).TYPE = *Param*

Param = SCPI.SENSe(*Ch*).CORRection.COLLect.CKIT.STAN(*Std*).TYPE

Description

This command sets/gets the standard type of the selected standard (*Std*), for the calibration kit selected for the selected channel (*Ch*). Before defining this, the calibration kit should be selected kit by the SCPI.SENSe(Ch).CORRection.COLLect.CKIT.SELect.

Variable

Parameter	*Param*
Description	Standard type
Data type	Character string type (String)
Range	Select from either of the following: • "OPEN": Specifies open. • "SHORt": Specifies short. • "LOAD": Specifies load. • "THRU": Specifies Thru. • "UTHRu": Specifies Unknown Thru. • "ARBI": Specifies arbitrary impedance. • "NONE": Specifies DUT of which theoretical value is 0.
Preset value	Varies depending on the specified calibration kit and standard.

Examples

```
Dim StanType As String
SCPI.SENSe(1).CORRection.COLLect.CKIT.STAN(5).TYPE = "OPEN"
StanType = SCPI.SENSe(1).CORRection.COLLect.CKIT.STAN(5).TYPE
```

Related objects

SCPI.SENSe(Ch).CORRection.COLLect.CKIT.SELect

Equivalent key

Cal > **Modify Cal Kit** > **Define STDs** > **no. name** > **STD Type** > **Open|Short|Load|Delay/Thru|Arbitrary|None|Unknown Thru**

Equivalent SCPI command

Syntax

:SENSe{[1]-4}:CORRection:COLLect:CKIT:STAN{[1]-21}:TYPE {OPEN|SHORt|LOAD|THRU|UTHRu|ARBI|NONE}

:SENSe{[1]-4}:CORRection:COLLect:CKIT:STAN{[1]-21}:TYPE?

Query response

{OPEN|SHOR|LOAD|THRU|UTHR|ARBI|NONE}<newline><^END>

Example of use

```
10 OUTPUT 717;":SENS1:CORR:COLL:CKIT:STAN1:TYPE OPEN"
20 OUTPUT 717;":SENS1:CORR:COLL:CKIT:STAN1:TYPE?"
30 ENTER 717;A$
```

SCPI.SENSe(*Ch*).CORRection.COLLect.CKIT.STAN(*Std*).Z0

Object type

Property (**Read-Write**)

Syntax

SCPI.SENSe(*Ch*).CORRection.COLLect.CKIT.STAN(*Std*).Z0 = *Value*

Value = SCPI.SENSe(*Ch*).CORRection.COLLect.CKIT.STAN(*Std*).Z0

Description

This command sets/gets the value of the offset Z0 of the selected standard (*Std*), for the calibration kit selected for the selected channel (*Ch*). Before defining this, the calibration kit should be selected kit by the SCPI.SENSe(Ch).CORRection.COLLect.CKIT.SELect.

Variable

Parameter	*Value*
Description	Offset Z0
Data type	Double precision floating point type (Double)
Range	-1E18 to 1E18
Preset value	Varies depending on the specified calibration kit and standard.
Unit	Ω
Note	If the specified variable 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.

Examples

```
Dim StanZ0 As Double
SCPI.SENSe(1).CORRection.COLLect.CKIT.STAN(5).Z0 = 50
StanZ0 = SCPI.SENSe(1).CORRection.COLLect.CKIT.STAN(5).Z0
```

Related objects

SCPI.SENSe(Ch).CORRection.COLLect.CKIT.SELect

Equivalent key

Cal > **Modify Cal Kit** > **Define STDs** > **no. name** > **Offset Z0**

Equivalent SCPI command

Syntax

:SENSe{[1]-4}:CORRection:COLLect:CKIT:STAN{[1]-21}:Z0 <numeric>

:SENSe{[1]-4}:CORRection:COLLect:CKIT:STAN{[1]-21}:Z0?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":SENS1:CORR:COLL:CKIT:STAN1:Z0 50"
20 OUTPUT 717;":SENS1:CORR:COLL:CKIT:STAN1:Z0?"
30 ENTER 717;A
```

SCPI.SENSe(*Ch*).CORRection.COLLect.CKIT.ZORDer.LOAD

Object Type

Property (**Read-Write**)

Syntax

SCPI.SENSe(*Ch*).CORRection.COLLect.CKIT.ZORDer.LOAD = *Std*

Std = SCPI.SENSe(*Ch*).CORRection.COLLect.CKIT.ZORDer.LOAD

Description

This command sets/gets standard number for load in impedance calibration for the selected channel (*Ch*). Before defining this, the calibration kit should be selected kit by the SCPI.SENSe(Ch).CORRection.COLLect.CKIT.SELect. This command is available when option 005 is installed.

Variable

Parameter	*Std*
Description	standard number for load
Data Type	Long integer type (Long)
Range	1 ~ 21
Preset Value	3 (for registered calibration kit)
Unit	-
Resolution	-

Examples

```
Dim Var as Long
SCPI.SENSe(1).CORRection.COLLect.CKIT.ZORDer.LOAD = 3
Var = SCPI.SENSe(1).CORRection.COLLect.CKIT.ZORDer.LOAD
```

Related Objects

SCPI.SENSe(Ch).CORRection.COLLect.CKIT.SELect

SCPI.SENSe(Ch).CORRection.COLLect.CKIT.ZORDer.OPEN

SCPI.SENSe(Ch).CORRection.COLLect.CKIT.ZORDer.SHORt

Equivalent Key

Cal > **Modify Cal Kit** > **Specify CLSs** > **Load** > **Impedance Cal**

Equivalent SCPI Command

Syntax

:SENSe{[1]-4}:CORRection:COLLect:CKIT:ZORDer:LOAD <numeric>

:SENSe{[1]-4}:CORRection:COLLect:CKIT:ZORDer:LOAD?

Query Response

<numeric><newline><^END>

Example of use

```
10 OUTPUT 717;":SENS1:CORR:COLL:CKIT:ZORD:LOAD 3"
20 OUTPUT 717;":SENS1:CORR:COLL:CKIT:ZORD:LOAD?"
30 ENTER 717;A
```

SCPI.SENSe(*Ch*).CORRection.COLLect.CKIT.ZORDer.OPEN

Object Type

Property (**Read-Write**)

Syntax

SCPI.SENSe(*Ch*).CORRection.COLLect.CKIT.ZORDer.OPEN = *Std*

Std = SCPI.SENSe(*Ch*).CORRection.COLLect.CKIT.ZORDer.OPEN

Description

This command sets/gets standard number for open in impedance calibration for the selected channel (*Ch*). Before defining this, the calibration kit should be selected kit by the SCPI.SENSe(Ch).CORRection.COLLect.CKIT.SELect. This command is available when option 005 is installed.

Variable

Parameter	*Std*
Description	Standard number for open
Data Type	Long integer type (Long)
Range	1 ~ 21
Preset Value	1 (for registered calibration kit)
Unit	-
Resolution	-

Examples

```
Dim Var as Long
SCPI.SENSe(1).CORRection.COLLect.CKIT.ZORDer.OPEN = 1
Var = SCPI.SENSe(1).CORRection.COLLect.CKIT.ZORDer.OPEN
```

Related Objects

SCPI.SENSe(Ch).CORRection.COLLect.CKIT.SELect

SCPI.SENSe(Ch).CORRection.COLLect.CKIT.ZORDer.LOAD

SCPI.SENSe(Ch).CORRection.COLLect.CKIT.ZORDer.SHORt

Equivalent Key

Cal > **Modify Cal Kit** > **Specify CLSs** > **Open** > **Impedance Cal**

Equivalent SCPI Command

Syntax

:SENSe{[1]-4}:CORRection:COLLect:CKIT:ZORDer:OPEN <numeric>

:SENSe{[1]-4}:CORRection:COLLect:CKIT:ZORDer:OPEN?

Query Response

<numeric><newline><^END>

Example of use

```
10 OUTPUT 717;":SENS1:CORR:COLL:CKIT:ZORD:OPEN 1"
20 OUTPUT 717;":SENS1:CORR:COLL:CKIT:ZORD:OPEN?"
30 ENTER 717;A
```

SCPI.SENSe(*Ch*).CORRection.COLLect.CKIT.ZORDer.SHORt

Object Type

Property (**Read-Write**)

Syntax

SCPI.SENSe(*Ch*).CORRection.COLLect.CKIT.ZORDer.SHORt = *Std*

Std = SCPI.SENSe(*Ch*).CORRection.COLLect.CKIT.ZORDer.SHORt

Description

This command sets/gets standard number for short in impedance calibration for the selected channel (*Ch*). Before defining this, the calibration kit should be selected kit by the SCPI.SENSe(Ch).CORRection.COLLect.CKIT.SELect. This command is available when option 005 is installed.

Variable

Parameter	*Std*
Description	Standard number for short
Data Type	Long integer type (Long)
Range	1 ~ 21
Preset Value	2 (for registered calibration kit)
Unit	-
Resolution	-

Examples

```
Dim Var as Long
SCPI.SENSe(1).CORRection.COLLect.CKIT.ZORDer.SHORt = 2
Var = SCPI.SENSe(1).CORRection.COLLect.CKIT.ZORDer.SHORt
```

Related Objects

SCPI.SENSe(Ch).CORRection.COLLect.CKIT.SELect

SCPI.SENSe(Ch).CORRection.COLLect.CKIT.ZORDer.OPEN

SCPI.SENSe(Ch).CORRection.COLLect.CKIT.ZORDer.LOAD

Equivalent Key

Cal > **Modify Cal Kit** > **Specify CLSs** > **Short** > **Impedance Cal**

Equivalent SCPI Command

Syntax

:SENSe{[1]-4}:CORRection:COLLect:CKIT:ZORDer:SHORt <numeric>

:SENSe{[1]-4}:CORRection:COLLect:CKIT:ZORDer:SHORt?

Query Response

<numeric><newline><^END>

Example of use

```
10 OUTPUT 717;":SENS1:CORR:COLL:CKIT:ZORD:SHOR 2"
20 OUTPUT 717;":SENS1:CORR:COLL:CKIT:ZORD:SHOR?"
30 ENTER 717;A
```

SCPI.SENSe(*Ch*).CORRection.COLLect.CLEar

Type of object

Method (**Write Only**)

Syntax

SCPI.SENSe(*Ch*).CORRection.COLLect.CLEar

Description

This command clears the calibration measurement data when the frequency offset feature is off, for the selected channel (*Ch*).

NOTE Settings that have been temporarily changed due to measurement for each standard (number of traces, measurement parameters, and so on) return to their original values.

Example of use

SCPI.SENSe(1).CORRection.COLLect.CLEar

Related objects

SCPI.SENSe(Ch).OFFSet.STATe

Equivalent key

Cal > **Calibrate** > **Responce(Open)** > **Cancel** > **OK**

Cal > **Calibrate** > **Responce(Short)** > **Cancel** > **OK**

Cal > **Calibrate** > **Responce(Thru)** > **Cancel** > **OK**

Cal > **Calibrate** > **Enhanced Responce** > **Cancel** > **OK**

Cal> **Calibrate** > **1-Port Cal** > **Cancel** > **OK**

Cal> **Calibrate** > **2-Port Cal** > **Cancel** > **OK**

Cal > **Calibrate** > **Adapter Removal** > **Port1|Port2** > **Cancel** > **OK**

Equivalent SCPI command

Syntax

:SENSe{[1]-4}:CORRection:COLLect:CLEar

Example of use

10 OUTPUT 717;":SENS1:CORR:COLL:CLE"

SCPI.SENSe(*Ch*).CORRection.COLLect.ECAL.CCHeck.ACQuire

Type of object

Method (**Write Only**)

Syntax

SCPI.SENSe(*Ch*).CORRection.COLLect.ECAL.CCHeck.ACQuire

Description

This command executes the confidence check of the calibration coefficients for selected channel (*Ch*), using ECal (Electronic Calibration). In other words, this command sets the data measured with the analyzer and the data stored in ECal so that they can be compared).

NOTE If you execute this object when the ECal module is not connected or when ports are not connected each other appropriately, a runtime error occurs.

Example of use

```
SCPI.SENSe(1).CORRection.COLLect.ECAL.CCHeck.ACQuire
```

Equivalent key

Cal > **ECal** > **Confidence Check**

Equivalent SCPI command

Syntax

:SENSe{[1]-4}:CORRection:COLLect:ECAL:CCHeck[:ACQuire]

Example of use

```
10 OUTPUT 717;":SENS1:CORR:COLL:ECAL:CCH"
20 OUTPUT 717;"*OPC?"
30 ENTER 717;A
```

SCPI.SENSe(*Ch*).CORRection.COLLect.ECAL.ERESponse

Object type

Method (**Write Only**)

Syntax

SCPI.SENSe(*Ch*).CORRection.COLLect.ECAL.ERESponse = *Eports*

Description

This command executes enhanced response calibration between the two specified ports of the selected channel using the ECal (Electrical Calibration) module.

NOTE If you execute this command when the ECal module is not connected or when ports are not connected each other appropriately, an error occurs and the command is ignored.

Variable

Parameter	*Eports*
Description	Indicates 2-element array data (port number). • *EPorts(0):* Specifies the response port. • *EPorts(1):* Specifies the stimulus port. The index of the array starts from 0.
Data type	Variant type (Variant)
Range	1 to 2
Resolution	1
Note	For each parameter, you must specify a different port number. If you specify the same port number for 2 or more parameters, an error occurs and the command is ignored.

For information on the variable (*Ch*), see Ch.

Examples

```
Dim ERESPort(1) As Variant
ERESPort(0) = 1
ERESPort(1) = 2
SCPI.SENSe(1).CORRection.COLLect.ECAL.ERESponse = ERESPort
```

Equivalent key

Cal > ECal > Enhanced Response > 1-2 (S12 S22)|2-1 (S21 S11)

Equivalent SCPI command

Syntax

```
:SENSe{[1]-4}:CORRection:COLLect:ECAL:ERESponse <numeric 1>,<numeric 2>
```

Example of use

```
10 OUTPUT 717;":SENS1:CORR:COLL:ECAL:ERES 1,2"
20 OUTPUT 717;"*OPC?"
30 ENTER 717;A
```

SCPI.SENSe(*Ch*).CORRection.COLLect.ECAL.INFormation

Object type

Property (**Read Only**)

Syntax

Param = SCPI.SENSe(*Ch*).CORRection.COLLect.ECAL.INFormation?

Description

This command gets information of the connected ECAL modules to the E5061B.

Variable

The command return information in a string variable <param> in the following syntax:

[For 2-port ECal]

- ModelNumber
- SerialNumber
- PortAConnector
- PortBConnector
- MinFreq
- MaxFreq
- NumberOfPoints
- Calibrated
- ID
- PortAExtension
- PortBExtension
- Analyzer
- Operator
- Location

[For 4-port ECal]

- ModelNumber
- SerialNumber
- PortAConnector
- PortBConnector
- PortCConnector
- PortDConnector
- MinFreq
- MaxFreq

- NumberOfPoints
- Calibrated
- ID
- PortAExtension
- PortBExtension
- PortCExtension
- PortDExtension
- Analyzer
- Operator
- Location

Examples

```
Dim Cal_Info As String
Cal_Info = SCPI.SENS1.CORRection.COLLect.ECAL.INFormation
```

Related objects

SCPI.SENSe(Ch).CORRection.COLLect.SAVE

SCPI.SENSe(Ch).CORRection.TYPE(Tr)

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

:SENSe{[1]-4}:CORRection:COLLect:ECAL:INFormation?

Query response

{Model number,Serial number,Connector type, Calibration date, Min and max frequency}<newline><^END>

Example of use

```
10 OUTPUT 717;":SENS1:CORR:COLL:ECAL:INF?"
20 ENTER 717;A$
```

SCPI.SENSe(*Ch*).CORRection.COLLect.ECAL.ISOLation.STATe

Object type

Property (**Read-Write**)

Syntax

SCPI.SENSe(*Ch*).CORRection.COLLect.ECAL.ISOLation.STATe = *Status*

Status = SCPI.SENSe(*Ch*).CORRection.COLLect.ECAL.ISOLation.STATe

Description

This command ignores ON/OFF of the isolation measurement when executing Ecal for the selected channel (*Ch*), because the isolation of the E5061B is better than that of the ECal. This command takes no action and only exists to maintain backward compatibility.

Variable

Parameter	*Status*
Description	ON/OFF of the isolation measurement when executing ECal
Data type	Boolean type (Boolean)
Range	Select from either of the following: • True or ON: Turns ON the isolation measurement. • False or OFF: Turns OFF the isolation measurement.
Preset value	False or OFF

Examples

```
Dim EcalIso As Boolean
SCPI.SENSe1.CORRection.COLLect.ECAL.ISOLation.STATe = False
EcalIso = SCPI.SENSe1.CORRection.COLLect.ECAL.ISOLation.STATe
```

Related objects

SCPI.SENSe(Ch).CORRection.COLLect.ECAL.SOLT1

SCPI.SENSe(Ch).CORRection.COLLect.ECAL.SOLT2

Equivalent key

No equivalent key is available on front panel.

Equivalent SCPI command

Syntax

:SENSe{[1]-4}:CORRection:COLLect:ECAL:ISOLation[:STATe] {ON|OFF|1|0}

:SENSe{[1]-4}:CORRection:COLLect:ECAL:ISOLation[:STATe]?

Query response

{1|0}<newline><^END>

Example of use

```
10 OUTPUT 717;":SENS1:CORR:COLL:ECAL:ISOL OFF"
20 OUTPUT 717;":SENS1:CORR:COLL:ECAL:ISOL?"
30 ENTER 717;A
```

SCPI.SENSe(*Ch*).CORRection.COLLect.ECAL.ORIentation.STATe

Object type

Property (**Read-Write**)

Syntax

SCPI.SENSe(*Ch*).CORRection.COLLect.ECAL.ORIentation.STATe = *Status*

Status = SCPI.SENSe(*Ch*).CORRection.COLLect.ECAL.ORIentation.STATe

Description

This command turns ON/OFF the ECal auto detect function.

Variable

Parameter	*Status*
Description	ON/OFF the ECal auto detect funcion.
Data type	Boolean type (Boolean)
Range	Select from either of the following: • True or ON: Turns ON the auto detect function. • False or OFF: Turns OFF the auto detect function.
Preset value	True or ON

For information on the variable (*Ch*), see *Ch*.

Examples

```
Dim EcalOri As Boolean
SCPI.SENSe(1).CORRection.COLLect.ECAL.ORIentation.STATe = True
EcalOri = SCPI.SENSe(1).CORRection.COLLect.ECAL.ORIentation.STATe
```

Related objects

SCPI.SENSe(Ch).CORRection.COLLect.ECAL.PATH(Cpt)

Equivalent key

Cal > **ECal** > **Orientation** > **Orientation**

Equivalent SCPI command

Syntax

:SENSe{[1]-4}:CORRection:COLLect:ECAL:ORIentation[:STATe] {ON|OFF|1|0}

:SENSe{[1]-4}:CORRection:COLLect:ECAL:ORIentation[:STATe]?

Query response

{0|1}<newline><^END>

Example of use

```
10 OUTPUT 717;":SENS1:CORR:COLL:ECAL:ORI ON"
20 OUTPUT 717;":SENS1:CORR:COLL:ECAL:ORI?"
30 ENTER 717;A
```

SCPI.SENSe(*Ch*).CORRection.COLLect.ECAL.PATH(*Cpt*)

Object type

Property (**Read-Write**)

Syntax

SCPI.SENSe(Ch).CORRection.COLLect.ECAL.PATH(*Cpt*) = *Value*

Value = SCPI.SENSe(Ch).CORRection.COLLect.ECAL.PATH(*Cpt*)

Description

This command sets/gets the ECal module n port number which is connected to a specified port.

Variable

Parameter	*Value*
Description	Port of ECal module.
Data type	Long integer type (Long)
Range	One of the following is set/get: • 0: Nothing is connected. • 1: Port A is connected. • 2: Port B is connected. • 3: Port C is connected. • 4: Port D is connected.

For information on the variable (*Cpt*), see *Cpt*.

Examples

```
Dim ECalPort As Long
SCPI.SENSe.CORRection.COLLect.ECAL.PATH(1) = 3
ECalPort = SCPI.SENSe.CORRection.COLLect.ECAL.PATH(1)
```

Related objects

SCPI.SENSe(Ch).CORRection.COLLect.ECAL.ORIentation.STATe

Equivalent key

Cal > **ECal** > **Orientation** > **Port1|Port2** > **A|B|C|D|None**

Equivalent SCPI command

Syntax

:SENSe{[1]-4}:CORRection:COLLect:ECAL:PATH <numeric 1>,<numeric 2>

:SENSe{[1]-4}:CORRection:COLLect:ECAL:PATH? <numeric 2>

Example of use

```
10 OUTPUT 717;":SENS1:CORR:COLL:ECAL:PATH 1,2"
```

SCPI.SENSe(*Ch*).CORRection.COLLect.ECAL.SOLT1

Object type

Method (**Write Only**)

Syntax

SCPI.SENSe(*Ch*).CORRection.COLLect.ECAL.SOLT1 = *Eport*

Description

This command executes 1-port calibration of the specified port of selected channel (*Ch*) using the ECal (Electronic Calibration) module.

NOTE If this command is executed when the ECal module is not connected or when ports are not connected each other appropriately, an error occurs and the command is ignored.

Variable

Parameter	*Eport*
Description	Port number
Data type	Long integer type (Long)
Range	1 to 2
Note	If the specified variable is out of the allowable setup range, an error occurs when executed.

Examples

```
SCPI.SENSe(1).CORRection.COLLect.ECAL.SOLT1 = 1
```

Equivalent key

Cal > **ECal** > **1-Port Cal** > **Port 1|Port 2**

Equivalent SCPI command

Syntax

:SENSe{[1]-4}:CORRection:COLLect:ECAL:SOLT1 <numeric>

Example of use

```
10 OUTPUT 717;":SENS1:CORR:COLL:ECAL:SOLT1 1"
20 OUTPUT 717;"*OPC?"
30 ENTER 717;A
```

SCPI.SENSe(*Ch*).CORRection.COLLect.ECAL.SOLT2

Object type

Method (**Write Only**)

Syntax

SCPI.SENSe(*Ch*).CORRection.COLLect.ECAL.SOLT2 = *Eports*

Description

Executes full 2-port calibration between the specified 2 ports of channels 1 to 16 (*Ch*) using the ECal (Electronic Calibration) module.

NOTE If this command is executed when the ECal module is not connected or when ports are not connected each other appropriately, an error occurs and the command is ignored.

Variable

Parameter	*Eports*
Description	Indicates 2-element array data (port number). • *EPorts(0)* *EPorts*(1) Specifies the port numbers for 2-port ECal. The index of the array starts from 0.
Data type	Variant type (Variant)
Range	1 to 2
Resolution	1
Note	If the specified variable is out of the allowable setup range, an error occurs when executed. If you specify the same port number to 2 port numbers, an error occurs when executed. The order of the 2 port numbers to be specified is arbitrary.

Examples

```
SCPI.SENSe(1).CORRection.COLLect.ECAL.SOLT2 = Array(1,2)
```

```
Dim EcalPort(1) As Variant
EcalPort(0) = 1
EcalPort(1) = 2
SCPI.SENSe(1).CORRection.COLLect.ECAL.SOLT2 = EcalPort
```

Equivalent key

Cal > **ECal** > **2-Port Cal**

Equivalent SCPI command

Syntax

:SENSe{[1]-4}:CORRection:COLLect:ECAL:SOLT2 <numeric 1>,<numeric 2>

Example of use

```
10 OUTPUT 717;":SENS1:CORR:COLL:ECAL:SOLT2 1,2"
20 OUTPUT 717;"*OPC?"
30 ENTER 717;A
```

SCPI.SENSe(*Ch*).CORRection.COLLect.ECAL.THRU

Object type

Method (**Write Only**)

Syntax

SCPI.SENSe(*Ch*).CORRection.COLLect.ECAL.THRU = *Eports*

Description

This command executes response calibration (thru) between the specified 2 ports of the selected channel (*Ch*) using the ECal (Electronic Calibration) module.

If this command is executed when the ECal module is not connected or when ports are not connected to each other appropriately, an error occurs and the command is ignored.

Variable

Parameter	*Eports*
Description	Indicates 2-element array data (port number). • *Ports(0):* Specifies the response port number. • *Ports(1):* Specifies the stimulus port number. The index of the array starts from 0.
Data type	Variant type (Variant)
Range	1 to 4
Resolution	1
Note	If the specified variable is out of the allowable setup range, an error occurs when executed. If you specify the same port number to 2 port numbers, an error occurs when executed. The order of the 2 port numbers to be specified is arbitrary.

Examples

```
SCPI.SENSe(1).CORRection.COLLect.ECAL.THRU = Array(1,2)

Dim EcalPort(1) As Variant
EcalPort(0) = 1
```

```
EcalPort(1) = 2
SCPI.SENSe(1).CORRection.COLLect.ECAL.THRU = EcalPort
```

Equivalent key

Cal > **ECal** > **Thru Cal** > **1-2 (S12)|2-1 (S21)**

Equivalent SCPI command

Syntax

```
:SENSe{[1]-4}:CORRection:COLLect:ECAL:THRU <numeric 1>,<numeric 2>
```

Example of use

```
10 OUTPUT 717;":SENS1:CORR:COLL:ECAL:THRU 1,2"
20 OUTPUT 717;"*OPC?"
30 ENTER 717;A
```

SCPI.SENSe(*Ch*).CORRection.COLLect.ECAL.UCHar

Object type

Property (**Read-Write**)

Syntax

SCPI.SENSe(*Ch*).CORRection.COLLect.ECAL.UCHar = *Param*

Param = SCPI.SENSe(*Ch*).CORRection.COLLect.ECAL.UCHar

Description

This command sets the ECal characteristic used when executing the user-defined ECal, for the selected channel (*Ch*).

NOTE The user-defined ECal is a type of ECal that is executed using the characteristic that has been acquired by the user and stored in the memory for ECal.

When the ECal module is not connected or the characteristic is not stored at the specified location number, executing this object causes a runtime error.

Variable

Parameter	*Param*
Description	Characteristic used when executing ECal (user characterization)
Data type	Character string type (String)
Range	Select from either of the following: • "CHAR0": Uses the factory-default characteristic. (Normal ECal) • "CHAR1": Uses the characteristic stored at location number 1 in the ECal's flash memory. • "CHAR2": Uses the characteristic stored at location number 2 in the ECal's flash memory. • "CHAR3": Uses the characteristic stored at location number 3 in the ECal's flash memory. • "CHAR4": Uses the characteristic stored at location number 4 in the ECal's flash memory. • "CHAR5": Uses the characteristic stored at

	location number 5 in the ECal's flash memory.
Preset value	"CHAR0"

Examples

```
Dim UserChar As String
SCPI.SENSe(1).CORRection.COLLect.ECAL.UCHar = "CHAR2"
UserChar = SCPI.SENSe(1).CORRection.COLLect.ECAL.UCHar
```

Equivalent key

Cal > **ECal** > **Characterization** > **Factory|User1|User2|User3|User4|User5**

Equivalent SCPI command

Syntax

:SENSe{[1]-4}:CORRection:COLLect:ECAL:UCHar {CHAR0|CHAR1|CHAR2|CHAR3|CHAR4|CHAR5}

:SENSe{[1]-4}:CORRection:COLLect:ECAL:UCHar?

Query response

{CHAR0|CHAR1|CHAR2|CHAR3|CHAR4|CHAR5}<newline><^END>

Example of use

```
10 OUTPUT 717;":SENS1:CORR:COLL:ECAL:UCH CHAR1"
20 OUTPUT 717;":SENS1:CORR:COLL:ECAL:UCH?"
30 ENTER 717;A$
```

SCPI.SENSe(*Ch*).CORRection.COLLect.GPACquire.ISOLation

Object Type

Method (**Write Only**)

Syntax

SCPI.SENSe(*Ch*).CORRection.COLLect.GPACquire.ISOLation

Description

This command measures the calibration data of the isolation from the T-port to the R-port using gain phase measurement, for the selected channel (*Ch*).

Examples

SCPI.SENSe(1).CORRection.COLLect.GPACquire.ISOLation

Related Objects

SCPI.SENSe(Ch).CORRection.COLLect.GPACquire.LOAD

SCPI.SENSe(Ch).CORRection.COLLect.GPACquire.OPEN

SCPI.SENSe(Ch).CORRection.COLLect.GPACquire.SHORt

SCPI.SENSe(Ch).CORRection.COLLect.GPACquire.THRU

Equivalent Key

Cal > **Calibrate** > **Response (Thru)** > **Isolation (Optional)**

Equivalent SCPI Command

Syntax

:SENSe{[1]-4}:CORRection:COLLect:GPACquire:ISOLation

Example of use

10 OUTPUT 717;":SENS1:CORR:COLL:GPAC:ISOL"

SCPI.SENSe(*Ch*).CORRection.COLLect.GPACquire.LOAD

Object Type

Method (**Write Only**)

Syntax

SCPI.SENSe(*Ch*).CORRection.COLLect.GPACquire.LOAD

Description

This command measures the calibration data of the load standard for the T-port using gain phase measurement, for the selected channel (*Ch*).

Examples

SCPI.SENSe(2).CORRection.COLLect.GPACquire.LOAD

Related Objects

SCPI.SENSe(Ch).CORRection.COLLect.GPACquire.ISOLation

SCPI.SENSe(Ch).CORRection.COLLect.GPACquire.OPEN

SCPI.SENSe(Ch).CORRection.COLLect.GPACquire.SHORt

SCPI.SENSe(Ch).CORRection.COLLect.GPACquire.THRU

Equivalent Key

Cal > **Calibrate** > **Response (Open)** > **Load (Optional)**

Cal > **Calibrate** > **Response (Short)** > **Load (Optional)**

Cal > **Calibrate** > **1-Port Cal** > **Load**

Equivalent SCPI Command

Syntax

:SENSe{[1]-4}:CORRection:COLLect:GPACquire:LOAD

Example of use

10 OUTPUT 717;":SENS1:CORR:COLL:GPAC:LOAD"

SCPI.SENSe(*Ch*).CORRection.COLLect.GPACquire.OPEN

Object Type

Method (**Write Only**)

Syntax

SCPI.SENSe(*Ch*).CORRection.COLLect.GPACquire.OPEN

Description

This command measures the calibration data of the open standard for the T-port using gain phase measurement, for the selected channel (*Ch*).

Examples

SCPI.SENSe(1).CORRection.COLLect.GPACquire.OPEN

Related Objects

SCPI.SENSe(Ch).CORRection.COLLect.GPACquire.ISOLation

SCPI.SENSe(Ch).CORRection.COLLect.GPACquire.LOAD

SCPI.SENSe(Ch).CORRection.COLLect.GPACquire.SHORt

SCPI.SENSe(Ch).CORRection.COLLect.GPACquire.THRU

Equivalent Key

Cal > **Calibrate** > **Response (Open)** > **Open**

Cal > **Calibrate** > **1-Port Cal** > **Open**

Equivalent SCPI Command

Syntax

:SENSe{[1]-4}:CORRection:COLLect:GPACquire:OPEN

Example of use

10 OUTPUT 717;":SENS1:CORR:COLL:GPAC:OPEN"

SCPI.SENSe(*Ch*).CORRection.COLLect.GPACquire.SHORt

Object Type

Method (**Write Only**)

Syntax

SCPI.SENSe(*Ch*).CORRection.COLLect.GPACquire.SHORt

Description

This command measures the calibration data of the short standard for the T-port using gain phase measurement, for the selected channel (*Ch*).

Examples

SCPI.SENSe(1).CORRection.COLLect.GPACquire.SHORt

Related Objects

SCPI.SENSe(Ch).CORRection.COLLect.GPACquire.ISOLation

SCPI.SENSe(Ch).CORRection.COLLect.GPACquire.LOAD

SCPI.SENSe(Ch).CORRection.COLLect.GPACquire.OPEN

SCPI.SENSe(Ch).CORRection.COLLect.GPACquire.THRU

Equivalent Key

Cal > **Calibrate** > **Response (Short)** > **Short**

Cal > **Calibrate** > **1-Port Cal** > **Short**

Equivalent SCPI Command

Syntax

:SENSe{[1]-4}:CORRection:COLLect:GPACquire:SHORt

Example of use

10 OUTPUT 717;":SENS1:CORR:COLL:GPAC:SHOR"

SCPI.SENSe(*Ch*).CORRection.COLLect.GPACquire.THRU

Object Type

Method (**Write Only**)

Syntax

SCPI.SENSe(*Ch*).CORRection.COLLect.GPACquire.THRU

Description

This command measures the calibration data of the thru standard from the T-port to the R-port using gain phase measurement, for the selected channel (*Ch*).

Examples

```
SCPI.SENSe(1).CORRection.COLLect.GPACquire.THRU
```

Related Objects

SCPI.SENSe(Ch).CORRection.COLLect.GPACquire.ISOLation

SCPI.SENSe(Ch).CORRection.COLLect.GPACquire.LOAD

SCPI.SENSe(Ch).CORRection.COLLect.GPACquire.OPEN

SCPI.SENSe(Ch).CORRection.COLLect.GPACquire.SHORt

Equivalent Key

Cal > **Calibrate** > **Response (Thru)** > **Thru**

Equivalent SCPI Command

Syntax

:SENSe{[1]-4}:CORRection:COLLect:GPACquire:THRU

Example of use

```
10 OUTPUT 717;":SENS1:CORR:COLL:GPAC:THRU"
```

SCPI.SENSe(*Ch*).CORRection.COLLect.METHod.ADAPter.REMoval

Object type

Property (**Write Only**)

Syntax

SCPI.SENSe(*Ch*).CORRection.COLLect.METHod.ADAPter.REMoval = *Port*

Description

This command sets the port in which adaptor needs to be added/removed.

Variable

See *Port*.

Examples

SCPI.SENSe(Ch).CORRection.COLLect.METHod.ADAPter.REMoval = 1

Related objects

SCPI.SENSe(Ch).CORRection.COLLect.ADAPter(Pt).LENGth

Equivalent key

Cal > **Calibrate** > **Adapter Removal** > **Port1|Port2**

Equivalent SCPI command

Syntax

:SENSe{[1]-4}:CORRection:COLLect:METHod:ADAPter:REMoval {1-2}

Example of use

Call viVPrintf(vi, ":SENS1:CORR:COLL:METH:ADAP:REM 1"+vbcr, 0)

SCPI.SENSe(*Ch*).CORRection.COLLect.METHod.ERESponse

Object type

Property (**Write Only**)

Syntax

SCPI.SENSe(*Ch*).CORRection.COLLect.METHod.ERESponse = *Ports*

Description

This command sets the calibration type to the enhanced response calibration between the two specified ports, for the selected channel.

Variable

Parameter	*Ports*
Description	Indicates 2-element array data (port number). • *Ports(0):* Specifies the response port. • *Ports(1):* Specifies the stimulus port. The index of the array starts from 0.
Data type	Variant type (Variant)
Range	1 to 2
Resolution	1
Note	For each parameter, you must specify a different port number. If you specify the same port number for 2 or more parameters, an error occurs and the command is ignored.

For information on the variable (*Ch*), see Ch.

Examples

```
Dim ERESPort(1) As Variant
ERESPort(0) = 1
ERESPort(1) = 2
SCPI.SENSe(1).CORRection.COLLect.METHod.ERESponse = ERESPort
```

Related objects

SCPI.SENSe(Ch).CORRection.COLLect.METHod.TYPE

Equivalent key

Cal > **Calibrate** > **Enhanced Response** > **Select Ports** > **1-2 (S12 S22)|2-1 (S21 S11)**

Equivalent SCPI command

Syntax

```
:SENSe{[1]-4}:CORRection:COLLect:METHod:ERESponse <numeric 1>,<numeric 2>
```

Example of use

```
10 OUTPUT 717;":SENS1:CORR:COLL:METH:ERES 1,2"
```

SCPI.SENSe(*Ch*).CORRection.COLLect.METHod.GPResponse.OPEN

Object Type

Method (**Write Only**)

Syntax

SCPI.SENSe(*Ch*).CORRection.COLLect.METHod.GPResponse.OPEN

Description

This command sets the calibration type to the response calibration (open) of the T-port using gain phase measurement, for the selected channel (*Ch*).

Examples

SCPI.SENSe(1).CORRection.COLLect.METHod.GPResponse.OPEN

Related Objects

SCPI.SENSe(Ch).CORRection.COLLect.METHod.GPResponse.SHORt

SCPI.SENSe(Ch).CORRection.COLLect.METHod.GPResponse.THRU

SCPI.SENSe(Ch).CORRection.COLLect.METHod.GPS1

Equivalent Key

Cal > **Calibrate** > **Response (Open)** > **Select Port** > **Gain Phase**

Equivalent SCPI Command

Syntax

:SENSe{[1]-4}:CORRection:COLLect:METHod:GPResponse:OPEN

Example of use

```
10 OUTPUT 717;":SENS1:CORR:COLL:METH:GPR:OPEN"
```

SCPI.SENSe(*Ch*).CORRection.COLLect.METHod.GPResponse.SHORt

Object Type

Method (**Write Only**)

Syntax

SCPI.SENSe(*Ch*).CORRection.COLLect.METHod.GPResponse.SHORt

Description

This command sets the calibration type to the response calibration (short) of the T-port using gain phase measurement, for the selected channel (*Ch*).

Examples

SCPI.SENSe(1).CORRection.COLLect.METHod.GPResponse.SHORt

Related Objects

SCPI.SENSe(Ch).CORRection.COLLect.METHod.GPResponse.OPEN

SCPI.SENSe(Ch).CORRection.COLLect.METHod.GPResponse.THRU

SCPI.SENSe(Ch).CORRection.COLLect.METHod.GPS1

Equivalent Key

Cal > **Calibrate** > **Response (Short)** > **Select Port** > **Gain Phase**

Equivalent SCPI Command

Syntax

:SENSe{[1]-4}:CORRection:COLLect:METHod:GPResponse:SHORt

Example of use

10 OUTPUT 717;":SENS1:CORR:COLL:METH:GPR:SHOR"

SCPI.SENSe(*Ch*).CORRection.COLLect.METHod.GPResponse.THRU

Object Type

Method (**Write Only**)

Syntax

SCPI.SENSe(*Ch*).CORRection.COLLect.METHod.GPResponse.THRU

Description

This command sets the calibration type to the response calibration (thru) between the T-port and the R-port using gain phase measurement, for the selected channel (*Ch*).

Examples

SCPI.SENSe(1).CORRection.COLLect.METHod.GPResponse.THRU

Related Objects

SCPI.SENSe(Ch).CORRection.COLLect.METHod.GPResponse.OPEN

SCPI.SENSe(Ch).CORRection.COLLect.METHod.GPResponse.SHORt

SCPI.SENSe(Ch).CORRection.COLLect.METHod.GPS1

Equivalent Key

Cal > **Calibrate** > **Response (Thru)** > **Select Ports** > **GP Port**

Equivalent SCPI Command

Syntax

:SENSe{[1]-4}:CORRection:COLLect:METHod:GPResponse:THRU

Example of use

10 OUTPUT 717;":SENS1:CORR:COLL:METH:GPR:THRU"

SCPI.SENSe(*Ch*).CORRection.COLLect.METHod.GPS1

Object Type

Method (**Write Only**)

Syntax

SCPI.SENSe(*Ch*).CORRection.COLLect.METHod.GPS1

Description

This command sets the calibration type to the full 1-port SOLT calibration of the T-port using gain phase measurement, for the selected channel (*Ch*).

Examples

SCPI.SENSe(2).CORRection.COLLect.METHod.GPS1

Related Objects

SCPI.SENSe(Ch).CORRection.COLLect.METHod.GPResponse.OPEN

SCPI.SENSe(Ch).CORRection.COLLect.METHod.GPResponse.SHORt

SCPI.SENSe(Ch).CORRection.COLLect.METHod.GPResponse.THRU

Equivalent Key

Cal > **Calibrate** > **1-Port Cal** > **Select Port** > **Gain Phase**

Equivalent SCPI Command

Syntax

:SENSe{[1]-4}:CORRection:COLLect:METHod:GPS1

Example of use

10 OUTPUT 717;":SENS1:CORR:COLL:METH:GPS1"

SCPI.SENSe(*Ch*).CORRection.COLLect.METHod.RESPonse.OPEN

Object type

Property (**Write Only**)

Syntax

SCPI.SENSe(*Ch*).CORRection.COLLect.METHod.RESPonse.OPEN = *Port*

Description

This command sets the calibration type to the response calibration (open) of the specified port, for the selected channel (*Ch*).

Examples

SCPI.SENSe(1).CORRection.COLLect.METHod.RESPonse.OPEN = 1

Related objects

SCPI.SENSe(Ch).CORRection.COLLect.METHod.TYPE

Equivalent key

Cal > **Calibrate** > **Response (Open)** > **Select Port** > **1|2**

Equivalent SCPI command

Syntax

:SENSe{[1]-4}:CORRection:COLLect:METHod[:RESPonse]:OPEN <numeric>

Example of use

```
10 OUTPUT 717;":SENS1:CORR:COLL:METH:OPEN 1"
```

SCPI.SENSe(*Ch*).CORRection.COLLect.METHod.RESPonse.SHORt

Object type

Property (**Write Only**)

Syntax

SCPI.SENSe(*Ch*).CORRection.COLLect.METHod.RESPonse.SHORt = *Port*

Description

This command sets the calibration type to the response calibration (short) of the specified port, for the selected channel (*Ch*).

Examples

SCPI.SENSe(1).CORRection.COLLect.METHod.RESPonse.SHORt = 1

Related objects

SCPI.SENSe(Ch).CORRection.COLLect.METHod.TYPE

Equivalent key

Cal > **Calibrate** > **Response (Short)** > **Select Port** > **1|2**

Equivalent SCPI command

Syntax

:SENSe{[1]-4}:CORRection:COLLect:METHod[:RESPonse]:SHORt <numeric>

Example of use

```
10 OUTPUT 717;":SENS1:CORR:COLL:METH:SHOR 1"
```

SCPI.SENSe(*Ch*).CORRection.COLLect.METHod.RESPonse.THRU

Object type

Property (**Write Only**)

Syntax

SCPI.SENSe(*Ch*).CORRection.COLLect.METHod.RESPonse.THRU = *Ports*

Description

This command sets the calibration type to the response calibration (thru) between the specified 2 ports, for the selected channel (*Ch*).

Variable

For information on the variable (*Ch*) and the variable (*Port*), see Ch and Port.

Examples

```
SCPI.SENSe(1).CORRection.COLLect.METHod.RESPonse.THRU = Array(2,1)

Dim ThruPort(1) As Variant
ThruPort(0) = 2
ThruPort(1) = 1
SCPI.SENSe(1).CORRection.COLLect.METHod.RESPonse.THRU = ThruPort
```

Related objects

SCPI.SENSe(Ch).CORRection.COLLect.METHod.TYPE

Equivalent key

Cal > **Calibrate** > **Response (Thru)** > **Select Ports** > **1-2 (S12)|2-1 (S21)**

Equivalent SCPI command

Syntax

:SENSe{[1]-4}:CORRection:COLLect:METHod[:RESPonse]:THRU <numeric 1>,<numeric 2>

Example of use

```
10 OUTPUT 717;":SENS1:CORR:COLL:METH:THRU 1,2"
```

SCPI.SENSe(*Ch*).CORRection.COLLect.METHod.SOLT1

Object type

Property (**Write Only**)

Syntax

SCPI.SENSe(*Ch*).CORRection.COLLect.METHod.SOLT1 = *Port*

Description

This command sets the calibration type to the 1-port calibration of the specified port, for the selected channel (*Ch*).

Examples

SCPI.SENSe(1).CORRection.COLLect.METHod.SOLT1 = 1

Related objects

SCPI.SENSe(Ch).CORRection.COLLect.METHod.TYPE

Equivalent key

Cal > **Calibrate** > **1-Port Cal > Select Port** > **1|2**

Equivalent SCPI command

Syntax

:SENSe{[1]-4}:CORRection:COLLect:METHod:SOLT1 <numeric>

Example of use

```
10 OUTPUT 717;":SENS1:CORR:COLL:METH:SOLT1 1"
```

SCPI.SENSe(*Ch*).CORRection.COLLect.METHod.SOLT2

Object type

Property (**Write Only**)

Syntax

SCPI.SENSe(*Ch*).CORRection.COLLect.METHod.SOLT2 = *Ports*

Description

This command sets the calibration type to the full 2-port calibration between the specified 2 ports, for the selected channel (*Ch*).

Variable

Parameter	*Ports*
Description	Indicates 2-element array data (port number). • *Ports(0):* Specifies a port for full 2-port calibration. • *Ports(1):* Specifies the other port for full 2-port calibration. The index of the array starts from 0.
Data type	Variant type (Variant)
Range	1 to 2
Resolution	1
Note	If the specified variable is out of the allowable setup range, an error occurs when executed. If you specify the same port number to 2 port numbers, an error occurs when executed. The order of the 2 port numbers to be specified is arbitrary.

Examples

```
SCPI.SENSe(1).CORRection.COLLect.METHod.SOLT2 = Array(1,2)

Dim CalPort(1) As Variant
CalPort(0) = 1
CalPort(1) = 2
SCPI.SENSe(1).CORRection.COLLect.METHod.SOLT2 = CalPort
```

Related objects

SCPI.SENSe(Ch).CORRection.COLLect.METHod.TYPE

Equivalent key

Cal > Calibrate > 2-Port Cal

Equivalent SCPI command

Syntax

```
:SENSe{[1]-4}:CORRection:COLLect:METHod:SOLT2 <numeric
1>,<numeric 2>
```

Example of use

```
10 OUTPUT 717;":SENS1:CORR:COLL:METH:SOLT2 1,2"
```

SCPI.SENSe(*Ch*).CORRection.COLLect.METHod.TYPE

Object type

Property (**Read Only**)

Syntax

Param = SCPI.SENSe(*Ch*).CORRection.COLLect.METHod.TYPE

Description

This command reads the selected calibration type of selected channel (*Ch*).

NOTE This object is used to check the selected calibration type for calculating the calibration coefficients. To check the applied calibration type (error correction on), use the SCPI.SENSe(Ch).CORRection.TYPE(Tr) object.

Variable

Parameter	*Param*	
Description	Calibration type	
Range	AREM	Adaptor Removal
	ERES	Enhanced response calibration
	NONE	None
	RESPO	Response calibration (open)
	RESPS	Response calibration (short)
	RESPT	Response calibration (thru)
	SOLT1	1-port calibration.
	SOLT2	Full 2-port calibration.
Preset Value	NONE	

Data type	Character string type (String)

Examples

```
Dim CalType As String
CalType = SCPI.SENSe(1).CORRection.COLLect.METHod.TYPE
```

Related objects

SCPI.SENSe(Ch).CORRection.COLLect.SAVE

SCPI.SENSe(Ch).CORRection.TYPE(Tr)

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

:SENSe{[1]-4}:CORRection:COLLect:METHod:TYPE?

Query response

{AREM|ERES|NONE|RESPO|RESPS|RESPT|SOLT1|SOLT2}<newline><^END>

Example of use

```
10 OUTPUT 717;":SENS1:CORR:COLL:METH:TYPE?"
20 ENTER 717;A$
```

SCPI.SENSe(*Ch*).CORRection.COLLect.PARTial.SAVE

Object type

Method (**Write Only**)

Syntax

SCPI.SENSe(*Ch*).CORRection.COLLect.PARTial.SAVE

Description

This command, used for partial overwrite, recalculates the calibration coefficients depending on the selected calibration type from the measured calibration data.

NOTE Calculating the calibration coefficients clears all calibration data regardless of whether they are used for the calculation and also clears the calibration type selections.

If partial overwrite is executed before selecting the calibration type, an error occurs and the command is ignored.

Variable

For information on the variable (*Ch*), see Ch.

Examples

SCPI.SENSe(1).CORRection.COLLect.PARTial.SAVE

Equivalent key

Cal > **Calibrate** > **1-Port Cal|2-Port Cal** > **Overwrite**

Equivalent SCPI command

Syntax

:SENSe{[1]-4}:CORRection:COLLect:PARTial:SAVE

Example of use

10 OUTPUT 717;":SENS1:CORR:COLL:PART:SAVE"

SCPI.SENSe(*Ch*).CORRection.COLLect.SAVE

Object type

Method (**Write Only**)

Syntax

SCPI.SENSe(*Ch*).CORRection.COLLect.SAVE

Description

This command calculates the calibration coefficients depending on the calibration type selection, from the measured calibration data.

> **NOTE** Calculating the calibration coefficients clears all the measured calibration data whether or not used for the calculation and also clears the calibration type selection.
>
> If this command is executed before all necessary calibration data for calculating the calibration coefficients is measured, an error occurs and the command is ignored.

Variable

For information on the variable (*Ch*), see Ch.

Examples

```
Dim Dmy As Long
SCPI.SENSe(1).CORRection.COLLect.METHod.RESPonse.THRU = Array(2,1)
SCPI.SENSe(1).CORRection.COLLect.ACQuire.THRU = Array(2,1)
Dmy = SCPI.IEEE4882.OPC
SCPI.SENSe(1).CORRection.COLLect.SAVE
```

Related objects

SCPI.SENSe(Ch).CORRection.COLLect.METHod.RESPonse.OPEN

SCPI.SENSe(Ch).CORRection.COLLect.METHod.RESPonse.SHORt

SCPI.SENSe(Ch).CORRection.COLLect.METHod.RESPonse.THRU

SCPI.SENSe(Ch).CORRection.COLLect.METHod.SOLT1

SCPI.SENSe(Ch).CORRection.COLLect.METHod.SOLT2

Equivalent key

Cal > **Calibrate** > **Response(XXXX)|n-Port Cal|Enhanced Response** > **Done**

Cal > **Calibrate** > **Adapter Removal** > **Port1|Port2** > **Done**

Equivalent SCPI command

Syntax

:SENSe{[1]-4}:CORRection:COLLect:SAVE

Example of use

```
10 OUTPUT 717;":SENS1:CORR:COLL:SAVE"
```

SCPI.SENSe(*Ch*).CORRection.EXTension.AUTO.CONFig

Object type

Property (**Read-Write**)

Syntax

SCPI.SENSe(*Ch*).CORRection.EXTension.AUTO.CONFig = *Param*

Param = SCPI.SENSe(*Ch*).CORRection.EXTension.AUTO.CONFig

Description

This command sets/gets the frequency point to calculate the auto port extension, for the selected channel (*Ch*).

Variable

Parameter	*Param*
Description	The frequency point to calculate the auto port extension
Data type	Character string type (String)
Range	Select from either of the following: • "CSPN": Uses the frequency of the current sweep range. • "AMKR": Use the frequency of the active marker. This is applied to Loss 1 and Loss 2 is ignored. • "USPN": This is executed with the arbitrary specified start frequency and stop frequency.
Preset value	"CSPN"

Examples

```
Dim Conf As String
SCPI.SENSe(1).CORRection.EXTension.AUTO.CONFig = "AMKR"
Conf = SCPI.SENSe(1).CORRection.EXTension.AUTO.CONFig
```

Related objects

SCPI.SENSe(Ch).CORRection.EXTension.STATe

SCPI.SENSe(Ch).CORRection.EXTension.AUTO.DCOFfset

SCPI.SENSe(Ch).CORRection.EXTension.AUTO.LOSS

SCPI.SENSe(Ch).CORRection.EXTension.AUTO.MEASure

SCPI.SENSe(Ch).CORRection.EXTension.AUTO.PORT(Pt)

SCPI.SENSe(Ch).CORRection.EXTension.AUTO.RESet

SCPI.SENSe(Ch).CORRection.EXTension.AUTO.STARt

SCPI.SENSe(Ch).CORRection.EXTension.AUTO.STOP

Equivalent key

Cal > **Port Extensions** > **Auto Port Extension** > **Method** > **Current Span|Active Marker|User Span**

Equivalent SCPI command

Syntax

:SENSe{[1]-4}:CORRection:EXTension:AUTO:CONFig {CSPN|AMKR|USPN}

:SENSe{[1]-4}:CORRection:EXTension:AUTO:CONFig?

Query response

{CSPN|AMKR|USPN}<newline><^END>

Example of use

```
10 OUTPUT 717;":SENS1:CORR:EXT:AUTO:CONF CSPN"
20 OUTPUT 717;":SENS1:CORR:EXT:AUTO:CONF?"
30 ENTER 717;A$
```

SCPI.SENSe(*Ch*).CORRection.EXTension.AUTO.DCOFfset

Object type

Property (**Read-Write**)

Syntax

SCPI.SENSe(*Ch*).CORRection.EXTension.AUTO.DCOFfset = *Status*

Status = SCPI.SENSe(*Ch*).CORRection.EXTension.AUTO.DCOFfset

Description

This command enables/disables or gets the usage of DC Offset value for the results of the auto port extension, for the selected channel (*Ch*).

Variable

Parameter	*Status*
Description	ON/OFF the usage of DC Offset value for the results of the auto port extension
Data type	Boolean type (Boolean)
Range	Select from either of the following: • True or ON: Uses the DC Offset value for the results. • False or OFF: Does not use the DC Offset value for the results.
Preset value	False or OFF

Examples

```
Dim Dcof As Boolean
SCPI.SENSe(1).CORRection.EXTension.AUTO.DCOFfset = True
Dcof = SCPI.SENSe(1).CORRection.EXTension.AUTO.DCOFfset
```

Related objects

SCPI.CALCulate(Ch).PARameter(Tr).SELect

SCPI.CALCulate(Ch).SELected.CONVersion.FUNCtion

SCPI.SENSe(Ch).CORRection.EXTension.STATe

SCPI.SENSe(Ch).CORRection.EXTension.AUTO.CONFig

SCPI.SENSe(Ch).CORRection.EXTension.AUTO.LOSS

SCPI.SENSe(Ch).CORRection.EXTension.AUTO.MEASure

SCPI.SENSe(Ch).CORRection.EXTension.AUTO.PORT(Pt)

SCPI.SENSe(Ch).CORRection.EXTension.AUTO.RESet

SCPI.SENSe(Ch).CORRection.EXTension.AUTO.STARt

SCPI.SENSe(Ch).CORRection.EXTension.AUTO.STOP

Equivalent key

Cal > **Port Extensions** > **Auto Port Extension** > **Adjust Mismatch**

Equivalent SCPI command

Syntax

:SENSe{[1]-4}:CORRection:EXTension:AUTO:DCOFfset {ON|OFF|1|0}

:SENSe{[1]-4}:CORRection:EXTension:AUTO:DCOFfset?

Query response

{1|0}<newline><^END>

Example of use

```
10 OUTPUT 717;":SENS1:CORR:EXT:AUTO:DCOF ON"
20 OUTPUT 717;":SENS1:CORR:EXT:AUTO:DCOF?"
30 ENTER 717;A
```

SCPI.SENSe(*Ch*).CORRection.EXTension.AUTO.LOSS

Object type

Property (**Read-Write**)

Syntax

SCPI.SENSe(*Ch*).CORRection.EXTension.AUTO.LOSS = *Status*

Status = SCPI.SENSe(*Ch*).CORRection.EXTension.AUTO.LOSS

Description

This command turns ON/OFF or gets the status of the loss compensation for the results of the auto port extension, for the selected channel (*Ch*).

Variable

Parameter	*Status*
Description	ON/OFF the loss compensation for the results of the auto port extension
Data type	Boolean type (Boolean)
Range	Select from either of the following: • True or ON: Turns ON the loss compensation • False or OFF: Turns OFF the loss compensation
Preset value	False or OFF

Examples

```
Dim AutoLoss As Boolean
SCPI.SENSe(1).CORRection.EXTension.AUTO.LOSS = True
AutoLoss = SCPI.SENSe(1).CORRection.EXTension.AUTO.LOSS
```

Related objects

SCPI.SENSe(Ch).CORRection.EXTension.AUTO.CONFig

SCPI.SENSe(Ch).CORRection.EXTension.AUTO.DCOFfset

SCPI.SENSe(Ch).CORRection.EXTension.AUTO.MEASure

SCPI.SENSe(Ch).CORRection.EXTension.AUTO.PORT(Pt)

SCPI.SENSe(Ch).CORRection.EXTension.AUTO.RESet

SCPI.SENSe(Ch).CORRection.EXTension.AUTO.STARt

SCPI.SENSe(Ch).CORRection.EXTension.AUTO.STOP

Equivalent key

Cal > **Port Extensions** > **Auto Port Extension** > **Include Loss**

Equivalent SCPI command

Syntax

:SENSe{[1]-4}:CORRection:EXTension:AUTO:LOSS {ON|OFF|1|0}

:SENSe{[1]-4}:CORRection:EXTension:AUTO:LOSS?

Query response

{1|0}<newline><^END>

Example of use

```
10 OUTPUT 717;":SENS1:CORR:EXT:AUTO:LOSS ON"
20 OUTPUT 717;":SENS1:CORR:EXT:AUTO:LOSS?"
30 ENTER 717;A
```

SCPI.SENSe(*Ch*).CORRection.EXTension.AUTO.MEASure

Object type

Method (**Write Only**)

Syntax

SCPI.SENSe(*Ch*).CORRection.EXTension.AUTO.MEASure = *Param*

Description

This command measures the calibration data of the OPEN standard or SHORT standard of the auto port extension, for the selected channel (*Ch*).

Variable

Parameter	*Param*
Description	Standard type of the auto port extension
Data type	Character string type (String)
Range	Select from either of the following: • "OPEN": Measures the calibration data of the OPEN standard • "SHORt": Measures the calibration data of the SHORT standard
Preset value	"SHORt"

Examples

```
Dim AutoMeas As String
SCPI.SENSe(1).CORRection.EXTension.AUTO.MEASure = "OPEN"
AutoLoss = SCPI.SENSe(1).CORRection.EXTension.AUTO.LOSS
```

Related objects

SCPI.SENSe(Ch).CORRection.EXTension.AUTO.CONFig

SCPI.SENSe(Ch).CORRection.EXTension.AUTO.DCOFfset

SCPI.SENSe(Ch).CORRection.EXTension.AUTO.LOSS

SCPI.SENSe(Ch).CORRection.EXTension.AUTO.PORT(Pt)

SCPI.SENSe(Ch).CORRection.EXTension.AUTO.RESet

SCPI.SENSe(Ch).CORRection.EXTension.AUTO.STARt

SCPI.SENSe(Ch).CORRection.EXTension.AUTO.STOP

Equivalent key

Cal > **Port Extensions** > **Auto Port Extension** > **Measure OPEN|Measure Short** > **All|Port 1|Port 2**

Equivalent SCPI command

Syntax

:SENSe{[1]-4}:CORRection:EXTension:AUTO:MEASure {OPEN|SHORt}

:SENSe{[1]-4}:CORRection:EXTension:AUTO:MEASure?

Query response

{OPEN|SHOR}<newline><^END>

Example of use

```
10 OUTPUT 717;":SENS1:CORR:EXT:AUTO:MEAS OPEN"
20 OUTPUT 717;":SENS1:CORR:EXT:AUTO:MEAS?"
30 ENTER 717;A$
```

SCPI.SENSe(*Ch*).CORRection.EXTension.AUTO.PORT(*Pt*)

Object type

Property (**Read-Write**)

Syntax

SCPI.SENSe(*Ch*).CORRection.EXTension.AUTO.PORT(*Pt*) = *Status*

Status = SCPI.SENSe(*Ch*).CORRection.EXTension.AUTO.PORT(*Pt*)

Description

This command turns ON/OFF or gets the status of the auto port extension, for the selected channel (*Ch*).

Variable

Parameter	*Status*
Description	ON/OFF of the auto port extension
Data type	Boolean type (Boolean)
Range	Select from either of the following: • True or ON: Turns ON the auto port extension • False or OFF: Turns OFF the auto port extension
Preset value	True or ON

For information on the variable (*Ch*) or Pt, refer to Ch or Pt.

Examples

```
Dim APort As Boolean
SCPI.SENSe(1).CORRection.EXTension.AUTO.PORT(1) = True
APort = SCPI.SENSe(1).CORRection.EXTension.AUTO.PORT(1)
```

Related objects

SCPI.SENSe(Ch).CORRection.EXTension.AUTO.CONFig

SCPI.SENSe(Ch).CORRection.EXTension.AUTO.DCOFfset

SCPI.SENSe(Ch).CORRection.EXTension.AUTO.LOSS

SCPI.SENSe(Ch).CORRection.EXTension.AUTO.MEASure

SCPI.SENSe(Ch).CORRection.EXTension.AUTO.RESet

SCPI.SENSe(Ch).CORRection.EXTension.AUTO.STARt

SCPI.SENSe(Ch).CORRection.EXTension.AUTO.STOP

Equivalent key

Cal > **Port Extensions** > **Auto Port Extension** > **Select Ports** > **Port 1|Port 2**

Equivalent SCPI command

Syntax

:SENSe{[1]-4}:CORRection:EXTension:AUTO:PORT{[1]|2} {ON|OFF|1|0}

:SENSe{[1]-4}:CORRection:EXTension:AUTO:PORT{[1]|2}?

Query response

{1|0}<newline><^END>

Example of use

```
10 OUTPUT 717;":SENS1:CORR:EXT:AUTO:PORT1 ON"
20 OUTPUT 717;":SENS1:CORR:EXT:AUTO:PORT1?"
30 ENTER 717;A$
```

SCPI.SENSe(*Ch*).CORRection.EXTension.AUTO.RESet

Object type

Method (**Write Only**)

Syntax

SCPI.SENSe(*Ch*).CORRection.EXTension.AUTO.RESet

Description

This command deletes the finished measurement data (OPEN and SHORT), for the selected channel (*Ch*).

Examples

```
SCPI.SENSe(1).CORRection.EXTension.AUTO.RESet
```

Related objects

SCPI.SENSe(Ch).CORRection.EXTension.STATe

SCPI.SENSe(Ch).CORRection.EXTension.AUTO.CONFig

SCPI.SENSe(Ch).CORRection.EXTension.AUTO.DCOFfset

SCPI.SENSe(Ch).CORRection.EXTension.AUTO.LOSS

SCPI.SENSe(Ch).CORRection.EXTension.AUTO.MEASure

SCPI.SENSe(Ch).CORRection.EXTension.AUTO.PORT(Pt)

SCPI.SENSe(Ch).CORRection.EXTension.AUTO.STARt

SCPI.SENSe(Ch).CORRection.EXTension.AUTO.STOP

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

:SENSe{[1]-4}:CORRection:EXTension:AUTO:RESet

Example of use

```
10 OUTPUT 717;":SENS1:CORR:EXT:AUTO:RES"
```

SCPI.SENSe(*Ch*).CORRection.EXTension.AUTO.STARt

Object type

Property (**Read-Write**)

Syntax

SCPI.SENSe(*Ch*).CORRection.EXTension.AUTO.STARt = *Value*

Value = SCPI.SENSe(*Ch*).CORRection.EXTension.AUTO.STARt

Description

This command gets/sets the start frequency within the frequency range of the user specified auto port extension, for the selected channel (*Ch*).

Variable

Parameter	*Value*
Description	Start frequency
Data type	Double precision floating point type (Double)
Range	5 to 3E9
Preset value	5
Unit	Hz (hertz)
Note	If the specified variable 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.

Examples

```
Dim AStart As Double
SCPI.SENSe(1).CORRection.EXTension.AUTO.STARt = 1E9
AStart = SCPI.SENSe(1).CORRection.EXTension.AUTO.STARt
```

Related objects

SCPI.SENSe(Ch).CORRection.EXTension.STATe

SCPI.SENSe(Ch).CORRection.EXTension.AUTO.CONFig

SCPI.SENSe(Ch).CORRection.EXTension.AUTO.DCOFfset

SCPI.SENSe(Ch).CORRection.EXTension.AUTO.LOSS

SCPI.SENSe(Ch).CORRection.EXTension.AUTO.MEASure

SCPI.SENSe(Ch).CORRection.EXTension.AUTO.PORT(Pt)

SCPI.SENSe(Ch).CORRection.EXTension.AUTO.RESet

SCPI.SENSe(Ch).CORRection.EXTension.AUTO.STOP

Equivalent key

Cal > Port Extensions > Auto Port Extension > Method > User Span Start

Equivalent SCPI command

Syntax

:SENSe{[1]-4}:CORRection:EXTension:AUTO:STARt <numeric>

:SENSe{[1]-4}:CORRection:EXTension:AUTO:STARt?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":SENS1:CORR:EXT:AUTO:STAR 1.2E9"
20 OUTPUT 717;":SENS1:CORR:EXT:AUTO:STAR?"
30 ENTER 717;A
```

SCPI.SENSe(*Ch*).CORRection.EXTension.AUTO.STOP

Object type

Property (**Read-Write**)

Syntax

SCPI.SENSe(*Ch*).CORRection.EXTension.AUTO.STOP = *Value*

Value = SCPI.SENSe(*Ch*).CORRection.EXTension.AUTO.STOP

Description

This command sets/gets the stop frequency within the frequency range of the user specified auto port extension, for the selected channel (*Ch*).

Variable

Parameter	*Value*
Description	Stop frequency
Data type	Double precision floating point type (Double)
Range	5 to 3E9
Preset value	3000000000
Unit	Hz (hertz)
Note	If the specified variable 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.

Examples

```
Dim AStop As Double
SCPI.SENSe(1).CORRection.EXTension.AUTO.STOP = 1E9
AStop = SCPI.SENSe(1).CORRection.EXTension.STOP.STARt
```

Related objects

SCPI.SENSe(Ch).CORRection.EXTension.STATe

SCPI.SENSe(Ch).CORRection.EXTension.AUTO.CONFig

SCPI.SENSe(Ch).CORRection.EXTension.AUTO.DCOFfset

SCPI.SENSe(Ch).CORRection.EXTension.AUTO.LOSS

SCPI.SENSe(Ch).CORRection.EXTension.AUTO.MEASure

SCPI.SENSe(Ch).CORRection.EXTension.AUTO.PORT(Pt)

SCPI.SENSe(Ch).CORRection.EXTension.AUTO.RESet

SCPI.SENSe(Ch).CORRection.EXTension.AUTO.STARt

Equivalent key

Cal > **Port Extensions** > **Auto Port Extension** > **Method** > **User Span Stop**

Equivalent SCPI command

Syntax

:SENSe{[1]-4}:CORRection:EXTension:AUTO:STOP <numeric>

:SENSe{[1]-4}:CORRection:EXTension:AUTO:STOP?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":SENS1:CORR:EXT:AUTO:STOP 1E9"
20 OUTPUT 717;":SENS1:CORR:EXT:AUTO:STOP?"
30 ENTER 717;A
```

SCPI.SENSe(*Ch*).CORRection.EXTension.PORT(*Pt*).FREQuency(*Fq*)

Object type

Property (**Read-Write**)

Syntax

SCPI.SENSe(*Ch*).CORRection.EXTension.PORT(*Pt*).FREQuency(*Fq*) = *Value*

Value = SCPI.SENSe(*Ch*).CORRection.EXTension.PORT(*Pt*).FREQuency(*Fq*)

Description

This command sets/gets the frequency used for calculation of the loss value of the frequency 1 and 2 (*Fq*) of the selected port (*Pt*), for the selected channel (*Ch*).

Variable

Parameter	*Fq*
Description	Frequency number
Data type	Long integer type (Long)
Range	1 to 2
Preset value	1
Note	If the specified variable is out of the allowable setup range, an error occurs when executed.

Parameter	*Value*
Description	Frequency
Data type	Double precision floating point type (Double)
Range	5 to 3E9
Preset value	1E9

Unit	Hz (hertz)
Note	If the specified variable 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.

Examples

```
Dim PortFreq As Double
SCPI.SENSe(1).CORRection.EXTension.PORT(1).FREQuency(1) = 500E6
PortFreq = SCPI.SENSe(1).CORRection.EXTension.PORT(1).FREQuency(1)
```

Related objects

SCPI.SENSe(Ch).CORRection.EXTension.STATe

SCPI.SENSe(Ch).CORRection.EXTension.PORT(Pt).INCLude(Il).STATe

SCPI.SENSe(Ch).CORRection.EXTension.PORT(Pt).LDC

SCPI.SENSe(Ch).CORRection.EXTension.PORT(Pt).LOSS(Loss)

Equivalent key

Cal > **Port Extensions** > **Extension Port 1|Extension Port 2** > **Loss** > **Freq1|Freq2**

Equivalent SCPI command

Syntax

:SENSe{[1]-4}:CORRection:EXTension:PORT{[1]|2}:FREQuency{[1]|2} <numeric>

:SENSe{[1]-4}:CORRection:EXTension:PORT{[1]|2}:FREQuency{[1]|2}?

Query response

<numeric><newline><^END>

Example of use

```
10 OUTPUT 717;":SENS1:CORR:EXT:PORT1:FREQ1 10E6"
20 OUTPUT 717;":SENS1:CORR:EXT:PORT1:FREQ1?"
30 ENTER 717;A
```

SCPI.SENSe(*Ch*).CORRection.EXTension.PORT(*Pt*).INCLude(*Il*).STATe

Object type

Property (**Read-Write**)

Syntax

SCPI.SENSe(*Ch*).CORRection.EXTension.PORT(*Pt*).INCLude(*Il*).STATe = *Status*

Status = SCPI.SENSe(*Ch*).CORRection.EXTension.PORT(*Pt*).INCLude(*Il*).STATe

Description

This command turns ON/OFF the set of loss value and frequency value of include 1 and 2 (*Il*) of the port 1 to 2 (*Pt*), for the selected channel (*Ch*).

Variable

Parameter	*Il*
Description	Include number
Data type	Long integer type (Long)
Range	1 to 2
Preset value	1
Note	If the specified variable is out of the allowable setup range, an error occurs when executed.

Parameter	*Status*
Description	ON/OFF the set of loss value and frequency value.
Data type	Boolean type (Boolean)
Range	Select from either of the following: • True or ON: Turns ON the loss value and frequency value.

	• False or OFF: Turns OFF the loss value and frequency value.
Preset value	False or OFF

Examples

```
Dim PortIncl As Double
SCPI.SENSe(1).CORRection.EXTension.PORT(1).INCLude(1).STATe = 500E6
PortIncl = SCPI.SENSe(1).CORRection.EXTension.PORT(1).INCLude(1).STATe
```

Related objects

SCPI.SENSe(Ch).CORRection.EXTension.STATe

SCPI.SENSe(Ch).CORRection.EXTension.PORT(Pt).FREQuency(Fq)

SCPI.SENSe(Ch).CORRection.EXTension.PORT(Pt).LDC

SCPI.SENSe(Ch).CORRection.EXTension.PORT(Pt).LOSS(Loss)

SCPI.SENSe(Ch).CORRection.EXTension.PORT(Pt).TIME

Equivalent key

Cal > **Port Extensions** > **Extension Port 1|Extension Port 2** > **Loss** > **Loss1|Loss2**

Equivalent SCPI command

Syntax

:SENSe{[1]-4}:CORRection:EXTension:PORT{[1]|2}:INCLude{[1]|2}[:STATe] {ON|OFF|1|0}

:SENSe{[1]-4}:CORRection:EXTension:PORT{[1]|2}:INCLude{[1]|2}[:STATe]?

Query response

{1|0}<newline><^END>

Example of use

```
10 OUTPUT 717;":SENS1:CORR:EXT:PORT1:INCL1 ON"
20 OUTPUT 717;":SENS1:CORR:EXT:PORT1:INCL1?"
30 ENTER 717;A
```

SCPI.SENSe(*Ch*).CORRection.EXTension.PORT(*Pt*).LDC

Object type

Property (**Read-Write**)

Syntax

SCPI.SENSe(*Ch*).CORRection.EXTension.PORT(*Pt*).LDC = *Value*

Value = SCPI.SENSe(*Ch*).CORRection.EXTension.PORT(*Pt*).LDC

Description

This command sets/gets the DC loss value of the port 1 to 2 (*Pt*), for the selected channel (*Ch*).

Variable

Parameter	*Value*
Description	The loss value of DC.
Data type	Double precision floating point type (Double)
Range	-90 to 90
Preset value	0
Unit	dB
Note	If the specified variable 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.

Examples

```
Dim PLdc As Double
SCPI.SENSe(1).CORRection.EXTension.PORT(1).LDC = 45
PLdc = SCPI.SENSe(1).CORRection.EXTension.PORT(1).LDC
```

Related objects

SCPI.SENSe(Ch).CORRection.EXTension.STATe

SCPI.SENSe(Ch).CORRection.EXTension.PORT(Pt).FREQuency(Fq)

SCPI.SENSe(Ch).CORRection.EXTension.PORT(Pt).INCLude(Il).STATe

SCPI.SENSe(Ch).CORRection.EXTension.PORT(Pt).LOSS(Loss)

SCPI.SENSe(Ch).CORRection.EXTension.PORT(Pt).TIME

Equivalent key

Cal > **Port Extensions** > **Extension Port 1|Extension Port 2** > **Loss** > **Loss at DC**

Equivalent SCPI command

Syntax

:SENSe{[1]-4}:CORRection:EXTension:PORT{[1]|2}:LDC <numeric>

:SENSe{[1]-4}:CORRection:EXTension:PORT{[1]|2}:LDC?

Query response

<numeric><newline><^END>

Example of use

```
10 OUTPUT 717;":SENS1:CORR:EXT:PORT1:LDC 1.2"
20 OUTPUT 717;":SENS1:CORR:EXT:PORT1:LDC?"
30 ENTER 717;A
```

SCPI.SENSe(*Ch*).CORRection.EXTension.PORT(*Pt*).LOSS(Loss)

Object type

Property (**Read-Write**)

Syntax

SCPI.SENSe(*Ch*).CORRection.EXTension.PORT(*Pt*).LOSS(*Loss*) = *Value*

Value = SCPI.SENSe(*Ch*).CORRection.EXTension.PORT(*Pt*).LOSS(*Loss*)

Description

This command sets/gets the loss value of the loss 1 to 2 of the port 1 to 2 (*Pt*), for the selected channel (*Ch*).

Variable

Parameter	*Loss*
Description	Loss number
Data type	Long integer type (Long)
Range	1 to 2
Preset value	1
Note	If the specified variable is out of the allowable setup range, an error occurs when executed.

Parameter	*Value*
Description	The loss value
Data type	Double precision floating point type (Double)
Range	-90 to 90
Preset value	0

Unit	dB
Note	If the specified variable 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.

Examples

```
Dim PLoss As Double
SCPI.SENSe(1).CORRection.EXTension.PORT(1).LOSS(1) = -45
PLoss = SCPI.SENSe(1).CORRection.EXTension.PORT(1).LOSS(1)
```

Related objects

SCPI.SENSe(Ch).CORRection.EXTension.STATe

SCPI.SENSe(Ch).CORRection.EXTension.PORT(Pt).FREQuency(Fq)

SCPI.SENSe(Ch).CORRection.EXTension.PORT(Pt).INCLude(Il).STATe

SCPI.SENSe(Ch).CORRection.EXTension.PORT(Pt).LDC

SCPI.SENSe(Ch).CORRection.EXTension.PORT(Pt).TIME

Equivalent key

Cal > **Port Extensions** > **Extension Port 1|Extension Port 2** > **Loss** > **Loss1|Loss2**

Equivalent SCPI command

Syntax

:SENSe{[1]-4}:CORRection:EXTension:PORT{[1]|2}:LOSS{[1]|2} <numeric>

:SENSe{[1]-4}:CORRection:EXTension:PORT{[1]|2}:LOSS{[1]|2}?

Query response

<numeric><newline><^END>

Example of use

```
10 OUTPUT 717;":SENS1:CORR:EXT:PORT1:LOSS1 0.8"
20 OUTPUT 717;":SENS1:CORR:EXT:PORT1:LOSS1?"
30 ENTER 717;A
```

SCPI.SENSe(*Ch*).CORRection.EXTension.PORT(*Pt*).TIME

Object type

Property (**Read-Write**)

Syntax

SCPI.SENSe(*Ch*).CORRection.EXTension.PORT(*Pt*).TIME = *Value*

Value = SCPI.SENSe(*Ch*).CORRection.EXTension.PORT(*Pt*).TIME

Description

This command sets/gets the delay time for the port extension of ports 1 and 2 (*Pt*), for the selected channel (*Ch*).

Variable

Parameter	*Value*
Description	Delay time
Data type	Double precision floating point type (Double)
Range	-10 to 10
Preset value	0
Unit	s (second)
Note	If the specified variable 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.

Examples

```
Dim PortExt As Double
SCPI.SENSe(1).CORRection.EXTension.PORT(1).TIME = 1E-3
PortExt = SCPI.SENSe(1).CORRection.EXTension.PORT(1).TIME
```

Related objects

SCPI.SENSe(Ch).CORRection.EXTension.STATe

SCPI.SENSe(Ch).CORRection.EXTension.PORT(Pt).FREQuency(Fq)

SCPI.SENSe(Ch).CORRection.EXTension.PORT(Pt).INCLude(Il).STATe

SCPI.SENSe(Ch).CORRection.EXTension.PORT(Pt).LDC

SCPI.SENSe(Ch).CORRection.EXTension.PORT(Pt).LOSS(Loss)

Equivalent key

Cal > **Port Extensions** > **Extension Port 1|Extension Port 2** > **Extension**

Equivalent SCPI command

Syntax

:SENSe{[1]-4}:CORRection:EXTension:PORT{[1]|2}:TIME <numeric>
:SENSe{[1]-4}:CORRection:EXTension:PORT{[1]|2}:TIME?

Query response

<numeric><newline><^END>

Example of use

```
10 OUTPUT 717;":SENS1:CORR:EXT:PORT1:TIME 1e-3"
20 OUTPUT 717;":SENS1:CORR:EXT:PORT1:TIME?"
30 ENTER 717;A
```

SCPI.SENSe(*Ch*).CORRection.EXTension.STATe

Object type

Property (**Read-Write**)

Syntax

SCPI.SENSe(*Ch*).CORRection.EXTension.STATe = *Status*

Status = SCPI.SENSe(*Ch*).CORRection.EXTension.STATe

Description

This command turns ON/OFF or returns the status of the port extension, for the selected channel (*Ch*).

Variable

Parameter	*Status*
Description	ON/OFF of the port extension correction
Data type	Boolean type (Boolean)
Range	Select from either of the following. • True or ON: Turns ON the port extension. • False or OFF: Turns OFF the port extension.
Preset value	False or OFF

Examples

```
Dim Ext As Boolean
SCPI.SENSe(1).CORRection.EXTension.STATe = True
Ext = SCPI.SENSe(1).CORRection.EXTension.STATe
```

Related objects

SCPI.SENSe(Ch).CORRection.EXTension.PORT(Pt).TIME

SCPI.SENSe(Ch).CORRection.EXTension.AUTO.DCOFfset

SCPI.SENSe(Ch).CORRection.EXTension.AUTO.LOSS

SCPI.SENSe(Ch).CORRection.EXTension.AUTO.MEASure

SCPI.SENSe(Ch).CORRection.EXTension.AUTO.PORT(Pt)

SCPI.SENSe(Ch).CORRection.EXTension.AUTO.RESet

SCPI.SENSe(Ch).CORRection.EXTension.PORT(Pt).FREQuency(Fq)

SCPI.SENSe(Ch).CORRection.EXTension.PORT(Pt).INCLude(Il).STATe

SCPI.SENSe(Ch).CORRection.EXTension.PORT(Pt).LDC

SCPI.SENSe(Ch).CORRection.EXTension.PORT(Pt).LOSS(Loss)

SCPI.SENSe(Ch).CORRection.EXTension.PORT(Pt).TIME

Equivalent key

Cal > **Port Extensions** > **Extensions**

Equivalent SCPI command

Syntax

:SENSe{[1]-4}:CORRection:EXTension[:STATe] {ON|OFF|1|0}

:SENSe{[1]-4}:CORRection:EXTension[:STATe]?

Query response

{1|0}<newline><^END>

Example of use

```
10 OUTPUT 717;":SENS1:CORR:EXT ON"
20 OUTPUT 717;":SENS1:CORR:EXT?"
30 ENTER 717;A
```

SCPI.SENSe(*Ch*).CORRection.PROPerty

Object type

Property (**Read-Write**)

Syntax

SCPI.SENSe(*Ch*).CORRection.PROPerty = *Status*

Status = SCPI.SENSe(*Ch*).CORRection.PROPerty

Description

This command turns ON/OFF or returns the display of the calibration property, for the active trace of selected channel (*Ch*).

Variable

Parameter	*Status*
Description	ON/OFF of the display of the calibration property
Data type	Boolean type (Boolean)
Range	Select from either of the following: • True or ON: Turns ON the display of the calibration property. • False or OFF: Turns OFF the display of the calibration property.
Preset value	False or OFF

Examples

```
Dim CalProp As Boolean
SCPI.SENSe(1).CORRection.PROPerty = True
CalProp = SCPI.SENSe(1).CORRection.PROPerty
```

Equivalent key

Cal > **Property**

Equivalent SCPI command

Syntax

:SENSe{[1]-4}:CORRection:PROPerty {ON|OFF|1|0}

:SENSe{[1]-4}:CORRection:PROPerty?

Query response

{1|0}<newline><^END>

Example of use

```
10 OUTPUT 717;":SENS1:CORR:PROP ON"
20 OUTPUT 717;":SENS1:CORR:PROP?"
30 ENTER 717;A
```

SCPI.SENSe(*Ch*).CORRection.RVELocity.COAX

Object type

Property (**Read-Write**)

Syntax

SCPI.SENSe(*Ch*).CORRection.RVELocity.COAX = *Value*

Value = SCPI.SENSe(*Ch*).CORRection.RVELocity.COAX

Description

This command sets/gets the velocity factor for the selected channel (*Ch*).

Variable

Parameter	*Value*
Description	Velocity factor
Data type	Double precision floating point type (Double)
Range	0.01 to 10
Preset value	1
Note	If the specified variable 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.

Examples

```
Dim Vel As Double
SCPI.SENSe(1).CORRection.RVELocity.COAX = 0.5
Vel = SCPI.SENSe(1).CORRection.RVELocity.COAX
```

Equivalent key

Cal > **Velocity Factor**

Equivalent SCPI command

Syntax

:SENSe{[1]-4}:CORRection:RVELocity:COAX <numeric>

:SENSe{[1]-4}:CORRection:RVELocity:COAX?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":SENS1:CORR:RVEL:COAX 0.7"
20 OUTPUT 717;":SENS1:CORR:RVEL:COAX?"
30 ENTER 717;A
```

SCPI.SENSe(*Ch*).CORRection.STATe

Object type

Property (**Read-Write**)

Syntax

SCPI.SENSe(*Ch*).CORRection.STATe = *Status*

Status = SCPI.SENSe(*Ch*).CORRection.STATe

Description

This command turns ON/OFF or gets the status of the error correction, for the active trace of the selected channel (*Ch*).

Variable

Parameter	*Status*
Description	ON/OFF of the error correction
Data type	Boolean type (Boolean)
Range	Select from either of the following: • True or ON: Turns ON the error correction. • False or OFF: Turns OFF the error correction.
Preset value	False or OFF

Examples

```
Dim Corr As Boolean
SCPI.SENSe(1).CORRection.STATe = True
Corr = SCPI.SENSe(1).CORRection.STATe
```

Equivalent key

Cal > **Correction**

Equivalent SCPI command

Syntax

:SENSe{[1]-4}:CORRection:STATe {ON|OFF|1|0}

:SENSe{[1]-4}:CORRection:STATe?

Query response

{1|0}<newline><^END>

Example of use

```
10 OUTPUT 717;":SENS1:CORR:STAT ON"
20 OUTPUT 717;":SENS1:CORR:STAT?"
30 ENTER 717;A
```

SCPI.SENSe(*Ch*).CORRection.TYPE(*Tr*)

Object type

Property (**Read Only**)

Syntax

Data = SCPI.SENSe(*Ch*).CORRection.TYPE(*Tr*)

Description

This command reads the information (calibration type, port numbers) of the applied calibration coefficients for the actual error correction, for selected trace (*Tr*) of the selected channel (*Ch*).

Variable

Parameter	*Data*
Description	Indicates 5 array data items (the calibration type and the port information to which the calibration is applied). • *Data(0):* The calibration type applied. For detail, refer to the Range section. • *Data(1):* The port number to which the calibration is applied (0 when the calibration type is NONE, GPRO, GPRS, GPRT, or GPS1). • *Data(2)":* The port number to which the calibration is applied (0 when the calibration type is not SOLT2). The array index starts from 0.
Range	One of the following is read out as *Data(0)*. • "ERES": The enhanced response calibration is applied. • "NONE": Nothing is applied. • "RESPO": The response calibration (open) is applied. • "RESPS": The response calibration (short) is applied. • "RESPT": The response calibration (thru) is applied. • "SOLT1": The 1-port calibration is applied. • "SOLT2": The full 2-port calibration is applied. • "GPRO": The gain phase response calibration

	(open) is applied. • "GPRS": The gain phase response calibration (short) is applied. • "GPRT": The gain phase response calibration (thru) is applied. • "GPS1": The gain phase 1-port calibration is applied.
Data type	Variant type (Variant)

Examples

```
Dim CalType As Variant
CalType = SCPI.SENSe(1).CORRection.TYPE(1)
```

Related objects

None.

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

:SENSe{[1]-4}:CORRection:TYPE{[1]-4}?

Query response

{ERES|NONE|RESPO|RESPS|RESPT|SOLT1|SOLT2|GPRO|GPRS|GPRT|GPS1},{numeric 1},{numeric 2}<newline><^END>

	Description
ERES	The enhanced response calibration is applied.
NONE	Any calibration is not applied.
RESPO	The response calibration (open) is applied.
RESPS	The response calibration (short) is applied.
RESPT	The response calibration (thru) is applied.

SOLT1	The 1-port calibration is applied.
SOLT2	The 2-port calibration is applied.
GPRO	The gain phase response calibration (open) is applied.
GPRS	The gain phase response calibration (short) is applied.
GPRT	The gain phase response calibration (thru) is applied.
GPS1	The gain phase 1-port calibration is applied.

{numeric 1}:

the calibration port number
(This parameter is 0 when the first parameter is NONE, GPRO, GPRS, GPRT, or GPS1.)

{numeric 2}:

the calibration port number
(This parameter is 0 when the first parameter is not ERES, RESPT, and SOLT2.)

Example of use

```
10 OUTPUT 717;":SENS1:CORR:TYPE1?"
20 ENTER 717;A$
```

SCPI.SENSe(*Ch*).DC.MEASure.CLEar

Object Type

Method (**Write Only**)

Syntax

SCPI.SENSe(*Ch*).DC.MEASure.CLEar

Description

This command clears the measurement data of DC monitor sweep end, for the selected channel (*Ch*).

Examples

SCPI.SENSe(1).DC.MEASure.CLEar

Related Objects

SCPI.SENSe(Ch).DC.MEASure.ENABle

SCPI.SENSe(Ch).DC.PARameter

Equivalent Key

Meas > **DC Monitor Setup** > **DC Monitor On Sweep End** > **Clear**

Equivalent SCPI Command

Syntax

:SENSe{[1]-4}:DC:MEASure:CLEar

Example of use

10 OUTPUT 717;":SENS1:DC:MEAS:CLE"

SCPI.SENSe(*Ch*).DC.MEASure.DATA

Object Type

Property (**Read Only**)

Syntax

Value = SCPI.SENSe(*Ch*).DC.MEASure.DATA

Description

This command gets the measurement data of DC monitor sweep end, for the selected channel (*Ch*).

NOTE The unit of *value* is decided by the measurement parameter decided depending on the SCPI.SENSe(Ch).DC.PARameter command.

Variable

Parameter	*Velue*
Description	DC monitor measurement data
Data Type	Double precision floating point type (Double)
Unit	V (voltage) or A (ampere)

Examples

```
Dim DCmon as Double
DCmon = SCPI.SENSe(2).DC.MEASure.DATA
```

Related Objects

SCPI.SENSe(Ch).DC.MEASure.CLEar

Equivalent Key

No equivalent key is available on the front panel.

Equivalent SCPI Command

Syntax

:SENSe{[1]-4}:DC:MEASure:DATA?

Query Response

{numeric} <newline><^END>

Example of use

```
10 OUTPUT 717;":SENS1:DC:MEAS:DATA?"
20 ENTER 717;A
```

SCPI.SENSe(*Ch*).DC.MEASure.ENABle

Object Type

Property (**Read-Write**)

Syntax

SCPI.SENSe(*Ch*).DC.MEASure.ENABle = *Status*

Status = SCPI.SENSe(*Ch*).DC.MEASure.ENABle

Description

This command turns ON/OFF DC monitor measurement on sweep end, for the selected channel (*Ch*).

Variable

Parameter	*Status*
Description	ON/OFF the DC monitor measurement on sweep end.
Data Type	Boolean type (Boolean)
Range	Select from the following: • True or ON: Turns ON the DC monitor measurement on sweep end function. • False or OFF: Turns OFF the DC monitor measurement on sweep end function.
Preset Value	False or OFF

Examples

```
Dim StatDCmon as Boolean
SCPI.SENSe(1).DC.MEASure.ENABle = True
StatDCmon = SCPI.SENSe(1).DC.MEASure.ENABle
```

Related Objects

Equivalent Key

Meas > **DC Monitor Setup** > **DC Monitor On Sweep End** > **Monitor**

Equivalent SCPI Command

Syntax

:SENSe{[1]-4}:DC:MEASure:ENABle {ON|OFF|1|0}

:SENSe{[1]-4}:DC:MEASure:ENABle?

Query Response

{1|0} <newline><^END>

Example of use

```
10 OUTPUT 717;":SENS1:DC:MEAS:ENAB ON"
20 OUTPUT 717;":SENS1:DC:MEAS:ENAB?"
30 ENTER 717;A
```

SCPI.SENSe(*Ch*).DC.PARameter

Object Type

Property (**Read-Write**)

Syntax

SCPI.SENSe(*Ch*).DC.PARameter = *Param*

Param = SCPI.SENSe(*Ch*).DC.PARameter

Description

This command sets/gets the port of DC monitor measurement, for the selected channel (*Ch*).

Variable

Parameter	*Param*
Description	Port of DC monitor measurement
Data Type	Character string type (String)
Range	Select from either of the following: • "DCV": Sets the DC measurement port to DC output port (V). • "DCI": Sets the DC measurement port to DC output port (A). • "R": Sets the DC measurement port to R-port. • "T": Sets the DC measurement port to T-port.
Preset Value	"DCV"

Examples

```
Dim DCPar as String
SCPI.SENSe(1).DC.PARameter = "DCI"
DCPar = SCPI.SENSe(1).DC.PARameter
```

Related Objects

SCPI.SENSe(Ch).DC.MEASure.DATA

SCPI.SENSe(Ch).DC.MEASure.ENABle

Equivalent Key

Meas > **DC Monitor Setup** > **Function** > **Vdc Bias|Idc Bias|Vdc R|Vdc T**

Equivalent SCPI Command

Syntax

:SENSe{[1]-4}:DC:PARameter {DCV|DCI|R|T}

:SENSe{[1]-4}:DC:PARameter?

Query Response

{DCV|DCI|R|T} <newline><^END>

Example of use

```
10 OUTPUT 717;":SENS1:DC:PAR DCI"
20 OUTPUT 717;":SENS1:DC:PAR?"
30 ENTER 717;A$
```

SCPI.SENSe(*Ch*).FREQuency.CENTer

Object type

Property (**Read-Write**)

Syntax

SCPI.SENSe(*Ch*).FREQuency.CENTer = *Value*

Value = SCPI.SENSe(*Ch*).FREQuency.CENTer

Description

This command sets/gets the center value of the sweep range of the selected channel (*Ch*).

Variable

Parameter	*Value*
Description	Center value
Data type	Double precision floating point type (Double)
Range	5 to 3E9
Preset value	1.50005G
Unit	Hz (hertz)
Resolution	1E-3 or 5E-4
Note	If the specified variable 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.

Examples

```
Dim Cntr As Double
SCPI.SENSe(1).FREQuency.CENTer = 2E9
Cntr = SCPI.SENSe(1).FREQuency.CENTer
```

Related objects

SCPI.SENSe(Ch).FREQuency.SPAN

Equivalent key

Center

Equivalent SCPI command

Syntax

:SENSe{[1]-4}:FREQuency:CENTer <numeric>

:SENSe{[1]-4}:FREQuency:CENTer?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":SENS1:FREQ:CENT 2E9"
20 OUTPUT 717;":SENS1:FREQ:CENT?"
30 ENTER 717;A
```

SCPI.SENSe(*Ch*).FREQuency.CW

Object type

Property (**Read-Write**)

Syntax

SCPI.SENSe(*Ch*).FREQuency.CW = *Value*

Value = SCPI.SENSe(*Ch*).FREQuency.CW

Description

This command sets/gets the fixed frequency (CW frequency) for the power/dc bias sweep for the selected channel (*Ch*).

NOTE This object provides the same function as the SCPI.SENSe(Ch).FREQuency.FIXed object.

Variable

Parameter	*Value*
Description	Fixed frequency
Data type	Double precision floating point type (Double)
Range	5 to 3E9
Preset value	1E5
Unit	Hz (hertz)
Resolution	1E-3
Note	If the specified variable 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.

Examples

```
Dim CwFreq As Double
SCPI.SENSe(1).FREQuency.CW = 1E9
CwFreq = SCPI.SENSe(1).FREQuency.CW
```

Related objects

SCPI.SENSe(Ch).FREQuency.FIXed

SCPI.SENSe(Ch).SWEep.TYPE

Equivalent key

Sweep Setup > **Power** > **CW Freq**

Equivalent SCPI command

Syntax

:SENSe{[1]-4}:FREQuency[:CW|FIXed] <numeric>

:SENSe{[1]-4}:FREQuency[:CW|FIXed]?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":SENS1:FREQ 1E9"
20 OUTPUT 717;":SENS1:FREQ?"
30 ENTER 717;A
```

SCPI.SENSe(*Ch*).FREQuency.DATA

Object type

Property (**Read Only**)

Syntax

Data = SCPI.SENSe(Ch).FREQuency.DATA

Description

This command reads the frequencies at all measurement points for the selected channel (Ch).

Variable

Parameter	*Data*
Description	Indicates the array data (frequency) of NOP (number of measurement points). Where n is an integer between 1 and NOP. • *Data(n-1):* Frequency at the n-th measurement point The index of the array starts from 0.
Data type	Variant type (Variant)

Examples

```
Dim FreqData As Variant
SCPI.SENSe(1).SWEep.POINts = 201
FreqData = SCPI.SENSe(1).FREQuency.DATA
```

Related objects

SCPI.SENSe(Ch).SWEep.POINts

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

:SENSe{[1]-4}:FREQuency:DATA?

Query response

{numeric 1},… ,{numeric NOP}<newline><^END>

	Description

{numeric n}	Frequency at the n-th measurement point

Where NOP is the number of measurement points and n is an integer between 1 and NOP.

Example of use

```
10 DIM A(1:201)
20 OUTPUT 717;":SENS1:FREQ:DATA?"
30 ENTER 717;A(*)
```

SCPI.SENSe(*Ch*).FREQuency.FIXed

Object type

Property (**Read-Write**)

Syntax

SCPI.SENSe(*Ch*).FREQuency.FIXed = *Value*

Value = SCPI.SENSe(*Ch*).FREQuency.FIXed

Description

This command sets/gets the fixed frequency (CW frequency) for the power/dc bias sweep for the selected channel (*Ch*).

NOTE This command is similar to SCPI.SENSe(Ch).FREQuency.CW.

Variable

Parameter	*Value*
Description	Fixed frequency
Data type	Double precision floating point type (Double)
Range	5 to 3E9
Preset value	1E5
Unit	Hz (hertz)
Resolution	1E-3
Note	If the specified variable 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.

Examples

```
Dim CwFreq As Double
SCPI.SENSe(1).FREQuency.FIXed = 1E9
CwFreq = SCPI.SENSe(1).FREQuency.FIXed
```

Related objects

SCPI.SENSe(Ch).FREQuency.CW

SCPI.SENSe(Ch).SWEep.TYPE

Equivalent key

Sweep Setup > **Power** > **CW Freq**

Equivalent SCPI command

Syntax

:SENSe{[1]-4}:FREQuency[:CW|FIXed] <numeric>

:SENSe{[1]-4}:FREQuency[:CW|FIXed]?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":SENS1:FREQ 1E9"
20 OUTPUT 717;":SENS1:FREQ?"
30 ENTER 717;A
```

SCPI.SENSe(*Ch*).FREQuency.SPAN

Object type

Property (**Read-Write**)

Syntax

SCPI.SENSe(*Ch*).FREQuency.SPAN = *Value*

Value = SCPI.SENSe(*Ch*).FREQuency.SPAN

Description

This command sets/gets the span value of the sweep range of the selected channel (*Ch*).

Variable

Parameter	*Value*
Description	Span value
Data type	Double precision floating point type (Double)
Range	0 to 2.999999995E9
Preset value	2.9999E9
Unit	Hz (hertz)
Resolution	1E-3
Note	If the specified variable 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.

Examples

```
Dim Span As Double
SCPI.SENSe(1).FREQuency.SPAN = 1E9
Span = SCPI.SENSe(1).FREQuency.SPAN
```

Related objects

SCPI.SENSe(Ch).FREQuency.CENTer

Equivalent key

Span

Equivalent SCPI command

Syntax

```
:SENSe{[1]-4}:FREQuency:SPAN <numeric>
:SENSe{[1]-4}:FREQuency:SPAN?
```

Query response

```
{numeric}<newline><^END>
```

Example of use

```
10 OUTPUT 717;":SENS1:FREQ:SPAN 1E9"
20 OUTPUT 717;":SENS1:FREQ:SPAN?"
30 ENTER 717;A
```

SCPI.SENSe(*Ch*).FREQuency.STARt

Object type

Property (**Read-Write**)

Syntax

SCPI.SENSe(*Ch*).FREQuency.STARt = *Value*

Value = SCPI.SENSe(*Ch*).FREQuency.STARt

Description

This command sets/gets the start value of the sweep range of the selected channel (*Ch*).

Variable

Parameter	*Value*
Description	Start value
Data type	Double precision floating point type (Double)
Range	5 to 3E9
Preset value	1E5
Unit	Hz (hertz)
Resolution	1E-3
Note	If the specified variable 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.

Examples

```
Dim Start As Double
SCPI.SENSe(1).FREQuency.STARt = 100E6
Start = SCPI.SENSe(1).FREQuency.STARt
```

Related objects

SCPI.SENSe(Ch).FREQuency.STOP

Equivalent key

Start

Equivalent SCPI command

Syntax

```
:SENSe{[1]-4}:FREQuency:STARt <numeric>
:SENSe{[1]-4}:FREQuency:STARt?
```

Query response

```
{numeric}<newline><^END>
```

Example of use

```
10 OUTPUT 717;":SENS1:FREQ:STAR 100E6"
20 OUTPUT 717;":SENS1:FREQ:STAR?"
30 ENTER 717;A
```

SCPI.SENSe(*Ch*).FREQuency.STOP

Object type

Property (**Read-Write**)

Syntax

SCPI.SENSe(*Ch*).FREQuency.STOP = *Value*

Value = SCPI.SENSe(*Ch*).FREQuency.STOP

Description

This command sets/gets the stop value of the sweep range of the selected channel (*Ch*).

Variable

Parameter	*Value*
Description	Stop value
Data type	Double precision floating point type (Double)
Range	5 to 3E9
Preset value	3E9
Unit	Hz (hertz)
Resolution	1E-3
Note	If the specified variable 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.

Examples

```
Dim Stp As Double
SCPI.SENSe(1).FREQuency.STOP = 3E9
Stp = SCPI.SENSe(1).FREQuency.STOP
```

Related objects

SCPI.SENSe(Ch).FREQuency.STARt

Equivalent key

Stop

Equivalent SCPI command

Syntax

:SENSe{[1]-4}:FREQuency:STOP <numeric>

:SENSe{[1]-4}:FREQuency:STOP?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":SENS1:FREQ:STOP 100E6"
20 OUTPUT 717;":SENS1:FREQ:STOP?"
30 ENTER 717;A
```

SCPI.SENSe(*Ch*).ROSCillator.SOURce

Object type

Property (**Read Only**)

Syntax

Param = SCPI.SENSe(*Ch*).ROSCillator.SOURce

Description

This command reads whether the external reference signal is inputted to the Ref In connector on the rear panel. The return values are the same for all of Ch.

Variable

Parameter	*Param*
Description	Whether the external reference signal is inputted or not.
Data type	Character string type (String)
Range	Select from either of the following: • "INTernal": The external reference signal is not inputted. • "EXTernal": The external reference signal is inputted.

Examples

```
Dim Ref As String
Ref = SCPI.SENSe(1).ROSCillator.SOURce
```

Equivalent key

Displayed on the instrument status bar (at the bottom of the LCD display).

Equivalent SCPI command

Syntax

:SENSe{[1]-4}:ROSCillator:SOURce?

Query response

{INTernal|EXTernal}<newline><^END>

Example of use

```
10 OUTPUT 717;":SENS1:ROSC:SOUR?"
20 ENTER 717;A$
```

SCPI.SENSe(*Ch*).SEGMent.BWIDth.RESolution.CONTrol

Object Type

Property (**Read-Write**)

Syntax

SCPI.SENSe(*Ch*).SEGMent.BWIDth.RESolution.CONTrol = *Status*

Status = SCPI.SENSe(*Ch*).SEGMent.BWIDth.RESolution.CONTrol

Description

This command turns ON/OFF whether to add the IF bandwidth to the segment table, for the selected channel (*Ch*).

Variable

Parameter	*Status*
Description	ON/OFF of the IF bandwidth in the segment table
Data Type	Boolean type (Boolean)
Range	Select from the following: • True or ON: Turns ON the IF bandwidth in the segment table. • False or OFF: Turns OFF the IF bandwidth in the segment table.
Preset Value	False or OFF

Examples

```
Dim IFBWStat as Boolean
SCPI.SENSe(1).SEGMent.BWIDth.RESolution.CONTol = True
IFBWStat = SCPI.SENSe(1).SEGMent.BWIDth.RESolution.CONTol
```

Related Objects

Equivalent Key

Sweep Setup > **Edit Segment Table** > **List IFBW**

Equivalent SCPI Command

Syntax

:SENSe{[1]-4}:SEGMent:BWIDth:RESolution:CONTrol {ON|OFF|1|0}

:SENSe{[1]-4}:SEGMent:BWIDth:RESolution:CONTrol?

Query Response

{1|0} <newline><^END>

Example of use

```
10 OUTPUT 717;":SENS1:SEGM:BWID:RES:CONT ON"
20 OUTPUT 717;":SENS1:SEGM:BWID:RES:CONT?"
30 ENTER 717;A
```

SCPI.SENSe(*Ch*).SEGMent.DATA

Object type

Property (**Read-Write**)

Syntax

SCPI.SENSe(*Ch*).SEGMent.DATA = *Data*

Data = SCPI.SENSe(*Ch*).SEGMent.DATA

Description

This command creates the segment sweep table of the selected channel (*Ch*).

Variable

<table>
<tr><th>Parameter</th><td>Data</td></tr>
<tr><th>Description</th><td>Indicates the array data arranged in the following order (for the segment sweep table); where N is the number of segments (specified with <segm>) and n is an integer between 1 and N.

Data =
{<buf>,<stim>,<ifbw>,<pow>,<del>,<time>,<segm>,
<star 1>,<stop 1>,<nop 1>,<ifbw 1>,<pow 1>,<del 1>,<time 1>,... ,
<star n>,<stop n>,<nop n>,<ifbw n>,<pow n>,<del n>,<time n>,.... ,
<star N>,<stop N>,<nop N>,<ifbw N>,<pow N>,<del N>,<time N>}

Each parameter in the above array data is detailed below:
• <buf>: Always specify 5.
• <stim>: Stimulus setting mode
0: Specifies with start/stop values
1: Specifies with center/span values
• <ifbw>: ON/OFF of the IF bandwidth setting for each segment
0: OFF, 1: ON
• <pow>: ON/OFF of the power setting for each segment
0: OFF, 1: ON
• <del>: ON/OFF of the sweep delay time setting for each segment</td></tr>
</table>

	0: OFF, 1: ON • <time>: ON/OFF of the sweep time setting for each segment 0: OFF, 1: ON • <segm>: Number of segments Specify an integer ranging 1 to 201. • <star n>: Start value/center value of the n-th segment • <stop n>: Stop value/span value of the n-th segment • <nop n>: Number of measurement points of the n-th segment • <ifbw n>: IF bandwidth of the n-th segment is not necessary when the IF bandwidth setting for each segment is OFF (<ifbw>:0). • <pow n>: Power of the n-th segment is not necessary when the power setting for each segment is OFF (<pow>:0). • <del n>: Sweep delay time of the n-th segment is not necessary when the sweep delay time setting for each segment is OFF (<del>:0). • <time n>: Sweep time of the n-th segment is not necessary when the sweep time setting for each segment is OFF (<time>:0).
Data type	Variant type (Variant)
Note	If the necessary amount of array data for the specified number of segments is not available while setting the segment sweep table, an error occurs when its executed and the object is ignored. For <stim>, <ifbw>, <pow>, <del>, and <time>, if the specified value is not the allowable integer, an error occurs when its executed. For <star n>, <stop n>, <nop n>, <ifbw n>, <pow n>, <del n>, and <time n> in the array data, if the specified value 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.

Examples

```
Dim SegmData As Variant
SCPI.SENSe(1).SEGMent.DATA = Array(5,0,0,1,0,0,2, _
100E6,1E9,31,0,2E9,3E9,51,-10)
SegmData = SCPI.SENSe(1).SEGMent.DATA
```

```
Dim SegmData(14) As Variant
Dim Ref As Variant
SegmData(0) = 5
SegmData(1) = 0
SegmData(2) = 0
SegmData(3) = 1
SegmData(4) = 0
SegmData(5) = 0
SegmData(6) = 2
SegmData(7) = 100E6
SegmData(8) = 1E9
SegmData(9) = 31
SegmData(10) = 0
SegmData(11) = 2E9
SegmData(12) = 3E9
SegmData(13) = 51
SegmData(14) = -10
SCPI.SENSe(1).SEGMent.DATA = SegmData
Ref = SCPI.SENSe(1).SEGMent.DATA
```

Related objects

SCPI.SENSe(Ch).SWEep.TYPE

SCPI.SENSe(Ch).SEGMent.FMODe

Equivalent key

Sweep Setup > **Edit Segment Table**

Equivalent SCPI command

Syntax

:SENSe{[1]-4}:SEGMent:DATA
5,<mode>,<ifbw>,<pow>,<del>,<time>,<segm>,
<star 1>,<stop 1>,<nop 1>,<ifbw 1>,<pow 1>,<del 1>,<time 1>,... ,
<star n>,<stop n>,<nop n>,<ifbw n>,<pow n>,<del n>,<time n>,... ,
<star N>,<stop N>,<nop N>,<ifbw N>,<pow N>,<del N>,<time N>

:SENSe{[1]-4}:SEGMent:DATA?

Where N is the number of segments (specified with <segm>) and n is an integer between 1 and N.

Query response

5,{mode},{ifbw},{pow},{del},{time},{segm},
{star 1},{stop 1},{nop 1},{pow 1},{del 1},{time 1},... ,
{star n},{stop n},{nop n},{pow n},{del n},{time n},... ,
{star N},{stop N},{nop N},{pow N},{del N},{time N}<newline><^END>

Example of use

```
10 DIM H(1:3,1:4)
20 OUTPUT 717;":SENS1:SEGM:DATA 5,0,1,0,0,0,3,";
30 OUTPUT 717;"1E9,3E9,11,70e3,";
40 OUTPUT 717;"3E9,4E9,51,7e3,";
50 OUTPUT 717;"4E9,6E9,11,70e3"
60 OUTPUT 717;":SENS1:SEGM:DATA?"
70 ENTER 717;A,B,C,D,E,F,G,H(*)
```

SCPI.SENSe(*Ch*).SEGMent.FMODe

Object Type

Property (**Read-Write**)

Syntax

SCPI.SENSe(*Ch*).SEGMent.FMODe = *Param*

Param = SCPI.SENSe(*Ch*).SEGMent.FMODe

Description

This command sets/gets the stimulus setting mode of stimulus range setting in segment table, for the selected channel (*Ch*). This command is the same operation as the second parameter of SCPI.SENSe(Ch).SEGMent.DATA command.

Variable

Parameter	*Param*
Description	Stimulus setting mode in segment table
Data Type	Character string type (String)
Range	Select from either of the following: • "ENDS": Specified stimulus setting mode to "Start/Stop". • "SPAN": Specified stimulus setting mode to "Center/Span".
Preset Value	"ENDS"

Examples

```
Dim FreqMode as String
SCPI.SENSe(1).SEGMent.FMODe = "SPAN"
FreqMode = SCPI.SENSe(1).SEGMent.FMODe
```

Related Objects

SCPI.SENSe(Ch).SEGMent.DATA

Equivalent Key

Sweep Setup > **Edit Segment Table** > **Freq Mode** > **Start/Stop|Center/Span**

Equivalent SCPI Command

Syntax

:SENSe{[1]-4}:SEGMent:FMODe {ENDS|SPAN}

:SENSe{[1]-4}:SEGMent:FMODe?

Query Response

{ENDS|SPAN} <newline><^END>

Example of use

```
10 OUTPUT 717;":SENS1:SEGM:FMOD SPAN"
20  OUTPUT 717;":SENS1:SEGM:FMOD?"
30 ENTER 717;A$
```

SCPI.SENSe(*Ch*).SEGMent.POWer.LEVel.CONTrol

Object Type

Property (**Read-Write**)

Syntax

SCPI.SENSe(*Ch*).SEGMent.POWer.LEVel.CONTrol = *Status*

Status = SCPI.SENSe(*Ch*).SEGMent.POWer.LEVel.CONTrol

Description

This command turns ON/OFF whether to add the power level to the segment table, for the selected channel (*Ch*).

Variable

Parameter	*Status*
Description	ON/OFF of the power level in the segment table
Data Type	Boolean type (Boolean)
Range	Select from the following: • True or ON: Turns ON the power level in the segment table. • False or OFF: Turns OFF the power level in the segment table.
Preset Value	False or OFF

Examples

```
Dim PowStat as Boolean
SCPI.SENSe(1).SEGMent.POWer.LEVel.CONTol = True
PowStat = SCPI.SENSe(1).SEGMent.POWer.LEVel.CONTol
```

Related Objects

Equivalent Key

Sweep Setup > **Edit Segment Table** > **List Power**

Equivalent SCPI Command

Syntax

:SENSe{[1]-4}:SEGMent:POWer:LEVel:CONTrol {ON|OFF|1|0}

:SENSe{[1]-4}:SEGMent:POWer:LEVel:CONTrol?

Query Response

{1|0} <newline><^END>

Example of use

```
10 OUTPUT 717;":SENS1:SEGM:POW:LEV:CONT ON"
20 OUTPUT 717;":SENS1:SEGM:POW:LEV:CONT?"
30 ENTER 717;A
```

SCPI.SENSe(*Ch*).SEGMent.SWEep.DELay.CONTrol

Object Type

Property (**Read-Write**)

Syntax

SCPI.SENSe(*Ch*).SEGMent.SWEep.DELay.CONTrol = *Status*

Status = SCPI.SENSe(*Ch*).SEGMent.SWEep.DELay.CONTrol

Description

This command turns ON/OFF whether to add the sweep delay time to the segment table, for the selected channel (*Ch*).

Variable

Parameter	*Status*
Description	ON/OFF of the sweep delay time in the segment table
Data Type	Boolean type (Boolean)
Range	Select from the following: • True or ON: Turns ON the sweep delay time in the segment table. • False or OFF: Turns OFF the sweep delay time in the segment table.
Preset Value	False or OFF

Examples

```
Dim DelStat as Boolean
SCPI.SENSe(1).SEGMent.SWEep.DELay.CONTol = True
DelStat = SCPI.SENSe(1).SEGMent.SWEep.DELay.CONTol
```

Related Objects

Equivalent Key

Sweep Setup > **Edit Segment Table** > **List Delay**

Equivalent SCPI Command

Syntax

:SENSe{[1]-4}:SEGMent:SWEep:DELay:CONTrol {ON|OFF|1|0}

:SENSe{[1]-4}:SEGMent:SWEep:DELay:CONTrol?

Query Response

{1|0} <newline><^END>

Example of use

```
10 OUTPUT 717;":SENS1:SEGM:SWE:DEL:CONT ON"
20 OUTPUT 717;":SENS1:SEGM:SWE:DEL:CONT?"
30 ENTER 717;A
```

SCPI.SENSe(*Ch*).SEGMent.SWEep.POINts

Object type

Property (**Read Only**)

Syntax

Value = SCPI.SENSe(*Ch*).SEGMent.SWEep.POINts

Description

This command reads the total number of the measurement points of all segments, for the segment sweep table of the selected channel (*Ch*).

Variable

Parameter	*Value*
Description	Total number of measurement points of all the segments
Data type	Long integer type (Long)
Preset Value	2

Examples

```
Dim SegmPoin As Long
SegmPoin = SCPI.SENSe(1).SEGMent.SWEep.POINts
```

Related objects

SCPI.SENSe(Ch).SEGMent.DATA

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

:SENSe{[1]-4}:SEGMent:SWEep:POINts?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":SENS1:SEGM:SWE:POIN?"
20 ENTER 717;A
```

SCPI.SENSe(Ch).SEGMent.SWEep.TIME.CONTrol

Object Type

Property (**Read-Write**)

Syntax

SCPI.SENSe(*Ch*).SEGMent.SWEep.TIME.CONTrol = *Status*

Status = SCPI.SENSe(*Ch*).SEGMent.SWEep.TIME.CONTrol

Description

This command turns ON/OFF whether to add the sweep time to the segment table, for the selected channel (*Ch*).

Variable

Parameter	*Status*
Description	ON/OFF of the sweep time in the segment table
Data Type	Boolean type (Boolean)
Range	Select from the following: • True or ON: Turns ON the sweep time in the segment table. • False or OFF: Turns OFF the sweep time in the segment table.
Preset Value	False or OFF

Examples

```
Dim SwTimeStat as Boolean
SCPI.SENSe(1).SEGMent.SWEep.TIME.CONTol = True
SwTimeStat = SCPI.SENSe(1).SEGMent.SWEep.TIME.CONTol
```

Related Objects

Equivalent Key

Sweep Setup > **Edit Segment Table** > **List Time**

Equivalent SCPI Command

Syntax

:SENSe{[1]-4}:SEGMent:SWEep:TIME:CONTrol {ON|OFF|1|0}

:SENSe{[1]-4}:SEGMent:SWEep:TIME:CONTrol?

Query Response

{1|0} <newline><^END>

Example of use

```
10 OUTPUT 717;":SENS1:SEGM:SWE:TIME:CONT ON"
20 OUTPUT 717;":SENS1:SEGM:SWE:TIME:CONT?"
30 ENTER 717;A
```

SCPI.SENSe(*Ch*).SEGMent.SWEep.TIME.DATA

Object type

Property (**Read Only**)

Syntax

Value = SCPI.SENSe(*Ch*).SEGMent.SWEep.TIME.DATA

Description

This command reads the total sweep time (including sweep delay time) of all the segments, for the segment sweep table of the selected channel (*Ch*).

Variable

Parameter	*Value*
Description	Total sweep time of all segments
Data type	Double precision floating point type (Double)

Examples

```
Dim SegmTime As Double
SegmTime = SCPI.SENSe(1).SEGMent.SWEep.TIME.DATA
```

Related objects

SCPI.SENSe(Ch).SEGMent.DATA

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

:SENSe{[1]-4}:SEGMent:SWEep:TIME[:DATA]?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":SENS1:SEGM:SWE:TIME?"
20 ENTER 717;A
```

SCPI.SENSe(Ch).SWEep.BAND.WAIT

Object Type

Property (**Read-Write**)

Syntax

SCPI.SENSe(Ch).SWEep.BAND.WAIT = *Value*

Value = SCPI.SENSe(Ch).SWEep.BAND.WAIT

Description

This command sets/gets the additional band wait time for the selected channel (*Ch*).

Variable

Parameter	*Value*
Description	Waiting time
Data Type	Double precision floating point type (Double)
Range	0 to 1
Preset Value	0
Unit	sec
Resolution	100u

Examples

```
Dim Var as Double
Var= 0.1
SCPI.SENSe(1).SWEep.BAND.WAIT = Var
Var = SCPI.SENSe(1).SWEep.BAND.WAIT
```

Equivalent Key

System > **Service Menu** > **Band Wait**

Equivalent SCPI Command

Syntax

:SENSe{[1]-4}:SWEep:BAND:WAIT <numeric>

:SENSe{[1]-4}:SWEep:BAND:WAIT?

Query Response

<numeric><newline><^END>

Example of use

```
10 OUTPUT 717;":SENS1:SWE:BAND:WAIT 0.1"
20 OUTPUT 717;":SENS1:SWE:BAND:WAIT ?"
30 ENTER 717;A
```

SCPI.SENSe(*Ch*).SWEep.DELay

Object type

Property (**Read-Write**)

Syntax

SCPI.SENSe(*Ch*).SWEep.DELay = *Value*

Value = SCPI.SENSe(*Ch*).SWEep.DELay

Description

This command sets/gets the sweep delay time of the selected channel (*Ch*).

Variable

Parameter	*Value*
Description	Sweep delay time
Data type	Double precision floating point type (Double)
Range	0 to 1
Preset value	0
Unit	s (second)
Resolution	0.001
Note	If the specified variable 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.

Examples

```
Dim SweDel As Double
SCPI.SENSe(1).SWEep.DELay = 0.05
SweDel = SCPI.SENSe(1).SWEep.DELay
```

Equivalent key

Sweep Setup > **Sweep Delay**

Equivalent SCPI command

Syntax

:SENSe{[1]-4}:SWEep:DELay <numeric>

:SENSe{[1]-4}:SWEep:DELay?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":SENS1:SWE:DEL 0.05"
20 OUTPUT 717;":SENS1:SWE:DEL?"
30 ENTER 717;A
```

SCPI.SENSe(*Ch*).SWEep.POINts

Object type

Property (**Read-Write**)

Syntax

SCPI.SENSe(*Ch*).SWEep.POINts = *Value*

Value = SCPI.SENSe(*Ch*).SWEep.POINts

Description

This command sets/gets the number of measurement points of the selected channel (*Ch*).

Variable

Parameter	*Value*
Description	Number of measurement points
Data type	Long integer type (Long)
Range	2 to 1601
Preset value	201
Note	If the specified variable 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.

Examples

```
Dim Nop As Long
SCPI.SENSe(1).SWEep.POINts = 801
Nop = SCPI.SENSe(1).SWEep.POINts
```

Equivalent key

Sweep Setup > **Points**

Equivalent SCPI command

Syntax

:SENSe{[1]-4}:SWEep:POINts <numeric>

:SENSe{[1]-4}:SWEep:POINts?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":SENS1:SWE:POIN 801"
20 OUTPUT 717;":SENS1:SWE:POIN?"
30 ENTER 717;A
```

SCPI.SENSe(*Ch*).SWEep.TIME.AUTO

Object type

Property (**Read-Write**)

Syntax

SCPI.SENSe(*Ch*).SWEep.TIME.AUTO = *Status*

Status = SCPI.SENSe(*Ch*).SWEep.TIME.AUTO

Description

This command sets/gets whether to automatically set the sweep time of the selected channel (*Ch*).

Variable

Parameter	*Status*
Description	ON/OFF of the auto setting of the sweep time
Data type	Boolean type (Boolean)
Range	Select from either of the following: • True or ON: Turns ON the auto setting. • False or OFF: Turns OFF the auto setting.
Preset value	True or ON

Examples

```
Dim SweAuto As Boolean
SCPI.SENSe(1).SWEep.TIME.AUTO = False
SweAuto = SCPI.SENSe(1).SWEep.TIME.AUTO
```

Related objects

SCPI.SENSe(Ch).SWEep.TIME.DATA

Equivalent key

Sweep Setup > **Sweep Time Auto**

NOTE When performing the operation from the front panel, the auto setting of the sweep time is turned ON by setting the sweep time to 0 s.

Equivalent SCPI command

Syntax

:SENSe{[1]-4}:SWEep:TIME:AUTO {ON|OFF|1|0}

:SENSe{[1]-4}:SWEep:TIME:AUTO?

Query response

{1|0}<newline><^END>

Example of use

```
10 OUTPUT 717;":SENS1:SWE:TIME:AUTO ON"
20 OUTPUT 717;":SENS1:SWE:TIME:AUTO?"
30 ENTER 717;A
```

SCPI.SENSe(*Ch*).SWEep.TIME.DATA

Object type

Property (**Read-Write**)

Syntax

SCPI.SENSe(*Ch*).SWEep.TIME.DATA = *Value*

Value = SCPI.SENSe(*Ch*).SWEep.TIME.DATA

Description

This command sets/gets the sweep time of the selected channel (*Ch*).

NOTE Before using this object to set the sweep time, turns OFF the auto setting of the sweep time (specify False with the SCPI.SENSe(Ch).SWEep.TIME.AUTO object).

Variable

Parameter	*Value*
Description	Sweep time
Data type	Double precision floating point type (Double)
Range	Varies depending on the measurement conditions
Preset value	Varies depending on the measurement conditions
Unit	s (second)
Note	If the specified variable 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.

Examples

```
Dim SweTime As Double
SCPI.SENSe(1).SWEep.TIME.AUTO = False
SCPI.SENSe(1).SWEep.TIME.DATA = 1.5
SweTime = SCPI.SENSe(1).SWEep.TIME.DATA
```

Related objects

SCPI.SENSe(Ch).SWEep.TIME.AUTO

Equivalent key

Sweep Setup > Sweep Time

Equivalent SCPI command

Syntax

:SENSe{[1]-4}:SWEep:TIME[:DATA] <numeric>

:SENSe{[1]-4}:SWEep:TIME[:DATA]?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":SENS1:SWE:TIME 1.5"
20 OUTPUT 717;":SENS1:SWE:TIME?"
30 ENTER 717;A
```

SCPI.SENSe(*Ch*).SWEep.TYPE

Object type

Property (**Read-Write**)

Syntax

SCPI.SENSe(*Ch*).SWEep.TYPE = *Param*

Param = SCPI.SENSe(*Ch*).SWEep.TYPE

Description

This command sets/gets the sweep type of the selected channel (*Ch*).

Variable

Parameter	*Param*
Description	Sweep type
Data type	Character string type (String)
Range	Select from either of the following: • "LINear": Sets the sweep type to the linear sweep. • "LOGarithmic": Sets the sweep type to the log sweep. • "SEGMent": Sets the sweep type to the segment sweep. • "POWer": Sets the sweep type to the power sweep. • "BIAS": Sets the sweep type to the DC bias sweep.
Preset value	"LINear"

Examples

```
Dim SweType As String
SCPI.SENSe(1).SWEep.TYPE = "segm"
SweType = SCPI.SENSe(1).SWEep.TYPE
```

Equivalent key

Sweep Setup > **Sweep Type** > **Lin Freq|Log Freq|Segment|Power Sweep|DC Bias Sweep**

Equivalent SCPI command

Syntax

:SENSe{[1]-4}:SWEep:TYPE {LINear|LOGarithmic|SEGMent|POWer|BIAS}

:SENSe{[1]-4}:SWEep:TYPE?

Query response

{LIN|LOG|SEGM|POW|BIAS}<newline><^END>

Example of use

```
10 OUTPUT 717;":SENS1:SWE:TYPE SEGM"
20 OUTPUT 717;":SENS1:SWE:TYPE?"
30 ENTER 717;A$
```

SCPI.SENSe.CORRection.IMPedance.INPut.MAGNitude

Object type

Property (**Read-Write**)

Syntax

SCPI.SENSe.CORRection.IMPedance.INPut.MAGNitude = *Value*

Value = SCPI.SENSe.CORRection.IMPedance.INPut.MAGNitude

Description

This command sets/gets the system characteristic impedance (Z0) value.

NOTE The system impedance is a common parameter in all the channels.

Variable

Parameter	*Value*
Description	System Z0 value
Data type	Double precision floating point type (Double)
Range	1E-3 to 1000
Preset value	50
Unit	Ω
Resolution	0.001
Note	If the specified variable 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.

Examples

```
Dim SysZ0 As Double
SCPI.SENSe.CORRection.IMPedance.INPut.MAGNitude = 75
SysZ0 = SCPI.SENSe.CORRection.IMPedance.INPut.MAGNitude
```

Equivalent key

Cal > **Set Z0**

Equivalent SCPI command

Syntax

```
:SENSe:CORRection:IMPedance[:INPut][:MAGNitude] <numeric>
:SENSe:CORRection:IMPedance[:INPut][:MAGNitude]?
```

Query response

```
{numeric}<newline><^END>
```

Example of use

```
10 OUTPUT 717;":SENS:CORR:IMP 75"
20 OUTPUT 717;":SENS:CORR:IMP?"
30 ENTER 717;A
```

SCPI.SENSe(Ch).Z.COMPensation.CLEar

Object Type

Method (**Write Only**)

Syntax

SCPI.SENSe(Ch).Z.COMPensation.CLEar

Description

This command clears the fixture compensation table for the selected channel.

Examples

SCPI.SENSe(1).Z.COMPensation.CLEar

Related Objects

SCPI.SENSe(Ch).Z.CORRection.COLLect.ACQuire

Equivalent Key

Calibration > **Fixture Compen** > **Clear** > **OK**

Equivalent SCPI Command

Syntax

:SENSe{[1]-4}:Z:COMPensation:CLEar

Example of use

```
10 OUTPUT 717;":SENS1:Z:COMP:CLE"
```

SCPI.SENSe(*Ch*).Z.COMPensation.COEFficient.DATA

Object Type

Property (**Read-Write**)

Syntax

SCPI.SENSe(*Ch*).Z.COMPensation.COEFficient.DATA(*Std) = Array*

Array = SCPI.SENSe(*Ch*).Z.COMPensation.COEFficient.DATA(*Std)*

Description

This command sets/gets the measured impedance data for each fixture compensation standard for the selected channel (*Ch*).

Variable

Parameter	*Std*
Description	Standrad
Data Type	Character string type (String)
Range	OPEN: Open SHORt Short LOAD: Load

Parameter	*Array*
Description	Indicates the array data (corrected data array) of NOP (number of measurement points)×2. Where n is an integer between 1 and NOP. *Data(n×2-2)* Real part of data (complex number) at the n-th measurement point. *Data(n×2-1)* Imaginary part of data (complex number) at the n-th measurement point. The index of the array starts from 0.
Data type	Variant type (Variant)

Examples

```
Dim CoefAray As Variant, NoOfPoint  As Long, dmy As Long

MsgBox "Connect Open"
SCPI.SENSe(1).Z.COMPensation.COLLect.ACQuire = "OPEN"
dmy = SCPI.IEEE4882.OPC
SCPI.SENSe(1).Z.COMPensation.COLLect.SAVE
SCPI.SENSe(2).Z.COMPensation.COLLect.ACQuire = "OPEN"
dmy = SCPI.IEEE4882.OPC
SCPI.SENSe(2).Z.COMPensation.COLLect.SAVE

NoOfPoint = SCPI.SENSe.Z.COMPensation.COEFficient.POINts("OPEN")

ReDim CoefAray(NoOfPoint)
CoefAray = SCPI.SENSe(1).Z.COMPensation.COEFficient.DATA("OPEN")
SCPI.SENSe(2).Z.COMPensation.COEFficient.DATA("OPEN") = CoefAray
```

Related Objects

SCPI.SENSe(Ch).Z.COMPensation.COLLect.ACQuire

SCPI.SENSe(Ch).Z.COMPensation.COEFficient.POINts

Equivalent Key

No equivalent key is available on the front panel.

Equivalent SCPI Command

Syntax

```
:SENSe{[1]-4}:Z:COMPensation:COEFficient[:DATA]
{OPEN|SHORt|LOAD}, <numeric 1>, ... ,<numeric NOP×2>
```

```
:SENSe{[1]-4}:Z:COMPensation:COEFficient[:DATA]?
{OPEN|SHORt|LOAD}
```

Query Response

```
<numeric 1>, ... ,<numeric NOP×2><newline><^END>
```

	Description
<numeric n×2-1>	Real part of data (complex number) at the n-th measurement point.
<numeric n×2>	Imaginary part of data (complex number) at the n-th measurement point.

Because the calibration coefficient array is expressed by a complex number, the real part and the imaginary part of one measurement point are returned and obtained as a value. Here, NOP is the number of measurement points and n is an integer between 1 and NOP.

Example of use

```
10 OUTPUT 717;":SENS:Z:COMP:COEF OPEN, 50,0, 50.1, 0, 50.2, 0"
20 OUTPUT 717;":SENS:Z:COMP:COEF? OPEN"
30 ENTER 717;A(*)
```

SCPI.SENSe(*Ch*).Z.COMPensation.COEFficient.POINts

Object Type

Property (**Read Only**)

Syntax

Value = SCPI.SENSe(*Ch*).Z.COMPensation.COEFficient.POINts(*Std*)

Description

This command returns the size of array of fixture compensation coefficient data of Impedance measurement which is retuned by SCPI.SENSe(Ch).Z.COMPensation.COEFficient.DATA.

Variable

Parameter	*Std*
Description	Standrad
Data Type	Character string type (String)
Range	OPEN: Open SHORt Short LOAD: Load

Parameter	*Value*
Description	Size of array
Data Type	Long integer type (Long)
Range	0 to 1601
Preset Value	0
Unit	- -
Resolution	- -

Examples

See SCPI.SENSe(Ch).Z.COMPensation.COEFficient.DATA

Related Objects

SCPI.SENSe(Ch).Z.COMPensation.COEFficient.DATA

Equivalent Key

No equivalent key is available on the front panel.

Equivalent SCPI Command

Syntax

:SENSe{[1]-4}:Z:COMPensation:COEFficient:POINts? {OPEN|SHORt|LOAD}

Query Response

<numeric><newline><^END>

Example of use

```
10 OUTPUT 717;":SENS1:Z:COMP:COEF:POIN? OPEN"
20 ENTER 717;A
```

SCPI.SENSe(*Ch*).Z.COMPensation.COLLect.ACQuire

Object Type

Method (**Write Only**)

Syntax

SCPI.SENSe(*Ch*).Z.COMPensation.COLLect.ACQuire = *Standard*

Description

This command executes the open, short or load of the fixture compensation for the selected channel (*Ch*).

Variable

Parameter	*Standard*
Description	Compensation Type
Data Type	Character string type (String)
Range	OPEN: Open SHORt: Short LOAD: Load
Preset Value	-
Unit	-
Resolution	-

Examples

```
Dim Dmy as Long
SCPI.SENSe(1).Z.COMPensation.COLLect.ACQuire = "OPEN"
Dmy = SCPI.IEEE4882.OPC
SCPI.SENSe(1).Z.COMPensation.COLLect.SAVE
```

Related Objects

SCPI.SENSe(Ch).Z.COMPensation.COLLect.SAVE

SCPI.SENSe(Ch).Z.COMPensation.CLEar

Equivalent Key

Calibration > **Fixture Compen** > **Compensate** > **Open**

Calibration > **Fixture Compen** > **Compensate** > **Short**

Calibration > **Fixture Compen** > **Compensate** > **Load**

Equivalent SCPI Command

Syntax

:SENSe{[1]-4}:Z:COMPensation:COLLect[:ACQuire] {OPEN|SHORt|LOAD}

Example of use

```
10 OUTPUT 717;":SENS1:Z:COMP:COLL OPEN"
20 OUTPUT 717;"*OPC?"
30 ENTER 717;A
40 OUTPUT 717;":SENS1:Z:COMP:COLL:SAVE"
```

SCPI.SENSe(*Ch*).Z.COMPensation.COLLect.SAVE

Object Type

Method (**Write Only**)

Syntax

SCPI.SENSe(*Ch*).Z.COMPensation.COLLect.SAVE

Description

This command calculates the calibration coefficients for fixture compensation for the selected channel (*Ch*).

Examples

See SCPI.SENSe(Ch).Z.COMPensation.COLLect.ACQuire

Related Objects

SCPI.SENSe(Ch).Z.COMPensation.COLLect.ACQuire

Equivalent Key

Calibration > **Fixture Compen** > **Compensate** > **Done**

Equivalent SCPI Command

Syntax

:SENSe{[1]-4}:Z:COMPensation:COLLect:SAVE

Example of use

See :SENSe{[1]-4}:Z:COMPensation:COLLect[:ACQuire]

SCPI.SENSe(*Ch*).Z.COMPensation.COLLect.STAN.LOAD.CP

Object Type

Property (**Read-Write**)

Syntax

SCPI.SENSe(*Ch*).Z.COMPensation.COLLect.STAN.LOAD.CP = *Value*

Value = SCPI.SENSe(*Ch*).Z.COMPensation.COLLect.STAN.LOAD.CP

Description

This command sets/gets the Cp value of Load Standard for the fixture compensation for the selected channel (*Ch*). This is independent from the calibration kit.

Variable

Parameter	*Value*
Description	Cp value of Load Standard
Data Type	Double precision floating point type (Double)
Range	-1M ~ 1M
Preset Value	0
Unit	F
Resolution	-

Examples

```
Dim Var as Double
Var= 1E-13
SCPI.SENSe(1).Z.COMPensation.COLLect.STAN.LOAD.CP = Var
Var = SCPI.SENSe(1).Z.COMPensation.COLLect.STAN.LOAD.CP
```

Related Objects

SCPI.SENSe(Ch).Z.COMPensation.COLLect.STAN.LOAD.LS

SCPI.SENSe(Ch).Z.COMPensation.COLLect.STAN.LOAD.RS

SCPI.SENSe(Ch).Z.COMPensation.COLLect.ACQuire

SCPI.SENSe(Ch).Z.COMPensation.COLLect.SAVE

Equivalent Key

Calibration > **Fixture Compen** > **Compen STDs** > **Load Cp**

Equivalent SCPI Command

Syntax

```
:SENSe{[1]-4}:Z:COMPensation:COLLect:STAN:LOAD:CP <numeric>
:SENSe{[1]-4}:Z:COMPensation:COLLect:STAN:LOAD:CP?
```

Query Response

```
<numeric><newline><^END>
```

Example of use

```
10 OUTPUT 717;":SENS1:Z:COMP:COLL:STAN:LOAD:CP 1E-13"
20 OUTPUT 717;":SENS1:Z:COMP:COLL:STAN:LOAD:CP?"
30 ENTER 717;A
```

SCPI.SENSe(*Ch*).Z.COMPensation.COLLect.STAN.LOAD.LS

Object Type

Property (**Read-Write**)

Syntax

SCPI.SENSe(*Ch*).Z.COMPensation.COLLect.STAN.LOAD.LS = *Value*

Value = SCPI.SENSe(*Ch*).Z.COMPensation.COLLect.STAN.LOAD.LS

Description

This command sets/gets the Ls value of Load Standard for the fixture compensation for the selected channel (*Ch*). This is independent from the calibration kit.

Variable

Parameter	*Value*
Description	Ls value of Load Standard
Data Type	Double precision floating point type (Double)
Range	-1M ~ 1M
Preset Value	0
Unit	H
Resolution	-

Examples

```
Dim Var as Double
Var= 1E-10
SCPI.SENSe(1).Z.COMPensation.COLLect.STAN.LOAD.LS = Var
Var = SCPI.SENSe(1).Z.COMPensation.COLLect.STAN.LOAD.LS
```

Related Objects

SCPI.SENSe(Ch).Z.COMPensation.COLLect.STAN.LOAD.CP

SCPI.SENSe(Ch).Z.COMPensation.COLLect.STAN.LOAD.RS

SCPI.SENSe(Ch).Z.COMPensation.COLLect.ACQuire

Equivalent Key

Calibration > **Fixture Compen** > **Compen STDs** > **Load Ls**

Equivalent SCPI Command

Syntax

:SENSe{[1]-4}:Z:COMPensation:COLLect:STAN:LOAD:LS <numeric>

:SENSe{[1]-4}:Z:COMPensation:COLLect:STAN:LOAD:LS?

Query Response

<numeric><newline><^END>

Example of use

```
10 OUTPUT 717;":SENSe1:Z:COMPensation:COLLect:STAN:LOAD:LS 1E-10"
20 OUTPUT 717;":SENSe1:Z:COMPensation:COLLect:STAN:LOAD:LS?"
30 ENTER 717;A
```

SCPI.SENSe(*Ch*).Z.COMPensation.COLLect.STAN.LOAD.RS

Object Type

Property (**Read-Write**)

Syntax

SCPI.SENSe(*Ch*).Z.COMPensation.COLLect.STAN.LOAD.RS = *Value*

Value = SCPI.SENSe(*Ch*).Z.COMPensation.COLLect.STAN.LOAD.RS

Description

This command sets/gets the Rs value of Load Standard for the fixture compensation for the selected channel (*Ch*). This is independent from the calibration kit.

Variable

Parameter	*Value*
Description	Rs value of Load Standard
Data Type	Double precision floating point type (Double)
Range	-1M ~ 1M
Preset Value	50
Unit	Ω
Resolution	-

Examples

```
Dim Var as Double
Var= 50.5
SCPI.SENSe(1).Z.COMPensation.COLLect.STAN.LOAD.RS = Var
Var = SCPI.SENSe(1).Z.COMPensation.COLLect.STAN.LOAD.RS
```

Related Objects

SCPI.SENSe(Ch).Z.COMPensation.COLLect.STAN.LOAD.CP

SCPI.SENSe(Ch).Z.COMPensation.COLLect.STAN.LOAD.RS

SCPI.SENSe(Ch).Z.COMPensation.COLLect.ACQuire

Equivalent Key

Calibration > **Fixture Compen** > **Compen STDs** > **Load Rs**

Equivalent SCPI Command

Syntax

:SENSe{[1]-4}:Z:COMPensation:COLLect:STAN:LOAD:RS <numeric>

:SENSe{[1]-4}:Z:COMPensation:COLLect:STAN:LOAD:RS?

Query Response

<numeric><newline><^END>

Example of use

```
10 OUTPUT 717;":SENSe1:Z:COMP:COLL:STAN:LOAD:RS 50.5"
20 OUTPUT 717;":SENSe1:Z:COMP:COLL:STAN:LOAD:RS?"
30 ENTER 717;A
```

SCPI.SENSe(*Ch*).Z.COMPensation.COLLect.STAN.OPEN.CP

Object Type

Property (**Read-Write**)

Syntax

SCPI.SENSe(*Ch*).Z.COMPensation.COLLect.STAN.OPEN.CP = *Value*

Value = SCPI.SENSe(*Ch*).Z.COMPensation.COLLect.STAN.OPEN.CP

Description

This command sets/gets the Cp value of Open Standard for the fixture compensation for the selected channel (*Ch*). This is independent from the calibration kit.

Variable

Parameter	*Value*
Description	Cp value of Open Standard
Data Type	Double precision floating point type (Double)
Range	-1M ~ 1M
Preset Value	0
Unit	F
Resolution	-

Examples

```
Dim Var as Double
Var= 1E-13
SCPI.SENSe(1).Z.COMPensation.COLLect.STAN.OPEN.CP = Var
Var = SCPI.SENSe(1).Z.COMPensation.COLLect.STAN.OPEN.CP
```

Related Objects

SCPI.SENSe(Ch).Z.COMPensation.COLLect.STAN.OPEN.GS

SCPI.SENSe(Ch).Z.COMPensation.COLLect.STAN.OPEN.LS

SCPI.SENSe(Ch).Z.COMPensation.COLLect.ACQuire

Equivalent Key

Calibration > **Fixture Compen** > **Compen STDs** > **Open Cp**

Equivalent SCPI Command

Syntax

:SENSe{[1]-4}:Z:COMPensation:COLLect:STAN:OPEN:CP <numeric>

:SENSe{[1]-4}:Z:COMPensation:COLLect:STAN:OPEN:CP?

Query Response

<numeric><newline><^END>

Example of use

```
10 OUTPUT 717;":SENS1:Z:COMP:COLL:STAN:OPEN:CP 1E-13"
20 OUTPUT 717;":SENS1:Z:COMP:COLL:STAN:OPEN:CP?"
30 ENTER 717;A
```

SCPI.SENSe(*Ch*).Z.COMPensation.COLLect.STAN.OPEN.LS

Object Type

Property (**Read-Write**)

Syntax

SCPI.SENSe(*Ch*).Z.COMPensation.COLLect.STAN.OPEN.LS = *Value*

Value = SCPI.SENSe(*Ch*).Z.COMPensation.COLLect.STAN.OPEN.LS

Description

This command sets/gets the Ls value of Open Standard for the fixture compensation for the selected channel (*Ch*). This is independent from the calibration kit.

Variable

Parameter	*Value*
Description	Ls value of Open Standard
Data Type	Double precision floating point type (Double)
Range	-1M ~ 1M
Preset Value	0
Unit	H
Resolution	-

Examples

```
Dim Var as Double
Var= 1E-10
SCPI.SENSe(1).Z.COMPensation.COLLect.STAN.OPEN.LS = Var
Var = SCPI.SENSe(1).Z.COMPensation.COLLect.STAN.OPEN.LS
```

Related Objects

SCPI.SENSe(Ch).Z.COMPensation.COLLect.STAN.OPEN.GS

SCPI.SENSe(Ch).Z.COMPensation.COLLect.STAN.OPEN.CP

SCPI.SENSe(Ch).Z.COMPensation.COLLect.ACQuire

Equivalent Key

Calibration > **Fixture Compen** > **Compen STDs** > **Open Ls**

Equivalent SCPI Command

Syntax

:SENSe{[1]-4}:Z:COMPensation:COLLect:STAN:OPEN:LS <numeric>

:SENSe{[1]-4}:Z:COMPensation:COLLect:STAN:OPEN:LS?

Query Response

<numeric><newline><^END>

Example of use

```
10 OUTPUT 717;":SENS1:Z:COMP:COLL:STAN:OPEN:LS 1E-10"
20 OUTPUT 717;":SENS1:Z:COMP:COLL:STAN:OPEN:LS?"
30 ENTER 717;A
```

SCPI.SENSe(*Ch*).Z.COMPensation.COLLect.STAN.OPEN.GS

Object Type

Property (**Read-Write**)

Syntax

SCPI.SENSe(*Ch*).Z.COMPensation.COLLect.STAN.OPEN.GS = *Value*

Value = SCPI.SENSe(*Ch*).Z.COMPensation.COLLect.STAN.OPEN.GS

Description

This command sets/gets the Gs value of Open Standard for the fixture compensation for the selected channel (*Ch*). This is independent from the calibration kit.

Variable

Parameter	*Value*
Description	Gs value of Open Standard
Data Type	Double precision floating point type (Double)
Range	-1M ~ 1M
Preset Value	0
Unit	S
Resolution	-

Examples

```
Dim Var as Double
Var= 0.01
SCPI.SENSe(1).Z.COMPensation.COLLect.STAN.OPEN.GS = Var
Var = SCPI.SENSe(1).Z.COMPensation.COLLect.STAN.OPEN.GS
```

Related Objects

SCPI.SENSe(Ch).Z.COMPensation.COLLect.STAN.OPEN.CP

SCPI.SENSe(Ch).Z.COMPensation.COLLect.STAN.OPEN.LS

SCPI.SENSe(Ch).Z.COMPensation.COLLect.ACQuire

Equivalent Key

Calibration > **Fixture Compen** > **Compen STDs** > **Open Gs**

Equivalent SCPI Command

Syntax

:SENSe{[1]-4}:Z:COMPensation:COLLect:STAN:OPEN:GS <numeric>

:SENSe{[1]-4}:Z:COMPensation:COLLect:STAN:OPEN:GS?

Query Response

<numeric><newline><^END>

Example of use

```
10 OUTPUT 717;":SENS1:Z:COMP:COLL:STAN:OPEN:GS 0.01"
20 OUTPUT 717;":SENS1:Z:COMP:COLL:STAN:OPEN:GS?"
30 ENTER 717;A
```

SCPI.SENSe(*Ch*).Z.COMPensation.COLLect.STAN.SHORt.CP

Object Type

Property (**Read-Write**)

Syntax

SCPI.SENSe(*Ch*).Z.COMPensation.COLLect.STAN.SHORt.CP = *Value*

Value = SCPI.SENSe(*Ch*).Z.COMPensation.COLLect.STAN.SHORt.CP

Description

This command sets/gets the Cp value of Short Standard for the fixture compensation for the selected channel (*Ch*). This is independent from the calibration kit.

Variable

Parameter	*Value*
Description	Cp value of Short Standard
Data Type	Double precision floating point type (Double)
Range	-1M ~ 1M
Preset Value	0
Unit	F
Resolution	-

Examples

```
Dim Var as Double
Var= 1E-13
SCPI.SENSe(1).Z.COMPensation.COLLect.STAN.SHORt.CP = Var
Var = SCPI.SENSe(1).Z.COMPensation.COLLect.STAN.SHORt.CP
```

Related Objects

SCPI.SENSe(Ch).Z.COMPensation.COLLect.STAN.SHORt.LS

SCPI.SENSe(Ch).Z.COMPensation.COLLect.STAN.SHORt.RS

SCPI.SENSe(Ch).Z.COMPensation.COLLect.ACQuire

Equivalent Key

Calibration > **Fixture Compen** > **Compen STDs** > **Short Cp**

Equivalent SCPI Command

Syntax

:SENSe{[1]-4}:Z:COMPensation:COLLect:STAN:SHORt:CP <numeric>

:SENSe{[1]-4}:Z:COMPensation:COLLect:STAN:SHORt:CP?

Query Response

<numeric><newline><^END>

Example of use

```
10 OUTPUT 717;":SENS1:Z:COMP:COLL:STAN:SHORt:CP 1E-13"
20 OUTPUT 717;":SENS1:Z:COMP:COLL:STAN:SHORt:CP?"
30 ENTER 717;A
```

SCPI.SENSe(*Ch*).Z.COMPensation.COLLect.STAN.SHORt.LS

Object Type

Property (**Read-Write**)

Syntax

SCPI.SENSe(*Ch*).Z.COMPensation.COLLect.STAN.SHORt.LS = *Value*

Value = SCPI.SENSe(*Ch*).Z.COMPensation.COLLect.STAN.SHORt.LS

Description

This command sets/gets the Ls value of Short Standard for the fixture compensation for the selected channel (*Ch*). This is independent from the calibration kit.

Variable

Parameter	*Value*
Description	Ls value of Short Standard
Data Type	Double precision floating point type (Double)
Range	-1M ~ 1M
Preset Value	0
Unit	H
Resolution	-

Examples

```
Dim Var as Double
Var= 1E-10
SCPI.SENSe(1).Z.COMPensation.COLLect.STAN.SHORt.LS = Var
Var = SCPI.SENSe(1).Z.COMPensation.COLLect.STAN.SHORt.LS
```

Related Objects

SCPI.SENSe(Ch).Z.COMPensation.COLLect.STAN.SHORt.CP

SCPI.SENSe(Ch).Z.COMPensation.COLLect.STAN.SHORt.RS

SCPI.SENSe(Ch).Z.COMPensation.COLLect.ACQuire

Equivalent Key

Calibration > **Fixture Compen** > **Compen STDs** > **Short Ls**

Equivalent SCPI Command

Syntax

:SENSe{[1]-4}:Z:COMPensation:COLLect:STAN:SHORt:LS <numeric>

:SENSe{[1]-4}:Z:COMPensation:COLLect:STAN:SHORt:LS?

Query Response

<numeric><newline><^END>

Example of use

```
10 OUTPUT 717;":SENS1:Z:COMP:COLL:STAN:SHORt:LS 1E-10"
20 OUTPUT 717;":SENS1:Z:COMP:COLL:STAN:SHORt:LS?"
30 ENTER 717;A
```

SCPI.SENSe(*Ch*).Z.COMPensation.COLLect.STAN.SHORt.RS

Object Type

Property (**Read-Write**)

Syntax

SCPI.SENSe(*Ch*).Z.COMPensation.COLLect.STAN.SHORt.RS = *Value*

Value = SCPI.SENSe(*Ch*).Z.COMPensation.COLLect.STAN.SHORt.RS

Description

This command sets/gets the Rs value of Short Standard for the fixture compensation for the selected channel (*Ch*). This is independent from the calibration kit.

Variable

Parameter	*Value*
Description	Rs value of Short Standard
Data Type	Double precision floating point type (Double)
Range	-1M ~ 1M
Preset Value	0
Unit	Ω
Resolution	-

Examples

```
Dim Var as Double
Var= 0.001
SCPI.SENSe(1).Z.COMPensation.COLLect.STAN.SHORt.RS = Var
Var = SCPI.SENSe(1).Z.COMPensation.COLLect.STAN.SHORt.RS
```

Related Objects

Equivalent Key

Calibration > **Fixture Compen** > **Compen STDs** > **Short Rs**

Equivalent SCPI Command

Syntax

:SENSe{[1]-4}:Z:COMPensation:COLLect:STAN:SHORt:RS <numeric>

:SENSe{[1]-4}:Z:COMPensation:COLLect:STAN:SHORt:RS?

Query Response

<numeric><newline><^END>

Example of use

```
10 OUTPUT 717;":SENS1:Z:COMP:COLL:STAN:SHORt:RS 0.001"
20 OUTPUT 717;":SENS1:Z:COMP:COLL:STAN:SHORt:RS?"
30 ENTER 717;A
```

SCPI.SENSe(*Ch*).Z.COMPensation.EDELay.TIME

Object Type

Property (**Read-Write**)

Syntax

SCPI.SENSe(*Ch*).Z.COMPensation.EDELay.TIME = *Value*

Value = SCPI.SENSe(*Ch*).Z.COMPensation.EDELay.TIME

Description

This command sets/gets the Z port extension delay value for the selected channel (*Ch*).

Variable

Parameter	*Value*
Description	Z port extension
Data Type	Double precision floating point type (Double)
Range	-1M ~ 1M
Preset Value	0
Unit	sec
Resolution	-

Examples

```
Dim Var as Double
Var= 0.001
SCPI.SENSe(1).Z.COMPensation.EDELay.TIME = Var
Var = SCPI.SENSe(1).Z.COMPensation.EDELay.TIME
```

Related Objects

SCPI.SENSe(Ch).Z.COMPensation.FIXTure.EDELay.USER.DISTance

Equivalent Key

Calibration > **Fixture Compen** > **Z Port Extension**

Equivalent SCPI Command

Syntax

:SENSe{[1]-4}:Z:COMPensation:EDELay:TIME <numeric>

:SENSe{[1]-4}:Z:COMPensation:EDELay:TIME?

Query Response

<numeric><newline><^END>

Example of use

```
10 OUTPUT 717;":SENSe1:Z:COMP:EDEL:TIME 0.001"
20 OUTPUT 717;":SENSe1:Z:COMP:EDEL:TIME?"
30 ENTER 717;A
```

SCPI.SENSe(*Ch*).Z.COMPensation.FIXTure.EDELay.MODel.DISTance

Object Type

Property (**Read Only**)

Syntax

Value = SCPI.SENSe(*Ch*).Z.COMPensation.FIXTure.EDELay.MODel.DISTance("NONE"|"FXT16191A"|"FXT16192A"|"FXT16193A"|"FXT16194A"|"FXT16196A"|"FXT16196B"|"FXT16196C"|"FXT16197")

Description

This command gets the registered electrical length of specified fixture for the selected channel (*Ch*).

Variable

Parameter	*Value*
Description	Electrical Length
Data Type	Double precision floating point type (Double)
Range	-
Preset Value	-
Unit	meter
Resolution	-

Examples

```
Dim Var as Double
Var = SCPI.SENSe(1).Z.COMPensation.FIXTure.EDELay.MODel.DISTance ("FXT16191A")
```

Related Objects

SCPI.SENSe(Ch).Z.COMPensation.FIXTure.SELect

Equivalent Key

Calibration > **Fixture Compen** > **Fixture**

This command returns the electrical length value on the softkey for each fixture.

Equivalent SCPI Command

Syntax

:SENSe{[1]-4}:Z:COMPensation:FIXTure:EDELay:MODel:DISTance {NONE|FXT16191A|FXT16192A|FXT16193A|FXT16194A|FXT16196A|FXT16196B|FXT16196C|FXT16197A}

Query Response

<numeric><newline><^END>

Example of use

```
10 OUTPUT 717;":SENS1:Z:COMP:FIXT:EDEL:MOD:DIST? FXT16191A"
30 ENTER 717;A
```

SCPI.SENSe(*Ch*).Z.COMPensation.FIXTure.EDELay.USER.DISTance

Object Type

Property (**Read-Write**)

Syntax

SCPI.SENSe(*Ch*).Z.COMPensation.FIXTure.EDELay.USER.DISTance = *Value*

Value = SCPI.SENSe(*Ch*).Z.COMPensation.FIXTure.EDELay.USER.DISTance

Description

This command sets/gets the electrical lenght of user fixture for the selected channel (*Ch*).

Variable

Parameter	*Value*
Description	Electrical Length
Data Type	Double precision floating point type (Double)
Range	-1k ~ 1k
Preset Value	0
Unit	m
Resolution	-

Examples

```
Dim Var as Double
Var= 0.015
SCPI.SENSe(1).Z.COMPensation.FIXTure.EDELay.USER.DISTance = Var
Var = SCPI.SENSe(1).Z.COMPensation.FIXTure.EDELay.USER.DISTance
```

Related Objects

SCPI.SENSe(Ch).Z.COMPensation.FIXTure.SELect

Equivalent Key

Calibration > **Fixture Compen** > **Modify User Fixt.** > **User Fixture**

Equivalent SCPI Command

Syntax

:SENSe{[1]-4}:Z:COMPensation:FIXTure:EDELay:USER:DISTance <numeric>

:SENSe{[1]-4}:Z:COMPensation:FIXTure:EDELay:USER:DISTance?

Query Response

<numeric><newline><^END>

Example of use

```
10 OUTPUT 717;":SENSe1:Z:COMPensation:FIXTure:EDELay:USER:DISTance 0.015"
20 OUTPUT 717;":SENSe1:Z:COMPensation:FIXTure:EDELay:USER:DISTance?"
30 ENTER 717;A
```

SCPI.SENSe(*Ch*).Z.COMPensation.FIXTure.SELect

Object Type

Property (**Read-Write**)

Syntax

SCPI.SENSe(*Ch*).Z.COMPensation.FIXTure.SELect = *Fixture*

Fixture = SCPI.SENSe(*Ch*).Z.COMPensation.FIXTure.SELect

Description

This command sets/gets the fixture model number to compensate the electrical length for the selected channel (*Ch*). Whne user fixture is selected, the electrical length for user fixture should be specified by SCPI.SENSe(Ch).Z.COMPensation.FIXTure.EDELay.USER.DISTance.

Variable

Parameter	*Fixture*
Description	Fixture model number
Data Type	Character string type (String)
Range	NONE|FXT16191A|FXT16192A|FXT16193A|FXT16194A|FXT16196A|FXT16196B|FXT16196C|FXT16197A|USER
Preset Value	NONE
Unit	-
Resolution	-

Examples

```
Dim Fixture as String
Fixture= "FXT16191A"
SCPI.SENSe(1).Z.COMPensation.FIXTure.SELect = Fixture
Fixture = SCPI.SENSe(1).Z.COMPensation.FIXTure.SELect
```

Related Objects

SCPI.SENSe(Ch).Z.COMPensation.FIXTure.EDELay.USER.DISTance

SCPI.SENSe(Ch).Z.COMPensation.FIXTure.EDELay.MODel.DISTance

Equivalent Key

Calibration > **Fixture Compen** > **Fixture**

Equivalent SCPI Command

Syntax

:SENSe{[1]-4}:Z:COMPensation:FIXTure[:SELect] {NONE|FXT16191A|FXT16192A|FXT16193A|FXT16194A|FXT16196A|FXT16196B|FXT16196C|FXT16197A|USER}

:SENSe{[1]-4}:Z:COMPensation:FIXTure[:SELect]?

Query Response

{NONE|FXT16191A|FXT16192A|FXT16193A|FXT16194A|FXT16196A|FXT16196B|FXT16196C|FXT16197A|USER} <newline><^END>

Example of use

```
10 OUTPUT 717;":SENS1:Z:COMP:FIXT FXT16191A"
20 OUTPUT 717;":SENS1:Z:COMP:FIXT?"
30 ENTER 717;A$
```

SCPI.SENSe(*Ch*).Z.COMPensation.STATe.ALL

Object Type

Property (**Read-Write**)

Syntax

SCPI.SENSe(*Ch*).Z.COMPensation.STATe.ALL = *Status*

Status = SCPI.SENSe(*Ch*).Z.COMPensation.STATe.ALL

Description

This command turns On/Off the fixture compensation for the selected channel (*Ch*).

Variable

Parameter	*Status*
Description	Fixture compensation on/off
Data Type	Boolean type (Boolean)
Range	Select from either of the following: • True or ON(1): Turns ON. • False or OFF(0): Turns OFF.
Preset Value	OFF
Unit	-
Resolution	-

Examples

```
Dim CompStatus as Boolean
SCPI.SENSe(1).Z.COMPensation.STATe.ALL = True
CompStatus = SCPI.SENSe(1).Z.COMPensation.STATe.ALL
```

Related Objects

SCPI.SENSe(Ch).Z.COMPensation.STATe.LOAD

SCPI.SENSe(Ch).Z.COMPensation.STATe.OPEN

SCPI.SENSe(Ch).Z.COMPensation.STATe.SHORt

Equivalent Key

Calibration > **Fixture Compen** > **Fixture Compen**

Equivalent SCPI Command

Syntax

```
:SENSe{[1]-4}:Z:COMPensation:STATe[:ALL] {ON|OFF|1|0}
:SENSe{[1]-4}:Z:COMPensation:STATe[:ALL]?
```

Query Response

```
{1|0}<newline><^END>
```

Example of use

```
10 OUTPUT 717;":SENS1:Z:COMP:STAT ON"
20 OUTPUT 717;":SENS1:Z:COMP:STAT?"
30 ENTER 717;A
```

SCPI.SENSe(*Ch*).Z.COMPensation.STATe.LOAD

Object Type

Property (**Read-Write**)

Syntax

SCPI.SENSe(*Ch*).Z.COMPensation.STATe.LOAD = *Status*

Status = SCPI.SENSe(*Ch*).Z.COMPensation.STATe.LOAD

Description

This command turns On/Off the load of fixture compensation for the selected channel (*Ch*).

Variable

Parameter	*Status*
Description	Load compensation on/off
Data Type	Boolean type (Boolean)
Range	Select from either of the following: • True or ON(1): Turns ON. • False or OFF(0): Turns OFF.
Preset Value	OFF
Unit	-
Resolution	-

Examples

```
Dim LoadCompStatus as Boolean
SCPI.SENSe(1).Z.COMPensation.STATe.LOAD = True
LoadCompStatus = SCPI.SENSe(1).Z.COMPensation.STATe.LOAD
```

Related Objects

SCPI.SENSe(Ch).Z.COMPensation.STATe.ALL

SCPI.SENSe(Ch).Z.COMPensation.STATe.OPEN

SCPI.SENSe(Ch).Z.COMPensation.STATe.SHORt

Equivalent Key

Calibration > **Fixture Compen** > **Compensate** > **Compen Load**

Equivalent SCPI Command

Syntax

:SENSe{[1]-4}:Z:COMPensation:STATe:LOAD {ON|OFF|1|0}

:SENSe{[1]-4}:Z:COMPensation:STATe:LOAD?

Query Response

{1|0}<newline><^END>

Example of use

```
10 OUTPUT 717;":SENSe1:Z:COMPensation:STATe:LOAD ON"
20 OUTPUT 717;":SENSe1:Z:COMPensation:STATe:LOAD?"
30 ENTER 717;A
```

SCPI.SENSe(*Ch*).Z.COMPensation.STATe.OPEN

Object Type

Property (**Read-Write**)

Syntax

SCPI.SENSe(*Ch*).Z.COMPensation.STATe.OPEN = *Status*

Status = SCPI.SENSe(*Ch*).Z.COMPensation.STATe.OPEN

Description

This command turns On/Off the open of fixture compensation for the selected channel (*Ch*).

Variable

Parameter	*Status*
Description	Open compensation on/off
Data Type	Boolean type (Boolean)
Range	Select from either of the following: • True or ON(1): Turns ON. • False or OFF(0): Turns OFF.
Preset Value	OFF
Unit	-
Resolution	-

Examples

```
Dim OpenCompStatus as Boolean
SCPI.SENSe(1).Z.COMPensation.STATe.OPEN = Ture
OpenCompStatus = SCPI.SENSe(1).Z.COMPensation.STATe.OPEN
```

Related Objects

SCPI.SENSe(Ch).Z.COMPensation.STATe.ALL

SCPI.SENSe(Ch).Z.COMPensation.STATe.LOAD

SCPI.SENSe(Ch).Z.COMPensation.STATe.SHORt

Equivalent Key

Calibration > **Fixture Compen** > **Compensate** > **Compen Open**

Equivalent SCPI Command

Syntax

:SENSe{[1]-4}:Z:COMPensation:STATe:OPEN {ON|OFF|1|0}

:SENSe{[1]-4}:Z:COMPensation:STATe:OPEN?

Query Response

{1|0}<newline><^END>

Example of use

```
10 OUTPUT 717;":SENS1:Z:COMP:STAT:OPEN ON"
20 OUTPUT 717;":SENS1:Z:COMP:STAT:OPEN?"
30 ENTER 717;A
```

SCPI.SENSe(*Ch*).Z.COMPensation.STATe.SHORt

Object Type

Property (**Read-Write**)

Syntax

SCPI.SENSe(*Ch*).Z.COMPensation.STATe.SHORt = *Value*

Value = SCPI.SENSe(*Ch*).Z.COMPensation.STATe.SHORt

Description

This command turns On/Off the short of fixture compensation for the selected channel (*Ch*).

Variable

Parameter	*Status*
Description	Short compensation on/off
Data Type	Boolean type (Boolean)
Range	Select from either of the following: • True or ON(1): Turns ON. • False or OFF(0): Turns OFF.
Preset Value	OFF
Unit	-
Resolution	-

Examples

```
Dim ShortCompStatus as Boolean
SCPI.SENSe(1).Z.COMPensation.STATe.SHORt = Ture
ShortCompStatus= SCPI.SENSe(1).Z.COMPensation.STATe.SHORt
```

Related Objects

SCPI.SENSe(Ch).Z.COMPensation.STATe.ALL

SCPI.SENSe(Ch).Z.COMPensation.STATe.LOAD

SCPI.SENSe(Ch).Z.COMPensation.STATe.OPEN

Equivalent Key

Calibration > **Fixture Compen** > **Compensate** > **Compen Short**

Equivalent SCPI Command

Syntax

:SENSe{[1]-4}:Z:COMPensation:STATe:SHORt {ON|OFF|1|0}

:SENSe{[1]-4}:Z:COMPensation:STATe:SHORt?

Query Response

{1|0}<newline><^END>

Example of use

```
10 OUTPUT 717;":SENS1:Z:COMP:STAT:SHOR ON"
20 OUTPUT 717;":SENS1:Z:COMP:STAT:SHORt?"
30 ENTER 717;A
```

SCPI.SENSe(*Ch*).Z.CORRection.COEFficient.DATA

Object Type

Property (**Read-Write**)

Syntax

SCPI.SENSe(*Ch*).Z.CORRection.COEFficient.DATA(*Std*) = *Array*

Array = SCPI.SENSe(*Ch*).Z.CORRection.COEFficient.DATA(*Std*)

Description

This command sets/gets the calibration coefficient data of impedance calibration for the selected channel. Before writing the calibration coefficient array, the calibration should be performed in advance in that channel with the same setting. After writing the calibration coefficient array, the written value becomes effective immediately.

Variable

Parameter	*Std*
Description	Standrad
Data Type	String
Range	OPEN: Open SHORt Short LOAD: Load LOAD2: Low-loss capacitor

Parameter	*Array*
Description	Indicates the array data (corrected data array) of NOP (number of measurement points)×2. Where n is an integer between 1 and NOP. The size of array, NOP, is the number of points when the calibration is executed. *Data(n×2-2)* Real part of data (complex number) at the n-th measurement point. *Data(n×2-1)* Imaginary part of data (complex number) at the n-th measurement point.

	The index of the array starts from 0.
Data type	Variant type (Variant)

Examples

```
Dim CoefArray as Variant, NoOfPoint as Long
Dim Dmy As Long

MsgBox "Connect Open"
SCPI.SENSe(1).Z.CORRection.COLLect.ACQuire = "OPEN"
Dmy = SCPI.IEEE4882.OPC
MsgBox "Connect Short"
SCPI.SENSe(1).Z.CORRection.COLLect.ACQuire = "SHORT"
Dmy = SCPI.IEEE4882.OPC
MsgBox "Connect Load"
SCPI.SENSe(1).Z.CORRection.COLLect.ACQuire = "LOAD"
Dmy = SCPI.IEEE4882.OPC
SCPI.SENSe(1).Z.CORRection.COLLect.SAVE


MsgBox "Connect Open"
SCPI.SENSe(2).Z.CORRection.COLLect.ACQuire = "OPEN"
Dmy = SCPI.IEEE4882.OPC
MsgBox "Connect Short"
SCPI.SENSe(2).Z.CORRection.COLLect.ACQuire = "SHORT"
Dmy = SCPI.IEEE4882.OPC
MsgBox "Connect Load"
SCPI.SENSe(2).Z.CORRection.COLLect.ACQuire = "LOAD"
Dmy = SCPI.IEEE4882.OPC
SCPI.SENSe(2).Z.CORRection.COLLect.SAVE


NoOfPoint = SCPI.SENSe(1).Z.CORRection.COEFficient.POINts
Redim CoefArry(NoOfPoint)
CoefArray = SCPI.SENSe(2).Z.CORRection.COEFficient.DATA("OPEN")
SCPI.SENSe(1).Z.CORRection.COEFficient.DATA("OPEN") = CoefArray
```

Related Objects

SCPI.SENSe(Ch).Z.CORRection.COLLect.ACQuire

SCPI.SENSe(Ch).Z.CORRection.COEFficient.POINts

Equivalent Key

No equivalent key is available on the front panel.

Equivalent SCPI Command

Syntax

:SENSe{[1]-4}:Z:CORRection:COEFficient[:DATA] {OPEN|SHORt|LOAD|LOAD2},<numeric 1>, ... ,<numeric NOP×2>

:SENSe{[1]-4}:Z:CORRection:COEFficient[:DATA]?

Query Response

<numeric 1>, ... ,<numeric NOP×2><newline><^END>

	Description
<numeric n×2-1>	Real part of data (complex number) at the n-th measurement point.
<numeric n×2>	Imaginary part of data (complex number) at the n-th measurement point.

Because the calibration coefficient array is expressed by a complex number, the real part and the imaginary part of one measurement point are returned and obtained as a value. Here, NOP is the number of measurement points and n is an integer between 1 and NOP.

Example of use

```
10 DIM A(1:201)
20 OUTPUT 717;":SENS1:Z:CORR:COEF? OPEN"
30 ENTER 717;A(*)
```

SCPI.SENSe(*Ch*).Z.CORRection.COEFficient.POINts

Object Type

Property (**Read Only**)

Syntax

Value = SCPI.SENSe(*Ch*).Z.CORRection.COEFficient.POINts

Description

This command returns the size of array of calibration coefficient data of Impedance measurement which is retuned by SCPI.SENSe(Ch).Z.CORRection.COEFficient.DATA

Variable

Parameter	*Value*
Description	size of array
Data Type	Long integer type (Long)
Range	0 to 1601
Preset Value	0
Unit	-
Resolution	-

Examples

See SCPI.SENSe(Ch).Z.CORRection.COEFficient.DATA

Related Objects

SCPI.SENSe(Ch).Z.CORRection.COEFficient.DATA

Equivalent Key

No equivalent key is available on the front panel.

Equivalent SCPI Command

Syntax

:SENSe{[1]-4}:Z:CORRection:COEFficient:POINts?

Query Response

<numeric><newline><^END>

Example of use

```
10 OUTPUT 717;":SENS1:Z:CORR:COEF:POINt?"
20 ENTER 717;A
```

SCPI.SENSe(*Ch*).Z.CORRection.COLLect.ACQuire

Object Type

Method (**Write Only**)

Syntax

SCPI.SENSe(*Ch*).Z.CORRection.COLLect.ACQuire = *Standard*

Description

This command executes the open, short, load and low-loss capacitor of the impedance calibration for the selected channel (*Ch*).

Variable

Parameter	*Standard*
Description	Standard Type
Data Type	Character string type (String)
Range	OPEN: Open SHORt: Short LOAD: Load LOAD2: Low- Loss capacitor
Preset Value	-
Unit	-
Resolution	-

Examples

```
Dim Dmy As Long
MsgBox "Connect Open"
SCPI.SENSe.Z.CORRection.COLLect.ACQuire = "OPEN"
Dmy = SCPI.IEEE4882.OPC
MsgBox "Connect Short"
SCPI.SENSe.Z.CORRection.COLLect.ACQuire = "SHORT"
Dmy = SCPI.IEEE4882.OPC
MsgBox "Connect Load"
SCPI.SENSe.Z.CORRection.COLLect.ACQuire = "LOAD"
Dmy = SCPI.IEEE4882.OPC
```

SCPI.SENSe.Z.CORRection.COLLect.SAVE

Related Objects

SCPI.SENSe(Ch).Z.CORRection.COLLect.SAVE

Equivalent Key

Calibration > **Calibrate** > **Impedance Calibration** > **Open [Open(f)]**

Calibration > **Calibrate** > **Impedance Calibration** > **Short [Short(f)]**

Calibration > **Calibrate** > **Impedance Calibration** > **Load [Broadband]**

Calibration > **Calibrate** > **Impedance Calibration** > **Low-Loss C (optional)**

Equivalent SCPI Command

Syntax

:SENSe{[1]-4}:Z:CORRection:COLLect[:ACQuire]
{OPEN|SHORt|LOAD|LOAD2}

Example of use

```
10 OUTPUT 717;":SENS1:Z:CORR:COLL OPEN"
20 OUTPUT 717;"*OPC?"
30 ENTER 717;A
40 OUTPUT 717;":SENS1:Z:CORR:COLL SHORT"
50 OUTPUT 717;"*OPC?"
60 ENTER 717;A
70 OUTPUT 717;":SENS1:Z:CORR:COLL LOAD"
80 OUTPUT 717;"*OPC?"
90 ENTER 717;A
100 OUTPUT 717;":SENS1:Z:CORR:COLL:SAVE"
```

SCPI.SENSe(*Ch*).Z.CORRection.COLLect.SAVE

Object Type

Method (**Write Only**)

Syntax

SCPI.SENSe(*Ch*).Z.CORRection.COLLect.SAVE

Description

This command calculates the calibration coefficients for impedance calibration for the selected channel (*Ch*).

Examples

See SCPI.SENSe(Ch).Z.CORRection.COLLect.ACQuire

Related Objects

SCPI.SENSe(Ch).Z.CORRection.COLLect.ACQuire

Equivalent Key

Calibration > **Calibrate** > **Impedance Calibration** > **Done**

Equivalent SCPI Command

Syntax

:SENSe{[1]-4}:Z:CORRection:COLLect:SAVE

Example of use

See :SENSe{[1]-4}:Z:CORRection:COLLect[:ACQuire]

SCPI.SENSe(*Ch*).Z.METHod

Object Type

Property (**Read-Write**)

Syntax

SCPI.SENSe(*Ch*).Z.METHod = *Value*

Value = SCPI.SENSe(*Ch*).Z.METHod

Description

This command sets or gets the impedance method.

When the method changes, the Fixture Compensation table is cleared.

Variable

Parameter	*Value*
Description	Impedance method
Data Type	Character string type (String)
Range	Select one of the following options: P1Reflection:S-Parameter Port 1 reflection measurement (for simple impedance measurement) P2Reflection:S-Parameter Port 2 reflection measurement (for simple impedance measurement) TSERies:S-Parameter series-through measurement (for simple impedance measurement) TSHunt:S-Parameter shunt-through measurement (for PDN component characterization) GSERies:Gain-Phase series-through measurement (for simple impedance measurement) GSHunt:Gain-Phase shunt-through measurement (for PDN component characterization)
Preset Value	P1Reflection
Unit	-
Resolution	-

Examples

```
Dim Var as String
Var= "P1Reflection"
SCPI.SENSe(1).Z.METHod = Var
Var = SCPI.SENSe(1).Z.METHod
```

Equivalent Key

There is no equivalent key is available on the front panel. However, the similar key is:

Meas > **Method** > **Port 1 Refl** OR

Meas > **Method** > **Port 2 Refl** OR

Meas > **Method** > **Port 1-2 Series** OR

Meas > **Method** > **Port 1-2 Shunt** OR

Meas > **Method** > **GP Series T 50Ω R 1MΩ** OR

Meas > **Method** > **GP Shunt T 50Ω R 50Ω**

Equivalent SCPI Command

Syntax

```
:SENSe{[1]-4}:Z:METHod <string>
:SENSe{[1]-4}:Z:METHod?
```

Query Response

```
<string><newline><^END>
```

Example of use

```
10 OUTPUT 717;":SENS1:Z:METH P1R"
20 OUTPUT 717;":SENS1:Z:METH?"
30 ENTER 717;A$
```

SERVICE

SCPI.SERVice.ACHannel.ACTive

Object type

Property (**Read-Write**)

Syntax

SCPI.SERVice.ACHannel.ACTive = *Value*

Value = SCPI.SERVice.ACHannel.ACTive

Description

This command sets/gets the active channel number.

Variable

Parameter	*Value*
Description	Active channel number
Data type	Long integer type (Long)
Range	1 to 4

Examples

```
Dim ActChan As Long
ActChan = SCPI.SERVice.ACHannel.ACTive
SCPI.SERVice.ACHannel.ACTive = 2
```

Related objects

SCPI.SERVice.ACHannel.COUNt

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

:SERVice:ACHannel:ACTive <numeric>

:SERVice:ACHannel:ACTive?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":SERV:ACH:ACT 2"
20 OUTPUT 717;":SERV:ACH:ACT?"
30 ENTER 717;A
```

SCPI.SERVice.ACHannel.COUNt

Object type

Property (**Read Only**)

Syntax

Value = SCPI.SERVice.ACHannel.COUNt

Description

This command reads the upper limit of the number of channels of the E5061B.

Variable

Parameter	*Value*
Description	Upper limit of the number of channels
Data type	Long integer type (Long)
Range	4

Examples

```
Dim MaxChan As Long
MaxChan = SCPI.SERVice.ACHannel.COUNt
```

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

:SERVice:ACHannel:COUNt?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":SERV:ACH:COUN?"
20 ENTER 717;A
```

SCPI.SERVice.CHANnel(Ch).ATRace.ACTive

Object Type

Property (**Read-Write**)

Syntax

SCPI.SERVice.CHANnel(*Ch*).ATRace.ACTive = *Value*

Value = SCPI.SERVice.CHANnel(*Ch*).ATRace.ACTive

Description

This command sets/gets the active trace number of the selected channel (*Ch*).

Variable

Parameter	*Value*
Description	Active trace number
Data Type	Long integer type (Long)
Range	1 to 4

Examples

```
Dim NumTrac as Long
NumTrac = SCPI.SERVice.CHANnel(4).ATRace.ACTive
SCPI.SERVice.CHANnel(4).ATRace.ACTive = 3
```

Related Objects

SCPI.SERVice.CHANnel(Ch).ATRace.COUNt

Equivalent Key

No equivalent key is available on the front panel.

Equivalent SCPI Command

Syntax

:SERVice:CHANnel{[1]-4}:ATRace:ACTive?

Query Response

{numeric} <newline><^END>

Example of use

```
10 OUTPUT 717;":SERV:CHAN1:ATR:ACT 2"
20 OUTPUT 717;":SERV:CHAN2:ATR:ACT?"
30 ENTER 717;A
```

SCPI.SERVice.CHANNel(Ch).ATRace.COUNt

Object type

Property (**Read Only**)

Syntax

Value = SCPI.SERVice.CHANNel(*Ch*).ATRace.COUNt

Description

This command reads the upper limit of the number of the selected channel (*Ch*).

Variable

Parameter	*Value*
Description	Upper limit of the number of traces.
Data type	Long integer type (Long)
Range	4

Examples

```
Dim MaxTrac As Long
MaxTrac = SCPI.SERVice.CHANNel(1).ATRace.COUNt
```

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

:SERVice:CHANNel{[1]-4}:ATRace:COUNt?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":SERV:CHAN2:ATR:COUN?"
20 ENTER 717;A
```

SCPI.SERVice.CHANnel(*Ch*).DISPlay.ANNotation.XAXis.MODE.BIAS

Object Type

Property (**Read-Write**)

Syntax

SCPI.SERVice.CHANnel(*Ch*).DISPlay.ANNotation.XAXis.MODE.BIAS = *Value*

Value = SCPI.SERVice.CHANnel(*Ch*).DISPlay.ANNotation.XAXis.MODE.BIAS

Description

This command sets or gets the X-Axis DC Bias annotation mode display to Start/Stop or Center/Span.

Variable

Parameter	*Value*
Description	Display of Start/Stop or Center/Span value at the bottom of the screen.
Data Type	Character string type (String)
Range	Select from either of the following: "SSTop": Displays the Start and Stop value. "CSPan": Displays the Centre and Span value.
Preset Value	SSTop
Unit	-
Resolution	-

Examples

```
Dim Var as String
Var= "SSTop"
SCPI.SERVice.CHANnel(1).DISPlay.ANNotation.XAXis.MODE.BIAS = Var
Var = SCPI.SERVice.CHANnel(1).DISPlay.ANNotation.XAXis.MODE.BIAS
```

Related Objects

SCPI.SERVice.CHANnel(Ch).DISPlay.ANNotation.XAXis.MODE.FREQuency

SCPI.SERVice.CHANnel(Ch).DISPlay.ANNotation.XAXis.MODE.POWer

Equivalent Key

No equivalent key is available on the front panel.

Equivalent SCPI Command

Syntax

:SERVice:CHANnel{[1]-4}:DISPlay:ANNotation:XAXis:MODE:BIAS {SSTop|CSPan}

:SERVice:CHANnel{[1]-4}:DISPlay:ANNotation:XAXis:MODE:BIAS?

Query Response

{SST|CSP}<newline><^END>

Example of use

```
10 OUTPUT 717;":SERV:CHAN1:DISP:ANN:XAX:MODE:BIAS SSTop"
20 OUTPUT 717;":SERV:CHAN1:DISP:ANN:XAX:MODE:BIAS?"
30 ENTER 717;A$
```

SCPI.SERVice.CHANNel(*Ch*).DISPlay.ANNotation.XAXis.MODE.FREQuency

Object Type

Property (**Read-Write**)

Syntax

SCPI.SERVice.CHANNel(*Ch*).DISPlay.ANNotation.XAXis.MODE.FREQuency = *Value*

Value = SCPI.SERVice.CHANNel(*Ch*).DISPlay.ANNotation.XAXis.MODE.FREQuency

Description

This command sets or gets X-Axis frequency annotation mode display to Start/Stop or Center/Span.

Variable

Parameter	*Value*
Description	Display of Start/Stop or Center/Span value at the bottom of the screen.
Data Type	Character string type (String)
Range	Select from either of the following: "SSTop": Displays the Start and Stop value. "CSPan": Displays the Centre and Span value.
Preset Value	SSTop
Unit	-
Resolution	-

Examples

```
Dim Var as String
Var= "SSTop"
SCPI.SERVice.CHANNel(1).DISPlay.ANNotation.XAXis.MODE.FREQuency = Var
Var = SCPI.SERVice.CHANNel(1).DISPlay.ANNotation.XAXis.MODE.FREQuency
```

Related Objects

SCPI.SERVice.CHANNel(Ch).DISPlay.ANNotation.XAXis.MODE.BIAS

SCPI.SERVice.CHANNel(Ch).DISPlay.ANNotation.XAXis.MODE.POWer

Equivalent Key

E5061B

No equivalent key is available on the front panel.

Equivalent SCPI Command

Syntax

:SERVice:CHANnel{[1]-4}:DISPlay:ANNotation:XAXis:MODE:FREQuency {SSTop|CSPan}

:SERVice:CHANnel{[1]-4}:DISPlay:ANNotation:XAXis:MODE:FREQuency?

Query Response

{SST|CSP}<newline><^END>

Example of use

```
10 OUTPUT 717;":SERV:CHAN1:DISP:ANN:XAX:MODE:FREQ SSTop"
20 OUTPUT 717;":SERV:CHAN1:DISP:ANN:XAX:MODE:FREQ?"
30 ENTER 717;A$
```

SCPI.SERVice.CHANNel(*Ch*).DISPlay.ANNotation.XAXis.MODE.POWer

Object Type

Property (**Read-Write**)

Syntax

SCPI.SERVice.CHANNel(*Ch*).DISPlay.ANNotation.XAXis.MODE.POWer = *Value*

Value = SCPI.SERVice.CHANNel(*Ch*).DISPlay.ANNotation.XAXis.MODE.POWer

Description

This command sets or gets X-Axis power annotation mode display to Start/Stop or Center/Span.

Variable

Parameter	*Value*
Description	Display of Start/Stop or Center/Span value at the bottom of the screen.
Data Type	Character string type (String)
Range	Select from either of the following: "SSTop": Displays the Start and Stop value. "CSPan": Displays the Centre and Span value.
Preset Value	SSTop
Unit	-
Resolution	-

Examples

```
Dim Var as String
Var= "SSTop"
SCPI.SERVice.CHANNel(1).DISPlay.ANNotation.XAXis.MODE.POWer = Var
Var = SCPI.SERVice.CHANNel(1).DISPlay.ANNotation.XAXis.MODE.POWer
```

Related Objects

SCPI.SERVice.CHANNel(Ch).DISPlay.ANNotation.XAXis.MODE.FREQuency

SCPI.SERVice.CHANNel(Ch).DISPlay.ANNotation.XAXis.MODE.POWer

Equivalent Key

E5061B

No equivalent key is available on the front panel.

Equivalent SCPI Command

Syntax

:SERVice:CHANnel{[1]-4}:DISPlay:ANNotation:XAXis:MODE:POWer {SSTop|CSPan}

:SERVice:CHANnel{[1]-4}:DISPlay:ANNotation:XAXis:MODE:POWer?

Query Response

{SST|CSP}<newline><^END>

Example of use

```
10 OUTPUT 717;":SERV:CHAN1:DISP:ANN:XAX:MODE:POW SST"
20 OUTPUT 717;":SERV:CHANnel1:DISPlay:ANN:XAX:MODE:POW?"
30 ENTER 717;A$
```

SCPI.SERVice.CHANnel(*Ch*).SEGMent.DATA

Object Type

Property (**Read Only**)

Syntax

Value = SCPI.SERVice.CHANnel(*Ch*).SEGMent.DATA

Description

This command gets number of the segment data, for the selected channel (*Ch*).

Variable

Parameter	*Value*
Description	Number of the segment data
Data Type	Long integer type (Long)
Range	9 to 1413
Preset Value	9

Examples

```
Dim NumSegm as Long
NumSegm = SCPI.SERVice.CHANnel(2).SEGMent.DATA
```

Related Objects

Equivalent Key

No equivalent key is available on the front panel.

Equivalent SCPI Command

Syntax

:SERVice:CHANnel{[1]-4}:SEGMent:DATA?

Query Response

{numeric} <newline><^END>

Example of use

```
10 OUTPUT 717;":SERV:CHAN3:SEGM:DATA?"
20 ENTER 717;A
```

SCPI.SERVice.CHANNel(*Ch*).STAN(*Std*).TABLe

Object Type

Property (**Read Only**)

Syntax

Value = SCPI.SERVice.CHANNel(*Ch*).STAN(*Std*).TABLe

Description

This command returns the number of point of the table data which is set by SCPI.SENSe(Ch).CORRection.COLLect.CKIT.STAN(Std).TABLe.

Variable

Parameter	*Value*
Description	Number of standard table data
Data Type	Long integer type (Long)
Range	1 to 4804
Preset Value	1
Unit	-
Resolution	-

Examples

See SCPI.SENSe(Ch).CORRection.COLLect.CKIT.STAN(Std).TABLe

Related objects

SCPI.SENSe(Ch).CORRection.COLLect.CKIT.STAN(Std).TABLe

Equivalent Key

No equivalent key is available on the front panel.

Equivalent SCPI Command

Syntax

:SERVice:CHANNel{[1]-4}:STAN{[1]-21}:TABLe?

Query Response

<numeric><newline><^END>

Example of use

```
10 OUTPUT 717;":SERV:CHAN1:STAN1:TABL?"
20 ENTER 717;A
```

SCPI.SERVice.CHANNel(*Ch*).SWEep.POINts

Object Type

Property (**Read Only**)

Syntax

Value = SCPI.SERVice.CHANNel(*Ch*).SWEep.POINts

Description

This command gets the number of measurement points of the selected channel (*Ch*).

Variable

Parameter	*Value*
Description	Number of measurement points
Data Type	Long integer type (Long)

Examples

```
Dim Poin as Long
Poin = SCPI.SERVice.CHANNel(2).SWEep.POINts
```

Related Objects

SCPI.SENSe(Ch).SWEep.POINts

SCPI.SENSe(Ch).SEGMent.SWEep.POINts

Equivalent Key

No equivalent key is available on the front panel.

Equivalent SCPI Command

Syntax

:SERVice:CHANNel{[1]-4}:SWEep:POINts?

Query Response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":SERV:CHAN2:SWE:POIN?"
20 ENTER 717;A
```

SCPI.SERVice.CHANNel(Ch).TRACe(Tr).AMRKer.ACTive

Object Type

Property (**Read-Write**)

Syntax

SCPI.SERVice.CHANNel(*Ch*).TRACe(*Tr*).AMRKer.ACTive = *Value*

Value = SCPI.SERVice.CHANNel(*Ch*).TRACe(*Tr*).AMRKer.ACTive

Description

This command sets/gets the active marker number of selected trace (*Tr*) in the selected channel (*Ch*).

Variable

Parameter	*Value*
Description	Active marker number of selected trace in selected channel
Data Type	Long integer type (Long)
Range	0 to 10

Examples

```
Dim ActMkr as Long
ActMkr = SCPI.SERVice.CHANNel(4).TRACe.AMRKer.ACTive
SCPI.SERVice.CHANNel(4).TRACe.AMRKer.ACTive = 6
```

Equivalent Key

No equivalent key is available on the front panel.

Equivalent SCPI Command

Syntax

:SERVice:CHANNel{[1]-4}:TRACe{[1]-4}:AMRKer:ACTive?

Query Response

{numeric} <newline><^END>

Example of use

```
10 OUTPUT 717;":SERV:CHAN2:TRAC2:AMRK:ACT?"
20 ENTER 717;A
```

SCPI.SERVice.CHANNel(*Ch*).TRACe(Tr).DISPlay.ANNotation.XAXis.MODE.TRANsform

Object Type

Property (**Read-Write**)

Syntax

SCPI.SERVice.CHANNel(*Ch*).TRACe(Tr).DISPlay.ANNotation.XAXis.MODE.TRANsform = *Value*

Value = SCPI.SERVice.CHANNel(*Ch*).TRACe(Tr).DISPlay.ANNotation.XAXis.MODE.TRANsform

Description

This command sets or gets X-Axis annotation mode display of fault location measurements to Start/Stop or Center/Span.

Variable

Parameter	*Value*
Description	Display of Start/Stop or Center/Span value at the bottom of the screen.
Data Type	Character string type (String)
Range	Select from either of the following: "SSTop": Displays the Start and Stop value. "CSPan": Displays the Centre and Span value.
Preset Value	SSTop
Unit	-
Resolution	-

Examples

```
Dim Var as String
Var= "SSTop"
SCPI.SERVice.CHANNel(1).TRACe(1).DISPlay.ANNotation.XAXis.MODE.TRANsform = Var
Var = SCPI.SERVice.CHANNel(1).TRACe(1).DISPlay.ANNotation.XAXis.MODE.TRANsform
```

Equivalent Key

No equivalent key is available on the front panel.

Equivalent SCPI Command

Syntax

:SERVice:CHANnel{[1]-4}:TRACe{[1]-4}:DISPlay:ANNotation:XAXis:MODE:TRANsform {SSTop|CSPan}

:SERVice:CHANnel{[1]-4}:TRACe{[1]-4}:DISPlay:ANNotation:XAXis:MODE:TRANsform?

Query Response

{SST|CSP}<newline><^END>

Example of use

```
10 OUTPUT 717;":SERV:CHAN1:TRAC1:DISP:ANN:XAX:MODE:TRAN SST"
20 OUTPUT 717;":SERV:CHAN1:TRACe1:DISP:ANN:XAX:MODE:TRAN?"
30 ENTER 717;A$
```

SCPI.SERVice.CHANNel(*Ch*).TRACe(*Tr*).LIMit.DATA

Object Type

Property (**Read Only**)

Syntax

Value = SCPI.SERVice.CHANNel(*Ch*).TRACe(*Tr*).LIMit.DATA

Description

This command gets the number of array elements using SCPI.CALCulate(Ch).SELected.LIMit.DATA command, for the selected trace (*Tr*) of the selected channel (*Ch*).

Variable

Parameter	*Value*
Description	number of array elements
Data Type	Long integer type (Long)
Range	1 to 501

Examples

```
Dim NumLimTbl as Long
NumLimTbl = SCPI.SERVice.CHANNel(2).TRACe(4).LIMit.DATA
```

Related Objects

SCPI.CALCulate(Ch).SELected.LIMit.DATA

Equivalent Key

No equivalent key is available on the front panel.

Equivalent SCPI Command

Syntax

:SERVice:CHANNel{[1]-4}:TRACe{[1]-4}:LIMit:DATA?

Query Response

{numeric} <newline><^END>

Example of use

```
10 OUTPUT 717;":SERV:CHAN1:TRAC3:LIM:DATA?"
20 ENTER 717;A
```

SCPI.SERVice.CHANnel(Ch).TRACe(Tr).MEMValid

Object Type

Property (**Read-Write**)

Syntax

SCPI.SERVice.CHANnel(*Ch*).TRACe(*Tr*).MEMValid = *Status*

Status = SCPI.SERVice.CHANnel(*Ch*).TRACe(*Tr*).MEMValid

Description

This command turns ON/OFF the memory trace for the selected trace (*Tr*) of the selected channel (*Ch*). The memory trace can be turned on even if it is inactivated (no data).

NOTE This command does not copy the data trace to memory trace. When the data trace copies to the memory trace, use the SCPI.CALCulate(Ch).SELected.MATH.MEMorize command.

Variable

Parameter	*Status*
Description	ON/OFF the memory trace
Data Type	Boolean type (Boolean)
Range	Select from either of the following: • True or ON: Turns ON the memory trace • False or OFF: Turns OFF the memory trace
Preset Value	False or OFF

Examples

```
Dim StatMemTr as Boolean
SCPI.SERVice.CHANnel(4).TRACe(2).MEMValid = True
StatMemTr = SCPI.SERVice.CHANnel(4).TRACe(2).MEMValid
```

Related Objects

SCPI.CALCulate(Ch).SELected.MATH.MEMorize

Equivalent Key

No equivalent key is available on the front panel.

Equivalent SCPI Command

Syntax

:SERVice:CHANnel{[1]-4}:TRACe{[1]-4}:MEMValid {ON|OFF|1|0}

:SERVice:CHANnel{[1]-4}:TRACe{[1]-4}:MEMValid?

Query Response

{1|0} <newline><^END>

Example of use

```
10 OUTPUT 717;":SERV:CHAN1:TRAC2:MEMV ON"
20 OUTPUT 717;":SERV:CHAN1:TRAC3:MEMV?"
30 ENTER 717;A
```

SCPI.SERVice.CHANNel(*Ch*).TRACe(*Tr*).RLIMit.DATA

Object Type

Property (**Read Only**)

Syntax

Value = SCPI.SERVice.CHANNel(*Ch*).TRACe(*Tr*).RLIMit.DATA

Description

This command gets the number of array elements using SCPI.CALCulate(Ch).SELected.RLIMit.DATA command, for the selected trace (*Tr*) of the selected channel (*Ch*).

Variable

Parameter	*Value*
Description	Number of array elements
Data Type	Long integer type (Long)
Range	1 to 49

Examples

```
Dim NumRlimTbl as Long
NumRlimTbl = SCPI.SERVice.CHANNel(1).TRACe(2).RLIMit.DATA
```

Related Objects

SCPI.CALCulate(Ch).SELected.RLIMit.DATA

Equivalent Key

No equivalent key is available on the front panel.

Equivalent SCPI Command

Syntax

:SERVice:CHANNel{[1]-4}:TRACe{[1]-4}:RLIMit:DATA?

Query Response

{numeric} <newline><^END>

Example of use

```
10 OUTPUT 717;":SERV:CHAN1:TRAC3:RLIM:DATA?"
20 ENTER 717;A
```

SCPI.SERVice.LOGGing.CLEar

Object Type

Method (**Write Only**)

Syntax

SCPI.SERVice.LOGGing.CLEar

Description

This command clears all log files about Event, Power on test, Mech software, Overload, FW close, Recovery.

Examples

```
SCPI.SERVice.LOGGing.CLEar
```

Equivalent Key

No equivalent key is available on the front panel.

Equivalent SCPI Command

Syntax

:SERVice:LOGGing:CLEar

Example of use

```
10 OUTPUT 717;":SERV:LOGG:CLE"
```

SCPI.SERVice.POWer.OVERload.PROTect.CLEar

Object type

Method (**Write Only**)

Syntax

SCPI.SERVice.POWer.OVERload.PROTect.CLEar

Description

This command clears the overload protection.

Examples

SCPI.SERVice.POWer.OVERload.PROTect.CLEar

Related objects

SCPI.SERVice.POWer.OVERload.PROTect.STATe

Equivalent key

System > **Overload Recovery** > **Clear Overload Protection**

Equivalent SCPI command

Syntax

:SERVice:POWer:OVERload:PROTect:CLEar

Example of use

10 OUTPUT 717;":SERV:POW:OVER:PROT:CLE"

SCPI.SERVice.POWer.OVERload.PROTect.STATe

Object Type

Property (**Read Only**)

Syntax

Status = SCPI.SERVice.POWer.OVERload.PROTect.STATe

Description

This command queries if the overload protection is activated (Overload protection indicator is displayed).

Variable

Parameter	*Status*
Description	Status of overload protection
Data type	Boolean type (Boolean)
Range	Return from either of the following: • True: overload protection is activated. • False: overload protection is off.
Preset value	False

Examples

```
Dim OverLoadStatus As Boolean
OverLoadStatus = SCPI.SERVice.POWer.OVERload.PROTect.STATe
```

Related Objects

SCPI.SERVice.POWer.OVERload.PROTect.CLEar

Equivalent Key

No equivalent key is available on the front panel.

Equivalent SCPI Command

Syntax

:SERVice:POWer:OVERload:PROTect:STATe?

Query Response

<1|0>

Example of use

```
10 OUTPUT 717;":SERV:POW:OVER:PROT:STAT?"
20 ENTER 717;A
```

SCPI.SERVice.SREVision

Object type

Property (**Read Only**)

Syntax

Value = SCPI.SERVice.SREVision

Description

This command reads the system version (OS ID) of the E5061B.

Variable

Parameter	*Value*
Description	1: Windows Vista Business (downgraded to XP) 0: Windoes XP
Data type	Long integer type (Long)

Examples

```
Dim SystemRevision As Long
SystemRevision = SCPI.SERVice.SREVision
```

Equivalent key

System > **Firmware Revision**

OS ID shows this status.

Equivalent SCPI command

Syntax

:SERVice:SREVision?

Query response

<newmeric><newline><^END>

Example of use

```
10 OUTPUT 717;":SERV:SREV?"
20 ENTER 717;A
```

SCPI.SERVice.SWEep.FREQuency.MAXimum

Object type

Property (**Read Only**)

Syntax

Value = SCPI.SERVice.SWEep.FREQuency.MAXimum

Description

This command reads the upper limit of measurement frequency of E5061B.

Variable

Parameter	*Value*
Description	Upper limit of measurement frequency
Data type	Double precision floating point type (Double)

Examples

```
Dim MaxFreq As Double
MaxFreq = SCPI.SERVice.SWEep.FREQuency.MAXimum
```

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

:SERVice:SWEep:FREQuency:MAXimum?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":SERV:SWE:FREQ:MAX?"
20 ENTER 717;A
```

SCPI.SERVice.SWEep.FREQuency.MINimum

Object type

Property (**Read Only**)

Syntax

Value = SCPI.SERVice.SWEep.FREQuency.MINimum

Description

This command reads the lower limit of measurement frequency of E5061B.

Variable

Parameter	*Value*
Description	Lower limit of measurement frequency
Data type	Double precision floating point type (Double)

Examples

```
Dim MinFreq As Double
MinFreq = SCPI.SERVice.SWEep.FREQuency.MINimum
```

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

:SERVice:SWEep:FREQuency:MINimum?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":SERV:SWE:FREQ:MIN?"
20 ENTER 717;A
```

SCPI.SERVice.SWEep.POINts

Object type

Property (**Read Only**)

Syntax

Value = SCPI.SERVice.SWEep.POINts

Description

This command reads the upper limit of the number of measurement points at the current configuration.

Variable

Parameter	*Value*
Description	Upper limit of the number of points
Data type	Long integer type (Long)

Examples

```
Dim MaxPoin As Long
MaxPoin = SCPI.SERVice.SWEep.POINts
```

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

:SERVice:SWEep:POINts?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":SERV:SWE:POIN?"
20 ENTER 717;A
```

SCPI.SERVice.SYSTem.OPTion.GPHase

Object Type

Property (**Read Only**)

Syntax

Value = SCPI.SERVice.SYSTem.OPTion.GPHase

Description

This query command gets the availability of the Gain-Phase test port option.

Variable

Parameter	*Status*
Description	Gets the availability of the Gain-Phase test port option
Data Type	Boolean type (Boolean)
Range	The return values are: True: The Gain-Phase test port option is available. False: The Gain-Phase test port option is not available.
Preset Value	-
Unit	-
Resolution	-

Examples

```
Dim Var as Boolean
Var = SCPI.SERVice.SYSTem.OPTion.GPHase
```

Related Objects

SCPI.SERVice.SYSTem.OPTion.TPIMpedance

SCPI.SERVice.SYSTem.OPTion.TSET

Equivalent Key

No equivalent key is available on the front panel.

Equivalent SCPI Command

Syntax

:SERVice:SYSTem:OPTion:GPHase?

Query response

```
{1|0}<newline><^END>
```

Example of use

```
10 OUTPUT 717;":SERV:SYST:OPT:GPH?"
20 ENTER 717;A
```

SCPI.SERVice.SYSTem.OPTion.TPIMpedance

Object Type

Property (**Read Only**)

Syntax

Value = SCPI.SERVice.SYSTem.OPTion.TPIMpedance

Description

This query command gets the test port impedance option, either 50 or 75.

Variable

Parameter	*Value*
Description	Test port impedance
Data Type	Long integer type (Long)
Range	The return values are: 50: The value of test port impedance option is 50 Ohm. 75: The value of test port impedance option is 75 Ohm.
Preset Value	-
Unit	Ohm (Ω)
Resolution	-

Examples

```
Dim Var as Long
Var = SCPI.SERVice.SYSTem.OPTion.TPIMpedance
```

Related Objects

SCPI.SERVice.SYSTem.OPTion.GPHase

SCPI.SERVice.SYSTem.OPTion.TSET

Equivalent Key

No equivalent key is available on the front panel.

Equivalent SCPI Command

Syntax

:SERVice:SYSTem:OPTion:TPIMpedance?

Query Response

{50|75}<newline><^END>

Example of use

```
10 OUTPUT 717;":SERV:SYST:OPT:TPIM?"
20 ENTER 717;A
```

SCPI.SERVice.SYSTem.OPTion.TSET

Object Type

Property (**Read Only**)

Syntax

Value = SCPI.SERVice.SYSTem.OPTion.TSET

Description

This query command gets the value of the test set option, either S-Parameter or T/R.

Variable

Parameter	*Value*
Description	Test set option
Data Type	Character string type (String)
Range	The return values are: SPARAM: The value of test set option is S-Parameter. TR: The value of test set option is Transmission/Reflection.
Preset Value	-
Unit	-
Resolution	-

Examples

```
Dim Var as String
Var = SCPI.SERVice.SYSTem.OPTion.TSET
```

Related Objects

SCPI.SERVice.SYSTem.OPTion.GPHase

SCPI.SERVice.SYSTem.OPTion.TPIMpedance

Equivalent Key

No equivalent key is available on the front panel.

Equivalent SCPI Command

Syntax

:SERVice:SYSTem:OPTion:TSET?

Query Response

{SPARM|TR}><newline><^END>

Example of use

```
10 OUTPUT 717;":SERV:SYST:OPT:TSET?"
20 ENTER 717;A$
```

SCPI.SERVice.SYSTem.OS.REVision

Object Type

Property (**Read Only**)

Syntax

Value = SCPI.SERVice.SYSTem.OS.REVision

Description

This command gets the HDD Image revision. Returns 'EL11x' for EL100/EL110/112/114.

Variable

Parameter	*Value*
Description	Shows HDD image revision
Data Type	Character string type (String)
Range	254 chars
Preset Value	-
Unit	-
Resolution	-

Examples

```
Dim Var as String
Var = SCPI.SERVice.SYSTem.OS.REVision
```

Equivalent Key

System > **Firmware Revision**

HDD Image shows this status.

Equivalent SCPI Command

Syntax

:SERVice:SYSTem:OS:REVision?

Query Response

{String}<newline><^END>

Example of use

```
10 OUTPUT 717;":SERV:SYST:OS:REV?"
20 ENTER 717;A$
```

SOURCE

SCPI.SOURce(*Ch*).BIAS.CENTer

Object Type

Property (**Read-Write**)

Syntax

SCPI.SOURce(*Ch*).BIAS.CENTer = *Value*

Value = SCPI.SOURce(*Ch*).BIAS.CENTer

Description

This command sets/gets the center value of the bias sweep for the selected channel (*Ch*).

Variable

Parameter	*Value*
Description	Center voltage
Data Type	Double precision floating point type (Double)
Range	-40 to 40
Preset Value	0
Unit	V (voltage)
Unit	If the specified variable 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.

Examples

```
Dim CentVolt as Double
SCPI.SOURce(4).BIAS.CENTer = 3.0
CentVolt = SCPI.SOURce(4).BIAS.CENTer
```

Related Objects

SCPI.SOURce(Ch).BIAS.SPAN

SCPI.SOURce(Ch).BIAS.STARt

SCPI.SOURce(Ch).BIAS.STOP

Equivalent Key

Center

Equivalent SCPI Command

Syntax

:SOURce{[1]-4}:BIAS:CENTer <numeric>

:SOURce{[1]-4}:BIAS:CENTer?

Query Response

{numeric} <newline><^END>

Example of use

```
10 OUTPUT 717;":SOUR1:BIAS:CENT 1.2"
20 OUTPUT 717;":SOUR1:BIAS:CENT?"
30 ENTER 717;A
```

SCPI.SOURce(Ch).BIAS.RANGe.AUTO

Object type

Property (**Read-Write**)

Syntax

SCPI.SOURce(*Ch*).BIAS.RANGe.AUTO = *Status*

Status = SCPI.SOURce(*Ch*).BIAS.RANGe.AUTO

Description

This command sets/gets to fix the dc bias range at 40 V range for the selected channel (*Ch*).

Variable

Parameter	*Status*
Description	ON/OFF of the auto setting of the sweep time
Data type	Boolean type (Boolean)
Range	Select from either of the following: • True or ON: Auto. • False or OFF: Range is fixed at 40 V range.
Preset value	True or ON

Examples

```
Dim DCBiasRange As Boolean
SCPI.SOURce(1).BIAS.RANGe.AUTO = False
DCBiasRange = SCPI.SOURce(1).BIAS.RANGe.AUTO
```

Equivalent key

System > **Service Menu** > **DC Bias Range**

Equivalent SCPI command

Syntax

:SOURce{[1]-4}:BIAS:RANGe:AUTO {ON|OFF|1|0}

:SOURce{[1]-4}:BIAS:RANGe:AUTO?

Query response

{1|0}<newline><^END>

Example of use

```
10 OUTPUT 717;":SOUR1:BIAS:RANG:AUTO OFF"
20 OUTPUT 717;":SOUR1:BIAS:RANG:AUTO?"
30 ENTER 717;A
```

SCPI.SOURce(*Ch*).BIAS.SPAN

Object Type

Property (**Read-Write**)

Syntax

SCPI.SOURce(*Ch*).BIAS.SPAN = *Value*

Value = SCPI.SOURce(*Ch*).BIAS.SPAN

Description

This command sets/gets the span value of the bias sweep for the selected channel (*Ch*).

Variable

Parameter	*Value*
Description	Span voltage
Data Type	Double precision floating point type (Double)
Range	0 to 80
Preset Value	2
Unit	V (voltage)
Note	If the specified variable 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.

Examples

```
Dim SpanVolt as Double
SCPI.SOURce(3).BIAS.SPAN = 4.0
SpanVolt = SCPI.SOURce(3).BIAS.SPAN
```

Related Objects

SCPI.SOURce(Ch).BIAS.CENTer

SCPI.SOURce(Ch).BIAS.STARt

SCPI.SOURce(Ch).BIAS.STOP

Equivalent Key

Span

Equivalent SCPI Command

Syntax

```
:SOURce{[1]-4}:BIAS:SPAN <numeric>
:SOURce{[1]-4}:BIAS:SPAN?
```

Query Response

```
{numeric} <newline><^END>
```

Example of use

```
10 OUTPUT 717;":SOUR1:BIAS:SPAN 2.0"
20 OUTPUT 717;":SOUR1:BIAS:SPAN?"
30 ENTER 717;A
```

SCPI.SOURce(*Ch*).BIAS.STARt

Object Type

Property (**Read-Write**)

Syntax

SCPI.SOURce(*Ch*).BIAS.STARt = *Value*

Value = SCPI.SOURce(*Ch*).BIAS.STARt

Description

This command sets/gets the start value of the bias sweep for the selected channel (*Ch*).

Variable

Parameter	*Value*
Description	Start voltage
Data Type	Double precision floating point type (Double)
Range	-40 to 40
Preset Value	-1
Unit	V (voltage)
Note	If the specified variable 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.

Examples

```
Dim StarVolt as Double
SCPI.SOURce(1).BIAS.STARt = -1.5
StarVolt = SCPI.SOURce(1).BIAS.STARt
```

Related Objects

SCPI.SOURce(Ch).BIAS.CENTer

SCPI.SOURce(Ch).BIAS.SPAN

SCPI.SOURce(Ch).BIAS.STOP

Equivalent Key

Start

Equivalent SCPI Command

Syntax

:SOURce{[1]-4}:BIAS:STARt <numeric>

:SOURce{[1]-4}:BIAS:STARt?

Query Response

{numeric} <newline><^END>

Example of use

```
10 OUTPUT 717;":SOUR2:BIAS:STAR -12.0"
20 OUTPUT 717;":SOUR2:BIAS:STAR?"
30 ENTER 717;A
```

SCPI.SOURce(*Ch*).BIAS.STOP

Object Type

Property (**Read-Write**)

Syntax

SCPI.SOURce(*Ch*).BIAS.STOP = *Value*

Value = SCPI.SOURce(*Ch*).BIAS.STOP

Description

This command sets/gets the stop value of the bias sweep for the selected channel (*Ch*).

Variable

Parameter	*Value*
Description	Stop voltage
Data Type	Double precision floating point type (Double)
Range	-40 to 40
Preset Value	1
Unit	V (voltage)
Note	If the specified variable 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.

Examples

```
Dim StopVolt as Double
SCPI.SOURce(1).BIAS.STOP = 2.5
StopVolt = SCPI.SOURce(1).BIAS.STOP
```

Related Objects

SCPI.SOURce(Ch).BIAS.CENTer

SCPI.SOURce(Ch).BIAS.SPAN

SCPI.SOURce(Ch).BIAS.STARt

Equivalent Key

Stop

Equivalent SCPI Command

Syntax

:SOURce{[1]-4}:BIAS:STOP <numeric>

:SOURce{[1]-4}:BIAS:STOP?

Query Response

{numeric} <newline><^END>

Example of use

```
10 OUTPUT 717;":SOUR2:BIAS:STOP 5.0"
20 OUTPUT 717;":SOUR2:BIAS:STOP?"
30 ENTER 717;A
```

SCPI.SOURce(*Ch*).POWer.ATTenuation.DATA

Object type

Property (**Read-Write**)

Syntax

SCPI.SOURce(*Ch*).POWer.ATTenuation.DATA = *Value*

Value = SCPI.SOURce(*Ch*).POWer.ATTenuation.DATA

Description

This command does nothing. The E5062B has no attenuator setting for power. This command is provided because of its command compatibility with E5061A/E5062B.

Variable

Parameter	*Value*
Description	Dummy Parameter
Data type	Long integer type (Long)
Range	0 to 40
Preset value	0
Unit	dB
Resolution	10

Examples

```
Dim Att As Long
SCPI.SOURce(1).POWer.ATTenuation.DATA = 10
Att = SCPI.SOURce(1).POWer.ATTenuation.DATA
```

Related objects

None.

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

:SOURce{[1]-4}:POWer:ATTenuation[:DATA] <numeric>

:SOURce{[1]-4}:POWer:ATTenuation[:DATA]?

Query response

0<newline><^END>

Example of use

```
10 OUTPUT 717;":SOUR1:POW:ATT.DATA 20"
20 OUTPUT 717;":SOUR1:POW:ATT.DATA?"
30 ENTER 717;A
```

SCPI.SOURce(*Ch*).POWer.CENTer

Object type

Property (**Read-Write**)

Syntax

SCPI.SOURce(*Ch*).POWer.CENTer = *Value*

Value = SCPI.SOURce(*Ch*).POWer.CENTer

Description

This command sets/gets the center value of the power sweep for the selected channel (*Ch*).

Variable

Parameter	*Value*
Description	Center value
Data type	Double precision floating point type (Double)
Range	−45 to 10 dBm
Preset value	-2.5
Unit	dBm
Resolution	0.05 or 0.025
Note	If the specified variable 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.

Examples

```
Dim Pcntr As Double
SCPI.SOURce(1).POWer.CENTer = 0
Pcntr = SCPI.SOURce(1).POWer.CENTer
```

Related objects

SCPI.SENSe(Ch).SWEep.TYPE

SCPI.SOURce(Ch).POWer.SPAN

Equivalent key

Center

Equivalent SCPI command

Syntax

:SOURce{[1]-4}:POWer:CENTer <numeric>

:SOURce{[1]-4}:POWer:CENTer?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":SOUR1:POW:CENT 0"
20 OUTPUT 717;":SOUR1:POW:CENT?"
30 ENTER 717;A
```

SCPI.SOURce(*Ch*).POWer.GPPort.LEVel.IMMediate.AMPLitude

Object Type

Property (**Read-Write**)

Syntax

SCPI.SOURce(*Ch*).POWer.GPPort.LEVel.IMMediate.AMPLitude = *Value*

Value = SCPI.SOURce(*Ch*).POWer.GPPort.LEVel.IMMediate.AMPLitude

Description

This command sets/gets the power level for gain phase measurement (LF out) for the selected channel (*Ch*).

Variable

Parameter	*Value*
Description	Power level
Data Type	Double precision floating point type (Double)
Range	-45 to 10 dBm
Preset Value	0
Unit	dBm
Resolution	0.05
Note	If the specified variable 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.

Examples

```
Dim GPLev as Double
SCPI.SOURce(2).POWer.GPPort.LEVel.IMMediate.AMPLitude = -15.0
GPLev = SCPI.SOURce(1).POWer.GPPort.LEVel.IMMediate.AMPLitude
```

Related Objects

SCPI.SOURce(Ch).POWer.PORT(Pt).LEVel.IMMediate.AMPLitude

SCPI.OUTPut.STATe

Equivalent Key

Sweep Setup > **Power** > **Port Power** > **LF OUT Power**

Equivalent SCPI Command

Syntax

:SOURce{[1]-4}:POWer:GPPort[:LEVel][:IMMediate][:AMPLitude] <numeric>

:SOURce{[1]-4}:POWer:GPPort[:LEVel][:IMMediate][:AMPLitude]?

Query Response

{numeric} <newline><^END>

Example of use

```
10 OUTPUT 717;":SOUR1:POW:GPP -5.0"
20 OUTPUT 717;":SOUR1:POW:GPP?"
30 ENTER 717;A
```

SCPI.SOURce(*Ch*).POWer.LEVel.IMMediate.AMPLitude

Object type

Property (**Read-Write**)

Syntax

SCPI.SOURce(*Ch*).POWer.LEVel.IMMediate.AMPLitude = *Value*

Value = SCPI.SOURce(*Ch*).POWer.LEVel.IMMediate.AMPLitude

Description

This command sets/gets the power level of the selected channel (*Ch*).

Variable

Parameter	*Value*
Description	Power level
Data type	Double precision floating point type (Double)
Range	-45 to 10 dBm
Preset value	0
Unit	dBm
Resolution	0.05
Note	If the specified variable 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.

Examples

```
Dim PowLev As Double
SCPI.SOURce(1).POWer.LEVel.IMMediate.AMPLitude = -10
PowLev = SCPI.SOURce(1).POWer.LEVel.IMMediate.AMPLitude
```

Related objects

SCPI.SOURce(Ch).POWer.GPPort.LEVel.IMMediate.AMPLitude

SCPI.OUTPut.STATe

Equivalent key

Sweep Setup > **Power** > **Power**

Equivalent SCPI command

Syntax

:SOURce{[1]-4}:POWer[:LEVel][:IMMediate][:AMPLitude] <numeric>

:SOURce{[1]-4}:POWer[:LEVel][:IMMediate][:AMPLitude]?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":SOUR1:POW -12.5"
20 OUTPUT 717;":SOUR1:POW?"
30 ENTER 717;A
```

SCPI.SOURce(*Ch*).POWer.LEVel.SLOPe.DATA

Object type

Property (**Read-Write**)

Syntax

SCPI.SOURce(*Ch*).POWer.LEVel.SLOPe.DATA = *Value*

Value = SCPI.SOURce(*Ch*).POWer.LEVel.SLOPe.DATA

Description

This command sets/gets the correction value of the power slope feature of the selected channel (*Ch*).

Variable

Parameter	*Value*
Description	Correction value of the power slope feature
Data type	Double precision floating point type (Double)
Range	-2 to 2
Preset value	0
Unit	dB/GHz
Resolution	0.01
Note	If the specified variable 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.

Examples

```
Dim SlopLev As Double
SCPI.SOURce(1).POWer.LEVel.SLOPe.DATA = 0.1
SlopLev = SCPI.SOURce(1).POWer.LEVel.SLOPe.DATA
```

Related objects

SCPI.SOURce(Ch).POWer.LEVel.SLOPe.STATe

Equivalent key

Sweep Setup > **Power** > **Slope**

Equivalent SCPI command

Syntax

:SOURce{[1]-4}:POWer[:LEVel]:SLOPe[:DATA] <numeric>

:SOURce{[1]-4}:POWer[:LEVel]:SLOPe[:DATA]?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":SOUR1:POW:SLOP 0.1"
20 OUTPUT 717;":SOUR1:POW:SLOP?"
30 ENTER 717;A
```

SCPI.SOURce(*Ch*).POWer.LEVel.SLOPe.STATe

Object type

Property (**Read-Write**)

Syntax

SCPI.SOURce(*Ch*).POWer.LEVel.SLOPe.STATe = *Status*

Status = SCPI.SOURce(*Ch*).POWer.LEVel.SLOPe.STATe

Description

This command turns ON/OFF or gets the status of the power slope feature, for the selected channel (*Ch*). This command corrects the attenuation of simple power level proportional to the frequency (attenuation due to cables etc).

Variable

Parameter	*Status*
Description	On/off of the power slope feature
Data type	Boolean type (Boolean)
Range	Select from either of the following: • True or ON: Turns on the power slop feature. • False or OFF: Turns off the power slop feature.
Preset value	False or OFF

Examples

```
Dim Slop As Boolean
SCPI.SOURce(1).POWer.LEVel.SLOPe.STATe = True
Slop = SCPI.SOURce(1).POWer.LEVel.SLOPe.STATe
```

Related objects

SCPI.SOURce(Ch).POWer.LEVel.SLOPe.DATA

Equivalent key

Sweep Setup > **Power** > **Slope**

Equivalent SCPI command

Syntax

```
:SOURce{[1]-4}:POWer[:LEVel]:SLOPe:STATe {ON|OFF|1|0}
:SOURce{[1]-4}:POWer[:LEVel]:SLOPe:STATe?
```

Query response

```
{1|0}<newline><^END>
```

Example of use

```
10 OUTPUT 717;":SOUR1:POW:SLOP:STAT ON"
20 OUTPUT 717;":SOUR1:POW:SLOP:STAT?"
30 ENTER 717;A
```

SCPI.SOURce(*Ch*).POWer.PORT(*Pt*).LEVel.IMMediate.AMPLitude

Object type

Property (**Read-Write**)

Syntax

SCPI.SOURce(*Ch*).POWer.PORT(*Pt*).LEVel.IMMediate.AMPLitude = *Value*

Value = SCPI.SOURce(*Ch*).POWer.PORT(*Pt*).LEVel.IMMediate.AMPLitude

Description

This command sets/gets the power level for the selected port (*Pt*) of the selected channel (*Ch*).

Variable

Parameter	*Value*
Description	Power level at the specified port.
Data type	Double precision floating point type (Double)
Range	-45 to 10 dBm
Preset value	0
Unit	dBm
Resolution	0.05
Note	If the specified variable 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.

Examples

```
Dim PowLev As Double
SCPI.SOURce(1).POWer.PORT.COUPle = False
SCPI.SOURce(1).POWer.PORT(1).LEVel.IMMediate.AMPLitude = -12.5
PowLev = SCPI.SOURce(1).POWer.PORT(1).LEVel.IMMediate.AMPLitude
```

Related objects

SCPI.SOURce(Ch).POWer.PORT.COUPle

SCPI.SOURce(Ch).POWer.GPPort.LEVel.IMMediate.AMPLitude

Equivalent key

Sweep Setup > **Power** > **Port Power** > **Port 1 Power|Port 2 Power**

Equivalent SCPI command

Syntax

:SOURce{[1]-4}:POWer:PORT{[1]|2}[:LEVel][:IMMediate][:AMPLitude] <numeric>

:SOURce{[1]-4}:POWer:PORT{[1]|2}[:LEVel][:IMMediate][:AMPLitude]?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":SOUR1:POW:PORT1 -12.5"
20 OUTPUT 717;":SOUR1:POW:PORT1?"
30 ENTER 717;A
```

SCPI.SOURce(*Ch*).POWer.PORT.COUPle

Object type

Property (**Read-Write**)

Syntax

SCPI.SOURce(*Ch*).POWer.PORT.COUPle = *Status*

Status = SCPI.SOURce(*Ch*).POWer.PORT.COUPle

Description

This command sets/gets whether to output the same power level for each port of the selected channel (*Ch*).

When the power slope feature is ON, the same power level is always output to all ports regardless of this setting because different power levels cannot be output for each port.

Variable

Parameter	*Status*
Description	Turns ON/OFF the coupling between ports for the power level output
Data type	Boolean type (Boolean)
Range	Select from either of the following: • True or ON: Outputs the same power level to individual ports. • False or OFF: Outputs different power levels to individual ports.
Preset value	True or ON

Examples

```
Dim OutCpl As Boolean
SCPI.SOURce(1).POWer.PORT.COUPle = False
OutCpl = SCPI.SOURce(1).POWer.PORT.COUPle
```

Related objects

SCPI.SOURce(Ch).POWer.PORT(Pt).LEVel.IMMediate.AMPLitude

Equivalent key

Sweep Setup > **Power** > **Port Couple**

Equivalent SCPI command

Syntax

```
:SOURce{[1]-4}:POWer:PORT:COUPle {ON|OFF|1|0}
:SOURce{[1]-4}:POWer:PORT:COUPle?
```

Query response

```
{1|0}<newline><^END>
```

Example of use

```
10 OUTPUT 717;":SOUR1:POW:PORT:COUP OFF"
20 OUTPUT 717;":SOUR1:POW:PORT:COUP?"
30 ENTER 717;A
```

SCPI.SOURce(*Ch*).POWer.SPAN

Object type

Property (**Read-Write**)

Syntax

SCPI.SOURce(*Ch*).POWer.SPAN = *Value*

Value = SCPI.SOURce(*Ch*).POWer.SPAN

Description

This command sets/gets the span value of the power sweep for the selected channel (*Ch*).

Variable

Parameter	*Value*
Description	Span value
Data type	Double precision floating point type (Double)
Range	0 to 55 dB
Preset value	15
Unit	dB
Resolution	0.05
Note	If the specified variable 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.

Examples

```
Dim Pspan As Double
SCPI.SOURce(1).POWer.SPAN = 10
Pspan = SCPI.SOURce(1).POWer.SPAN
```

Related objects

SCPI.SENSe(Ch).SWEep.TYPE

SCPI.SOURce(Ch).POWer.CENTer

Equivalent key

Span

Equivalent SCPI command

Syntax

:SOURce{[1]-4}:POWer:SPAN <numeric>

:SOURce{[1]-4}:POWer:SPAN?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":SOUR1:POW:SPAN 10"
20 OUTPUT 717;":SOUR1:POW:SPAN?"
30 ENTER 717;A
```

SCPI.SOURce(*Ch*).POWer.STARt

Object type

Property (**Read-Write**)

Syntax

SCPI.SOURce(*Ch*).POWer.STARt = *Value*

Value = SCPI.SOURce(*Ch*).POWer.STARt

Description

This command sets/gets the start value of the power sweep for the selected channel (*Ch*).

Variable

Parameter	*Value*
Description	Start value
Data type	Double precision floating point type (Double)
Range	−45 to 10 dBm
Preset value	-5
Unit	dBm
Resolution	0.05

Examples

```
Dim Pstart As Double
SCPI.SOURce(1).POWer.STARt = -10
Pstart = SCPI.SOURce(1).POWer.STARt
```

Related objects

SCPI.SENSe(Ch).SWEep.TYPE

SCPI.SOURce(Ch).POWer.STOP

Equivalent key

Start

Equivalent SCPI command

Syntax

:SOURce{[1]-4}:POWer:STARt <numeric>

:SOURce{[1]-4}:POWer:STARt?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":SOUR1:POW:STAR -10"
20 OUTPUT 717;":SOUR1:POW:STAR?"
30 ENTER 717;A
```

SCPI.SOURce(*Ch*).POWer.STOP

Object type

Property (**Read-Write**)

Syntax

SCPI.SOURce(*Ch*).POWer.STOP = *Value*

Value = SCPI.SOURce(*Ch*).POWer.STOP

Description

This command sets/gets the stop value of the power sweep for the selected channel (*Ch*).

Variable

Parameter	*Value*
Description	Stop value
Data type	Double precision floating point type (Double)
Range	−45 to 10 dBm
Preset value	0
Unit	dBm
Resolution	0.05
Note	If the specified variable 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.

Examples

```
Dim Pstop As Double
SCPI.SOURce(1).POWer.STOP = 10
Pstop = SCPI.SOURce(1).POWer.STOP
```

Related objects

SCPI.SENSe(Ch).SWEep.TYPE

SCPI.SOURce(Ch).POWer.STARt

Equivalent key

Stop

Equivalent SCPI command

Syntax

:SOURce{[1]-4}:POWer:STOP <numeric>

:SOURce{[1]-4}:POWer:STOP?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":SOUR1:POW:STOP 10"
20 OUTPUT 717;":SOUR1:POW:STOP?"
30 ENTER 717;A
```

SCPI.SOURce.BIAS.ENABle

Object Type

Property (**Read-Write**)

Syntax

SCPI.SOURce.BIAS.ENABle = *Status*

Status = SCPI.SOURce.BIAS.ENABle

Description

This command turns ON/OFF the DC bias output.

Measurement cannot be made until the DC bias output is turned ON.

Variable

Parameter	*Status*
Description	ON/OFF of the stimulus signal output
Data type	Boolean type (Boolean)
Range	Select from the following: • True or ON: Turns ON the DC bias output. • False or OFF: Turns OFF the DC bias output.
Preset value	False or OFF

Examples

```
Dim StatBias as Boolean
SCPI.SOURce.BIAS.ENABle = True
StatBias = SCPI.SOURce.BIAS.ENABle
```

Related Objects

SCPI.SOURce.BIAS.PORT

SCPI.SOURce.BIAS.VOLTage

Equivalent Key

Sweep Setup > **DC Bias**

System > **Overload Recovery** > **DC Bias**

Equivalent SCPI Command

Syntax

:SOURce:BIAS:ENABle {ON|OFF|1|0}

:SOURce:BIAS:ENABle?

Query Response

{1|0} <newline><^END>

Example of use

```
10 OUTPUT 717;":SOUR:BIAS:ENAB ON"
20 OUTPUT 717;":SOUR:BIAS:ENAB?"
30 ENTER 717;A
```

SCPI.SOURce.BIAS.PORT

Object Type

Property (**Read-Write**)

Syntax

SCPI.SOURce.BIAS.PORT = *Param*

Param = SCPI.SOURce.BIAS.PORT

Description

This command sets/gets the port of DC bias output. If this parameter changed, DC bias output is turned OFF.

Variable

Parameter	*Param*
Description	DC bias output port
Data Type	Character string type (String)
Range	Select from either of the following: • "LFOut": Specifies the LF output port. • "P1": Specifies the Port 1.
Preset Value	"LFOut"

Examples

```
Dim DCPort as String
SCPI.SOURce.BIAS.PORT = "LFOut"
DCPort = SCPI.SOURce.BIAS.PORT
```

Related Objects

SCPI.SOURce.BIAS.ENABle

SCPI.SOURce.BIAS.VOLTage

Equivalent Key

Sweep Setup > **DC Bias Port** > **LF Out|Port 1**

System > **Overload Recovery** > **DC Bias Port** > **LF Out|Port 1**

Equivalent SCPI Command

Syntax

:SOURce:BIAS:PORT {LFOut|P1}

:SOURce:BIAS:PORT?

Query Response

{LFO|P1} <newline><^END>

Example of use

```
10 OUTPUT 717;":SOUR:BIAS:PORT P1"
20 OUTPUT 717;":SOUR:BIAS:PORT?"
30 ENTER 717;A$
```

SCPI.SOURce.BIAS.VOLTage

Object Type

Property (**Read-Write**)

Syntax

SCPI.SOURce.BIAS.VOLTage = *Value*

Value = SCPI.SOURce.BIAS.VOLTage

Description

This command sets/gets the output level for DC bias output.

Variable

Parameter	*Value*
Description	Output level
Data Type	Double precision floating point type (Double)
Range	-40 to 40
Preset Value	0
Unit	V (voltage)
Note	If the specified variable 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.

Examples

```
Dim DCLev as Double
SCPI.SOURce.BIAS.VOLTage = 8.5
DCLev = SCPI.SOURce.BIAS.VOLTage
```

Related Objects

SCPI.SOURce.BIAS.ENABle

SCPI.SOURce.BIAS.PORT

Equivalent Key

Sweep Setup > **DC Bias Level**

System > **Overload Recovery** > **DC Bias Level**

Equivalent SCPI Command

Syntax

:SOURce:BIAS:VOLTage <numeric>

:SOURce:BIAS:VOLTage?

Query Response

{numeric} <newline><^END>

Example of use

```
10 OUTPUT 717;":SOUR:BIAS:VOLT 5.2"
20 OUTPUT 717;":SOUR:BIAS:VOLT?"
30 ENTER 717;A
```

STATUS

SCPI.STATus.OPERation.CONDition

Object type

Property (**Read Only**)

Syntax

Value = SCPI.STATus.OPERation.CONDition

Description

This command reads the value of the Operation Status Condition Register.

Variable

Parameter	*Value*
Description	Value of the Operation Status Condition Register
Data type	Long integer type (Long)

Examples

```
Dim Stat As Long
Stat = SCPI.STATus.OPERation.CONDition
```

Related objects

SCPI.STATus.OPERation.NTRansition

SCPI.STATus.OPERation.PTRansition

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

:STATus:OPERation:CONDition?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":STAT:OPER:COND?"
20 ENTER 717;A
```

SCPI.STATus.OPERation.ENABle

Object type

Property (**Read-Write**)

Syntax

SCPI.STATus.OPERation.ENABle = *Value*

Value = SCPI.STATus.OPERation.ENABle

Description

This command sets/gets the value of Operation Status Enable Register.

Variable

Parameter	*Value*
Description	Value of the Operation Status Enable Register
Data type	Long integer type (Long)
Range	0 to 65535
Preset value	0
Note	The bit 0 to 3, bit 6 to 13 and bit 15 can not be set to 1.

Examples

```
Dim Stat As Long
SCPI.STATus.OPERation.ENABle = 16
Stat = SCPI.STATus.OPERation.ENABle
```

Related objects

SCPI.IEEE4882.SRE

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

:STATus:OPERation:ENABle <numeric>

:STATus:OPERation:ENABle?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":STAT:OPER:ENAB 16"
20 OUTPUT 717;":STAT:OPER:ENAB?"
30 ENTER 717;A
```

SCPI.STATus.OPERation.EVENt

Object type

Property (**Read Only**)

Syntax

Value = SCPI.STATus.OPERation.EVENt

Description

This command reads the value of the Operation Status Event Register.

Variable

Parameter	*Value*
Description	Value of the Operation Status Event Register
Data type	Long integer type (Long)

Examples

```
Dim Stat As Long
Stat = SCPI.STATus.OPERation.EVENt
```

Related objects

SCPI.IEEE4882.CLS

SCPI.STATus.OPERation.NTRansition

SCPI.STATus.OPERation.PTRansition

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

:STATus:OPERation[:EVENt]?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":STAT:OPER?"
20 ENTER 717;A
```

SCPI.STATus.OPERation.NTRansition

Object type

Property (**Read-Write**)

Syntax

SCPI.STATus.OPERation.NTRansition = *Value*

Value = SCPI.STATus.OPERation.NTRansition

Description

This command sets/gets the value of negative transition filter of the Operation Status Register.

Variable

Parameter	*Value*
Description	Value of the negative transition filter
Data type	Long integer type (Long)
Range	0 to 65535
Preset value	0
Note	The bit 0 to 3, bit 6 to 13 and bit 15 can not be set to 1.

Examples

```
Dim Stat As Long
SCPI.STATus.OPERation.NTRansition = 16
Stat = SCPI.STATus.OPERation.NTRansition
```

Related objects

SCPI.STATus.OPERation.EVENt

SCPI.STATus.OPERation.PTRansition

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

:STATus:OPERation:NTRansition <numeric>

:STATus:OPERation:NTRansition?

Query response

<numeric> is printed as {numeric}

```
{numeric}<newline><^END>
```

Example of use

```
10 OUTPUT 717;":STAT:OPER:NTR 16"
20 OUTPUT 717;":STAT:OPER:NTR?"
30 ENTER 717;A
```

SCPI.STATus.OPERation.PTRansition

Object type

Property (**Read-Write**)

Syntax

SCPI.STATus.OPERation.PTRansition = *Value*

Value = SCPI.STATus.OPERation.PTRansition

Description

This command sets/gets the value of positive transition filter of the Operation Status Register.

Variable

Parameter	*Value*
Description	Value of the positive transition filter
Data type	Long integer type (Long)
Range	0 to 65535
Preset value	16432
Note	The bit 0 to 3, bit 6 to 13 and bit 15 can not be set to 1.

Examples

```
Dim Stat As Long
SCPI.STATus.OPERation.PTRansition = 0
Stat = SCPI.STATus.OPERation.PTRansition
```

Related objects

SCPI.STATus.OPERation.EVENt

SCPI.STATus.OPERation.NTRansition

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

:STATus:OPERation:PTRansition <numeric>

:STATus:OPERation:PTRansition?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":STAT:OPER:PTR 16"
20 OUTPUT 717;":STAT:OPER:PTR?"
30 ENTER 717;A
```

SCPI.STATus.PRESet

Object type

Method (**Write Only**)

Syntax

SCPI.STATus.PRESet

Description

This command initializes all the status registers.

Examples

```
SCPI.STATus.PRESet
```

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

:STATus:PRESet

Example of use

```
10 OUTPUT 717;":STAT:PRES"
```

SCPI.STATus.QUEStionable.BLIMit.CHANnel(*Ch*).CONDition

Object type

Property (**Read Only**)

Syntax

Value = SCPI.STATUS.QUEStionable.BLIMit.CHANnel(*Ch*).CONDition

Description

This command reads the value of the Questionable Bandwidth Limit Channel Status Condition Register of the specified channel.

Variable

Parameter	*Value*
Description	The value of the Questionable Bandwidth Limit Channel Status Condition Register
Data type	Long integer type (Long)

For information on the variable (*Ch*), see Ch.

Examples

```
Dim Stat As Long
Stat = SCPI.STATus.QUEStionable.BLIMit.CHANnel(1).CONDition
```

Related objects

SCPI.STATus.QUEStionable.BLIMit.CHANnel(Ch).NTRansition

SCPI.STATus.QUEStionable.BLIMit.CHANnel(Ch).PTRansition

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

:STATus:QUEStionable:BLIMit:CHANnel{[1]-4}:CONDition?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":STAT:QUES:BLIM:CHAN1:COND?"
20 ENTER 717;A
```

SCPI.STATus.QUEStionable.BLIMit.CHANnel(*Ch*).ENABle

Object type

Property (**Read-Write**)

Syntax

SCPI.STATus.QUEStionable.BLIMit.CHANnel(*Ch*).ENABle = *Value*

Value = SCPI.STATus.QUEStionable.BLIMit.CHANnel(*Ch*).ENABle

Description

This command sets/gets the value of the "Questionable Bandwidth Limit Channel Status Enable Register" for the specified channel.

Variable

Parameter	*Value*
Description	The value of the Questionable Bandwidth Limit Channel Status Enable Register
Data type	Long integer type (Long)
Range	0 to 65535
Preset value	Varies depending on the upper limit setting of the number of channels/traces.
Note	The bit 0, and bit 5 to 15 can not be set to 1.

For information on the variable (*Ch*), see Ch.

Examples

```
Dim Stat As Long
SCPI.STATus.QUEStionable.BLIMit.CHANnel(1).ENABle = 16
Stat = SCPI.STATus.QUEStionable.BLIMit.CHANnel(1).ENABle
```

Related objects

SCPI.STATus.QUEStionable.BLIMit.ENABle

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

:STATus:QUEStionable:BLIMit:CHANnel{[1]-4}:ENABle <numeric>

:STATus:QUEStionable:BLIMit:CHANnel{[1]-4}:ENABle?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":STAT:QUES:BLIM:CHAN1:ENAB 16"
20 OUTPUT 717;":STAT:QUES:BLIM:CHAN1:ENAB?"
30 ENTER 717;A
```

SCPI.STATus.QUEStionable.BLIMit.CHANnel(*Ch*).EVENt

Object type

Property (**Read Only**)

Syntax

Value = SCPI.STATus.QUEStionable.BLIMit.CHANnel(*Ch*).EVENt

Description

This command reads the value of the "Questionable Bandwidth Limit Channel Status Event Register" of the specified channel.

Variable

Parameter	*Value*
Description	The value of the Questionable Bandwidth Limit Channel Status Event Register
Data type	Long integer type (Long)

For information on the variable (*Ch*), see Ch.

Examples

```
Dim Stat As Long
Stat = SCPI.STATus.QUEStionable.BLIMit.CHANnel(1).EVENt
```

Related objects

SCPI.IEEE4882.CLS

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

:STATus:QUEStionable:BLIMit:CHANnel{[1]-4}[:EVENt]?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":STAT:QUES:BLIM:CHAN1?"
20 ENTER 717;A
```

SCPI.STATus.QUEStionable.BLIMit.CHANnel(*Ch*).NTRansition

Object type

Property (**Read-Write**)

Syntax

SCPI.STATus.QUEStionable.BLIMit.CHANnel(*Ch*).NTRansition = *Value*

Value = SCPI.STATus.QUEStionable.BLIMit.CHANnel(*Ch*).NTRansition

Description

This command sets/gets the value of the negative transition filter of the "Questionable Bandwidth Limit Channel Status Register" for the specified channel.

Variable

Parameter	*Value*
Description	The value of the negative transition filter
Data type	Long integer type (Long)
Range	0 to 65535
Preset value	0
Note	The bit 0, and bit 5 to 15 can not be set to 1.

For information on the variable (*Ch*), see Ch.

Examples

```
Dim Stat As Long
SCPI.STATus.QUEStionable.BLIMit.CHANnel(1).NTRansition = 16
Stat = SCPI.STATus.QUEStionable.BLIMit.CHANnel(1).NTRansition
```

Related objects

SCPI.STATus.QUEStionable.BLIMit.CHANnel(Ch).EVENt

SCPI.STATus.QUEStionable.BLIMit.CHANnel(Ch).PTRansition

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

:STATus:QUEStionable:BLIMit:CHANnel{[1]-4}:NTRansition <numeric>

:STATus:QUEStionable:BLIMit:CHANnel{[1]-4}:NTRansition?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":STAT:QUES:BLIM:CHAN1:NTR 16"
20 OUTPUT 717;":STAT:QUES:BLIM:CHAN1:NTR?"
30 ENTER 717;A
```

SCPI.STATus.QUEStionable.BLIMit.CHANnel(*Ch*).PTRansition

Object type

Property (**Read-Write**)

Syntax

SCPI.STATus.QUEStionable.BLIMit.CHANnel(*Ch*).PTRansition = *Value*

Value = SCPI.STATus.QUEStionable.BLIMit.CHANnel(*Ch*).PTRansition

Description

This command sets/gets the value of the positive transition filter of the "Questionable Bandwidth Limit Channel Status Register" for the specified channel.

Variable

Parameter	*Value*
Description	The value of the positive transition filter
Data type	Long integer type (Long)
Range	0 to 65535
Preset value	Varies depending on the upper limit setting of the number of channels/traces.
Note	The bit 0, and bit 5 to 15 can not be set to 1.

For information on the variable (*Ch*), see Ch.

Examples

```
Dim Stat As Long
SCPI.STATus.QUEStionable.BLIMit.CHANnel(1).PTRansition = 0
Stat = SCPI.STATus.QUEStionable.BLIMit.CHANnel(1).PTRansition
```

Related objects

SCPI.STATus.QUEStionable.BLIMit.CHANnel(Ch).EVENt

SCPI.STATus.QUEStionable.BLIMit.CHANnel(Ch).NTRansition

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

:STATus:QUEStionable:BLIMit:CHANnel{[1]-4}:PTRansition <numeric>

:STATus:QUEStionable:BLIMit:CHANnel{[1]-4}:PTRansition?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":STAT:QUES:BLIM:CHAN1:PTR 16"
20 OUTPUT 717;":STAT:QUES:BLIM:CHAN1:PTR?"
30 ENTER 717;A
```

SCPI.STATus.QUEStionable.BLIMit.CONDition

Object type

Property (**Read Only**)

Syntax

Value = SCPI.STATus.QUEStionable.BLIMit.CONDition

Description

This command reads the value of the Questionable Bandwidth Limit Status Condition Register.

Variable

Parameter	*Value*
Description	The value of the Questionable Bandwidth Limit Status Condition Register.
Data type	Long integer type (Long)

Examples

```
Dim Stat As Long
Stat = SCPI.STATus.QUEStionable.BLIMit.CONDition
```

Related objects

SCPI.STATus.QUEStionable.BLIMit.NTRansition

SCPI.STATus.QUEStionable.BLIMit.PTRansition

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

:STATus:QUEStionable:BLIMit:CONDition?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":STAT:QUES:BLIM:COND?"
20 ENTER 717;A
```

SCPI.STATus.QUEStionable.BLIMit.ENABle

Object type

Property (**Read-Write**)

Syntax

SCPI.STATus.QUEStionable.BLIMit.ENABle = *Value*

Value = SCPI.STATus.QUEStionable.BLIMit.ENABle

Description

This command sets/gets the value of the Questionable Bandwidth Limit Status Enable Register.

Variable

Parameter	*Value*
Description	The value of the Questionable Bandwidth Limit Status Enable Register.
Data type	Long integer type (Long)
Range	0 to 65535
Preset value	Varies depending on the upper limit setting of the number of channels/traces.
Note	The bit 0, and bit 5 to 15 can not be set to 1.

Examples

```
Dim Stat As Long
SCPI.STATus.QUEStionable.BLIMit.ENABle = 16
Stat = SCPI.STATus.QUEStionable.BLIMit.ENABle
```

Related objects

SCPI.STATus.QUEStionable.ENABle

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

:STATus:QUEStionable:BLIMit:ENABle <numeric>

:STATus:QUEStionable:BLIMit:ENABle?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":STAT:QUES:BLIM:ENAB 16"
20 OUTPUT 717;":STAT:QUES:BLIM:ENAB?"
30 ENTER 717;A
```

SCPI.STATus.QUEStionable.BLIMit.EVENt

Object type

Property (**Read Only**)

Syntax

Value = SCPI.STATus.QUEStionable.BLIMit.EVENt

Description

This command reads the value of the Questionable Bandwidth Limit Status Event Register.

Variable

Parameter	*Value*
Description	The value of the Questionable Bandwidth Limit Status Event Register.
Data type	Long integer type (Long)

Examples

```
Dim Stat As Long
Stat = SCPI.STATus.QUEStionable.BLIMit.EVENt
```

Related objects

SCPI.IEEE4882.CLS

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

:STATus:QUEStionable:BLIMit[:EVENt]?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":STAT:QUES:BLIM?"
20 ENTER 717;A
```

SCPI.STATus.QUEStionable.BLIMit.NTRansition

Object type

Property (**Read-Write**)

Syntax

SCPI.STATus.QUEStionable.BLIMit.NTRansition = *Value*

Value = SCPI.STATus.QUEStionable.BLIMit.NTRansition

Description

This command sets/gets the value of the negative transition filter of the Questionable Bandwidth Limit Status Register.

Variable

Parameter	*Value*
Description	The value of the negative transition filter
Data type	Long integer type (Long)
Range	0 to 65535
Preset value	0
Note	The bit 0, and bit 5 to 15 can not be set to 1.

Examples

```
Dim Stat As Long
SCPI.STATus.QUEStionable.BLIMit.NTRansition = 6
Stat = SCPI.STATus.QUEStionable.BLIMit.NTRansition
```

Related objects

SCPI.STATus.QUEStionable.BLIMit.EVENt

SCPI.STATus.QUEStionable.BLIMit.PTRansition

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

:STATus:QUEStionable:BLIMit:NTRansition <numeric>

:STATus:QUEStionable:BLIMit:NTRansition?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":STAT:QUES:BLIM:NTR 16"
20 OUTPUT 717;":STAT:QUES:BLIM:NTR?"
30 ENTER 717;A
```

SCPI.STATus.QUEStionable.BLIMit.PTRansition

Object type

Property (**Read-Write**)

Syntax

SCPI.STATus.QUEStionable.BLIMit.PTRansition = *Value*

Value = SCPI.STATus.QUEStionable.BLIMit.PTRansition

Description

This command sets/gets the value of the positive transition filter of the Questionable Bandwidth Limit Status Register.

Variable

Parameter	*Value*
Description	The value of the positive transition filter
Data type	Long integer type (Long)
Range	0 to 65535
Preset value	Varies depending on the upper limit setting of the number of channels/traces.
Note	The bit 0, and bit 5 to 15 can not be set to 1.

Examples

```
Dim Stat As Long
SCPI.STATus.QUEStionable.BLIMit.PTRansition = 6
Stat = SCPI.STATus.QUEStionable.BLIMit.PTRansition
```

Related objects

SCPI.STATus.QUEStionable.BLIMit.EVENt

SCPI.STATus.QUEStionable.BLIMit.NTRansition

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

:STATus:QUEStionable:BLIMit:PTRansition <numeric>

:STATus:QUEStionable:BLIMit:PTRansition?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":STAT:QUES:BLIM:PTR 16"
20 OUTPUT 717;":STAT:QUES:BLIM:PTR?"
30 ENTER 717;A
```

SCPI.STATus.QUEStionable.CONDition

Object type

Property (**Read Only**)

Syntax

Value = SCPI.STATus.QUEStionable.CONDition

Description

This command reads the value of the Questionable Status Condition Register.

Variable

Parameter	*Value*
Description	Value of the Questionable Status Condition Register
Data type	Long integer type (Long)

Examples

```
Dim Stat As Long
Stat = SCPI.STATus.QUEStionable.CONDition
```

Related objects

SCPI.STATus.QUEStionable.NTRansition

SCPI.STATus.QUEStionable.PTRansition

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

:STATus:QUEStionable:CONDition?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":STAT:QUES:COND?"
20 ENTER 717;A
```

SCPI.STATus.QUEStionable.ENABle

Object type

Property (**Read-Write**)

Syntax

SCPI.STATus.QUEStionable.ENABle = *Value*

Value = SCPI.STATus.QUEStionable.ENABle

Description

This command sets/gets the value of the Questionable Status Enable Register.

Variable

Parameter	*Value*
Description	Value of the Questionable Status Enable Register
Data type	Long integer type (Long)
Range	0 to 65535
Preset value	0
Note	The bit 0 to 7 and bit 12 to 15 can not be set to 1.

Examples

```
Dim Stat As Long
SCPI.STATus.QUEStionable.ENABle = 1024
Stat = SCPI.STATus.QUEStionable.ENABle
```

Related objects

SCPI.IEEE4882.SRE

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

:STATus:QUEStionable:ENABle <numeric>

:STATus:QUEStionable:ENABle?

Query response

```
{numeric}<newline><^END>
```

Example of use

```
10 OUTPUT 717;":STAT:QUES:ENAB 1024"
20 OUTPUT 717;":STAT:QUES:ENAB?"
30 ENTER 717;A
```

SCPI.STATus.QUEStionable.EVENt

Object type

Property (**Read Only**)

Syntax

Value = SCPI.STATus.QUEStionable.EVENt

Description

This command reads the value of the Questionable Status Event Register.

Variable

Parameter	*Value*
Description	Value of the Questionable Status Event Register
Data type	Long integer type (Long)

Examples

```
Dim Stat As Long
Stat = SCPI.STATus.QUEStionable.EVENt
```

Related objects

SCPI.IEEE4882.CLS

SCPI.STATus.QUEStionable.NTRansition

SCPI.STATus.QUEStionable.PTRansition

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

:STATus:QUEStionable[:EVENt]?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":STAT:QUES?"
20 ENTER 717;A
```

SCPI.STATus.QUEStionable.LIMit.CHANnel(*Ch*).CONDition

Object type

Property (**Read Only**)

Syntax

Value = SCPI.STATus.QUEStionable.LIMit.CHANnel(*Ch*).CONDition

Description

This command reads the value of the Questionable Limit Channel Status Condition Register of the specified channel (*Ch*).

Variable

Parameter	*Value*
Description	Value of the Questionable Limit Channel Status Condition Register
Data type	Long integer type (Long)

Examples

```
Dim Stat As Long
Stat = SCPI.STATus.QUEStionable.LIMit.CHANnel(1).CONDition
```

Related objects

SCPI.STATus.QUEStionable.LIMit.CHANnel(Ch).NTRansition

SCPI.STATus.QUEStionable.LIMit.CHANnel(Ch).PTRansition

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

:STATus:QUEStionable:LIMit:CHANnel{[1]-4}:CONDition?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":STAT:QUES:LIM:CHAN1:COND?"
20 ENTER 717;A
```

SCPI.STATus.QUEStionable.LIMit.CHANnel(*Ch*).ENABle

Object type

Property (**Read-Write**)

Syntax

SCPI.STATus.QUEStionable.LIMit.CHANnel(*Ch*).ENABle = *Value*

Value = SCPI.STATus.QUEStionable.LIMit.CHANnel(*Ch*).ENABle

Description

This command sets/gets the value of the Questionable Limit Channel Status Enable Register of the specified channel (*Ch*).

Variable

Parameter	*Value*
Description	Value of the Questionable Limit Channel Status Enable Register
Data type	Long integer type (Long)
Range	0 to 65535
Preset value	Varies depending on the upper limit setting of the channel/trace number.
Note	The bit 0, and bit 5 to 15 can not be set to 1.

Examples

```
Dim Stat As Long
SCPI.STATus.QUEStionable.LIMit.CHANnel(1).ENABle = 16
Stat = SCPI.STATus.QUEStionable.LIMit.CHANnel(1).ENABle
```

Related objects

SCPI.STATus.QUEStionable.LIMit.ENABle

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

:STATus:QUEStionable:LIMit:CHANnel{[1]-4}:ENABle <numeric>

:STATus:QUEStionable:LIMit:CHANnel{[1]-4}:ENABle?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":STAT:QUES:LIM:CHAN1:ENAB 16"
20 OUTPUT 717;":STAT:QUES:LIM:CHAN1:ENAB?"
30 ENTER 717;A
```

SCPI.STATus.QUEStionable.LIMit.CHANnel(*Ch*).EVENt

Object type

Property (**Read Only**)

Syntax

Value = SCPI.STATus.QUEStionable.LIMit.CHANnel(*Ch*).EVENt

Description

This command reads the value of the Questionable Limit Channel Status Event Register of the specified channel (*Ch*).

Variable

Parameter	*Value*
Description	Value of the Questionable Limit Channel Status Event Register of the specified channel
Data type	Long integer type (Long)

Examples

```
Dim Stat As Long
Stat = SCPI.STATus.QUEStionable.LIMit.CHANnel(1).EVENt
```

Related objects

SCPI.IEEE4882.CLS

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

:STATus:QUEStionable:LIMit:CHANnel{[1]-4}[:EVENt]?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":STAT:QUES:LIM:CHAN1?"
20 ENTER 717;A
```

SCPI.STATus.QUEStionable.LIMit.CHANnel(*Ch*).NTRansition

Object type

Property (**Read-Write**)

Syntax

SCPI.STATus.QUEStionable.LIMit.CHANnel(*Ch*).NTRansition = *Value*

Value = SCPI.STATus.QUEStionable.LIMit.CHANnel(*Ch*).NTRansition

Description

This command sets/gets the value of the negative transition filter of the Questionable Limit Channel Status Register of the specified channel (*Ch*).

Variable

Parameter	*Value*
Description	Value of the negative transition filter
Data type	Long integer type (Long)
Range	0 to 65535
Preset value	0
Note	The bit 0, and bit 5 to 15 can not be set to 1.

Examples

```
Dim Stat As Long
SCPI.STATus.QUEStionable.LIMit.CHANnel(1).NTRansition = 16
Stat = SCPI.STATus.QUEStionable.LIMit.CHANnel(1).NTRansition
```

Related objects

SCPI.STATus.QUEStionable.LIMit.CHANnel(Ch).EVENt

SCPI.STATus.QUEStionable.LIMit.CHANnel(Ch).PTRansition

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

:STATus:QUEStionable:LIMit:CHANnel{[1]-4}:NTRansition <numeric>

:STATus:QUEStionable:LIMit:CHANnel{[1]-4}:NTRansition?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":STAT:QUES:LIM:CHAN1:NTR 16"
20 OUTPUT 717;":STAT:QUES:LIM:CHAN1:NTR?"
30 ENTER 717;A
```

SCPI.STATus.QUEStionable.LIMit.CHANnel(*Ch*).PTRansition

Object type

Property (**Read-Write**)

Syntax

SCPI.STATus.QUEStionable.LIMit.CHANnel(*Ch*).PTRansition = *Value*

Value = SCPI.STATus.QUEStionable.LIMit.CHANnel(*Ch*).PTRansition

Description

This command sets/gets the value of the positive transition filter of the Questionable Limit Channel Status Register of the specified channel (*Ch*).

Variable

Parameter	*Value*
Description	Value of the positive transition filter
Data type	Long integer type (Long)
Range	0 to 65535
Preset value	Varies depending on the upper limit setting of the channel/trace number.
Note	The bit 0, and bit 5 to 15 can not be set to 1.

Examples

```
Dim Stat As Long
SCPI.STATus.QUEStionable.LIMit.CHANnel(1).PTRansition = 0
Stat = SCPI.STATus.QUEStionable.LIMit.CHANnel(1).PTRansition
```

Related objects

SCPI.STATus.QUEStionable.LIMit.CHANnel(Ch).EVENt

SCPI.STATus.QUEStionable.LIMit.CHANnel(Ch).NTRansition

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

:STATus:QUEStionable:LIMit:CHANnel{[1]-4}:PTRansition <numeric>

:STATus:QUEStionable:LIMit:CHANnel{[1]-4}:PTRansition?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":STAT:QUES:LIM:CHAN1:PTR 16"
20 OUTPUT 717;":STAT:QUES:LIM:CHAN1:PTR?"
30 ENTER 717;A
```

SCPI.STATus.QUEStionable.LIMit.CONDition

Object type

Property (**Read Only**)

Syntax

Value = SCPI.STATus.QUEStionable.LIMit.CONDition

Description

This command reads the value of the Questionable Limit Status Condition Register.

Variable

Parameter	*Value*
Description	Value of the Questionable Limit Status Condition Register
Data type	Long integer type (Long)

Examples

```
Dim Stat As Long
Stat = SCPI.STATus.QUEStionable.LIMit.CONDition
```

Related objects

SCPI.STATus.QUEStionable.LIMit.NTRansition

SCPI.STATus.QUEStionable.LIMit.PTRansition

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

:STATus:QUEStionable:LIMit:CONDition?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":STAT:QUES:LIM:COND?"
20 ENTER 717;A
```

SCPI.STATus.QUEStionable.LIMit.ENABle

Object type

Property (**Read-Write**)

Syntax

SCPI.STATus.QUEStionable.LIMit.ENABle = *Value*

Value = SCPI.STATus.QUEStionable.LIMit.ENABle

Description

This command sets/gets the value of the Questionable Limit Status Enable Register.

Variable

Parameter	*Value*
Description	Value of the Questionable Limit Status Enable Register
Data type	Long integer type (Long)
Range	0 to 65535
Preset value	Varies depending on the upper limit setting for the channel/trace number.
Note	The bit 0, and bit 5 to 15 can not be set to 1.

Examples

```
Dim Stat As Long
SCPI.STATus.QUEStionable.LIMit.ENABle = 16
Stat = SCPI.STATus.QUEStionable.LIMit.ENABle
```

Related objects

SCPI.STATus.QUEStionable.ENABle

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

:STATus:QUEStionable:LIMit:ENABle <numeric>

:STATus:QUEStionable:LIMit:ENABle?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":STAT:QUES:LIM:ENAB 16"
20 OUTPUT 717;":STAT:QUES:LIM:ENAB?"
30 ENTER 717;A
```

SCPI.STATus.QUEStionable.LIMit.EVENt

Object type

Property (**Read Only**)

Syntax

Value = SCPI.STATus.QUEStionable.LIMit.EVENt

Description

This command reads the value of the Questionable Limit Status Event Register.

Variable

Parameter	*Value*
Description	Value of the Questionable Limit Status Event Register
Data type	Long integer type (Long)

Examples

```
Dim Stat As Long
Stat = SCPI.STATus.QUEStionable.LIMit.EVENt
```

Related objects

SCPI.IEEE4882.CLS

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

:STATus:QUEStionable:LIMit[:EVENt]?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":STAT:QUES:LIM?"
20 ENTER 717;A
```

SCPI.STATus.QUEStionable.LIMit.NTRansition

Object type

Property (**Read-Write**)

Syntax

SCPI.STATus.QUEStionable.LIMit.NTRansition = *Value*

Value = SCPI.STATus.QUEStionable.LIMit.NTRansition

Description

This command sets/gets the value of the negative transition filter of the Questionable Limit Status Register.

Variable

Parameter	*Value*
Description	Value of the negative transition filter
Data type	Long integer type (Long)
Range	0 to 65535
Preset value	0
Note	The bit 0, and bit 5 to 15 can not be set to 1.

Examples

```
Dim Stat As Long
SCPI.STATus.QUEStionable.LIMit.NTRansition = 6
Stat = SCPI.STATus.QUEStionable.LIMit.NTRansition
```

Related objects

SCPI.STATus.QUEStionable.LIMit.EVENt

SCPI.STATus.QUEStionable.LIMit.PTRansition

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

:STATus:QUEStionable:LIMit:NTRansition <numeric>

:STATus:QUEStionable:LIMit:NTRansition?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":STAT:QUES:LIM:NTR 16"
20 OUTPUT 717;":STAT:QUES:LIM:NTR?"
30 ENTER 717;A
```

SCPI.STATus.QUEStionable.LIMit.PTRansition

Object type

Property (**Read-Write**)

Syntax

SCPI.STATus.QUEStionable.LIMit.PTRansition = *Value*

Value = SCPI.STATus.QUEStionable.LIMit.PTRansition

Description

This command sets/gets the value of positive transition filter of the Questionable Limit Status Register.

Variable

Parameter	*Value*
Description	Value of the positive transition filter
Data type	Long integer type (Long)
Range	0 to 65535
Preset value	Varies depending on the upper limit setting of the channel/trace number.
Note	The bit 0, and bit 5 to 15 can not be set to 1.

Examples

```
Dim Stat As Long
SCPI.STATus.QUEStionable.LIMit.PTRansition = 6
Stat = SCPI.STATus.QUEStionable.LIMit.PTRansition
```

Related objects

SCPI.STATus.QUEStionable.LIMit.EVENt

SCPI.STATus.QUEStionable.LIMit.NTRansition

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

:STATus:QUEStionable:LIMit:PTRansition <numeric>

:STATus:QUEStionable:LIMit:PTRansition?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":STAT:QUES:LIM:PTR 16"
20 OUTPUT 717;":STAT:QUES:LIM:PTR?"
30 ENTER 717;A
```

SCPI.STATus.QUEStionable.NTRansition

Object type

Property (**Read-Write**)

Syntax

SCPI.STATus.QUEStionable.NTRansition = *Value*

Value = SCPI.STATus.QUEStionable.NTRansition

Description

This command sets/gets the value of negative transition filter of the Questionable Status Register.

Variable

Parameter	*Value*
Description	Value of the negative transition filter
Data type	Long integer type (Long)
Range	0 to 65535
Preset value	0
Note	The bit (0 to 7) and bit (11 to 15) can not be set to 1.

Examples

```
Dim Stat As Long
SCPI.STATus.QUEStionable.NTRansition = 512
Stat = SCPI.STATus.QUEStionable.NTRansition
```

Related objects

SCPI.STATus.QUEStionable.EVENt

SCPI.STATus.QUEStionable.PTRansition

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

:STATus:QUEStionable:NTRansition <numeric>

:STATus:QUEStionable:NTRansition?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":STAT:QUES:NTR 512"
20 OUTPUT 717;":STAT:QUES:NTR?"
30 ENTER 717;A
```

SCPI.STATus.QUEStionable.PTRansition

Object type

Property (**Read-Write**)

Syntax

SCPI.STATus.QUEStionable.PTRansition = *Value*

Value = SCPI.STATus.QUEStionable.PTRansition

Description

This command sets/gets the value of the positive transition filter of the Questionable Status Register.

Variable

Parameter	*Value*
Description	Value of the positive transition filter
Data type	Long integer type (Long)
Range	0 to 65535
Preset value	1024
Note	The bit 0 to 7 and bit 11 to 15 can not be set to 1.

Examples

```
Dim Stat As Long
SCPI.STATus.QUEStionable.PTRansition = 512
Stat = SCPI.STATus.QUEStionable.PTRansition
```

Related objects

SCPI.STATus.QUEStionable.EVENt

SCPI.STATus.QUEStionable.NTRansition

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

:STATus:QUEStionable:PTRansition <numeric>

:STATus:QUEStionable:PTRansition?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":STAT:QUES:PTR 512"
20 OUTPUT 717;":STAT:QUES:PTR?"
30 ENTER 717;A
```

SCPI.STATus.QUEStionable.RLIMit.CHANnel(*Ch*).CONDition

Object type

Property (**Read Only**)

Syntax

Value = SCPI.STATus.QUEStionable.RLIMit.CHANnel(*Ch*).CONDition

Description

This command reads the value of the Questionable Ripple Limit Channel Status Condition Register for the specified channel.

Variable

Parameter	*Value*
Description	The value of the Questionable Ripple Limit Channel Status Condition Register.
Data type	Long integer type (Long)

For information on the variable (*Ch*), see Ch.

Examples

```
Dim Stat As Long
Stat = SCPI.STATus.QUEStionable.RLIMit.CHANnel(1).CONDition
```

Related objects

SCPI.STATus.QUEStionable.RLIMit.CHANnel(Ch).NTRansition

SCPI.STATus.QUEStionable.RLIMit.CHANnel(Ch).PTRansition

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

:STATus:QUEStionable:RLIMit:CHANnel{[1]-4}:CONDition?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":STAT:QUES:RLIM:CHAN1:COND?"
20 ENTER 717;A
```

SCPI.STATus.QUEStionable.RLIMit.CHANnel(*Ch*).ENABle

Object type

Property (**Read-Write**)

Syntax

SCPI.STATus.QUEStionable.RLIMit.CHANnel(*Ch*).ENABle = *Value*

Value = SCPI.STATus.QUEStionable.RLIMit.CHANnel(*Ch*).ENABle

Description

This command sets/gets the value of the Questionable Ripple Limit Channel Status Enable Register for the specified channel.

Variable

Parameter	*Value*
Description	The value of the Questionable Ripple Limit Channel Status Enable Register.
Data type	Long integer type (Long)
Range	0 to 65535
Preset value	Varies depending on the upper limit setting of the number of channels/traces.
Note	The bit 0, and bit 5 to 15 can not be set to 1.

For information on the variable (*Ch*), see Ch.

Examples

```
Dim Stat As Long
SCPI.STATus.QUEStionable.RLIMit.CHANnel(1).ENABle = 16
Stat = SCPI.STATus.QUEStionable.RLIMit.CHANnel(1).ENABle
```

Related objects

SCPI.STATus.QUEStionable.RLIMit.ENABle

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

:STATus:QUEStionable:RLIMit:CHANnel{[1]-4}:ENABle <numeric>

:STATus:QUEStionable:RLIMit:CHANnel{[1]-4}:ENABle?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":STAT:QUES:RLIM:CHAN1:ENAB 16"
20 OUTPUT 717;":STAT:QUES:RLIM:CHAN1:ENAB?"
30 ENTER 717;A
```

SCPI.STATus.QUEStionable.RLIMit.CHANnel(*Ch*).EVENt

Object type

Property (**Read Only**)

Syntax

Value = SCPI.STATus.QUEStionable.RLIMit.CHANnel(*Ch*).EVENt

Description

This command reads the value of the Questionable Ripple Limit Channel Status Event Register for the specified channel.

Variable

Parameter	*Value*
Description	The value of the Questionable Ripple Limit Channel Status Event Register.
Data type	Long integer type (Long)

For information on the variable (*Ch*), see Ch.

Examples

```
Dim Stat As Long
Stat = SCPI.STATus.QUEStionable.RLIMit.CHANnel(1).EVENt
```

Related objects

SCPI.IEEE4882.CLS

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

:STATus:QUEStionable:RLIMit:CHANnel{[1]-4}[:EVENt]?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":STAT:QUES:RLIM:CHAN1?"
20 ENTER 717;A
```

SCPI.STATus.QUEStionable.RLIMit.CHANnel(*Ch*).NTRansition

Object type

Property (**Read-Write**)

Syntax

SCPI.STATus.QUEStionable.RLIMit.CHANnel(*Ch*).NTRansition = *Value*

Value = SCPI.STATus.QUEStionable.RLIMit.CHANnel(*Ch*).NTRansition

Description

This command sets/gets the value of the negative transition filter of the Questionable Ripple Limit Channel Status Register for the specified channel.

Variable

Parameter	*Value*
Description	The value of the negative transition filter
Data type	Long integer type (Long)
Range	0 to 65535
Preset value	0
Note	The bit 0, and bit 5 to 15 can not be set to 1.

For information on the variable (*Ch*), see Ch.

Examples

```
Dim Stat As Long
SCPI.STATus.QUEStionable.RLIMit.CHANnel(1).NTRansition = 16
Stat = SCPI.STATus.QUEStionable.RLIMit.CHANnel(1).NTRansition
```

Related objects

SCPI.STATus.QUEStionable.RLIMit.CHANnel(Ch).EVENt

SCPI.STATus.QUEStionable.RLIMit.CHANnel(Ch).PTRansition

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

:STATus:QUEStionable:RLIMit:CHANnel{[1]-4}:NTRansition <numeric>

:STATus:QUEStionable:RLIMit:CHANnel{[1]-4}:NTRansition?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":STAT:QUES:RLIM:CHAN1:NTR 16"
20 OUTPUT 717;":STAT:QUES:RLIM:CHAN1:NTR?"
30 ENTER 717;A
```

SCPI.STATus.QUEStionable.RLIMit.CHANnel(*Ch*).PTRansition

Object type

Property (**Read-Write**)

Syntax

SCPI.STATus.QUEStionable.RLIMit.CHANnel(*Ch*).PTRansition = *Value*

Value = SCPI.STATus.QUEStionable.RLIMit.CHANnel(*Ch*).PTRansition

Description

This command sets/gets the value of the positive transition filter of the Questionable Ripple Limit Channel Status Register for the specified channel.

Variable

Parameter	*Value*
Description	The value of the positive transition filter
Data type	Long integer type (Long)
Range	0 to 65535
Preset value	Varies depending on the upper limit setting of the number of channels/traces.
Note	The bit 0, and bit 5 to 15 can not be set to 1.

For information on the variable (*Ch*), see Ch.

Examples

```
Dim Stat As Long
SCPI.STATus.QUEStionable.RLIMit.CHANnel(1).PTRansition = 0
Stat = SCPI.STATus.QUEStionable.RLIMit.CHANnel(1).PTRansition
```

Related objects

SCPI.STATus.QUEStionable.RLIMit.CHANnel(Ch).EVENt

SCPI.STATus.QUEStionable.RLIMit.CHANnel(Ch).NTRansition

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

:STATus:QUEStionable:RLIMit:CHANnel{[1]-4}:PTRansition <numeric>

:STATus:QUEStionable:RLIMit:CHANnel{[1]-4}:PTRansition?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":STAT:QUES:RLIM:CHAN1:PTR 16"
20 OUTPUT 717;":STAT:QUES:RLIM:CHAN1:PTR?"
30 ENTER 717;A
```

SCPI.STATus.QUEStionable.RLIMit.CONDition

Object type

Property (**Read Only**)

Syntax

Value = SCPI.STATus.QUEStionable.RLIMit.CONDition

Description

This command reads the value of the Questionable Ripple Limit Status Condition Register.

Variable

Parameter	*Value*
Description	The value of the Questionable Ripple Limit Status Condition Register.
Data type	Long integer type (Long)

Examples

```
Dim Stat As Long
Stat = SCPI.STATus.QUEStionable.RLIMit.CONDition
```

Related objects

SCPI.STATus.QUEStionable.RLIMit.NTRansition

SCPI.STATus.QUEStionable.RLIMit.PTRansition

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

:STATus:QUEStionable:RLIMit:CONDition?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":STAT:QUES:RLIM:COND?"
20 ENTER 717;A
```

SCPI.STATus.QUEStionable.RLIMit.ENABle

Object type

Property (**Read-Write**)

Syntax

SCPI.STATus.QUEStionable.RLIMit.ENABle = *Value*

Value = SCPI.STATus.QUEStionable.RLIMit.ENABle

Description

This command sets/gets the value of the Questionable Ripple Limit Status Enable Register.

Variable

Parameter	*Value*
Description	The value of the Questionable Ripple Limit Status Enable Register.
Data type	Long integer type (Long)
Range	0 to 65535
Preset value	Varies depending on the upper limit setting of the number of channels/traces.
Note	The bit 0, and bit 5 to 15 can not be set to 1.

Examples

```
Dim Stat As Long
SCPI.STATus.QUEStionable.RLIMit.ENABle = 16
Stat = SCPI.STATus.QUEStionable.RLIMit.ENABle
```

Related objects

SCPI.STATus.QUEStionable.ENABle

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

:STATus:QUEStionable:RLIMit:ENABle <numeric>

:STATus:QUEStionable:RLIMit:ENABle?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":STAT:QUES:RLIM:ENAB 16"
20 OUTPUT 717;":STAT:QUES:RLIM:ENAB?"
30 ENTER 717;A
```

SCPI.STATus.QUEStionable.RLIMit.EVENt

Object type

Property (**Read Only**)

Syntax

Value = SCPI.STATus.QUEStionable.RLIMit.EVENt

Description

This command reads the value of the Questionable Ripple Limit Status Event Register.

Variable

Parameter	*Value*
Description	The value of the Questionable Ripple Limit Status Event Register.
Data type	Long integer type (Long)

Examples

```
Dim Stat As Long
Stat =SCPI.STATus.QUEStionable.RLIMit.EVENt
```

Related objects

SCPI.IEEE4882.CLS

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

:STATus:QUEStionable:RLIMit[:EVENt]?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":STAT:QUES:RLIM?"
20 ENTER 717;A
```

SCPI.STATus.QUEStionable.RLIMit.NTRansition

Object type

Property (**Read-Write**)

Syntax

SCPI.STATus.QUEStionable.RLIMit.NTRansition = *Value*

Value = SCPI.STATus.QUEStionable.RLIMit.NTRansition

Description

This command sets/gets the value of the negative transition filter of the Questionable Ripple Limit Status Register.

Variable

Parameter	*Value*
Description	The value of the negative transition filter
Data type	Long integer type (Long)
Range	0 to 65535
Preset value	0
Note	The bit 0, and bit 5 to 15 can not be set to 1.

Examples

```
Dim Stat As Long
SCPI.STATus.QUEStionable.RLIMit.NTRansition = 6
Stat = SCPI.STATus.QUEStionable.RLIMit.NTRansition
```

Related objects

SCPI.STATus.QUEStionable.RLIMit.EVENt

SCPI.STATus.QUEStionable.RLIMit.PTRansition

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

:STATus:QUEStionable:RLIMit:NTRansition <numeric>

:STATus:QUEStionable:RLIMit:NTRansition?

Query response

<numeric> is written as:

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":STAT:QUES:RLIM:NTR 16"
20 OUTPUT 717;":STAT:QUES:RLIM:NTR?"
30 ENTER 717;A
```

SCPI.STATus.QUEStionable.RLIMit.PTRansition

Object type

Property (**Read-Write**)

Syntax

SCPI.STATus.QUEStionable.RLIMit.PTRansition = *Value*

Value = SCPI.STATus.QUEStionable.RLIMit.PTRansition

Description

This command sets/gets the value of the positive transition filter of the Questionable Ripple Limit Status Register.

Variable

Parameter	*Value*
Description	The value of the positive transition filter
Data type	Long integer type (Long)
Range	0 to 65535
Preset value	Varies depending on the upper limit setting of the number of channels/traces.
Note	The bit 0, and bit 5 to 15 can not be set to 1.

Examples

```
Dim Stat As Long
SCPI.STATus.QUEStionable.RLIMit.PTRansition = 6
Stat = SCPI.STATus.QUEStionable.RLIMit.PTRansition
```

Related objects

SCPI.STATus.QUEStionable.RLIMit.EVENt

SCPI.STATus.QUEStionable.RLIMit.NTRansition

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

:STATus:QUEStionable:RLIMit:PTRansition <numeric>

:STATus:QUEStionable:RLIMit:PTRansition?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":STAT:QUES:RLIM:PTR 16"
20 OUTPUT 717;":STAT:QUES:RLIM:PTR?"
30 ENTER 717;A
```

SYSTEM

SCPI.SYSTem.BACKlight

Object type

Property (**Read-Write**)

Syntax

SCPI.SYSTem.BACKlight = *Status*

Status = SCPI.SYSTem.BACKlight

Description

This command turns ON/OFF or return the status of the backlight of the LCD display.

NOTE When the backlight is OFF, you cannot read the information on the display.

Variable

Parameter	*Status*
Description	ON/OFF of the backlight
Data type	Boolean type (Boolean)
Range	Select from either of the following: • True or ON: Turns ON the backlight. • False or OFF: Turns OFF the backlight.
Preset value	True or ON

Examples

```
Dim BckLght As Boolean
SCPI.SYSTem.BACKlight = False
BckLght = SCPI.SYSTem.BACKlight
```

Equivalent key

System > **Backlight**

NOTE To turn the backlight ON, press any key on the front panel.

Equivalent SCPI command

Syntax

:SYSTem:BACKlight {ON|OFF|1|0}

:SYSTem:BACKlight?

Query response

{1|0}<newline><^END>

Example of use

```
10 OUTPUT 717;":SYST:BACK OFF"
20 OUTPUT 717;":SYST:BACK?"
30 ENTER 717;A
```

SCPI.SYSTem.BEEPer.COMPlete.IMMediate

Object type

Method (**Write Only**)

Syntax

SCPI.SYSTem.BEEPer.COMPlete.IMMediate

Description

This command generates a beep for the notification of the completion of an operation.

Examples

```
SCPI.SYSTem.BEEPer.COMPlete.IMMediate
```

Related objects

SCPI.SYSTem.BEEPer.COMPlete.STATe

SCPI.SYSTem.BEEPer.WARNing.IMMediate

Equivalent key

System > **Misc Setup** > **Beeper** > **Test Beep Complete**

Equivalent SCPI command

Syntax

:SYSTem:BEEPer:COMPlete:IMMediate

Example of use

```
10 OUTPUT 717;":SYST:BEEP:COMP:IMM"
```

SCPI.SYSTem.BEEPer.COMPlete.STATe

Object type

Property (**Read-Write**)

Syntax

SCPI.SYSTem.BEEPer.COMPlete.STATe = *Status*

Status = SCPI.SYSTem.BEEPer.COMPlete.STATe

Description

This command turns ON/OFF or returns the status of the beeper for the notification of the completion of the operation.

Variable

Parameter	*Status*
Description	ON/OFF of the beeper for the notification of the completion of the operation
Data type	Boolean type (Boolean)
Range	Select from either of the following: • True or ON: Turns ON the beeper for the notification of the completion of the operation. • False or OFF: Turns OFF the beeper for the notification of the completion of the operation.
Preset value	True or ON

Examples

```
Dim BeepComp As Boolean
SCPI.SYSTem.BEEPer.COMPlete.STATe = False
BeepComp = SCPI.SYSTem.BEEPer.COMPlete.STATe
```

Related objects

SCPI.SYSTem.BEEPer.COMPlete.IMMediate

SCPI.SYSTem.BEEPer.WARNing.STATe

Equivalent key

System > **Misc Setup** > **Beeper** > **Beep Complete**

Equivalent SCPI command

Syntax

```
:SYSTem:BEEPer:COMPlete:STATe {ON|OFF|1|0}
:SYSTem:BEEPer:COMPlete:STATe?
```

Query response

```
{1|0}<newline><^END>
```

Example of use

```
10 OUTPUT 717;":SYST:BEEP:COMP:STAT OFF"
20 OUTPUT 717;":SYST:BEEP:COMP:STAT?"
30 ENTER 717;A
```

SCPI.SYSTem.BEEPer.WARNing.IMMediate

Object type

Method (**Write Only**)

Syntax

SCPI.SYSTem.BEEPer.WARNing.IMMediate

Description

This command generates a beep for the notification of warning/limit test results.

Examples

```
SCPI.SYSTem.BEEPer.WARNing.IMMediate
```

Related objects

SCPI.SYSTem.BEEPer.WARNing.STATe

SCPI.SYSTem.BEEPer.COMPlete.IMMediate

Equivalent key

System > **Misc Setup** > **Beeper** > **Test Beep Warning**

Equivalent SCPI command

Syntax

:SYSTem:BEEPer:WARNing:IMMediate

Example of use

```
10 OUTPUT 717;":SYST:BEEP:WARN:IMM"
```

SCPI.SYSTem.BEEPer.WARNing.STATe

Object type

Property (**Read-Write**)

Syntax

SCPI.SYSTem.BEEPer.WARNing.STATe = *Status*

Status = SCPI.SYSTem.BEEPer.WARNing.STATe

Description

This command turns ON/OFF or return the status of the beeper for the notification of warning/limit test results.

Variable

Parameter	*Status*
Description	ON/OFF of the beeper for the notification of warning/limit test result
Data type	Boolean type (Boolean)
Range	Select from either of the following: • True or ON: Turns ON the beeper for the notification of warning/limit test result. • False or OFF: Turns OFF the beeper for the notification of warning/limit test result.
Preset value	True or ON

Examples

```
Dim BeepWarn As Boolean
SCPI.SYSTem.BEEPer.WARNing.STATe = False
BeepWarn = SCPI.SYSTem.BEEPer.WARNing.STATe
```

Related objects

SCPI.SYSTem.BEEPer.WARNing.IMMediate

SCPI.SYSTem.BEEPer.COMPlete.STATe

Equivalent key

System > **Misc Setup** > **Beeper** > **Beep Warning**

Equivalent SCPI command

Syntax

:SYSTem:BEEPer:WARNing:STATe {ON|OFF|1|0}

:SYSTem:BEEPer:WARNing:STATe?

Query response

{1|0}<newline><^END>

Example of use

```
10 OUTPUT 717;":SYST:BEEP:WARN:STAT OFF"
20 OUTPUT 717;":SYST:BEEP:WARN:STAT?"
30 ENTER 717;A
```

SCPI.SYSTem.COMMunicate.LAN.CONTrol

Object Type

Property (**Read Only**)

Syntax

Value = SCPI.SYSTem.COMMunicate.LAN.CONTrol

Description

This command gets control port number of socket for LAN connection.

Variable

Parameter	*Value*
Description	control port number of socket
Data Type	Long integer type (Long)

Examples

```
Dim PortNum as Long
PortNum = SCPI.SYSTem.COMMunicate.LAN.CONTrol
```

Equivalent Key

No equivalent key is available on the front panel.

Equivalent SCPI Command

Syntax

:SYSTem:COMMunicate:LAN:CONTrol?

Query Response

{numeric} <newline><^END>

Example of use

```
10 OUTPUT 717;":SYST:COMM:LAN:CONT?"
20 ENTER 717;A
```

SCPI.SYSTem.DATE

Object type

Property (**Read-Write**)

Syntax

SCPI.SYSTem.DATE = *Data*

Data = SCPI.SYSTem.DATE

Description

This command sets/gets the date of the clock built in the E5061B.

Variable

Parameter	*Data*
Description	Indicates 3-element array data (date of the built-in clock). • *Data(0)* Sets year. • *Data(1)* Sets month. • *Data(2)* Sets day. The index of the array starts from 0.
Data type	Variant type (Variant)
Range	• *Data(0)* 1980 to 2099 • *Data(1)* 1 to 12 • *Data(2)* 1 to 31
Resolution	1
Note	If the specified variable 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.

Examples

```
Dim Day As Variant
SCPI.SYSTem.DATE = Array(2001,12,24)
Day = SCPI.SYSTem.DATE
```

```
Dim Day(2) As Variant
Dim Ref As Variant
Day(0) = 2001
Day(1) = 12
Day(2) = 24
SCPI.SYSTem.DATE = Day
Ref = SCPI.SYSTem.DATE
```

Related objects

SCPI.SYSTem.TIME

SCPI.DISPlay.CLOCk

Equivalent key

System > **Misc Setup** > **Clock Setup** > **Set Date and Time**

Equivalent SCPI command

Syntax

```
:SYSTem:DATE <numeric 1>,<numeric 2>,<numeric 3>
:SYSTem:DATE?
```

Query response

```
{numeric 1},{numeric 2},{numeric 3}<newline><^END>
```

Example of use

```
10 OUTPUT 717;":SYST:DATE 2002,1,1"
20 OUTPUT 717;":SYST:DATE?"
30 ENTER 717;A,B,C
```

SCPI.SYSTem.ERRor

Object type

Property (**Read Only**)

Syntax

Err = SCPI.SYSTem.ERRor

Description

This command reads the oldest error from the list of errors stored in the error queue of the E5061B. The read-out error is deleted from the error queue. The size of the error queue is 100.

Executing SCPI.IEEE4882.CLS command clears the errors stored in the error queue.

NOTE This object can not return an error that occurs by the manual operation or the SCPI command used in controlling the E5061B from the external con-troller.

Variable

Parameter	*Err*
Description	Indicates 2-element array data (for error). • *Err(0)* :Error number • *Err(1)* :Error message The index of the array starts from 0.
Data type	Variant type (Variant)
Note	If no error is stored in the error queue, 0 and "No error" are read out as the error number and the error message.

Examples

```
Dim Err As Variant
Err = SCPI.SYSTem.ERRor
```

Related objects

SCPI.IEEE4882.CLS

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

:SYSTem:ERRor?

Query response

{numeric},{string}<newline><^END>

{numeric}:

Error number

{string}:

Error message (a character string with double quotation marks ("))

If no error is stored in the error queue, 0 and "No error" are read out as the error number and the error message, respectively.

Example of use

```
10 OUTPUT 717;":SYST:ERR?"
20 ENTER 717;A,B$
```

SCPI.SYSTem.ISPControl.DCBias.STATe

Object type

Property (**Read-Write**)

Syntax

SCPI.SYSTem.ISPControl.DCBias.STATe = *Status*

Status = SCPI.SYSTem.ISPControl.DCBias.STATe

Description

This command turns ON/OFF or returns the status of the Initial Source Port Control feature for DC Bias (to switch the stimulus output in the trigger hold state to a test port).

Variable

Parameter	*Status*
Description	ON/OFF Initial Source Port Control feature for DC Bias
Data type	Boolean type (Boolean)
Range	Select from either of the following: • True or ON: ON Control feature. • False or OFF: OFF Control feature.
Preset value	True or ON

Examples

```
SCPI.INITiate(1).CONTinuous = False
SCPI.SYSTem.ISPControl.DCBias.STATe = True
```

Related objects

SCPI.SYSTem.ISPControl.STAT

Equivalent key

System > **Service Menu** > **Init Src Ctrl** > **DC Bias**

Equivalent SCPI command

Syntax

:SYSTem:ISPControl:DCBias:STATe {ON|OFF|1|0}

:SYSTem:ISPControl:DCBias:STATe ?

Query response

{1|0}<newline><^END>

Example of use

```
10 OUTPUT 717;":INIT:CONT OFF"
20 OUTPUT 717;":SYST:ISPC:DCBI:STAT ON"
```

SCPI.SYSTem.ISPControl.STATe

Object type

Property (**Read-Write**)

Syntax

SCPI.SYSTem.ISPControl.STATe = *Status*

Status = SCPI.SYSTem.ISPControl.STATe

Description

This command turns ON/OFF or returns the status of the Initial Source Port Control feature for RF signal (to switch the stimulus output in the trigger hold state to a test port).

Variable

Parameter	*Status*
Description	ON/OFF Initial Source Port Control feature for RF signal
Data type	Boolean type (Boolean)
Range	Select from either of the following: • True or ON: ON Control feature. • False or OFF: OFF Control feature.
Preset value	True or ON

Examples

```
SCPI.INITiate(1).CONTinuous = False
SCPI.SYSTem.ISPControl.STATe = True
```

Related objects

SCPI.SYSTem.ISPControl.DCBias.STATe

Equivalent key

System > **Service Menu** > **Init Src Ctrl** > **RF Out**

Equivalent SCPI command

Syntax

:SYSTem:ISPControl[:STATe] {ON|OFF|1|0}

:SYSTem:ISPControl[:STATe] ?

Query response

```
{1|0}<newline><^END>
```

Example of use

```
10 OUTPUT 717;":INIT:CONT OFF"
20 OUTPUT 717;":SYST:ISPC ON"
```

SCPI.SYSTem.KLOCk.KBD

Object type

Property (**Read-Write**)

Syntax

SCPI.SYSTem.KLOCk.KBD = *Status*

Status = SCPI.SYSTem.KLOCk.KBD

Description

This command sets/gets whether to lock the operation of the front panel (key and rotary knob) and keyboard.

Variable

Parameter	*Status*
Description	ON/OFF of lock
Data type	Boolean type (Boolean)
Range	Select from either of the following. • True or ON: Specifies lock. • False or OFF: Specifies unlock.
Preset value	False or OFF

Examples

```
Dim FKLock As Boolean
SCPI.SYSTem.KLOCk.KBD = True
FKLock = SCPI.SYSTem.KLOCk.KBD
```

Related objects

SCPI.SYSTem.KLOCk.MOUSe

Equivalent key

System > **Misc Setup** > **Key Lock** > **Front Panel & Keyboard Lock**

Equivalent SCPI command

Syntax

:SYSTem:KLOCk:KBD {ON|OFF|1|0}

```
:SYSTem:KLOCk:KBD?
```

Query response

```
{1|0}<newline><^END>
```

Example of use

```
10 OUTPUT 717;":SYST:KLOC:KBD ON"
20 OUTPUT 717;":SYST:KLOC:KBD?"
30 ENTER 717;A
```

SCPI.SYSTem.KLOCk.MOUSe

Object type

Property (**Read-Write**)

Syntax

SCPI.SYSTem.KLOCk.MOUSe = *Status*

Status = SCPI.SYSTem.KLOCk.MOUSe

Description

This command sets/gets whether to lock the operation of the mouse and touch screen.

Variable

Parameter	*Status*
Description	ON/OFF of lock
Data type	Boolean type (Boolean)
Range	Select from either of the following. • True or ON: Specifies lock. • False or OFF: Specifies unlock.
Preset value	False or OFF

Examples

```
Dim MTLock As Boolean
SCPI.SYSTem.KLOCk.MOUSe = True
MTLock = SCPI.SYSTem.KLOCk.MOUSe
```

Related objects

SCPI.SYSTem.KLOCk.KBD

Equivalent key

System > **Misc Setup** > **Key Lock** > **Touch Screen & Mouse Lock**

Equivalent SCPI command

Syntax

:SYSTem:KLOCk:MOUSe {ON|OFF|1|0}

:SYSTem:KLOCk:MOUSe?

Query response

{1|0}<newline><^END>

Example of use

```
10 OUTPUT 717;":SYST:KLOC:MOUS ON"
20 OUTPUT 717;":SYST:KLOC:MOUS?"
30 ENTER 717;A
```

SCPI.SYSTem.POFF

Object type

Method (**Write Only**)

Syntax

SCPI.SYSTem.POFF

Description

This command turns OFF the E5061B.

Examples

```
SCPI.SYSTem.POFF
```

Equivalent key

Standby switch

Equivalent SCPI command

Syntax

:SYSTem:POFF

Example of use

```
10 OUTPUT 717;":SYST:POFF"
```

SCPI.SYSTem.PRESet

Object type

Method (**Write Only**)

Syntax

SCPI.SYSTem.PRESet

Description

This command presets the setting state of the E5061B to the original factory setting (Default Conditions). This command is different from SCPI.IEEE4882.RSTas the continuous startup mode (see SCPI.INITiate(Ch).CONTinuous) of channel 1 is set to ON.

Examples

SCPI.SYSTem.PRESet

Related objects

SCPI.IEEE4882.RST

Equivalent key

Preset > **OK**

Equivalent SCPI command

Syntax

:SYSTem:PRESet

Example of use

```
10 OUTPUT 717;":SYST:PRES"
```

SCPI.SYSTem.SECurity.LEVel

Object type

Property (**Read-Write**)

Syntax

SCPI.SYSTem.SECurity.LEVel = *Param*

Param = SCPI.SYSTem.SECurity.LEVel

Description

This command sets/gets the security level.

Variable

Parameter	*Param*
Description	Security level.
Data type	Character string type (String)
Range	Select from either of the following: • "NONE": Specifies OFF to the security level. • "LOW": Specifies LOW level to the security level. • "HIGH": Specifies HIGH level to the security level.
Preset value	"NONE"
Note	When the setting is LOW, it is able to change to NONE or HIGH. But when this setting is HIGH, it is not able to change NONE or LOW. The setting can be turned NONE by executing the preset or recalling when the setting of frequency blank function is HIGH. Even if the setting is LOW and HIGH, the command that reads out the frequency is not influenced.

Examples

```
Dim SecLev As String
SCPI.SYSTem.SECurity.LEVel = "LOW"
SecLev = SCPI.SYSTem.SECurity.LEVel
```

Equivalent key

System > **Service Menu** > **Security Level** > **None|Low|High**

Equivalent SCPI command

Syntax

```
:SYSTem:SECurity:LEVel {NONE|LOW|HIGH}
:SYSTem:SECurity:LEVel?
```

Query response

```
{NONE|LOW|HIGH}<newline><^END>
```

Example of use

```
10 OUTPUT 717;":SYST:SEC:LEV LOW"
20 OUTPUT 717;":SYST:SEC:LEV?"
30 ENTER 717;A$
```

SCPI.SYSTem.SERVice

Object type

Property (**Read Only**)

Syntax

Status = SCPI.SYSTem.SERVice

Description

This command reads whether the E5061B is in the service mode or not.

Variable

Parameter	*Status*
Description	Whether to be in the service mode
Data type	Boolean type (Boolean)
Range	Select from either of the following: • True or ON: In the service mode. • False or OFF : Not in the service mode.
Preset Value	False or OFF

Examples

```
Dim SvMode As Boolean
SvMode = SCPI.SYSTem.SERVice
```

Equivalent key

Displayed on the instrument status bar (at the bottom of the LCD display).

Equivalent SCPI command

Syntax

:SYSTem:SERVice?

Query response

{1|0}<newline><^END>

Example of use

```
10 OUTPUT 717;":SYST:SERV?"
30 ENTER 717;A
```

SCPI.SYSTem.SET

Object Type

Method (**Write Only**)

Syntax

SCPI.SYSTem.SET = *Value*

Description

This command recalls the state of the instrument when the *LRN? query is executed. The contents to be recalled depends on the contents of the block data.

If the block data contains trace state, the trigger source (:SCPI.TRIGger.SEQuence.SOURce) becomes "MANual". The result of *LRN? query contains ":SYSTem:SET " prefix. Hence, the *LRN? simply executes this command.

This command requires instrument settings by binary block data (same as Save/Recall state file contents).

Variable

Parameter	*Value*
Description	This command recalls the state of the instrument when the *LRN? query is executed.
Data Type	Binary (byte)

Related objects

SCPI.IEEE4882.LRN

Equivalent Key

No equivalent key is available on the front panel.

Equivalent SCPI Command

Syntax

:SYSTem:SET

Example of use (VISA-COM)

```
Dim SETData() As Byte, NoofByte As Double

'*** Recall the State data from the file, State01.sta is a state file saved by E5061B
Open "C:\State01.sta" For Binary As #1
```

```
    NoofByte = LOF(1)
    ReDim SETData(NoofByte)
    Get #1, , SETData()
Close

'*** Send the State file data to E5061B
Age506x.WriteIEEEBlock ":SYST:SET ", SETData, True
```

SCPI.SYSTem.TIME

Object type

Property (**Read-Write**)

Syntax

SCPI.SYSTem.TIME = *Data*

Data = SCPI.SYSTem.TIME

Description

This command sets/gets the time of the clock built in the E5061B.

Variable

Parameter	*Data*
Description	Indicates 3-element array data (time of the built-in clock). *Data(0):* Sets hour (24-hour basis) *Data(1)* :Sets minute. *Data(2)* :Sets second. The index of the array starts from 0.
Data type	Variant type (Variant)
Range	*Data(0)* :0 to 23 *Data(1)* :0 to 59 *Data(2)* :0 to 59
Resolution	1
Note	If the specified variable 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.

Examples

```
Dim Time As Variant
SCPI.SYSTem.TIME = Array(21,30,0)
Time = SCPI.SYSTem.TIME
```

```
Dim Time(2) As Variant
Dim Ref As Variant
Time(0) = 21
Time(1) = 30
Time(2) = 0
SCPI.SYSTem.TIME = Time
Ref = SCPI.SYSTem.TIME
```

Related objects

SCPI.SYSTem.DATE

SCPI.DISPlay.CLOCk

Equivalent key

System > **Misc Setup** > **Clock Setup** > **Set Date and Time**

Equivalent SCPI command

Syntax

:SYSTem:TIME <numeric 1>,<numeric 2>,<numeric 3>

:SYSTem:TIME?

Query response

{numeric 1},{numeric 2},{numeric 3}<newline><^END>

Example of use

```
10 OUTPUT 717;":SYST:TIME 17,30,0"
20 OUTPUT 717;":SYST:TIME?"
30 ENTER 717;A,B,C
```

SCPI.SYSTem.UPReset

Object type

Method (**Write Only**)

Syntax

SCPI.SYSTem.UPReset

Description

This command presets the E5061B with the user settings. The command is executed regardless of the operation mode in preset state.

NOTE If you try to specify a file for a preset (D:\UserPreset.sta) that does not exist, a warning message is displayed and SCPI.SYSTem.PRESet is executed.

Examples

SCPI.SYSTem.UPReset

Related objects

SCPI.IEEE4882.RST

SCPI.SYSTem.PRESet

Equivalent key

Preset > **OK**

Equivalent SCPI command

Syntax

:SYSTem:UPReset

Example of use

```
10 OUTPUT 717;":SYST:UPR"
```

TRIGGER

SCPI.TRIGger.OUTPut.DURation

Object Type

Property (**Read-Write**)

Syntax

SCPI.TRIGger.OUTPut.DURation = *Value*

Value = SCPI.TRIGger.OUTPut.DURation

Description

This command sets/gets pulse width of output trigger signal.

Variable

Parameter	*Value*
Description	pulse width
Data Type	Double precision floating point type (Double)
Range	1μ to 1
Preset Value	1μ
Unit	sec
Resolution	1μ

Examples

```
Dim WidTrig as Double
SCPI.TRIGger.OUTPut.DURation = 2E-6
WidTrig = SCPI.TRIGger.OUTPut.DURation
```

Related Objects

Equivalent Key

Trigger > **Pulse Width**

Equivalent SCPI Command

Syntax

:TRIGger:OUTPut:DURation <numeric>

:TRIGger:OUTPut:DURation?

Query Response

{numeric} <newline><^END>

Example of use

```
10 OUTPUT 717;":TRIG:OUTP:DUR 3E-6"
20 OUTPUT 717;":TRIG:OUTP:DUR?"
30 ENTER 717;A
```

SCPI.TRIGger.OUTPut.POLarity

Object type

Property (**Read-Write**)

Syntax

SCPI.TRIGger.OUTPut.POLarity=*Param*

Param=SCPI.TRIGger.OUTPut.POLarity

Description

This command sets/gets the polarity of the pulse generated from the External Trigger (Output).

Variable

Parameter	*Param*
Description	Sets/Gets polarity of the pulse generated by the output trigger.
Data type	Character string type (String)
Range	Select from either of the following: • "POSitive": Generates a Positive pulse. • "NEGative": Generates a Negative pulse.
Preset value	"POSitive"

Examples

```
Dim TrigPol As String
TrigPol="NEGative"
SCPI.TRIGger.OUTPut.POLarity=TrigPol
```

Related objects

SCPI.TRIGger.OUTPut.POSition

SCPI.TRIGger.OUTPut.STATe

Equivalent key

Trigger > **Polarity**

Equivalent SCPI command

Syntax

:TRIGger:OUTPut:POLarity {POSitive|NEGative}

```
:TRIGger:OUTPut:POLarity?
```

Query response

```
{POS|NEG}<newline><^END>
```

Example of use

```
10 OUTPUT 717;":TRIG:OUTP:POL POS"
20 OUTPUT 717;":TRIG:OUTP:POL?"
30 ENTER 717;A$
```

SCPI.TRIGger.OUTPut.POSition

Object type

Property (**Read-Write**)

Syntax

SCPI.TRIGger.OUTPut.POSition=*Param*

Param=SCPI.TRIGger.OUTPut.POSition

Description

This command sets/gets the position of the External Trigger Output Port.

Variable

Parameter	*Param*
Description	Sets/Gets (after or before measurement point) position of the output trigger.
Data type	Character string type (String)
Range	Select from either of the following: • "AFTer": Generates a Pulse (trigger) after the measurement points. • "BEFore": Generates a Pulse (trigger) before the measurement points.
Preset value	"AFTer"

Examples

```
Dim TrigPos As String
TrigPos="BEFore"
SCPI.TRIGger.OUTPut.POSition=TrigPos
```

Related objects

SCPI.TRIGger.OUTPut.POLarity

SCPI.TRIGger.OUTPut.STATe

Equivalent key

Trigger > **Position**

Equivalent SCPI command

Syntax

:TRIGger:OUTPut:POSition {AFTer|BEFore}

:TRIGger:OUTPut:POSition?

Query response

{AFT|BEF}<newline><^END>

Example of use

```
10 OUTPUT 717;":TRIG:OUTP:POS AFT"
20 OUTPUT 717;":TRIG:OUTP:POS?"
30 ENTER 717;A$
```

SCPI.TRIGger.OUTPut.STATe

Object type

Property (**Read-Write**)

Syntax

SCPI.TRIGger.OUTPut.STATe=*Status*

Status=SCPI.TRIGger.OUTPut.STATe

Description

This command sets/gets the External Trigger (Output) Port state.

Variable

Parameter	*Status*
Description	Sets/Gets external trigger output state.
Data type	Boolean type (Boolean)
Range	Select from either of the following: • True or ON: Turns ON the External Trigger Output • False or OFF : Turns OFF the External Trigger Output
Preset value	False or OFF

Examples

```
Dim TrigStat As boolean
TrigStat = True
SCPI.TRIGger.OUTPut.STATe=TrigStat
```

Related objects

SCPI.TRIGger.OUTPut.POLarity

SCPI.TRIGger.OUTPut.POSition

Equivalent key

Trigger > **Ext Trig Output**

Equivalent SCPI command

Syntax

```
:TRIGger:OUTPut[:STATe] {ON|1|OFF|0}
:TRIGger:OUTPut[:STATe]?
```

Query response

```
{1|0}<newline><^END>
```

Example of use

```
10 OUTPUT 717;":TRIG:OUTP ON"
20 OUTPUT 717;":TRIG:OUTP?"
30 ENTER 717;A$
```

SCPI.TRIGger.SEQuence.AVERage

Type of object

Property (**Read-Write**)

Syntax

SCPI.TRIGger.SEQuence.AVERage = *Status*

Status = SCPI.TRIGger.SEQuence.AVERage

Description

This command turns ON/OFF or gets the status of the averaging trigger function.

NOTE The sweep averaging feature must be set to *ON* when turning ON the averaging trigger feature.

Variable

Parameter	*Status*
Description	ON/OFF status of the averaging trigger
Data type	Boolean type (Boolean)
Range	Select from either of the following. • True or ON: Turns ON the averaging trigger. • False or OFF: Turns OFF the averaging trigger.
Preset value	False or OFF

Example of use

```
Dim Avetrig As Boolean
SCPI.TRIGger.SEQuence.AVERage = True
Avetrig = TRIGger.SEQuence.AVERage
```

Related objects

SCPI.SENSe(Ch).AVERage.STATe

Equivalent key

Ave > **Avg Trigger**

Equivalent SCPI command

Syntax

:TRIGger[:SEQuence]:AVERage {ON|OFF|1|0}

```
:TRIGger[:SEQuence]:AVERage?
```

Query response

```
{1|0}<newline><^END>
```

Example of use

```
10 OUTPUT 717;":TRIG:AVER ON"
20 OUTPUT 717;":TRIG:AVER?"
30 ENTER 717;A
```

SCPI.TRIGger.SEQuence.EXTernal.DELay

Type of object

Property (**Read-Write**)

Syntax

SCPI.TRIGger.SEQuence.EXTernal.DELay = *Value*

Value = SCPI.TRIGger.SEQuence.EXTernal.DELay

Description

This command sets/gets the time that it takes from receiving the trigger to starting the measurement when the trigger source is external.

Variable

Parameter	*Value*
Description	External trigger delay time
Data type	Double precision floating point type (Double)
Range	0 to 1
Preset value	0
Unit	s (second)
Resolution	1E-5

Example of use

```
Dim Extdel As Double
SCPI.TRIGger.SEQuence.EXTernal.DELay = 0.05
Extdel = SCPI.TRIGger.SEQuence.EXTernal.DELay
```

Related objects

SCPI.TRIGger.SEQuence.POINt

Equivalent key

Trigger > **Trigger Delay**

Equivalent SCPI command

Syntax

:TRIGger[:SEQuence]:EXTernal:DELay <numeric>

:TRIGger[:SEQuence]:EXTernal:DELay?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":TRIG:EXT:DEL 0.05"
20 OUTPUT 717;":TRIG:EXT:DEL?"
30 ENTER 717;A
```

SCPI.TRIGger.SEQuence.EXTernal.SLOPe

Object type

Property (**Read-Write**)

Syntax

SCPI.TRIGger.SEQuence.EXTernal.SLOPe=*Param*

Param=SCPI.TRIGger.SEQuence.EXTernal.SLOPe

Description

This command sets/gets the polarity of the External Trigger (Input).

Variable

Parameter	*Param*
Description	Sets/Gets polarity of the external trigger (input).
Data type	Character string type (String)
Range	Select from either of the following: • "POSitive": Sets/Gets Positive polarity. • "NEGative": Sets/Gets Negative polarity.
Preset value	"NEGative"

Examples

```
Dim InputTrigPol As String
InputTrigPol="NEGative"
SCPI.TRIGger.SEQuence.EXTernal.SLOPe=InputTrigPol
```

Related objects

SCPI.TRIGger.SEQuence.EXTernal.DELay

Equivalent key

Trigger > **Ext Trig Input**

Equivalent SCPI command

Syntax

:TRIGger[:SEQuence]:EXTernal:SLOPe {POSitive|NEGative}

:TRIGger[:SEQuence]:EXTernal:SLOPe?

Query response

{POS|NEG}<newline><^END>

Example of use

```
10 OUTPUT 717;":TRIG:EXT:SLOP POS"
20 OUTPUT 717;":TRIG:EXT:SLOP?"
30 ENTER 717;A$
```

SCPI.TRIGger.SEQuence.IMMediate

Object type

Method (**Write Only**)

Syntax

SCPI.TRIGger.SEQuence.IMMediate

Description

This command generates a trigger immediately and executes a measurement, regardless of the setting of the trigger mode.

This command is different from SCPI.TRIGger.SEQuence.SINGle as the execution of the object finishes at the time of a trigger.

NOTE If you execute this object when the trigger system is not in the trigger wait state (trigger event detection state), an error occurs when executed and the object is ignored.

Examples

```
SCPI.TRIGger.SEQuence.SOURce = "bus"
SCPI.INITiate(1).CONTinuous = True
SCPI.TRIGger.SEQuence.IMMediate
```

Related objects

SCPI.TRIGger.SEQuence.SINGle

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

:TRIGger[:SEQuence][:IMMediate]

Example of use

```
10 OUTPUT 717;":TRIG:IMM"
20 OUTPUT 717;"*OPC?"
30 ENTER 717;A
```

SCPI.TRIGger.SEQuence.POINt

Type of object

Property (**Read-Write**)

Syntax

SCPI.TRIGger.SEQuence.POINt = *Status*

Status = SCPI.TRIGger.SEQuence.POINt

Description

This command turns ON/OFF or returns the status of the point trigger feature.

NOTE When the trigger source is set to the internal trigger (Internal), the setting is ignored.

Variable

Parameter	*Status*
Description	Turns ON/OFF the point trigger
Data type	Boolean type (Boolean)
Range	Select from either of the following: True or ON: Turns ON the point trigger. False or OFF: Turns OFF the point trigger.
Preset value	False or OFF

Example of use

```
Dim Ptrig As Boolean
SCPI.TRIGger.SEQuence.POINt = True
Ptrig = TRIGger.SEQuence.POINt
```

Related objects

SCPI.TRIGger.SEQuence.SOURce

Equivalent key

Trigger > **Trigger Event**

Equivalent SCPI command

Syntax

:TRIGger[:SEQuence]:POINt {ON|OFF|1|0}

:TRIGger[:SEQuence]:POINt?

Query response

{1|0}<newline><^END>

Example of use

```
10 OUTPUT 717;":TRIG:POIN ON"
20 OUTPUT 717;":TRIG:POIN?"
30 ENTER 717;A
```

SCPI.TRIGger.SEQuence.SCOPe

Object type

Property (**Read-Write**)

Syntax

SCPI.TRIGger.SEQuence.SCOPe = *Param*

Param = SCPI.TRIGger.SEQuence.SCOPe

Description

This command sets/gets the effective scope of triggering. When this function is enabled with a value of "ACTive", only active channel is triggered. When this function is enabled with a value of "ALL", all channels of the E5061B are triggered.

For example, when this function is set at"ACTive" and SCPI.INITiate(Ch).CONTinuous is ON for all channels, a measurement channel is automatically changed by switching over the active channel.

Variable

Parameter	*Param*
Description	Trigger source
Data type	Character string type (String)
Range	Select from either of the following: • "ALL": Specifies trigger to all channels. • "ACTive": Specifies trigger to active channel.
Preset value	"ALL"

Examples

```
Dim TrigScope As Enum
SCPI.TRIGger.SEQuence.SCOPe = "ACTive"
TrigScope = SCPI.TRIGger.SEQuence.SCOPe
```

Related Objects

SCPI.INITiate(Ch).CONTinuous

Equivalent key

Trigger > **Trigger Scope**

Equivalent SCPI command

Syntax

:TRIGger[:SEQuence]:SCOPe {ALL|ACTive}

:TRIGger[:SEQuence]:SCOPe?

Query response

{ALL|ACT}<newline><^END>

Example of use

```
10 OUTPUT 717;":TRIG:SCOP ACT"
20 OUTPUT 717;":TRIG:SCOP?"
30 ENTER 717;A$
```

SCPI.TRIGger.SEQuence.SINGle

Object type

Method (**Write Only**)

Syntax

SCPI.TRIGger.SEQuence.SINGle

Description

This command generates a trigger immediately and executes a measurement, regardless of the setting of the trigger mode.

This command is different from SCPI.TRIGger.SEQuence.IMMediate as the execution of the object finishes when the measurement (all of the sweep) initiated with this object is complete. In other words, you can wait for the end of the measurement using the SCPI.IEEE4882.OPC object.

If you execute this object when the trigger system is not in the trigger wait state (trigger event detection state), an error occurs when executed and the object is ignored.

For details about the trigger system, see Trigger System.

Examples

```
Dim Dmy As Long
SCPI.TRIGger.SEQuence.SOURce = "bus"
SCPI.INITiate(1).CONTinuous = True
SCPI.TRIGger.SEQuence.SINGle
Dmy = SCPI.IEEE4882.OPC
```

Related objects

SCPI.TRIGger.SEQuence.IMMediate

SCPI.IEEE4882.OPC

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

:TRIGger[:SEQuence]:SINGle

Example of use

```
10 OUTPUT 717;":TRIG:SING"
20 OUTPUT 717;"*OPC?"
30 ENTER 717;A
```

SCPI.TRIGger.SEQuence.SOURce

Object type

Property (**Read-Write**)

Syntax

SCPI.TRIGger.SEQuence.SOURce = *Param*

Param = SCPI.TRIGger.SEQuence.SOURce

Description

This command sets/gets the trigger source from the following 4 types:

Trigger	**Description**
Internal Trigger	Uses the internal trigger to generate continuous triggers automatically.
External Trigger	Generates a trigger when the trigger signal is inputted externally through the External Trigger Input connector or the 24 Bit I/O Port (handler interface).
Manual Trigger	Generates a trigger when the key operation of **Trigger** > **Trigger** is executed from the front panel.
Bus Trigger	Generates a trigger when the SCPI.IEEE4882.TRG object is executed.

NOTE When you change the trigger source during sweep, the sweep is aborted.

Variable

Parameter	*Param*
Description	Trigger source
Data type	Character string type (String)
Range	Select from either of the following: • "INTernal": Specifies internal trigger. • "EXTernal": Specifies external trigger.

	• "MANual": Specifies manual trigger. • "BUS": Specifies bus trigger.
Preset value	"INTernal"

Examples

```
Dim TrigSour As String
SCPI.TRIGger.SEQuence.SOURce = "bus"
TrigSour = SCPI.TRIGger.SEQuence.SOURce
```

Equivalent key

Trigger > **Trigger Source** > **Internal|External|Manual|Bus**

Equivalent SCPI command

Syntax

:TRIGger[:SEQuence]:SOURce {INTernal|EXTernal|MANual|BUS}

:TRIGger[:SEQuence]:SOURce?

Query response

{BUS|EXT|INT|MAN}<newline><^END>

Example of use

```
10 OUTPUT 717;":TRIG:SOUR BUS"
20 OUTPUT 717;":TRIG:SOUR?"
30 ENTER 717;A$
```