**Agilent E5061A/E5062A**
**ENA Series RF Network Analyzers**

# VBA Programmer's Guide

**Fourth Edition**

**FIRMWARE REVISIONS**
This manual applies directly to instruments that have the firmware revision A.03.00.
For additional information about firmware revisions, see Appendix A.

Agilent Technologies

**Manufacturing No. E5061-90033**
**February 2007**

# 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

UNIX is a registered trademark in U.S. and other countries, licensed exclusively through X/Open Company Limited.

Portions ©Copyright 1996, Microsoft Corporation. All rights reserved.

© Copyright 2004, 2005, 2006, 2007 Agilent Technologies

# Manual Printing History

The manual's printing date and part number indicate its current edition. The printing date changes when a new edition is printed (minor corrections and updates that are incorporated at reprint do not cause the date to change). The manual part number changes when extensive technical changes are incorporated.

| | |
|---|---|
| February 2004 | First Edition (part number: E5061-90003) |
| September 2005 | Second Edition (part number: E5061-90013, changes for firmeware version A.02.00) |
| June 2006 | Third Edition (part number: E5061-90023, changes for firmeware version A.02.10) |
| February 2007 | Fourth Edition (part number: E5061-90033, changes for firmeware version A.03.00) |

# Typeface Conventions

| | |
|---|---|
| **Sample (bold)** | Boldface type is used when a term is defined or emphasized. |
| *Sample (Italic)* | Italic type is used for emphasis and for titles of manuals and other publications. |
| **[Sample]** | Indicates the hardkey whose key label is “Sample“. |
| **[Sample] - Item** | Indicates a series of key operations in which you press the **[Sample]** key, make the item called “Item” on the displayed menu blink by using the **[ ↓]** or in other ways, and then press the **[Enter]** key. |

# Sample Program Disk

A VBA sample program disk (Agilent part number: E5061-180x1) is furnished with this manual. The disk contains the sample programs used in this manual.

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.

# Documentation Map

The following manuals are available for the Agilent E5061A/E5062A.

- ***User's Guide*** **(Part Number E5061-900x0, attached to Option ABA)**

  This manual describes most of the basic information needed to use the E5061A/E5062A. It provides a function overview, detailed operation procedure for each function (from preparation for measurement to analysis of measurement results), measurement examples, specifications, and supplemental information. For programming guidance on performing automatic measurement with the E5061A/E5062A, please see the *Programming Manual*.

- ***Installation and Quick Start Guide*** **(Part Number E5061-900x1, attached to Option ABA)**

  This manual describes installation of the instrument after it is delivered and the basic procedures for applications and analysis. Refer to this manual when you use the E5061A/E5062A for the first time.

- ***Programmer's Guide*** **(Part Number E5061-900x2, attached to Option ABA)**

  This manual provides programming information for performing automatic measurement with the E5061A/E5062A. It includes an outline of remote control, procedures for detecting measurement start (trigger) and end (sweep end), application programming examples, a command reference, and related information.

- ***VBA Programmer's Guide*** **(Part Number E5061-900x3, attached to Option ABA)**

  This manual describes programming information for performing automatic measurement with internal controller. It includes an outline of VBA programming, some sample programming examples, a COM object reference, and related information.

- ***Option 100 Fault Location and Structural Return Loss Measurement User's Guide Supplement*** **(Part Number E5061-900x4, attached to Option 100)**

  This manual describes information for using the fault location and structual return loss measurement functions.

**NOTE** The number position shown by "x" in the part numbers above indicates the edition number. This convention is applied to each manual, CD-ROM (for manuals), and sample programs disk issued.

# Contents


**1. Making Effective Use of This Manual**

Contents of This Manual . . . 18
How To Use This Manual . . . 20
Looking Up COM Objects . . . 20
Using Sample Programs . . . 20

**2. Introduction to VBA Programming**

Introduction of the E5061A/E5062A Macro Function . . . 24
An Overview of a Control System Based on the Macro Function . . . 25
Implementing a Control System . . . 25
Control Methods . . . 26
Overview of E5061A/E5062A COM Object . . . 27
About COM Object . . . 27
Using COM Object to Control the E5061A/E5062A . . . 27
Major Control Difference between COM Object and SCPI Command . . . 28

**3. Operation Basics of the E5061A/E5062A's VBA**

Displaying Visual Basic Editor . . . 30
Initial Screen of Visual Basic Editor . . . 30
Closing Visual Basic Editor . . . 32
Switching to the E5061A/E5062A Measurement Screen . . . 32
Making a Preparation Before Coding . . . 33
A Project and Three Types of Module . . . 33
Displaying a Code Window . . . 34
Coding a VBA Program . . . 38
User Interface Elements of a Code Window . . . 38
Creating a Simple VBA Program . . . 39
Auto-complete Feature . . . 41
Saving a VBA program . . . 42
Saving a Project . . . 42
Saving a Module (Exporting) . . . 43
Loading a VBA Program . . . 45
Loading a Project . . . 45
Loading a Module (Importing) . . . 46
Running a VBA Program . . . 48
Running a previous loaded VBA program . . . 48
Loading and executing program in batch process . . . 50
Stopping a VBA Program . . . 51
Stopping with the Dialog Box Appeared . . . 51
Abruptly Terminating the VBA Program . . . 52
Errors and Debugging . . . 53
Types of Error . . . 53
Using a Debug Tool . . . 54
Printing Output Values in the Echo Window . . . 59
Entering Values Output to the Echo Window . . . 59
Opening the Echo Window . . . 59
Clearing Values Output in the Echo Window . . . 59
Using VBA Online Help . . . 60

# Contents


Accessing VBA Online Help . . . 60
Uses Advanced Techniques . . . 62
Accessing a List of E5061A/E5062A COM Objects . . . 62
Using Automatic Library References . . . 63

**4. Controlling the E5061A/E5062A**
Detecting the End of Measurement . . . 66
Using the Status Register . . . 66
Using the SCPI.TRIGger.SEQuence.SINGle Object . . . 69
Reading/Writing Measurement Data . . . 71
Executing a Procedure with a Softkey (User Menu Function) . . . 76
Preparation for Using the User Menu Function . . . 76
How to Use the User Menu Function . . . 77
Simple Example . . . 78

**5. Controlling Peripherals**
Overview . . . 82
Preparation . . . 82
Programming with VISA . . . 83
STEP 1. Starting Up VISA System . . . 84
STEP 2. Connection . . . 84
STEP 3. Communication . . . 85
STEP 4. Disconnection . . . 86

**6. Application Programs**
Basic measurement (measuring a band-pass filter) . . . 88
Overview of the program . . . 88
Description of the program . . . 88
Connecting Hard Disk (Shared Folder) of External PC . . . 96
Using VBA program . . . 96
Description of operation in VBA program . . . 97

**7. COM Object Reference**
COM Object Model . . . 100
Application Objects . . . 100
SCPI Objects . . . 101
COM Object List . . . 102
List by Front Panel Key . . . 102
Notational Rules of COM Objects . . . 113
Object Type . . . 113
Syntax . . . 113
Description . . . 113
Variable . . . 114
Examples . . . 114
Related Objects . . . 114
Equivalent Key . . . 114
Application Objects . . . 115
ECHO . . . 115

NAME . . . 116
Parse . . . 117
Prompt . . . 118
UserMenu.Item(Key_id).Caption . . . 119
UserMenu.Item(Key_id).Enabled . . . 120
UserMenu_OnPress(ByVal Key_id As Long) . . . 121
UserMenu.PRESet . . . 121
UserMenu.Press(Key_id) . . . 122
UserMenu.Show . . . 122
VBAVersion . . . 123
WaitOnSRQ . . . 124
SCPI Objects . . . 125
SCPI.ABORt . . . 125
SCPI.CALCulate(Ch).PARameter.COUNt . . . 126
SCPI.CALCulate(Ch).PARameter(Tr).DEFine . . . 127
SCPI.CALCulate(Ch).PARameter(Tr).SELect . . . 128
SCPI.CALCulate(Ch).SELected.BLIMit.DB . . . 129
SCPI.CALCulate(Ch).SELected.BLIMit.DISPlay.MARKer . . . 130
SCPI.CALCulate(Ch).SELected.BLIMit.DISPlay.VALue . . . 131
SCPI.CALCulate(Ch).SELected.BLIMit.FAIL . . . 132
SCPI.CALCulate(Ch).SELected.BLIMit.MAXimum . . . 133
SCPI.CALCulate(Ch).SELected.BLIMit.MINimum . . . 134
SCPI.CALCulate(Ch).SELected.BLIMit.REPort.DATA . . . 135
SCPI.CALCulate(Ch).SELected.BLIMit.STATe . . . 136
SCPI.CALCulate(Ch).SELected.CONVersion.FUNCtion . . . 137
SCPI.CALCulate(Ch).SELected.CONVersion.STATe . . . 138
SCPI.CALCulate(Ch).SELected.CORRection.EDELay. TIME . . . 139
SCPI.CALCulate(Ch).SELected.CORRection.OFFSet. PHASe . . . 140
SCPI.CALCulate(Ch).SELected.DATA.FDATa . . . 141
SCPI.CALCulate(Ch).SELected.DATA.FMEMory . . . 142
SCPI.CALCulate(Ch).SELected.DATA.SDATa . . . 143
SCPI.CALCulate(Ch).SELected.DATA.SMEMory . . . 144
SCPI.CALCulate(Ch).SELected.FORMat . . . 145
SCPI.CALCulate(Ch).SELected.FUNCtion.DATA . . . 146
SCPI.CALCulate(Ch).SELected.FUNCtion.DOMain.COUPle . . . 147
SCPI.CALCulate(Ch).SELected.FUNCtion.DOMain.STARt . . . 148
SCPI.CALCulate(Ch).SELected.FUNCtion.DOMain.STATe . . . 149
SCPI.CALCulate(Ch).SELected.FUNCtion.DOMain.STOP . . . 150
SCPI.CALCulate(Ch).SELected.FUNCtion.EXECute . . . 151
SCPI.CALCulate(Ch).SELected.FUNCtion.PEXCursion . . . 152
SCPI.CALCulate(Ch).SELected.FUNCtion.POINts . . . 153
SCPI.CALCulate(Ch).SELected.FUNCtion.PPOLarity . . . 154
SCPI.CALCulate(Ch).SELected.FUNCtion.TARGet . . . 155
SCPI.CALCulate(Ch).SELected.FUNCtion.TTRansition . . . 156
SCPI.CALCulate(Ch).SELected.FUNCtion.TYPE . . . 157
SCPI.CALCulate(Ch).SELected.LIMit.DATA . . . 158
SCPI.CALCulate(Ch).SELected.LIMit.DISPlay.CLIP . . . 160
SCPI.CALCulate(Ch).SELected.LIMit.DISPlay.STATe . . . 161
SCPI.CALCulate(Ch).SELected.LIMit.FAIL . . . 162

# Contents

SCPI.CALCulate(Ch).SELected.LIMit.OFFSet.AMPLitude. . . . . 163
SCPI.CALCulate(Ch).SELected.LIMit.OFFSet.MARKer. . . . . 164
SCPI.CALCulate(Ch).SELected.LIMit.OFFSet.STIMulus . . . . . 165
SCPI.CALCulate(Ch).SELected.LIMit.REPort.ALL. . . . . 166
SCPI.CALCulate(Ch).SELected.LIMit.REPort.DATA . . . . . 167
SCPI.CALCulate(Ch).SELected.LIMit.REPort.POINts . . . . . 168
SCPI.CALCulate(Ch).SELected.LIMit.STATe . . . . . 169
SCPI.CALCulate(Ch).SELected.MARKer(Mk).ACTivate . . . . . 170
SCPI.CALCulate(Ch).SELected.MARKer(Mk).BWIDth. DATA . . . . . 171
SCPI.CALCulate(Ch).SELected.MARKer.BWIDth.STATe . . . . . 172
SCPI.CALCulate(Ch).SELected.MARKer(Mk).BWIDth. THReshold . . . . . 173
SCPI.CALCulate(Ch).SELected.MARKer.COUPle . . . . . 174
SCPI.CALCulate(Ch).SELected.MARKer(Mk).DISCrete . . . . . 175
SCPI.CALCulate(Ch).SELected.MARKer.FUNCtion. DOMain.COUPle . . . . . 176
SCPI.CALCulate(Ch).SELected.MARKer.FUNCtion. DOMain.STARt. . . . . 177
SCPI.CALCulate(Ch).SELected.MARKer.FUNCtion. DOMain.STATe . . . . . 178
SCPI.CALCulate(Ch).SELected.MARKer.FUNCtion. DOMain.STOP . . . . . 179
SCPI.CALCulate(Ch).SELected.MARKer(Mk).FUNCtion. EXECute . . . . . 180
SCPI.CALCulate(Ch).SELected.MARKer.FUNCtion.MULTi.PEXCursion. . . . . 181
SCPI.CALCulate(Ch).SELected.MARKer.FUNCtion.MULTi.PPOLarity . . . . . 182
SCPI.CALCulate(Ch).SELected.MARKer.FUNCtion.MULTi.TARGet . . . . . 183
SCPI.CALCulate(Ch).SELected.MARKer.FUNCtion.MULTi.TRACking . . . . . 184
SCPI.CALCulate(Ch).SELected.MARKer.FUNCtion.MULTi.TTRansition. . . . . 185
SCPI.CALCulate(Ch).SELected.MARKer.FUNCtion.MULTi.TYPE. . . . . 186
SCPI.CALCulate(Ch).SELected.MARKer(Mk).FUNCtion. PEXCursion . . . . . 187
SCPI.CALCulate(Ch).SELected.MARKer(Mk).FUNCtion. PPOLarity . . . . . 188
SCPI.CALCulate(Ch).SELected.MARKer(Mk).FUNCtion. TARGet . . . . . 189
SCPI.CALCulate(Ch).SELected.MARKer(Mk).FUNCtion. TRACking. . . . . 190
SCPI.CALCulate(Ch).SELected.MARKer(Mk).FUNCtion. TTRansition . . . . . 191
SCPI.CALCulate(Ch).SELected.MARKer(Mk).FUNCtion. TYPE . . . . . 192
SCPI.CALCulate(Ch).SELected.MARKer.MATH.FLATness.DATA . . . . . 194
SCPI.CALCulate(Ch).SELected.MARKer.MATH.FLATness.STATe . . . . . 195
SCPI.CALCulate(Ch).SELected.MARKer.MATH.FSTatistics.DATA . . . . . 196
SCPI.CALCulate(Ch).SELected.MARKer.MATH.FSTatistics.STATe . . . . . 197
SCPI.CALCulate(Ch).SELected.MARKer.MATH.STATistics.DATA. . . . . 198
SCPI.CALCulate(Ch).SELected.MARKer.MATH.STATistics.STATe . . . . . 199
SCPI.CALCulate(Ch).SELected.MARKer(Mk).NOTCh. DATA . . . . . 200
SCPI.CALCulate(Ch).SELected.MARKer.NOTCh.STATe . . . . . 201
SCPI.CALCulate(Ch).SELected.MARKer(Mk).NOTCh. THReshold . . . . . 202
SCPI.CALCulate(Ch).SELected.MARKer.REFerence. STATe . . . . . 203
SCPI.CALCulate(Ch).SELected.MARKer(Mk).SET . . . . . 204
SCPI.CALCulate(Ch).SELected.MARKer(Mk).STATe. . . . . 205
SCPI.CALCulate(Ch).SELected.MARKer(Mk).X . . . . . 206
SCPI.CALCulate(Ch).SELected.MARKer(Mk).Y . . . . . 207
SCPI.CALCulate(Ch).SELected.MATH.FUNCtion . . . . . 208
SCPI.CALCulate(Ch).SELected.MATH.MEMorize . . . . . 209
SCPI.CALCulate(Ch).SELected.MSTatistics.DATA . . . . . 209
SCPI.CALCulate(Ch).SELected.MSTatistics.STATe . . . . . 210
SCPI.CALCulate(Ch).SELected.RLIMit.DATA . . . . . 211

SCPI.CALCulate(Ch).SELected.RLIMit.DISPlay.LINE . . . 213
SCPI.CALCulate(Ch).SELected.RLIMit.DISPlay.SELect . . . 214
SCPI.CALCulate(Ch).SELected.RLIMit.DISPlay.VALue . . . 215
SCPI.CALCulate(Ch).SELected.RLIMit.FAIL . . . 216
SCPI.CALCulate(Ch).SELected.RLIMit.REPort.DATA . . . 217
SCPI.CALCulate(Ch).SELected.RLIMit.STATe . . . 218
SCPI.CALCulate(Ch).SELected.SMOothing.APERture . . . 219
SCPI.CALCulate(Ch).SELected.SMOothing.STATe . . . 220
SCPI.CONTrol.HANDler.A.DATA . . . 221
SCPI.CONTrol.HANDler.B.DATA . . . 222
SCPI.CONTrol.HANDler.C.DATA . . . 223
SCPI.CONTrol.HANDler.C.MODE . . . 224
SCPI.CONTrol.HANDler.D.DATA . . . 225
SCPI.CONTrol.HANDler.D.MODE . . . 226
SCPI.CONTrol.HANDler.E.DATA . . . 227
SCPI.CONTrol.HANDler.EXTension.INDex.STATe . . . 228
SCPI.CONTrol.HANDler.EXTension.RTRigger.STATe . . . 229
SCPI.CONTrol.HANDler.F.DATA . . . 230
SCPI.CONTrol.HANDler.OUTPut(Num).DATA . . . 231
SCPI.DISPlay.ANNotation.FREQuency.STATe . . . 232
SCPI.DISPlay.CCLear . . . 232
SCPI.DISPlay.CLOCk . . . 233
SCPI.DISPlay.COLor(Dnum).BACK . . . 234
SCPI.DISPlay.COLor(Dnum).GRATicule(Gnum) . . . 235
SCPI.DISPlay.COLor(Dnum).LIMit(Lnum) . . . 236
SCPI.DISPlay.COLor(Dnum).RESet . . . 237
SCPI.DISPlay.COLor(Dnum).TRACe(Tr).DATA . . . 238
SCPI.DISPlay.COLor(Dnum).TRACe(Tr).MEMory . . . 239
SCPI.DISPlay.ECHO.CLEar . . . 240
SCPI.DISPlay.ECHO.DATA . . . 240
SCPI.DISPlay.ENABle . . . 241
SCPI.DISPlay.FSIGn . . . 242
SCPI.DISPlay.IMAGe . . . 243
SCPI.DISPlay.MAXimize . . . 244
SCPI.DISPlay.SKEY.STATe . . . 245
SCPI.DISPlay.SPLit . . . 246
SCPI.DISPlay.TABLe.STATe . . . 248
SCPI.DISPlay.TABLe.TYPE . . . 249
SCPI.DISPlay.UPDate.IMMediate . . . 250
SCPI.DISPlay.WINDow(Ch).ACTivate . . . 250
SCPI.DISPlay.WINDow(Ch).ANNotation.MARKer.ALIGn.STATe . . . 251
SCPI.DISPlay.WINDow(Ch).ANNotation.MARKer.SINGle.STATe . . . 252
SCPI.DISPlay.WINDow(Ch).LABel . . . 253
SCPI.DISPlay.WINDow(Ch).MAXimize . . . 254
SCPI.DISPlay.WINDow(Ch).SPLit . . . 255
SCPI.DISPlay.WINDow(Ch).TITLe.DATA . . . 256
SCPI.DISPlay.WINDow(Ch).TITLe.STATe . . . 257
SCPI.DISPlay.WINDow(Ch).TRACe(Tr).ANNotation. MARKer. POSition.X . . . 258
SCPI.DISPlay.WINDow(Ch).TRACe(Tr).ANNotation. MARKer. POSition.Y . . . 259

# Contents

SCPI.DISPlay.WINDow(Ch).TRACe(Tr).ANNotation. YAXis.MODE . . . . . 260
SCPI.DISPlay.WINDow(Ch).TRACe(Tr).MEMory. STATe . . . . . 261
SCPI.DISPlay.WINDow(Ch).TRACe(Tr).STATe . . . . . 262
SCPI.DISPlay.WINDow(Ch).TRACe(Tr).Y.SCALe.AUTO . . . . . 262
SCPI.DISPlay.WINDow(Ch).TRACe(Tr).Y.SCALe. PDIVision. . . . . 263
SCPI.DISPlay.WINDow(Ch).TRACe(Tr).Y.SCALe.RLEVel . . . . . 264
SCPI.DISPlay.WINDow(Ch).TRACe(Tr).Y.SCALe. RPOSition . . . . . 265
SCPI.DISPlay.WINDow(Ch).TRACe(Tr).Y. TRACk. FREQuency . . . . . 266
SCPI.DISPlay.WINDow(Ch).TRACe(Tr).Y. TRACk. MODE . . . . . 267
SCPI.DISPlay.WINDow(Ch).X.SPACing . . . . . 268
SCPI.DISPlay.WINDow(Ch).Y.SCALe.DIVisions . . . . . 269
SCPI.FORMat.BORDer . . . . . 270
SCPI.FORMat.DATA . . . . . 271
SCPI.HCOPy.ABORt . . . . . 272
SCPI.HCOPy.IMAGe . . . . . 272
SCPI.HCOPy.IMMediate . . . . . 273
SCPI.IEEE4882.CLS . . . . . 274
SCPI.IEEE4882.ESE . . . . . 274
SCPI.IEEE4882.ESR . . . . . 275
SCPI.IEEE4882.IDN . . . . . 275
SCPI.IEEE4882.OPC . . . . . 276
SCPI.IEEE4882.OPT . . . . . 277
SCPI.IEEE4882.RST . . . . . 277
SCPI.IEEE4882.SRE . . . . . 278
SCPI.IEEE4882.STB . . . . . 279
SCPI.IEEE4882.TRG . . . . . 279
SCPI.IEEE4882.WAI . . . . . 279
SCPI.INITiate(Ch).CONTinuous . . . . . 280
SCPI.INITiate(Ch).IMMediate . . . . . 281
SCPI.MMEMory.CATalog(Dir) . . . . . 282
SCPI.MMEMory.COPY . . . . . 283
SCPI.MMEMory.DELete . . . . . 284
SCPI.MMEMory.LOAD.CHANnel.COEFficient . . . . . 285
SCPI.MMEMory.LOAD.CHANnel.STATe . . . . . 286
SCPI.MMEMory.LOAD.LIMit . . . . . 287
SCPI.MMEMory.LOAD.RLIMit . . . . . 288
SCPI.MMEMory.LOAD.SEGMent . . . . . 289
SCPI.MMEMory.LOAD.STATe . . . . . 290
SCPI.MMEMory.MDIRectory . . . . . 291
SCPI.MMEMory.STORe.CHANnel.CLEar . . . . . 292
SCPI.MMEMory.STORe.CHANnel.COEFficient . . . . . 292
SCPI.MMEMory.STORe.CHANnel.STATe . . . . . 293
SCPI.MMEMory.STORe.FDATa . . . . . 294
SCPI.MMEMory.STORe.IMAGe . . . . . 295
SCPI.MMEMory.STORe.LIMit . . . . . 296
SCPI.MMEMory.STORe.RLIMit. . . . . 297
SCPI.MMEMory.STORe.SALL . . . . . 298
SCPI.MMEMory.STORe.SEGMent . . . . . 299
SCPI.MMEMory.STORe.STATe . . . . . 300

SCPI.MMEMory.STORe.STYPe . . . 301
SCPI.OUTPut.STATe . . . 302
SCPI.SENSe(Ch).AVERage.CLEar . . . 303
SCPI.SENSe(Ch).AVERage.COUNt . . . 303
SCPI.SENSe(Ch).AVERage.STATe . . . 304
SCPI.SENSe(Ch).BANDwidth.RESolution . . . 305
SCPI.SENSe(Ch).BWIDth.RESolution . . . 306
SCPI.SENSe(Ch).CORRection.CLEar . . . 307
SCPI.SENSe(Ch).CORRection.COEFficient.DATA . . . 308
SCPI.SENSe(Ch).CORRection.COLLect.ACQuire. ISOLation . . . 310
SCPI.SENSe(Ch).CORRection.COLLect.ACQuire.LOAD . . . 311
SCPI.SENSe(Ch).CORRection.COLLect.ACQuire.OPEN . . . 312
SCPI.SENSe(Ch).CORRection.COLLect.ACQuire. SHORt . . . 312
SCPI.SENSe(Ch).CORRection.COLLect.ACQuire.THRU . . . 313
SCPI.SENSe(Ch).CORRection.COLLect.CKIT.LABel . . . 314
SCPI.SENSe(Ch).CORRection.COLLect.CKIT.ORDer. LOAD(Cpt) . . . 315
SCPI.SENSe(Ch).CORRection.COLLect.CKIT.ORDer. OPEN(Cpt) . . . 316
SCPI.SENSe(Ch).CORRection.COLLect.CKIT.ORDer. SHORt(Cpt) . . . 317
SCPI.SENSe(Ch).CORRection.COLLect.CKIT.ORDer. THRU(Cpt_m,Cpt_n) . . . 318
SCPI.SENSe(Ch).CORRection.COLLect.CKIT.RESet . . . 319
SCPI.SENSe(Ch).CORRection.COLLect.CKIT.SELect . . . 319
SCPI.SENSe(Ch).CORRection.COLLect.CKIT.STAN(Std).ARBitrary . . . 320
SCPI.SENSe(Ch).CORRection.COLLect.CKIT.STAN(Std).C0 . . . 321
SCPI.SENSe(Ch).CORRection.COLLect.CKIT.STAN(Std).C1 . . . 322
SCPI.SENSe(Ch).CORRection.COLLect.CKIT.STAN(Std).C2 . . . 323
SCPI.SENSe(Ch).CORRection.COLLect.CKIT.STAN(Std).C3 . . . 324
SCPI.SENSe(Ch).CORRection.COLLect.CKIT.STAN(Std).DELay . . . 325
SCPI.SENSe(Ch).CORRection.COLLect.CKIT.STAN(Std).L0 . . . 326
SCPI.SENSe(Ch).CORRection.COLLect.CKIT.STAN(Std).L1 . . . 327
SCPI.SENSe(Ch).CORRection.COLLect.CKIT.STAN(Std).L2 . . . 328
SCPI.SENSe(Ch).CORRection.COLLect.CKIT.STAN(Std).L3 . . . 329
SCPI.SENSe(Ch).CORRection.COLLect.CKIT.STAN(Std).LABel . . . 330
SCPI.SENSe(Ch).CORRection.COLLect.CKIT.STAN(Std).LOSS . . . 331
SCPI.SENSe(Ch).CORRection.COLLect.CKIT.STAN(Std).TYPE . . . 332
SCPI.SENSe(Ch).CORRection.COLLect.CKIT.STAN(Std).Z0 . . . 333
SCPI.SENSe(Ch).CORRection.COLLect.ECAL.ERESponse . . . 334
SCPI.SENSe(Ch).CORRection.COLLect.ECAL.ISOLation.STATe . . . 335
SCPI.SENSe.CORRection.COLLect.ECAL.PATH(Cpt) . . . 336
SCPI.SENSe(Ch).CORRection.COLLect.ECAL.SOLT1 . . . 337
SCPI.SENSe(Ch).CORRection.COLLect.ECAL.SOLT2 . . . 338
SCPI.SENSe(Ch).CORRection.COLLect.ECAL.THRU . . . 339
SCPI.SENSe(Ch).CORRection.COLLect.METHod. ERESponse . . . 340
SCPI.SENSe(Ch).CORRection.COLLect.METHod. RESPonse.OPEN . . . 341
SCPI.SENSe(Ch).CORRection.COLLect.METHod. RESPonse.SHORt . . . 341
SCPI.SENSe(Ch).CORRection.COLLect.METHod. RESPonse.THRU . . . 342
SCPI.SENSe(Ch).CORRection.COLLect.METHod. SOLT1 . . . 342
SCPI.SENSe(Ch).CORRection.COLLect.METHod. SOLT2 . . . 343
SCPI.SENSe(Ch).CORRection.COLLect.METHod.TYPE . . . 344
SCPI.SENSe(Ch).CORRection.COLLect.SAVE . . . 345

# Contents


SCPI.SENSe(Ch).CORRection.EXTension.PORT(Pt).TIME . . . . . 346
SCPI.SENSe(Ch).CORRection.EXTension.STATe . . . . . 347
SCPI.SENSe.CORRection.IMPedance.INPut.MAGNitude . . . . . 348
SCPI.SENSe(Ch).CORRection.PROPerty . . . . . 349
SCPI.SENSe(Ch).CORRection.RVELocity.COAX . . . . . 350
SCPI.SENSe(Ch).CORRection.STATe . . . . . 351
SCPI.SENSe(Ch).CORRection.TYPE(Tr) . . . . . 352
SCPI.SENSe(Ch).FREQuency.CENTer . . . . . 353
SCPI.SENSe(Ch).FREQuency.CW . . . . . 354
SCPI.SENSe(Ch).FREQuency.DATA . . . . . 355
SCPI.SENSe(Ch).FREQuency.FIXed . . . . . 356
SCPI.SENSe(Ch).FREQuency.SPAN . . . . . 357
SCPI.SENSe(Ch).FREQuency.STARt . . . . . 358
SCPI.SENSe(Ch).FREQuency.STOP . . . . . 359
SCPI.SENSe(Ch).ROSCillator.SOURce . . . . . 360
SCPI.SENSe(Ch).SEGMent.DATA . . . . . 361
SCPI.SENSe(Ch).SEGMent.SWEep.POINts . . . . . 363
SCPI.SENSe(Ch).SEGMent.SWEep.TIME.DATA . . . . . 363
SCPI.SENSe(Ch).SWEep.DELay . . . . . 364
SCPI.SENSe(Ch).SWEep.POINts . . . . . 365
SCPI.SENSe(Ch).SWEep.TIME.AUTO . . . . . 366
SCPI.SENSe(Ch).SWEep.TIME.DATA . . . . . 367
SCPI.SENSe(Ch).SWEep.TYPE . . . . . 368
SCPI.SERVice.CHANnel.ACTive . . . . . 369
SCPI.SERVice.CHANnel.COUNt . . . . . 369
SCPI.SERVice.CHANnel(Ch).TRACe.ACTive . . . . . 370
SCPI.SERVice.CHANnel.TRACe.COUNt. . . . . 370
SCPI.SERVice.PORT.COUNt. . . . . 371
SCPI.SOURce(Ch).POWer.ATTenuation.DATA . . . . . 372
SCPI.SOURce(Ch).POWer.CENTer . . . . . 373
SCPI.SOURce(Ch).POWer.LEVel.IMMediate. AMPLitude . . . . . 374
SCPI.SOURce(Ch).POWer.LEVel.SLOPe.DATA . . . . . 375
SCPI.SOURce(Ch).POWer.LEVel.SLOPe.STATe . . . . . 376
SCPI.SOURce(Ch).POWer.PORT.COUPle . . . . . 377
SCPI.SOURce(Ch).POWer.PORT(Pt).LEVel.IMMediate. AMPLitude . . . . . 378
SCPI.SOURce(Ch).POWer.SPAN . . . . . 379
SCPI.SOURce(Ch).POWer.STARt . . . . . 380
SCPI.SOURce(Ch).POWer.STOP . . . . . 381
SCPI.STATus.OPERation.CONDition . . . . . 382
SCPI.STATus.OPERation.ENABle . . . . . 382
SCPI.STATus.OPERation.EVENt . . . . . 383
SCPI.STATus.OPERation.NTRansition . . . . . 383
SCPI.STATus.OPERation.PTRansition . . . . . 384
SCPI.STATus.PRESet . . . . . 384
SCPI.STATus.QUEStionable.BLIMit.CHANnel(Ch).CONDition . . . . . 385
SCPI.STATus.QUEStionable.BLIMit.CHANnel(Ch).ENABle . . . . . 386
SCPI.STATus.QUEStionable.BLIMit.CHANnel(Ch).EVENt . . . . . 387
SCPI.STATus.QUEStionable.BLIMit.CHANnel(Ch).NTRansition . . . . . 388
SCPI.STATus.QUEStionable.BLIMit.CHANnel(Ch).PTRansition . . . . . 389

SCPI.STATus.QUEStionable.BLIMit.CONDition . . . 390
SCPI.STATus.QUEStionable.BLIMit.ENABle. . . . 391
SCPI.STATus.QUEStionable.BLIMit.EVENt. . . . 391
SCPI.STATus.QUEStionable.BLIMit.NTRansition . . . 392
SCPI.STATus.QUEStionable.BLIMit.PTRansition. . . . 393
SCPI.STATus.QUEStionable.CONDition . . . 394
SCPI.STATus.QUEStionable.ENABle . . . 394
SCPI.STATus.QUEStionable.EVENt . . . 395
SCPI.STATus.QUEStionable.LIMit.CHANnel(Ch). CONDition . . . 395
SCPI.STATus.QUEStionable.LIMit.CHANnel(Ch).ENABle. . . . 396
SCPI.STATus.QUEStionable.LIMit.CHANnel(Ch).EVENt. . . . 397
SCPI.STATus.QUEStionable.LIMit.CHANnel(Ch). NTRansition . . . 398
SCPI.STATus.QUEStionable.LIMit.CHANnel(Ch). PTRansition . . . 399
SCPI.STATus.QUEStionable.LIMit.CONDition . . . 400
SCPI.STATus.QUEStionable.LIMit.ENABle . . . 401
SCPI.STATus.QUEStionable.LIMit.EVENt . . . 401
SCPI.STATus.QUEStionable.LIMit.NTRansition . . . 402
SCPI.STATus.QUEStionable.LIMit.PTRansition . . . 403
SCPI.STATus.QUEStionable.NTRansition . . . 404
SCPI.STATus.QUEStionable.PTRansition . . . 405
SCPI.STATus.QUEStionable.RLIMit.CHANnel(Ch).CONDition . . . 406
SCPI.STATus.QUEStionable.RLIMit.CHANnel(Ch).ENABle . . . 407
SCPI.STATus.QUEStionable.RLIMit.CHANnel(Ch).EVENt . . . 408
SCPI.STATus.QUEStionable.RLIMit.CHANnel(Ch).NTRansition . . . 409
SCPI.STATus.QUEStionable.RLIMit.CHANnel(Ch).PTRansition . . . 410
SCPI.STATus.QUEStionable.RLIMit.CONDition . . . 411
SCPI.STATus.QUEStionable.RLIMit.ENABle. . . . 412
SCPI.STATus.QUEStionable.RLIMit.EVENt. . . . 412
SCPI.STATus.QUEStionable.RLIMit.NTRansition . . . 413
SCPI.STATus.QUEStionable.RLIMit.PTRansition. . . . 414
SCPI.SYSTem.BACKlight . . . 415
SCPI.SYSTem.BEEPer.COMPlete.IMMediate . . . 416
SCPI.SYSTem.BEEPer.COMPlete.STATe . . . 416
SCPI.SYSTem.BEEPer.WARNing.IMMediate . . . 417
SCPI.SYSTem.BEEPer.WARNing.STATe . . . 417
SCPI.SYSTem.DATE . . . 418
SCPI.SYSTem.ERRor . . . 419
SCPI.SYSTem.KLOCk.KBD . . . 420
SCPI.SYSTem.KLOCk.MOUSe . . . 421
SCPI.SYSTem.POFF . . . 421
SCPI.SYSTem.PRESet . . . 422
SCPI.SYSTem.SECurity.LEVel . . . 423
SCPI.SYSTem.SERVice . . . 424
SCPI.SYSTem.TIME . . . 425
SCPI.SYSTem.UPReset . . . 426
SCPI.TRIGger.SEQuence.IMMediate . . . 427
SCPI.TRIGger.SEQuence.SINGle . . . 428
SCPI.TRIGger.SEQuence.SOURce . . . 429

# Contents


**8. Waveform Analysis Library**

Ripple Analysis Library . . . 432
Flow of Programming Using the Ripple Analysis Library . . . 432
Condition Setting Before Using the Ripple Analysis Library . . . 432
List of the Ripple Analysis Library . . . 433
Simple Use Example . . . 434
Procedure Reference . . . 435
FirstLeftGap(Chan) . . . 435
FirstLeftInterval(Chan) . . . 436
FirstRightGap(Chan) . . . 437
FirstRightInterval(Chan) . . . 438
GapMean(Chan) . . . 439
MaxEnvelopeGap(Chan) . . . 440
MaxGap(Chan) . . . 441
MaxLeftGap(Chan) . . . 442
MaxPeakToPeak(Chan) . . . 443
MaxRightGap(Chan) . . . 444
MaxRipplePoint(Chan,Stim) . . . 445
MaxRippleValue(Chan) . . . 446
Pole(Chan,D,LeftStim,LeftValue,RightStim,RightValue) . . . 447

**9. Complex Operation Library**

Complex operation library . . . 450
Data of the complex type . . . 450
List of procedures . . . 450
Procedure Reference . . . 451
ComplexAbs(x) . . . 451
ComplexAdd(x,y) . . . 451
ComplexArg(x) . . . 451
ComplexConj(x) . . . 452
ComplexCos(x) . . . 452
ComplexCosh(x) . . . 452
ComplexDiv(x,y) . . . 452
ComplexExp(x) . . . 453
ComplexLog(x) . . . 453
ComplexLog10(x) . . . 453
ComplexMul(x,y) . . . 453
ComplexNorm(x) . . . 454
ComplexPolar(x,y) . . . 454
ComplexSet(x,y) . . . 454
ComplexSetArray(x) . . . 455
ComplexSin(x) . . . 455
ComplexSinh(x) . . . 455
ComplexSqrt(x) . . . 456
ComplexSub(x,y) . . . 456
Sample Program . . . 457

**A. Manual Changes**

Manual Changes . . . . . 460
Change 3 . . . . . 461
Change 2 . . . . . 461
Change 1 . . . . . 462

# Contents

# 1 Making Effective Use of This Manual

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

# Contents of This Manual

This is a VBA programming guide with Agilent E5061A/E5062A.

This guide describes programming method mainly aiming at learning how to write a program that controls the E5061A/E5062A using COM objects, focusing on the macro function of the E5061A/E5062A and sample usage with the built-in VBA.

Controlling the E5061A/E5062A using an external controller is not covered by this guide; it is described in *Programmer's Guide*. For remote control using an external controller, see *Programmer's Guide*.

Description in this guide assumes that the reader has learned manual operation of the E5061A/E5062A.Thus, this guide does not describe each feature of the E5061A/E5062A in detail. For detailed information on each feature, see *User's Guide*.

The chapter-by-chapter contents of this manual are as follows.

Chapter 1, “Making Effective Use of This Manual.”

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

Chapter 2, “Introduction to VBA Programming.”

This chapter introduces you to the E5061A/E5062A's VBA macro function, describes how you can implement your system using the VBA macro function, and provides an overview of the COM objects that come with the E5061A/E5062A.

Chapter 3, “Operation Basics of the E5061A/E5062A's VBA.”

This chapter provides descriptive information on basic operations for creating VBA programs within the E5061A/E5062A's VBA environment; topics include launching Visual Basic Editor, creating, saving, and running VBA programs, and so on.

Chapter 4, “Controlling the E5061A/E5062A.”

This chapter describes how to use the E5061A/E5062A's VBA to control the E5061A/E5062A itself.

Chapter 5, “Controlling Peripherals.”

This chapter explains how to control peripherals connected to the E5061A/E5062A with GPIB by using the software (VISA library) installed in the E5061A/E5062A.

Chapter 6, “Application Programs.”

This chapter describes sample programs (VBA programs) based on actual measurement examples.

Chapter 7, “COM Object Reference.”

This chapter describes the COM object model of the Agilent E5061A/E5062A and the COM object reference in alphabetical order. If you want to look up COM objects by corresponding front panel keys, see “COM object list by front panel key.”

Chapter 8, “Waveform Analysis Library.”

This chapter describes how to use the ripple analysis library and the procedures in the ripple analysis library.

Chapter 9, “Complex Operation Library.”

This chapter describes the complex operation library.

Appendix A, “Manual Changes.”

This appendix contains the information required to adapt this manual to versions or configurations of the E5061A/E5062A manufactured earlier than the current printing date of this manual.

# How To Use This Manual

Chapter 3 provides the basic operation of VBA when coding VBA programs, and Chapter 4 provides the description of controlling the E5061A/E5062A and sample program examples that you can use to develop your custom programs. For more information on individual COM object, see Chapter 7, “COM Object Reference.”

## Looking Up COM Objects

Chapter 7, “COM Object Reference.” contains a complete reference of COM objects. You can look up a particular COM object in any of the following ways:

Lookup by Abbreviated COM Object Name

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

Lookup by Front panel key

Table 7-1 on page 102 provides a complete list of COM objects that correspond to the front panel key tree and indicates the page numbers where the COM objects appear in the COM object reference.

## Using Sample Programs

The manual comes with a sample program disk, which contains the source files of the sample programs described in this manual. The disk is DOS-formatted.

### Loading a Sample Program

For the method to load a sample program into the E5061A/E5062A VBA, see Section “Loading a VBA Program” on page 45 in the Chapter 3 “Operation Basics of the E5061A/E5062A's VBA”.

### List of the Sample Programs

Table 1-1 shows the file list contained with the VBA sample program disk. To look up the description of a sample program, see the listings under “Sample program” in the index.

**Table 1-1 List of the sample programs**

| Project | Object names of modules in the project | Module type | Content |
|---|---|---|---|
| apl_bsc.vba | mdlBscMeas | Standard module | Program for the basic measurement of the bandpass filter |
| map_drive.vba | Module1<br>frmMapDrive | Standard module<br>UserForm | Program for connecting a hard disk (a shared folder) of an external PC to the E5061A/E5062A. |
| meas_sing.vba | mdlSingMeas<br>frmSingMeas | Standard module<br>UserForm | Program for detecting the end of the measurement using **SCPI.TRIGger.SEQuence.SINGle** object and **SCPI.IEEE4882.OPC** object. |
| meas_srq.vba | mdlSrqMeas<br>frmSrqMeas | Standard module<br>UserForm | Program for detecting the end of the measurement through the status register |
| meas_user.vba | mdlUserMenu | Standard module | Program for utilizing the user menu function (interrupt processing by the assigned softkey) |
| read_write.vba | mdlReadWrite<br>frmReadWrite | Standard module<br>UserForm | Program for reading / displaying / writing a formatted data array |

**NOTE** The sample program disk also contains two definition file for controlling peripherals with VISA library, named “visa32.bas” and “vpptype.bas.”

# 2 Introduction to VBA Programming

This chapter introduces you to the E5061A/E5062A's VBA macro function, describes how you can implement your system using the VBA macro function, and provides an overview of the COM objects that come with the E5061A/E5062A.

# Introduction of the E5061A/E5062A Macro Function

The E5061A/E5062A has a built-in macro function that allows a single instruction to substitute for multiple instructions. You can have the E5061A/E5062A 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 E5061A/E5062A 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 is decreased some of the VB's features and added characteristic features for each application. The E5061A/E5062A VBA is added features for controlling the E5061A/E5062A. For details of 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 E5061A/E5062A's VBA, see Chapter 3, "Operation Basics of the E5061A/E5062A's VBA," on page 29. 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.

The macro function allows you to control the E5061A/E5062A itself as well as various peripherals. You can do the following:

1. Automate repetitive tasks

   You can use the E5061A/E5062A'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 E5061A/E5062A VBA supports user forms (see "User Form" on page 33) that simplify creating a visual user interface. User forms guide users through common tasks such as performing measurement and entering data, without requiring familiarity with the E5061A/E5062A, thus minimizing the possibility of human error.

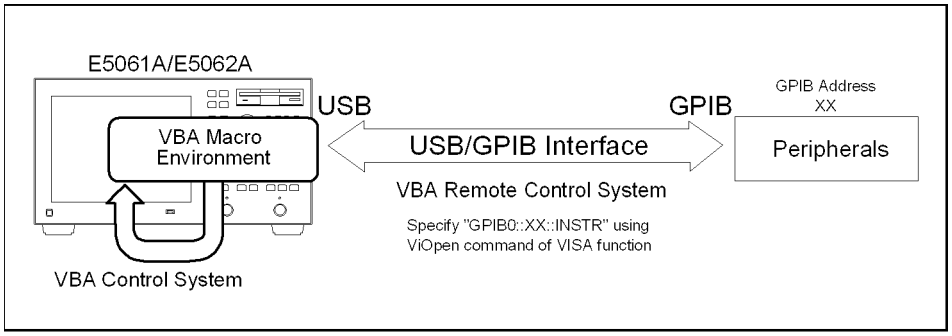
# An Overview of a Control System Based on the Macro Function

This section describes how you can use the E5061A/E5062A's built-in VBA macro function to implement a system that controls the E5061A/E5062A 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 Figure 2-1, a VBA control system controls the E5061A/E5062A itself while a VBA remote control system controls peripherals. When you use the macro function to control peripherals, you must connect the E5061A/E5062A with the peripherals through USB/GPIB interface, and configure them to communicate over VISA (Virtual Instrument Software Architecture). For information on programming using the VISA library, refer to “Programming with VISA” on page 83.

**Figure 2-1** **Configuration example of control system using macro environment**




e5061ave001

### Required Equipment

1. E5061A/E5062A
2. Peripherals and/or other instruments that serve your purpose
3. USB/GPIB interface

**NOTE** To use the VBA remote control system, you need to set the USB/GPIB interface correctly. For detail, refer to *User's Guide*.

**NOTE** Do not connect two or more USB/GPIB interfaces.

## Control Methods

The command set you can use differs depending on whether you use the macro function to control the E5061A/E5062A or a peripheral.

### Controlling the E5061A/E5062A

When you want to control the E5061A/E5062A itself, you can create a program using COM objects within the E5061A/E5062A VBA environment. COM objects that come with the E5061A/E5062A include seven objects specific to the COM interface and COM objects that correspond to SCPI commands.

For information on using E5061A/E5062A's COM objects, see Chapter 7, “COM Object Reference,” on page 99. For information on using SCPI commands, see the “SCPI Command Reference” in the *E5061A/E5062A Programmer's Guide*.

### Controlling a Peripheral

When you want to control a peripheral, you can create a program using VISA library functions within the E5061A/E5062A VBA environment.

For information on using the VISA library, see Chapter 5, “Controlling Peripherals,” on page 81. For a complete description of VISA functions, refer to the VISA library's online help. You can access this online help by double-clicking a file named visa.hlp contained in the CD-ROM (Agilent part No. E5061-905xx).

For information on the GPIB commands available with a particular peripheral, refer to the documentation that comes with the peripheral.

# Overview of E5061A/E5062A COM Object

The E5061A/E5062A VBA environment provides COM objects that support controlling the E5061A/E5062A. This section provides an overview of COM objects as well as considerations for using the E5061A/E5062A's COM objects. For more information on the E5061A/E5062A's COM objects and the comparison with SCPI commands, refer to Chapter 7, “COM Object Reference,” on page 99.

The definitions and specifications of COM are beyond the scope of this guide. Such in-depth information is covered in a variety of books on COM.

## About COM Object

When you control the E5061A/E5062A through the macro function, you can use COM objects as components of your application. The functionality of the E5061A/E5062A'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 E5061A/E5062A, you can use properties to set or read the settings of the E5061A/E5062A.

You can find properties in the list of object types in Chapter 7, “COM Object Reference,” on page 99.

### Method

A method allows you to manipulate an object in a particular way. With the E5061A/E5062A, you can use methods to perform specific tasks.

You can find methods in the list of object types in Chapter 7, “COM Object Reference,” on page 99.

### Event

An event means an operation from outside that the program can recognize such as clicking a mouse. The E5061A/E5062A detects events that a specific softkey is pressed using the UserMenu_OnPress(ByVal Key_id As Long) on page 121 procedure to execute the assigned procedure.

## Using COM Object to Control the E5061A/E5062A

When you want to control the E5061A/E5062A, you can use COM objects alone or in conjunction with SCPI commands and the Parse on page 117 object. The latter method is a little slower than the former method because the Parse on page 117 object is used to parse the messages of SCPI commands. For instructions on using the E5061A/E5062A's VBA Editor to create a program that uses COM objects, refer to Chapter 3, “Operation Basics of the E5061A/E5062A's VBA,” on page 29.

## Major Control Difference between COM Object and SCPI Command

While the control using SCPI commands allows SRQ (Service Request) interrupts through the status reporting mechanism, the control using COM objects does not support SRQ interrupts. Instead of SRQ interrupts, you can use the **WaitOnSRQ** object to suspend the program until the E5061A/E5062A is put into the desired state. For a detailed example of use, see “WaitOnSRQ” on page 124.

# 3 Operation Basics of the E5061A/E5062A's VBA

This chapter provides descriptive information on basic operations for creating VBA programs within the E5061A/E5062A's VBA environment; topics include launching Visual Basic Editor, creating, saving, and running VBA programs, and so on.

# Displaying Visual Basic Editor

This section describes how to launch Visual Basic Editor.

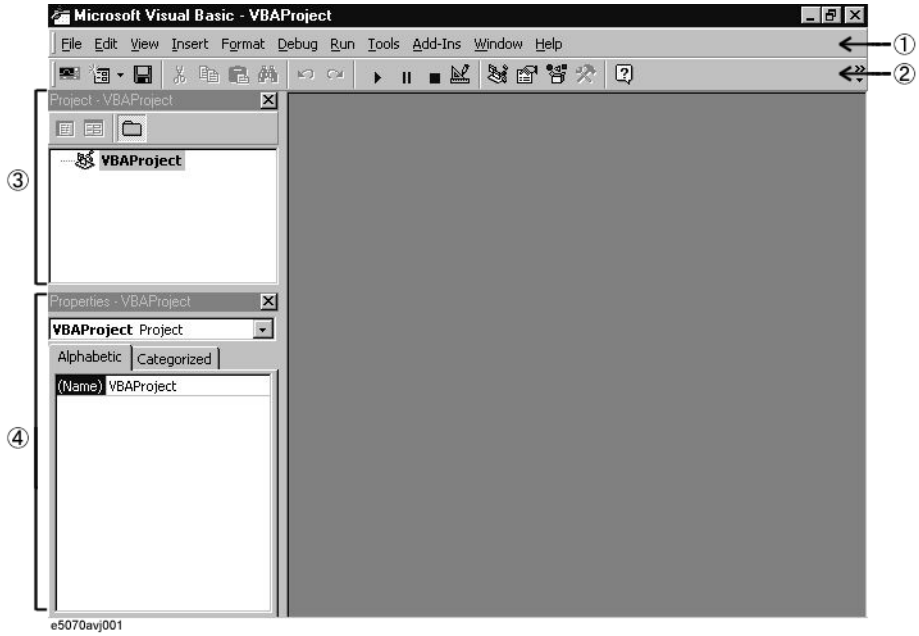
**Step 1.** From the E5061A/E5062A measurement screen, launch Visual Basic Editor using one of the following methods:

- **[Macro Setup] - VBA Editor**
- Press **[Alt] + [F11]** 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 Figure 3-1. The initial screen provides the following GUI elements:

**Figure 3-1 Example of Visual Basic Editor initial screen**



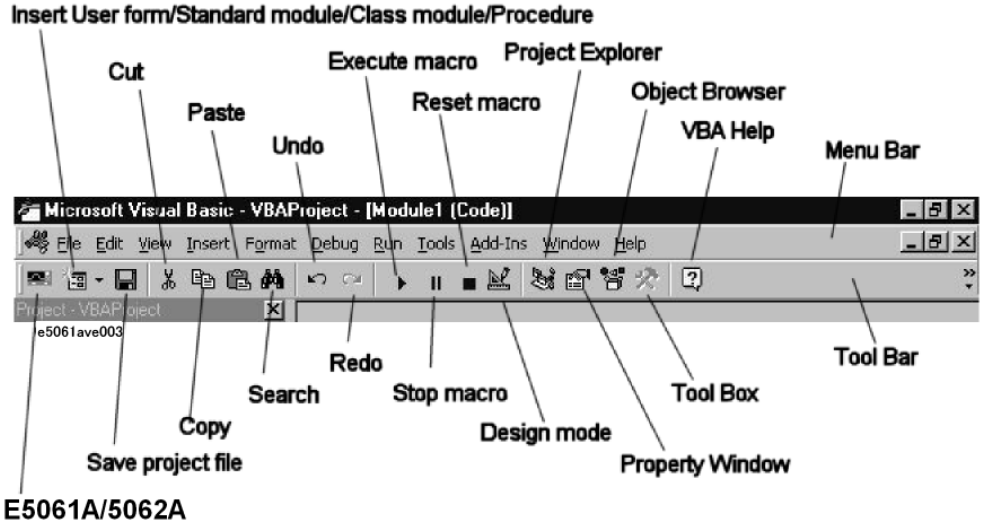

### 1. Menu Bar

Clicking one of the menu labels brings up the corresponding menu. The menu bar can be used as the primary method to navigate through E5061A/E5062A's VBA environment.

### 2. 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. For the description of the buttons on the standard toolbar, see Figure 3-2.

**Figure 3-2** **Buttons on the standard toolbar**




### 3. Project Explorer

Within the E5061A/E5062A'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" on page 33.

**Step 1.** To display the project explorer, do one of the following:

- On the **View** menu, click **Project Explorer**.
- Press **[Ctrl] + [R]** on the keyboard.
- On the toolbar, click "Project Explorer" icon (Figure 3-2).

### 4. 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" on page 33.

You can also set properties by programming in the code window.

**Step 1.** To display the project explorer, do one of the following:

- On the **View** menu, click **Properties Window**.
- Press **[F4]** on the keyboard.
- On the toolbar, click "Property Window" icon (Figure 3-2).

## Closing Visual Basic Editor

This section describes how to quit Visual Basic Editor.

**Step 1.** Close the Visual Basic Editor using one of the following methods:

- On Visual Basic Editor's **File** menu, click **Close and Return to E5062**.
- Within Visual Basic Editor, press **[Alt] + [Q]** on the keyboard.
- **[Macro Setup] - Close Editor**(E5061A/E5062A 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 E5061A/E5062A, 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.

## Switching to the E5061A/E5062A Measurement Screen

You can switch to the E5061A/E5062A measurement screen without closing Visual Basic Editor.

**Step 1.** To switch to the E5061A/E5062A measurement screen, do one of the following:

- On the **View** menu, click **E5062**.
- Press **[Alt] + [F11]** on the keyboard.
- On the toolbar, click “E5061A/E5062A” icon (Figure 3-2).
- Press the **[Focus]** key on the E5061A/E5062A front panel.

# Making a Preparation Before Coding

## A Project and Three Types of Module

Project Explorer (Figure 3-1) displays a list of files (modules) that are used in the E5061A/E5062A 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 purposes as described below.

### Project

When you develop an application within the E5061A/E5062A'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 windows appear on the Visual Basic Editor by inserting the modules in a project. You can do coding (programming) on this code windows practically.

The E5061A/E5062A's VBA environment does not allow you to manage multiple projects. When the current project is existing 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 from the E5061A/E5062A measurement screen.

- **[Macro Setup] - New Project**

**NOTE** When you replace the current project with a new project, the message whether or not the current project is saved may appear. If you want to save the project, click **Yes** button to display a dialog box for saving (Figure 3-6 on page 42). For saving the project, see “Saving a Project” on page 42.

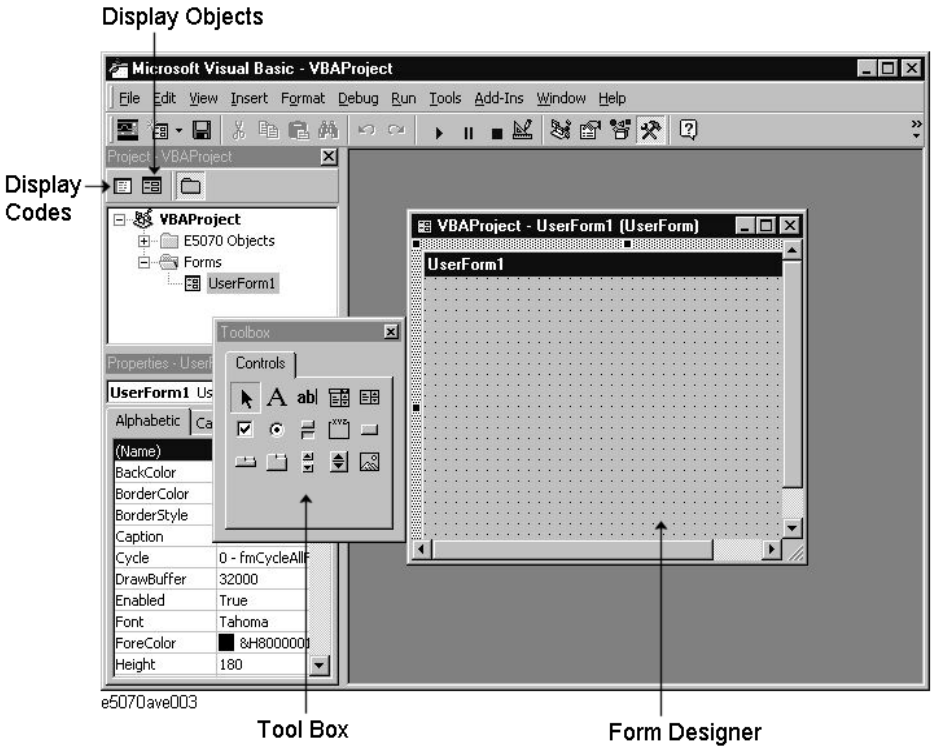
### Inserting the User Form

Within Visual Basic Editor, do one of the following to add a user form to your project (this brings up such a window as shown in Figure 3-3):

- On the **Insert** menu, click **UserForm**.
- On the toolbar, click “Insert User Form/Standard Module/Class Module/Procedure” icon (Figure 3-2), and click **UserForm**.
- In Project Explorer (Figure 3-1), 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 “Show Code” icon (Figure 3-3) in Project Explorer (Figure 3-1) or double-click a control placed on the user form.

**Figure 3-3** **Adding a user form**

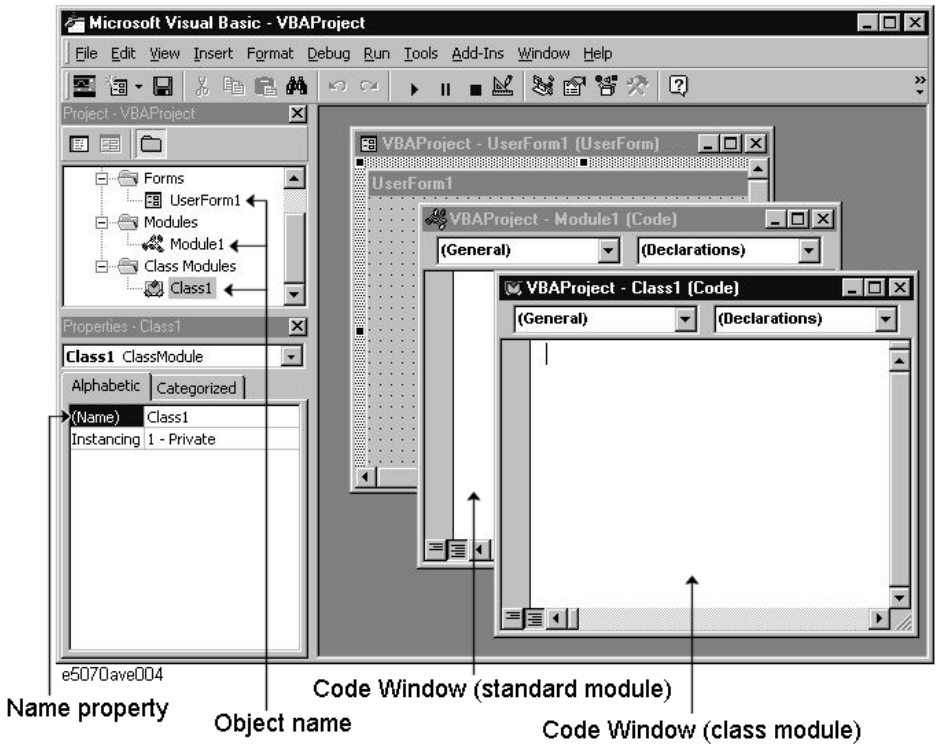


### Inserting the Standard Module

Within Visual Basic Editor, do one of the following to add a standard module to your project (this brings up such a window as shown in Figure 3-4):

- On the **Insert** menu, click **Module**.
- On the toolbar, click "Insert User Form/Standard Module/Class Module/Procedure" icon (Figure 3-2), and click **Module**.
- In Project Explorer (Figure 3-1), right-click the "VBAProject" icon, and click **Insert - Module**.

**Figure 3-4** **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 (this brings up such a window as shown in Figure 3-4):

- On the **Insert** menu, click **ClassModule**.
- On the toolbar, click "Insert User Form/Standard Module/Class Module/Procedure" icon (Figure 3-2), and click **ClassModule**.
- In Project Explorer (Figure 3-1), 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”.

**Step 1.** In Project Explorer (Figure 3-1), click the “Class1” class module under the “Class Modules” icon to highlight it.

**Step 2.** Delete the “Class1” class module using one of the following methods:

- On the **File** menu, click **Remove Class1...**.
- Click the right mouse button, and click **Remove Class1...**.

**Step 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.

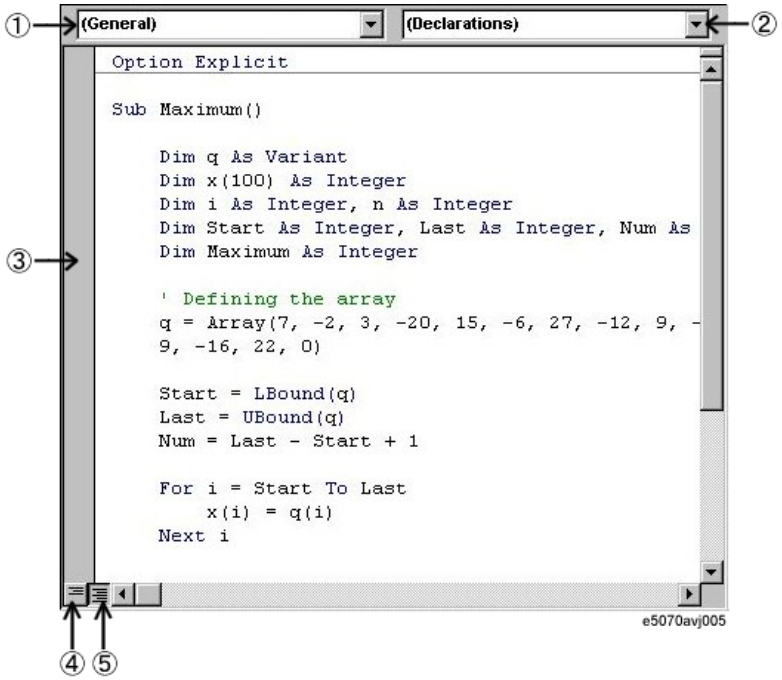
# Coding a VBA Program

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 (Figure 3-1).

**Figure 3-5** **Code window for a standard module**




### 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

This section walks through a sample program that finds the maximum value contained in an array while breaking down the code into a number of blocks and describing what they do. Line numbers are added for description purpose only, and do not appear in the actual program source code.

**Example 3-1** **Sample program that finds the maximum value contained in an array**

```
 10|  Option Explicit
 20|
 30|  Sub Maximum()
 40|
 50|      Dim q As Variant
 60|      Dim x(100) As Integer
 70|      Dim i As Integer, n As Integer
 80|      Dim Start As Integer, Last As Integer, Num As Integer
 90|      Dim Maximum As Integer
100|
110|      ' Defining the array
120|     q = Array(7, -2, 3, -20, 15, -6, 27, -12, 9, -5, 18, 23, _
130|      9, -16, 22, 0)
140|
150|      Start = LBound(q)
160|      Last = UBound(q)
170|      Num = Last - Start + 1
180|
190|      For i = Start To Last
200|          x(i) = q(i)
210|      Next i
220|
230|      Maximum = x(Start)
240|
250|      For n = Start + 1 To Last
260|          If x(n) > Maximum Then Maximum = x(n)
270|      Next n
280|
290|      MsgBox Maximum
300|
310|  End Sub
```

Let us break down the code into a number of blocks and see what they do.

| | |
|---|---|
| Line 10 | This instruction mandates explicit declaration of variables. |
| Lines 30 to 310 | The code enclosed between Sub Maximum() and End Sub will be executed within the E5061A/E5062A's macro environment. Thus enclosed code is called a procedure. In this example, "Maximum" is the procedure name. |
| Lines 50 to 90 | These lines declare data types of variables using Dim statements. A statement is the minimum instruction unit based on the syntax. The sample program declares the variable "q" as Variant, and the variables "x(100)", "i", "n", "Start", "Last", "Num", and "Maximum" as Integer. For a complete list of statements and data types supported by VBA, see VBA Online Help. |
| Line 110 | Any text preceded by a comment indicator (') is treated as a comment. |
| Lines 120 to 130 | These lines use VBA's Array function to initialize the array. The q() array contains elements delimited with commas in the ascending order of index numbers (zero-based). A combination of a space and underscore (_) is used to continue the statement across two or more lines. |
| Line 150 | Stores the starting index number of the q array into the Start variable. |
| Line 160 | Stores the last index number of the q array into the Last variable. |
| Line 170 | Stores the number of elements in the q array into the Num variable. |
| Lines 190 to 210 and Lines 250 to 270 | The code within each For ···Next statement is iterated until the counter reaches the specific number. |
| Line 200 | Stores the contents of the q array (Variant) into the x variable (Integer). |
| Line 230 | Uses the first element of the x array as the tentative maximum value. |
| Line 260 | Compares the tentative maximum value with each of elements that follow; if an element is larger than the tentative maximum value, then that element is used as the tentative maximum value. |
| Line 290 | Uses a message box function to display the maximum value. For a complete list of functions supported by VBA, see VBA Online Help. |

**NOTE** The sample program in Example 3-1 consists of a single procedure contained in a single module. However, when you deal with procedures and variables across multiple modules, you should be aware of the scope of variables and procedures.

### 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 on page 280 object.

**Step 1.** In a standard module, type **sub main** and press the **[Enter]** key. **End Sub** is automatically added.

**Step 2.** Typing **scpi** followed by a dot (**.**) brings up a list of classes under the SCPI class.

**Step 3.** Typing **in** automatically moves focus to **INITiate** in the list box.

**Step 4.** Typing **(** brings up a list of indexes.

**Step 5.** Typing **1).** brings up a list of classes under the INITiate class.

**Step 6.** Typing **c** automatically moves focus to **CONTinuous** in the list box.

**Step 7.** Typing **=** brings up a list box for setting a Boolean value (**True/False**).

**Step 8.** Typing **t** automatically moves focus to **True**.

**Step 9.** Pressing the **[Enter]** key completes the statement: SCPI.INITiate(1).CONTinuous = True.

# Saving a VBA program

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

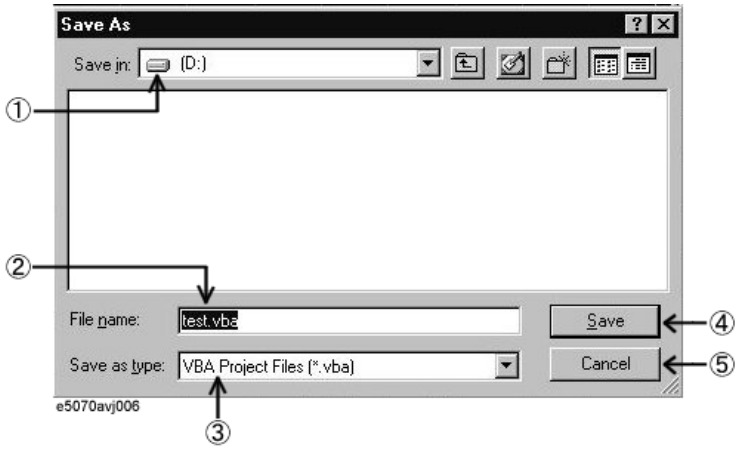
**Step 1.** Open the Save As dialog box by doing one of the following:

- On the **File** menu, click **Save xxx.VBA**. "**xxx**" represents the file name.
- On the toolbar, click "Save Project File" icon (Figure 3-2).
- Press **[Ctrl] + [S]** on the keyboard.

**Step 2.** The Save As dialog box (Figure 3-6) appears. Specify the file name and location (drive or folder) and click **Save**.

The Save As dialog box has the following user interface elements:

**Figure 3-6** **Save As dialog box**




**1. Save in:** Specify the location (drive or folder) where to save the file.

**2. File name** Type in the file name.

**3. Save as type:** Select the type of the file you are saving. Normally, you should select **VBA Project Files [*.vba]**.

**4. Save:** Clicking this button saves the project.

**5. Cancel:** Clicking this button closes the Save As dialog box and brings you back to the main screen.

**E5061A/E5062A Saving a Project from the E5061A/E5062A Measurement Screen**

**Step 1.** Display the E5061A/E5062A measurement screen following the instructions given in “Switching to the E5061A/E5062A Measurement Screen” on page 32.

**Step 2.** Open the Save As dialog box using the following key sequence:

- **[Macro Setup] - Save Project**

**Step 3.** The Save As dialog box (Figure 3-6) 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.

**Step 1.** In Project Explorer (Figure 3-1), click the file name that appears under the desired module icon to highlight it.

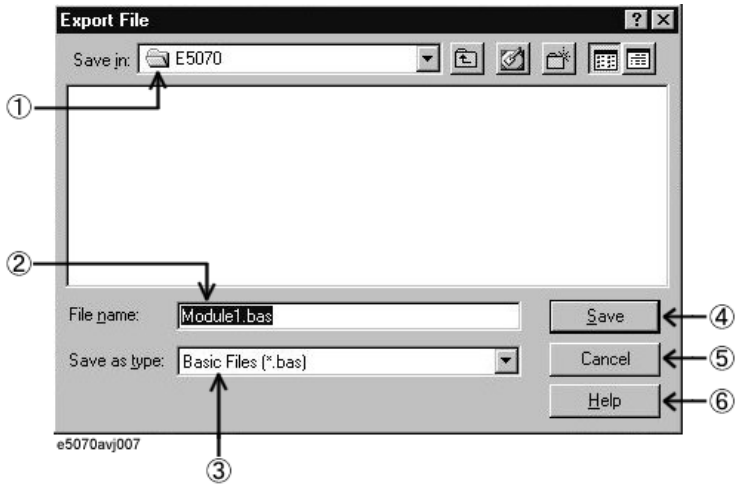
**Step 2.** Open the Export File dialog box by doing one of the following:

- On the **File** menu, click **Export File...**.
- Click the right mouse button, and click **Export File...**.
- Press **[Ctrl] + [E]** on the keyboard.

**Step 3.** The Export File dialog box (Figure 3-7) appears. Specify the file name and location (drive or folder) and click **Save**.

The Export File dialog box has the following user interface elements:

**Figure 3-7** **Export File dialog box**




**1. Save in:** Specify the location (drive or folder) where to save the file.

**2. File name** Type in the file name.

**3. Save as type:** Select the type of the module you are saving. The type that corresponds to the module you are saving is selected by default. Normally, you should use the default.

**4. Save:** Clicking this button saves the module.

**5. Cancel:** Clicking this button closes the Export File dialog box and brings you back to the main screen.

**6. Help:** Clicking this button brings up VBA Online Help.

# Loading a VBA Program

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 E5061A/E5062A measurement screen or by specifying that the project file be automatically loaded when the power is turned on.

### Loading a Project from the E5061A/E5062A Measurement Screen

**Step 1.** Access the Open dialog box using the following key sequence:

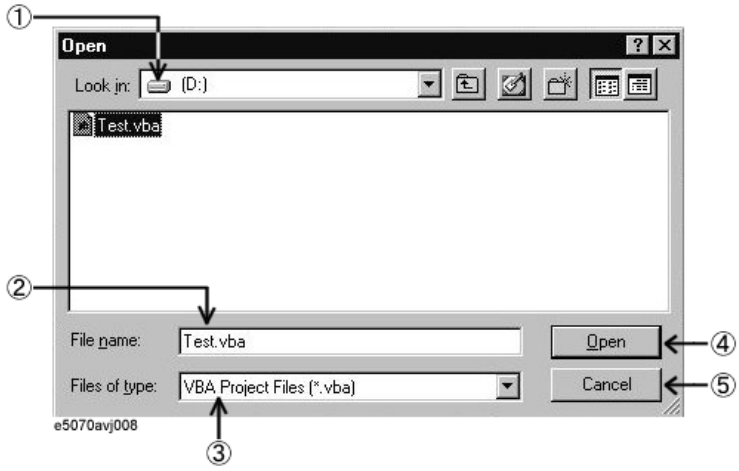
- **[Macro Setup] - Load Project**

**NOTE** When the another project has already been loaded on the Visual Basic Editor, the message whether or not the current project is saved may appear. If you want to save the project, click **Yes** button to display a dialog box for saving (Figure 3-6 on page 42). For saving the project, see "Saving a Project" on page 42.

**Step 2.** The Open dialog box (Figure 3-8) appears. Specify the file name and location (drive or folder) of the file you want to load and click **Open**.

The Open dialog box has the following user interface elements:

**Figure 3-8 Open dialog box**




**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 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 will be automatically loaded whenever the power is turned ON.

| Auto-loaded project | Conditions |
|---|---|
| Directory where the project resides. | A:\ (A:\) or D:\ (D:\) |
| Project file name | autoload.vba*1 |

*1. Upper/lower case insensitive.

**NOTE** 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.

**Step 1.** In Project Explorer (Figure 3-1), click the file name that appears under the desired module icon to highlight it.

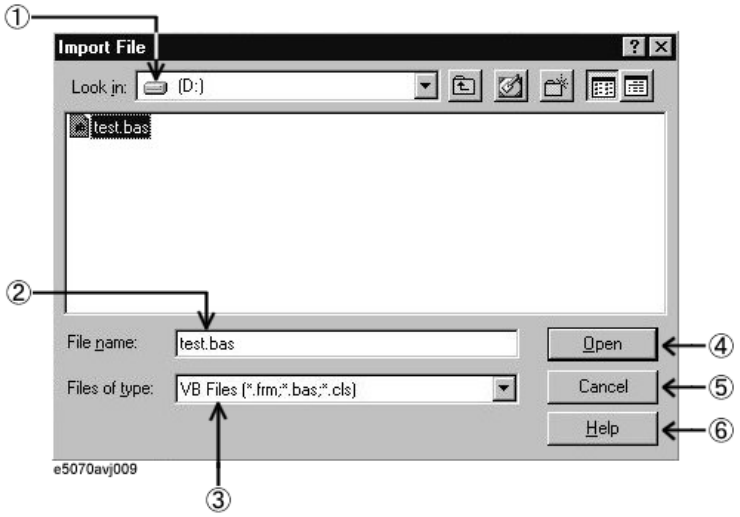
**Step 2.** Open the Import File dialog box by doing one of the following:

- On the **File** menu, click **Import File...**.
- In Project Explorer (Figure 3-1), right-click the “VBAProject” icon, and click **Import File...**.
- Press **[Ctrl] + [M]** on the keyboard.

**Step 3.** The Import File dialog box (Figure 3-9) appears. Specify the file name and location (drive or folder) of the file (module) you want to load and click **Open**.

The Import File dialog box has the following user interface elements:

**Figure 3-9** **Import File dialog box**




**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

The E5061A/E5062A provides 2 methods to execute a VBA program: executing a program that you previously loaded 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 Figure 3-10. “Run” indicates that the program is running while “Stop” indicates that the program is stopped.

**Figure 3-10** **Instrument status bar indicating the status of the VBA program**

## Running a previous loaded VBA program

The E5061A/E5062A allows you to run a previous loaded VBA program using one of the four methods listed below.

**Running a Program from Visual Basic Editor**

**Step 1.** Open the Macros dialog (Figure 3-11)box by doing one of the following:

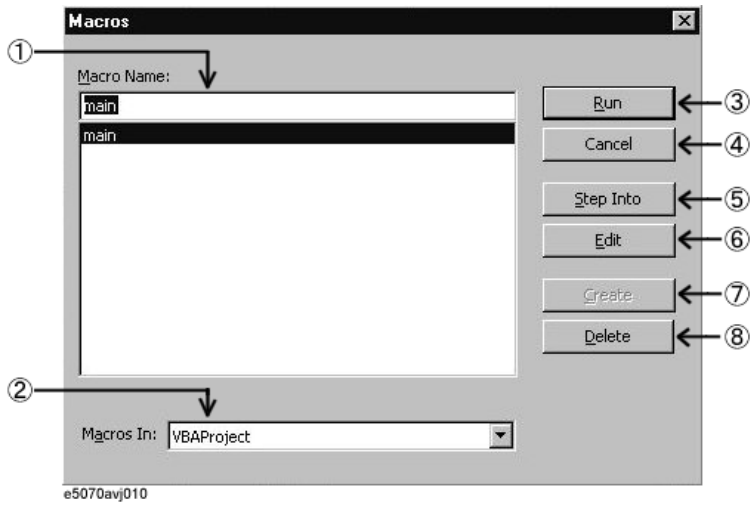
- On the **Run** menu, click **Run Macro**.
- On the **Tools** menu, click **Macros...**.
- On the toolbar, click “Run Macro” icon (Figure 3-2).
- Press **[F5]** on the keyboard.

**NOTE** Doing the above steps with the cursor positioned within a procedure in the code window immediately runs the program without displaying the Macros dialog box.

**Step 2.** In the Macros dialog box, select the VBA program (procedure name) you want to run, and click the **Run** button.

The Macros dialog box has the following user interface elements:

**Figure 3-11** **Macros dialog box**




**1. Macro Name:** Select the VBA program (procedure name) you want to run from the list box so 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” on page 54.

**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.

**NOTE** The Macros dialog provides access to subprograms (procedures enclosed between Sub and End Sub) created in a standard module.

**Running a Program from the E5061A/E5062A Measurement Screen**

The E5061A/E5062A allows you to run a program from E5061A/E5062A screen using one of the four methods listed below.

**Step 1.** Display the E5061A/E5062A measurement screen following the instructions given in “Switching to the E5061A/E5062A Measurement Screen” on page 32.

**Step 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: see Figure 3-4 on page 36) and “**xxx**” is the procedure name.

- Press the **[Macro Run]** key on the E5061A/E5062A front panel. For a program to be 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 E5061A/E5062A measurement screen, the E5061A/E5062A'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 program 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.

**Step 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.

**Step 2.** Press Macro Setup

**Step 3.** Press **Load & Run**.

**Step 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 whose procedure name is set to "Main" (subprogram enclosed between Sub Main() and End Sub) and whose object name (Name property as displayed in the property window) is set to "Module" 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 with the Dialog Box Appeared

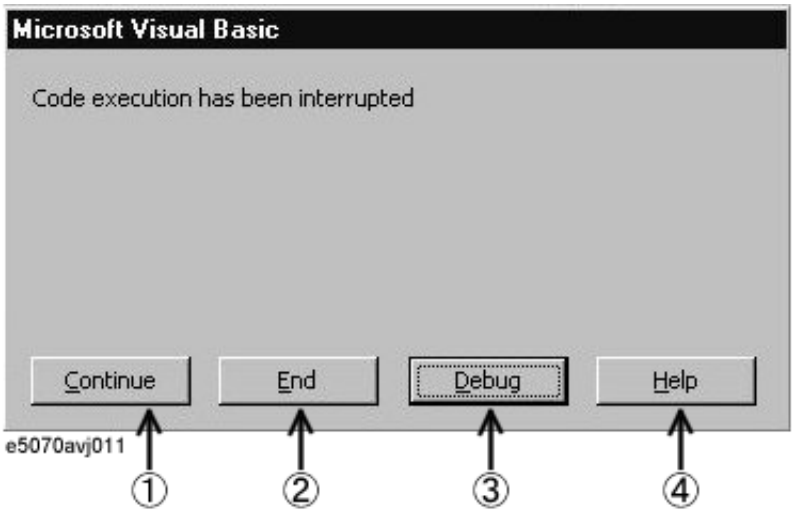
This section describes how to break a procedure during the execution of a VBA program (display a dialog box as shown in Figure 3-12 using forced interrupts).

**Step 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 (Figure 3-2).
- Press **[Ctrl] + [Break]** on the keyboard.
- **[Macro Setup] - Stop**(E5061A/E5062A measurement screen)
- Press the **[Macro Break]** key on the E5061A/E5062A front panel.

**Step 2.** A dialog box as shown in Figure 3-12 is displayed through forced interrupts, and the program is suspended.

**Figure 3-12 Dialog box that appears when a VBA program is suspended**




**1. Continue:** Resumes the execution of the program.

**2. End:** Terminates the VBA program.

**3. Debug:** Displays a run-time error.

**4. Help:** Brings up VBA Online Help.

## Abruptly Terminating the 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 in the instrument status bar on the bottom of the LCD display.

**Step 1.** To terminate the running VBA program, do one of the following:

- On the **Run** menu, click **Reset**.
- On the toolbar, click “Reset Macro” icon (Figure 3-2).
- Insert an *End* statement into your code.

# Errors and Debugging

## Types of Error

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 highlight 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 E5061A/E5062A VBA environment is by default configured to automatically check for syntax errors, but you can disable the auto syntax check feature using the following steps:

**Step 1.** On the **Tools** menu, click **Options...**.

**Step 2.** On the **Editor** tab, clear the **Auto Syntax Check** check box.

**Step 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 as shown in Figure 3-12 appears. You can terminate the program by clicking the **END** button in the error dialog box. Also, you can 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 run without occurring the 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 fail 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 fail to measure the calibration data because the ECal module is not appropriately connected to test ports, and so on. To avoid interruption of the program by these errors, you can handle these errors like lines 730 to 960 in Example 6-1 on page 91.

## Using a Debug Tool

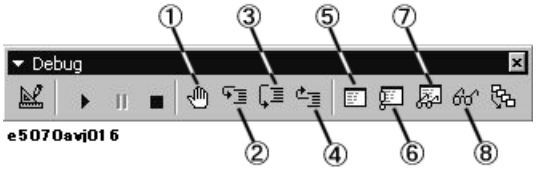
The E5061A/E5062A's VBA environment provides a variety of debug tools that help you identify logical errors. Detailed information on using the debug tools is covered in VBA Online Help and books on VBA.

### Debug Toolbar

The debug toolbar (Figure 3-13) provides tool buttons that allow you to easily access various debug tools. To display the debug toolbar, do the following:

**Step 1.** On the **View** menu, click **Toolbars - Debug**.

**Figure 3-13 Debug toolbar**




1. Set/clear break points (keyboard: **[F9]**)

   Puts 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. (keyboard: **[Shift]+[F9]**)

   Displays the current value of a specified expression in a dialog box.

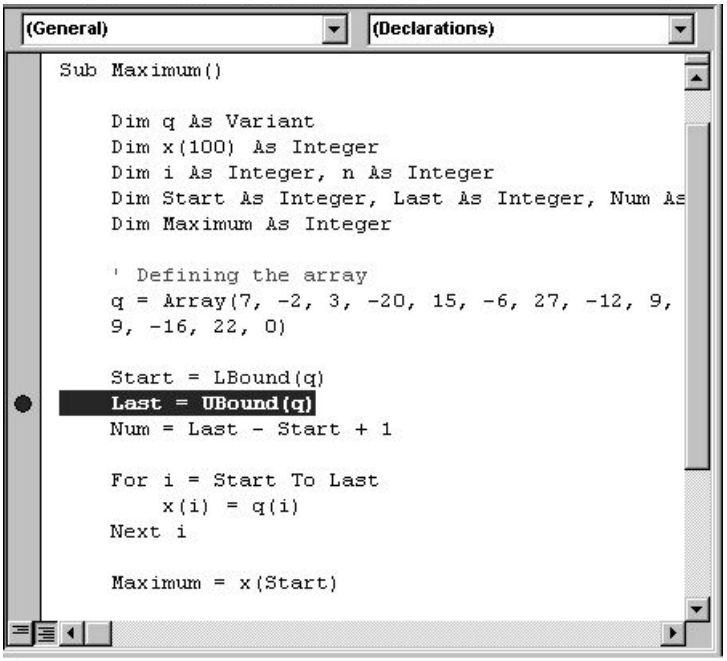
### Setting a Break Point

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.

**Step 1.** When you put a break point at a line, the line is highlighted in umber as shown in Figure 3-14. To set a break point do one of the following:

- Place the cursor at the desired line of code, and click the “Set/clear break points” button (Figure 3-13: 1) on the debug toolbar.
- Click anywhere in the margin indicator bar of the code window.

**Figure 3-14** **Setting a break point**




e5070avj015

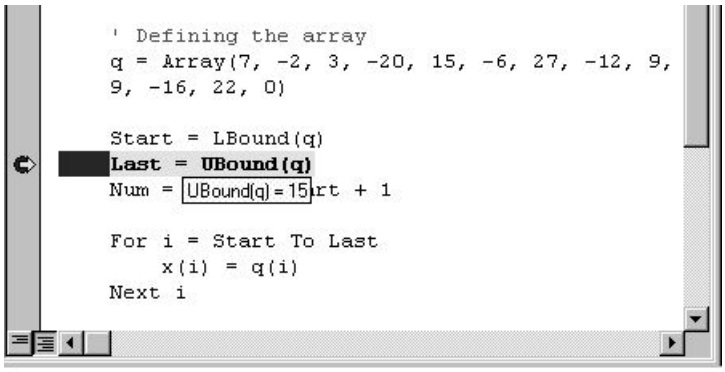
### Monitoring Variable or Property Values

With your VBA program suspended, 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 Figure 3-15.

**Figure 3-15** **Data Hint**




### Immediate Window

To display the immediate window, click the “Immediate Window” button (Figure 3-13:6) on the debug toolbar.

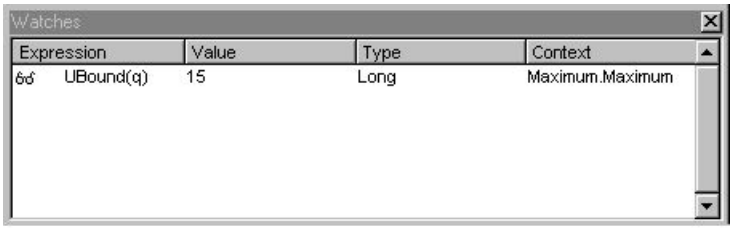
In the immediate window, enter a question mark (?) followed by the variable or expression whose value you want to check, and press the Enter key, as shown in Figure 3-16. The current value appears in the line that follows.

**Figure 3-16** **Immediate window**

### Watch Window

To display the watch window (Figure 3-17), click the “Watch Window” button (Figure 3-13: 7) on the debug toolbar.

**Figure 3-17** **Watch window**



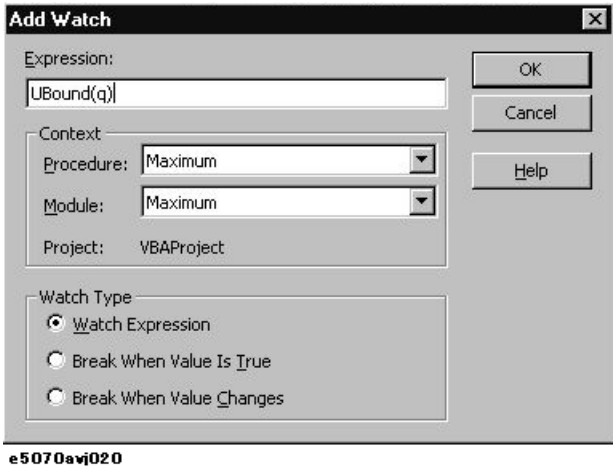

**Step 1.** To open the Add Watch dialog box (Figure 3-18), do the following:

- On the **Debug** menu, click **Add Watch...**.

**Step 2.** As shown in Figure 3-18, you can specify an expression of interest as a watch expression to always monitor its value.

**Step 3.** Click the **OK** button.

**Figure 3-18** **Add Watch dialog box**

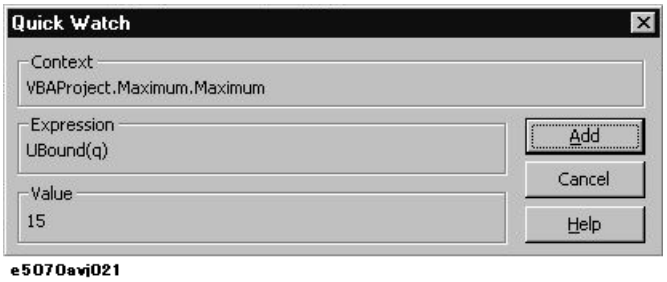


### Quick Watch

In the code window, select a variable or expression whose value you want to watch. On the debug toolbar, click the “Quick Watch” button (Figure 3-13:8) to open the Quick Watch dialog box (Figure 3-19). 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.

**Figure 3-19** **Quick watch**



# Printing Output Values in the Echo Window

The echo window, which appears in the lower part of the E5061A/E5062A measurement screen, can be used to display a message or the return value (data) of an object.

## Entering Values Output to the Echo Window

You can use the COM objects listed below to enter values output to the echo window. For more information on each object, see Chapter 7, “COM Object Reference.”.

- ECHO on page 115
- SCPI.DISPlay.ECHO.DATA on page 240

## Opening the Echo Window

You can use the COM objects listed below to open the echo window. For more information on each object, see Chapter 7, “COM Object Reference.”.

- SCPI.DISPlay.TABLe.TYPE on page 249
- SCPI.DISPlay.TABLe.STATe on page 248

Alternatively, you can also open the echo window using the following key sequence:

- **[Macro Setup] - Echo Window (ON)**

## Clearing Values Output in the Echo Window

You can use the COM object shown below to clear values output to the echo window. For more information on this object, see Chapter 7, “COM Object Reference.”.

- SCPI.DISPlay.ECHO.CLEar on page 240

Alternatively, you can also clear values output to the echo window using the following key sequence:

- **[Macro Setup] - Clear Echo**

# Using VBA Online Help

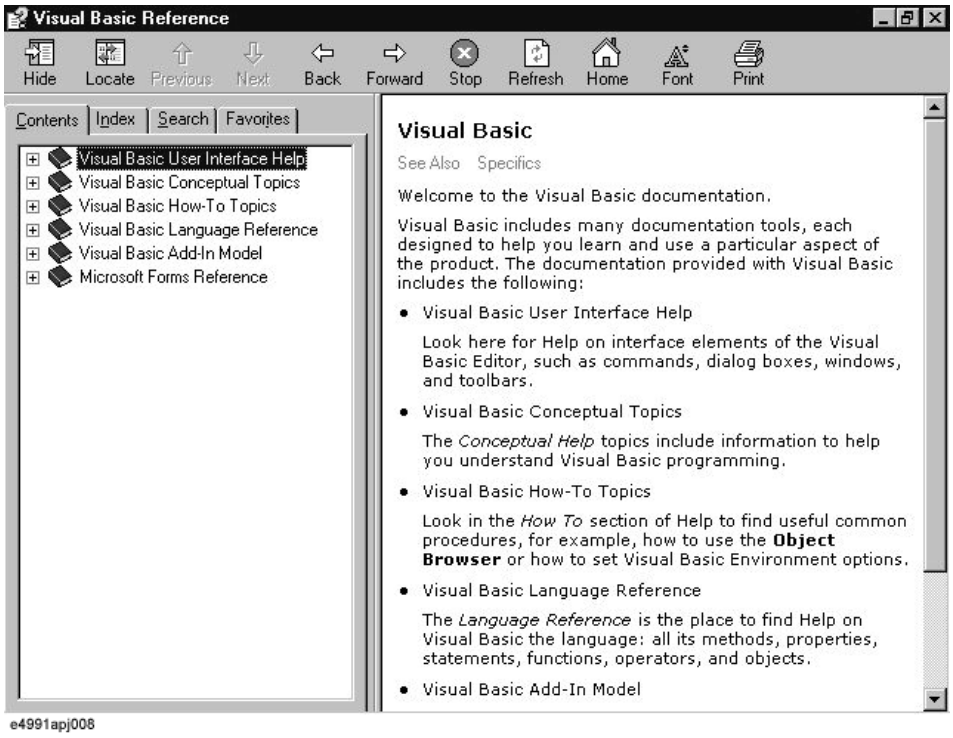
VBA Online Help provides useful topics, such as the VBA terminology or how to use a particular feature. In VBA Online Help, you can find a topic of interest through the Contents or by entering specific keywords.

## Accessing VBA Online Help

**Step 1.** From Visual Basic Editor, do one of the following to access the VBA Online Help screen (Figure 3-20):

- On the **Help** menu, click **Microsoft Visual Basic Help**.
- Press **[F1]** on the keyboard.
- On the toolbar, click "VBA Help" icon (Figure 3-2).

**Figure 3-20 VBA Online Help screen**



### Using the Contents Tab

**Step 1.** Clicking the **Contents** tab in the VBA Online Help screen brings up the items listed below. The E5061A/E5062A 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. Formats of VBA programs are 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

**Step 1.** 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 contained 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]**.

# Uses Advanced Techniques

## Accessing a List of E5061A/E5062A COM Objects

The E5061A/E5062A VBA environment provides COM objects that support controlling the E5061A/E5062A. When you are developing a program using E5061A/E5062A COM objects, you can access a list of E5061A/E5062A COM objects by opening Object Browser within Visual Basic Editor.

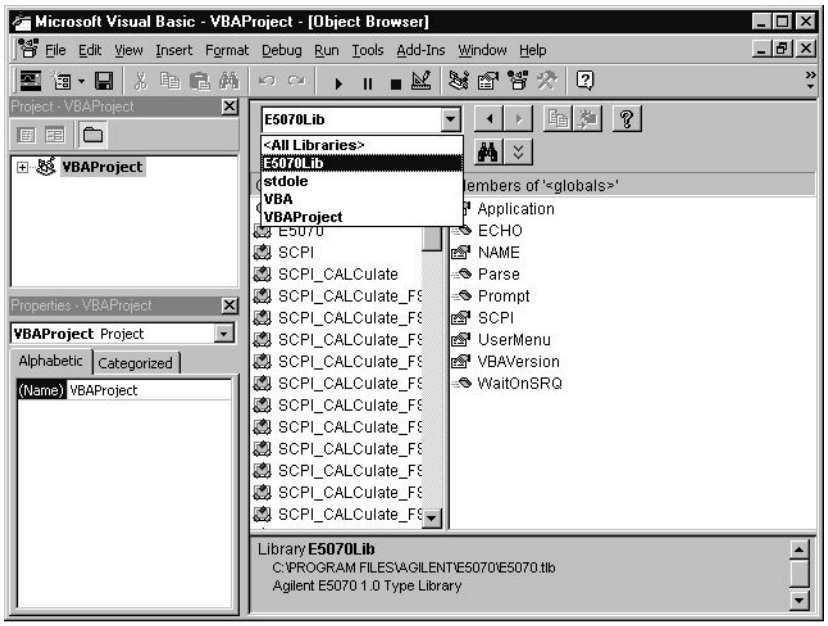
**Step 1.** To open Object Browser, do one of the following:

- On the **View** menu, click **Object Browser**.
- On the toolbar, click "Object Browser" icon (Figure 3-2).

**Step 2.** Select **E5062Lib** from the Project/Library box to display the E5061A/E5062A library as shown in Figure 3-21.

**NOTE** There are some COM objects NOT used in controlling with E5061A/E5062A VBA in the list of the E5061A/E5062A COM objects displayed on the Object Browser. The COM objects NOT used in controlling with E5061A/E5062A VBA are not described in the Chapter 7, "COM Object Reference," on page 99.

**Figure 3-21 How to use Object Browser**



e5070avj042

## Using Automatic Library References

For libraries that satisfy the following conditions, the library reference will be automatically set whenever a new project is created and loaded (**[Macro Setup] - New Project**).

| Automatically referenced libraries | Conditions |
|---|---|
| Directory where the library resides. | D:\Agilent (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.

# 4 Controlling the E5061A/E5062A

This chapter describes how to use the E5061A/E5062A's VBA to control the E5061A/E5062A itself.

# Detecting the End of Measurement

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. For a detailed description of the trigger system and the concept of triggering, see Chapter “Making a Measurement” in *E5061A/E5062A Programmer's Guide* gives a detailed description.

You can detect the end of measurement by using either the status register or the SCPI.TRIGger.SEQuence.SINGle on page 428 object.

## Using the Status Register

The status of the E5061A/E5062A can be detected through the status register. For a complete description of the status report mechanism, including the specifications of each bit of the status register, see Appendix “Status Reporting System” in *E5061A/E5062A Programmer's Guide*.

If your program is based on SPCI commands, you can use SRQ (Service Request) interrupts to detect the end of measurement. For more information, see Section “Waiting for the End of Measurement” in *E5061A/E5062A Programmer's Guide*.

On the other hand, if your program is based on COM objects, SRQ interrupts are not available; instead, you can use the following object to suspend the program until SRQs are generated upon completion of measurement.

- WaitOnSRQ on page 124

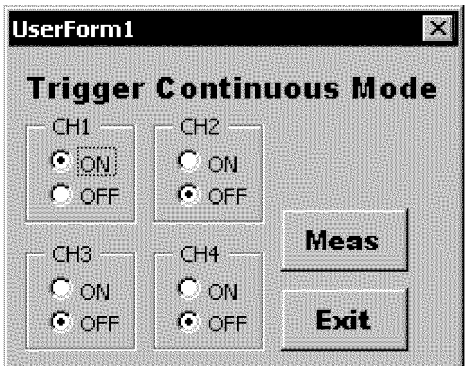
The sample program disk contains a sample program, named “meas_srq.vba”, that demonstrates how to use the status register to suspend the program until the end of measurement. This VBA program consists of the following modules:

**NOTE** For information on loading VBA programs, see “Loading a VBA Program” on page 45.

| Object name | Module type | Content |
|---|---|---|
| frmSrqMeas | UserForm | Uses the status register to wait for the end of measurement. |
| mdlSrqMeas | Standard module | Invokes a UserForm. |

When you run this VBA program, a UserForm as shown in Figure 4-1 appears. For how to use each element in Figure 4-1, see the following description.

**Figure 4-1** **The UserForm when running the Example 4-1 program**




1. The program turns on 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.

Procedure called when the user clicks the **Exit** button on the UserForm (lines 10 to 50)

| | |
|---|---|
| Line 30 | Unloads the UserForm from the memory, and terminates the program. |

Procedure called when the user clicks the **Meas** button on the UserForm (lines 70 to 340)

| | |
|---|---|
| Line 110 | Hides the UserForm (object name: frmSrqMeas) from the screen. |
| Line 130 | Displays 4 channel windows. |
| Line 140 | Sets the trigger source to "bus". |
| Lines 160 to 190 | 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. |
| Lines 210 to 220 | 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). |
| Line 230 | Enables the operation status event register's bit 4. |
| Line 240 | Enables the status byte register's bit 7. |
| Line 250 | Clears the status byte register and operation status event register. |
| Line 260 | Triggers the instrument to start a measurement cycle. |
| Line 270 | 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). |

Lines 280 to 300 — These lines display a measurement completion message upon detecting the end of measurement.

Line 320 — Displays the UserForm (object name :frmSrqMeas) on the screen.

**Example 4-1** **Using SRQs to detect the end of measurement (object name: frmSrqMeas)**

```
 10|  Private Sub cmdExit_Click()
 20|
 30|    Unload Me
 40|
 50|  End Sub
 60|
 70|  Private Sub cmdMeas_Click()
 80|
 90|    Dim Cond As Boolean
100|
110|  frmSrqMeas.Hide
120|
130|    SCPI.DISPlay.Split = "d12_34"
140|    SCPI.TRIGger.SEQuence.Source = "bus"
150|
160|    SCPI.INITiate(1).CONTinuous = optOn1.Value
170|    SCPI.INITiate(2).CONTinuous = optOn2.Value
180|    SCPI.INITiate(3).CONTinuous = optOn3.Value
190|    SCPI.INITiate(4).CONTinuous = optOn4.Value
200|
210|    SCPI.STATus.OPERation.PTRansition = 0
220|    SCPI.STATus.OPERation.NTRansition = 16
230|    SCPI.STATus.OPERation.ENABle = 16
240|    SCPI.IEEE4882.SRE = 128
250|    SCPI.IEEE4882.CLS
260|    SCPI.IEEE4882.TRG
270|    WaitOnSRQ Cond, 100000
280|    If Cond = True Then
290|      MsgBox "Measurement Completion"
300|    End If
310|
320|    frmSrqMeas.Show
330|
340|  End Sub
```

## Using the SCPI.TRIGger.SEQuence.SINGle Object

When you trigger the instrument by issuing the SCPI.TRIGger.SEQuence.SINGle on page 428 object, you can use the SCPI.IEEE4882.OPC on page 276 object to suspend the program until the end of measurement.

The sample program disk contains a sample program, named “meas_sing.vba”, that demonstrates how to use the SCPI.TRIGger.SEQuence.SINGle on page 428 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. |

When you run this VBA program, a window as shown in Figure 4-1 appears. For how to use each element, see Figure 4-1 in the previous section.

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.

Procedure called when the user clicks the **Exit** button on the UserForm (lines 10 to 50)

Line 30 Unloads the UserForm from the memory, and terminates the program.

Procedure called when the user clicks the **Meas** button on the UserForm (lines 70 to 280)

Line 110 Hides the UserForm (object name: frmSingMeas) from the screen.

Line 130 Displays 4 channel windows.

Line 140 Sets the trigger source to "bus".

Lines 160 to 190 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.

Line 210 Triggers the instrument to start a measurement cycle.

Line 220 Executes the **SCPI.IEEE4882.OPC** object to suspend the program until the value of 1 is returned indicating the end of measurement.

Line 240 Displays a measurement completion message.

Line 260 Displays the UserForm (object name: frmSingMeas) on the screen.

**Example 4-2** **Using the SCPI.TRIGger.SEQuence.SINGle object to suspend the program until the end of measurement (object name:frmSingMeas)**

```
 10|  Private Sub cmdExit_Click()
 20|
 30|    Unload Me
 40|
 50|  End Sub
 60|
 70|  Private Sub cmdMeas_Click()
 80|
 90|    Dim Dmy As Long
100|
110|    frmSingMeas.Hide
120|
130|    SCPI.DISPlay.Split = "d12_34"
140|    SCPI.TRIGger.SEQuence.Source = "bus"
150|
160|    SCPI.INITiate(1).CONTinuous = optOn1.Value
170|    SCPI.INITiate(2).CONTinuous = optOn2.Value
180|    SCPI.INITiate(3).CONTinuous = optOn3.Value
190|    SCPI.INITiate(4).CONTinuous = optOn4.Value
200|
210|    SCPI.TRIGger.SEQuence.SINGle
220|    Dmy = SCPI.IEEE4882.OPC
230|
240|    MsgBox "Measurement Completion"
250|
260|    frmSingMeas.Show
270|
280|  End Sub
```

# Reading/Writing Measurement Data

This section describes how to process the E5061A/E5062A'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 Section “Internal Data Processing” in *E5061A/E5062A Programmer's Guide*.

To read/write a formatted data array, formatted memory array, corrected data array, or corrected memory array use the following objects:

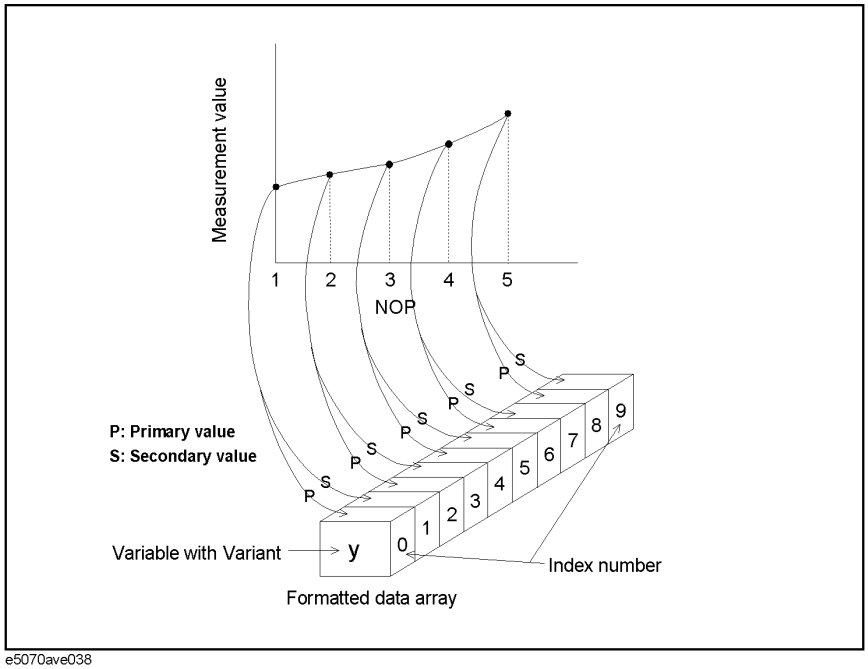
- SCPI.CALCulate(Ch).SELected.DATA.FDATa on page 141
- SCPI.CALCulate(Ch).SELected.DATA.FMEMory on page 142
- SCPI.CALCulate(Ch).SELected.DATA.SDATa on page 143
- SCPI.CALCulate(Ch).SELected.DATA.SMEMory on page 144

To read a c stimulus data array, use the following objects:

- SCPI.SENSe(Ch).FREQuency.DATA on page 355

The E5061A/E5062A 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 Figure 4-2. 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 Section “Internal Data Processing” in *E5061A/E5062A Programmer's Guide*; you can find a table that describes the relationship between contained data items and data formats.

**Figure 4-2** **Example storing data into a Variant variable**




e5070ave038

**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.

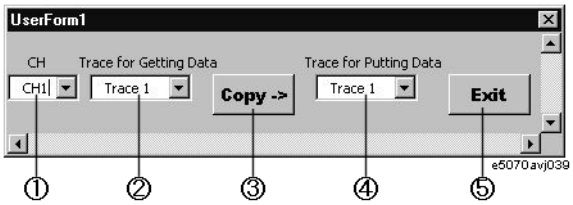
For example, you may wish to read the formatted data array for a particular trace in its entirety (including all measurement points), display the data in the echo window, and then write the data into another trace. How to implement such a process can be better understood with the aid of a sample program.

The sample program disk contains a sample program, 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. |

When you run this VBA program, a window as shown in Figure 4-3 appears. For how to use each element in Figure 4-3, see the following description.

**Figure 4-3 The UserForm when running the Example 4-3 program**




1. The program lets the user specify the channel to be controlled.
2. The program lets the user specify which trace's formatted data array to read (source trace).
3. 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.
4. The program lets the user specify which trace's formatted data array to overwrite (target trace).
5. 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.

Procedure called when the user clicks the **Copy** button on the UserForm (lines 10 to 520)

| | |
|---|---|
| Lines 90 to 160 | These lines identify the selected items in each list and store them into the variables TrGet, TrPut, and ActCh. |
| Lines 180 to 210 | If the specified target trace is not displayed, these lines display that trace. |
| Lines 230 to 250 | These lines make active the specified trace (TrGet: source trace) in the specified channel(ActCh) and hold the sweep. |
| Line 260 | Reads the number of measurement points for the specified channel (ActCh) and stores that number into the Nop variable. |
| Line 280 | Reads the formatted data array for the active trace (source trace) and store the data into the FmtData variable. |
| Line 290 | Reads the stimulus array for the specified channel (ActCh) and stores the data into the Freq variable. |
| Line 330 | Reads the data format for the active trace (source trace) and store it into the Fmt variable. |
| Lines 340 to 350 | These lines display the echo window in the lower part of the LCD screen. |
| Lines 360 to 470 | 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. |
| Line 490 | Makes active the specified trace (TrPut: target trace) in the specified channel(ActCh). |
| Line 500 | Writes the formatted data array (FmtData) into the active trace (target trace). |

Procedure called when the user clicks the **Exit** button on the UserForm (lines 540 to 580)

| | |
|---|---|
| Line 560 | Unloads the UserForm from the memory, and terminates the program. |

Procedure that initializes the UserForm (lines 600 to 870)

| | |
|---|---|
| Lines 620 to 850 | When the program is launched, these lines add each list item and set the default value for each list. |

**Example 4-3** **Reading/displaying/writing a formatted data array (read_write.frm)**

```
10|  Private Sub cmdCopy_Click()
20|
30|    Dim X As Integer, Y As Integer, Z As Integer, I As Integer
40|    Dim ActCh As Long, TrGet As Long, TrPut As Long
50|    Dim TrCont As Long, Nop As Long
60|    Dim FmtData As Variant, Freq As Variant
70|    Dim Fmt As String
80|
90|    X = cboCh.ListIndex
```

```
100|    ActCh = X + 1
110|
120|    Y = cboGet.ListIndex
130|    TrGet = Y + 1
140|
150|    Z = cboPut.ListIndex
160|    TrPut = Z + 1
170|
180|    TrCont = SCPI.CALCulate(ActCh).PARameter.Count
190|    If TrCont < TrPut Then
200|      SCPI.CALCulate(ActCh).PARameter.Count = TrPut
210|    End If
220|
230|    SCPI.CALCulate(ActCh).PARameter(TrGet).SELect
240|    SCPI.INITiate(ActCh).CONTinuous = False
250|    SCPI.ABORt
260|    Nop = SCPI.SENSe(ActCh).SWEep.POINts
270|
280|    FmtData = SCPI.CALCulate(ActCh).SELected.Data.FDATa
290|    Freq = SCPI.SENSe(ActCh).FREQuency.Data
300|
310|  '''Displays the formatted data
320|
330|    Fmt = SCPI.CALCulate(ActCh).SELected.Format
340|    SCPI.DISPlay.TABLe.TYPE = "ECHO"
350|    SCPI.DISPlay.TABLe.STATe = True
360|    Select Case Fmt
370|      Case "MLOG", "PHAS", "GDEL", "MLIN", "SWR", "REAL",
"IMAG", "UPH"
380|          ECHO "Nop", "Frequency(GHz)", "Data"
390|          For I = 0 To Nop - 1
400|              ECHO I + 1, Freq(I) / 1000000000#, FmtData(2 * I)
410|          Next I
420|      Case Else
430|          ECHO "Nop", "Frequency(GHz)", "Data1", "Data2"
440|          For I = 0 To Nop - 1
450|             ECHO I + 1, Freq(I) / 1000000000#, FmtData(2 * I),
FmtData(2 * I + 1)
460|          Next I
470|    End Select
480|
490|    SCPI.CALCulate(ActCh).PARameter(TrPut).SELect
500|    SCPI.CALCulate(ActCh).SELected.Data.FDATa = FmtData
510|
520|  End Sub
530|
540|  Private Sub cmdExit_Click()
550|
560|    Unload Me
570|
580|  End Sub
590|
600|  Private Sub UserForm_Initialize()
610|
620|    With cboCh
630|      .AddItem "CH1"
640|      .AddItem "CH2"
650|      .AddItem "CH3"
```

```
660|      .AddItem "CH4"
670|    End With
680|
690|    With cboGet
700|      .AddItem "Trace 1"
710|      .AddItem "Trace 2"
720|      .AddItem "Trace 3"
730|      .AddItem "Trace 4"
740|    End With
750|
760|    With cboPut
770|      .AddItem "Trace 1"
780|      .AddItem "Trace 2"
790|      .AddItem "Trace 3"
800|      .AddItem "Trace 4"
810|    End With
820|
830|    cboCh.ListIndex = 0
840|    cboGet.ListIndex = 0
850|    cboPut.ListIndex = 0
860|
870|  End Sub
```

# Executing a Procedure with a Softkey (User Menu Function)

The E5061A/E5062A 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 by an event that the 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.

## Preparation for Using the User Menu Function

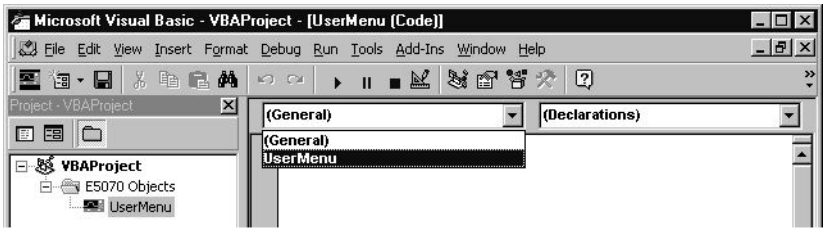
Before using the user menu function, perform the following preparation.

### 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 "E5062 Objects" folder.

**Step 1.** Double-click the "UserMenu" icon in the "E5062 Objects" folder to open the code window.

**Step 2.** In the object box in the code window, click **UserMenu** as shown below.




**Step 3.** In the UserMenu_OnPress(ByVal Key_id As Long) on page 121 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 Example 4-5 on page 79.

**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. For more information on this object, see Chapter 7, “COM Object Reference.”.

- UserMenu.Item(Key_id).Caption on page 119

When you want to set the softkey enabled/disabled for the user menu function, use the following COM object. For more information on this object, see Chapter 7, “COM Object Reference.”.

- UserMenu.Item(Key_id).Enabled on page 120

Moreover, when you want to preset the above settings for the user menu function, use the following COM object. For more information on this object, see Chapter 7, “COM Object Reference.”.

- UserMenu.PRESet on page 121

**NOTE** The above user menu setting is also preset by pressing **[Macro Setup] - Preset User Menu** on the E5061A/E5062A front panel.

## How to Use the 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

**Step 1.** 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. For more information on this object, see Chapter 7, “COM Object Reference.”.

- UserMenu.Press(Key_id) on page 122

## Simple Example

The sample program disk contains a sample program, 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 below:

| | |
|---|---|
| Line 70 | Stores True into the State variable. |
| Lines 90 to 150 | Sets the first to third softkey (*id*: 1 to 3) enabled, and sets the fourth to tenth softkey (*id*: 4 to 10) disabled. |
| Lines 170 to 190 | 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”. |
| Line 210 | Displays the buttons for the user menu function in the softkey area. |
| Lines 230 to 250 | Processing repeated until the State variable is True (State = True).<br>Line 240: Detects an event that a specific softkey is pressed and enables the interrupt from the event. |

**Example 4-4** **Sample program using user menu (object name: mdlUserMenu)**

```
 10|  Public State As Boolean
 20|
 30|  Sub Main()
 40|
 50|    Dim I As Long, J As Long
 60|
 70|    State = True
 80|
 90|    For I = 1 To 3
100|      UserMenu.Item(I).Enabled = True
110|    Next I
120|
130|    For J = 4 To 10
140|      UserMenu.Item(J).Enabled = False
150|    Next J
160|
170|    UserMenu.Item(1).Caption = "Setup"
180|    UserMenu.Item(2).Caption = "Meas"
190|    UserMenu.Item(3).Caption = "Exit"
200|
210|    UserMenu.Show
220|
230|    Do While State
240|      DoEvents
250|    Loop
260|
270|  End Sub
```

The procedures of the “UserMenu” object are described below.

Lines 70 to 190 The procedure when the first softkey (*id*: 1) is pressed.

Line 90: Returns the E5061A/E5062A to the preset state.

Lines 110 to 130 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.

Lines 150 to 170 After aborting the measurement, sets the trigger source to the bus trigger and turns on the continuous trigger startup mode for channel 1.

Line 190: Displays the buttons for the user menu function in the softkey area.

Lines 210 to 320 The procedure when the second softkey (*id*: 2) is pressed.

Lines 230 to 240 Generates a trigger to start a single sweep and waits until the measurement finishes (1 is read out with the **SCPI.IEEE4882.OPC** object).

Line 260: Retrieves the number of points in channel 1 and stores that number into the Nop variable.

Lines 280 to 290 Specifies trace 1 of channel 1 to the active trace, retrieves the formatted data array, and stores the data into the FmtData variable.

Lines 310 to 320 Displays the echo window in the lower part of the LCD screen.

Lines 340 to 360: Displays 2 measurement data values (primary value and secondary value) for each measurement point in the echo window.

Lines 380 to 430 The procedure when the third softkey (*id*: 3) is pressed.

Line 400: Displays a program closing message.

Line 410: Stores False into the sta variable to terminate the main program.

**Example 4-5 Sample program using user menu (“UserMenu” object)**

```
 10|  Private Sub UserMenu_OnPress(ByVal id As Long)
 20|
 30|  Dim I As Integer
 40|  Dim Nop As Long, Dmy As Long
 50|  Dim FmtData As Variant
 60|
 70|  If id = 1 Then
 80|
 90|      SCPI.SYSTem.PRESet
100|
110|      SCPI.SENSe(1).FREQuency.STARt = 1730000000#
120|      SCPI.SENSe(1).FREQuency.STOP = 1830000000#
130|      SCPI.SENSe(1).SWEep.POINts = 51
140|
150|      SCPI.ABORt
```

```
160|      SCPI.TRIGger.SEQuence.Source = "BUS"
170|      SCPI.INITiate(1).CONTinuous = True
180|
190|      UserMenu.Show
200|
210|  ElseIf id = 2 Then
220|
230|      SCPI.TRIGger.SEQuence.SINGle
240|      Dmy = SCPI.IEEE4882.OPC
250|
260|      Nop = SCPI.SENSe(1).SWEep.POINts
270|
280|      SCPI.CALCulate(1).PARameter(1).SELect
290|      FmtData = SCPI.CALCulate(1).SELected.DATA.FDATa
300|
310|      SCPI.DISPlay.TABLe.TYPE = "ECHO"
320|      SCPI.DISPlay.TABLe.State = True
330|
340|      For I = 1 To Nop - 1
350|      ECHO FmtData(2 * I - 2), FmtData(2 * I - 1)
360|      Next I
370|
380|  ElseIf id = 3 Then
390|
400|      MsgBox "Program ended!"
410|      State = False
420|
430|  End If
440|
450|  End Sub
```

# 5 Controlling Peripherals

This chapter explains how to control peripherals connected to the E5061A/E5062A with GPIB by using the software (VISA library) installed in the E5061A/E5062A.

# Overview

The E5061A/E5062A macro function (E5061A/E5062A VBA) can be used not only to automate measurements but also to control external measurement instruments connected via USB/GPIB interface cable by acting as a self-contained system controller (see “An Overview of a Control System Based on the Macro Function” on page 25).

The E5061A/E5062A macro function (E5061A/E5062A VBA) performs communications via the COM interface when controlling the E5061A/E5062A itself, but it communicates via VISA (Virtual Instrument Software Architecture) when controlling external measurement instruments.

To control peripherals connected to the E5061A/E5062A via USB/GPIB interface cable, the following preparation is required.

## Preparation

### Importing Definition Files

To use the VISA library in the E5061A/E5062A macro (E5061A/E5062A 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)” on page 43).

- visa32.bas
- vpptype.bas

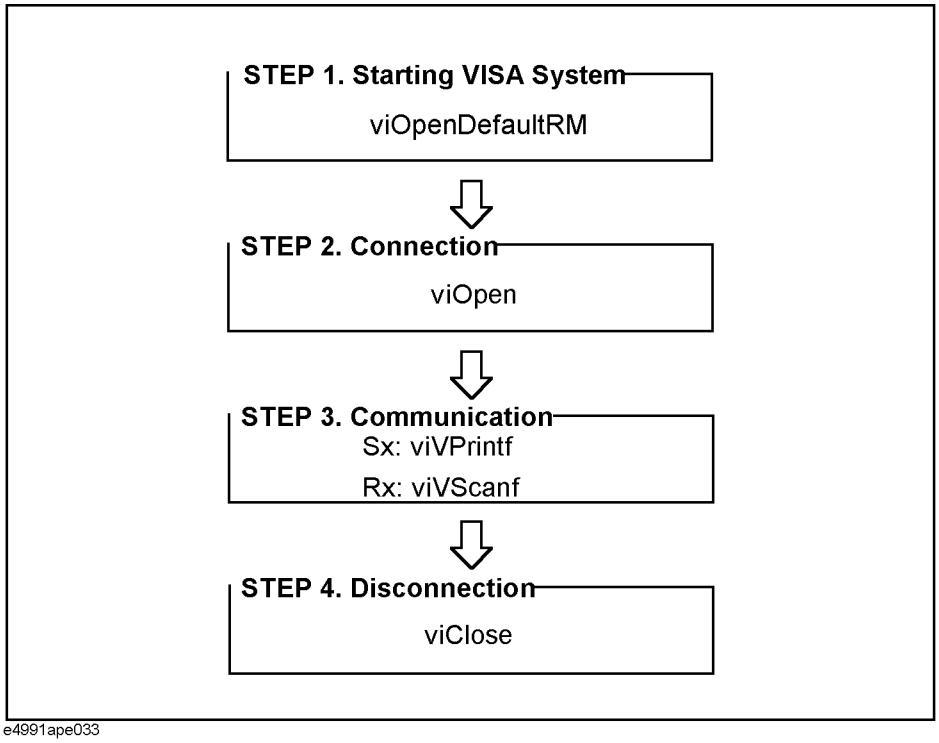
# Programming with VISA

Figure 5-1 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 E5061A/E5062A macro (E5061A/E5062A VBA), refer to the following files contained on the CD-ROM (Agilent part number: E5061-905xx).

- visa.hlp (on-line help for the VISA library)
- vbreadme.txt (notes on using the VISA library with VB)

**Figure 5-1 Flow of instrument control with VISA**




e4991ape033

## STEP 1. Starting Up VISA System

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 is startup information .

**Syntax** viOpenDefaultRM(*param*)

**Parameter**

| | (*param*) |
|---|---|
| Description | Startup information (output) |
| Data type | Long integer type |

## STEP 2. Connection

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 are startup information , the address information of the specified instrument , access mode , timeout , and connection information .

**Syntax** viOpen(*param1,param2,param3,param4,param5*)

**Parameters**

| | (*param1*) |
|---|---|
| Description | Startup information (input) |
| Data type | Long integer type |

| | (*param2*) |
|---|---|
| Description | Address information of the specified instrument (input) |
| Data type | Character string type |
| Syntax | GPIB[*board*]*1::*primary address**2::INSTR |

*1. GPIB0 for the E5061A/E5062A.
*2. The GPIB address of the instrument controlled by the E5061A/E5062A.

| | (*param3*) |
|---|---|
| Description | Access mode (Enter 0) |

| | (*param4*) |
|---|---|
| Description | Timeout (Enter 0) |

| | (*param5*) |
|---|---|
| Description | Connection information (output) |
| Data type | Long integer type |

## STEP 3. Communication

VISA's viVPrintf function sends a program message (GPIB command) to the specified instrument. The parameters of this function are connection information , the program message , and the variable to be formatted .

**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**

| | (*param1*) |
|---|---|
| Description | Connection information (input) |
| Data type | Long integer type |

| | (*param2*) |
|---|---|
| Description | Program message (input)*1 |
| Data type | Character string type |

*1. When sending a program message of the GPIB command, a message terminator is required at the end of the message .

| | (*param3*) |
|---|---|
| Description | A variable to be formatted*1 |
| Data type | Specified data type |

*1. If not applicable, enter 0.

VISA's viVScanf function receives the result from the specified instrument and stores it in the output variable. The parameters of this function are connection information , the format parameter for the output variable, and the output variable.

**Syntax** viVScanf(*param1,param2,param3*)

**Parameters**

| | (*param1*) |
|---|---|
| Description | Connection information (input) |
| Data type | Long integer type |

| | (*param2*) |
|---|---|
| Description | Format parameter for the output variable |
| Data type | Character string type |

| | (*param3*) |
|---|---|
| Description | Output variable (output) |
| Data type | Character string type |

## STEP 4. Disconnection

VISA's viClose function disconnects communication and terminates the VISA system. The parameter of this function is startup information.

**Syntax** viClose(*param*)

**Parameter**

| | (*param*) |
|---|---|
| Description | Startup information (input) |
| Data type | Long integer type |

# 6 Application Programs

This chapter describes sample programs (VBA programs) based on actual measurement examples.

# Basic measurement (measuring a band-pass filter)

Example 6-1 shows a sample program (VBA program) that demonstrates how to perform the basis measurement of the bandpass filter. You can find the source file of this program, named “apl_bsc.vba“, on the sample program disk. This VBA program consists of the following standard module.

| Object name | Module type | Content |
|---|---|---|
| mdlBscMeas | Standard module | Performs the basic measurement of the bandpass filter. |

## Overview of the program

The sample program performs full 2-port calibration using the 85032F calibration kit, measure 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 “Example of measuring a band-pass filter” in *Installation/Quick Start Guide* of the E5061A/E5062A. Therefore, for information on the flow of the measurement, the connection of the standard, and so on, refer to the description of *Installation/Quick Start Guide*.

## Description of the program

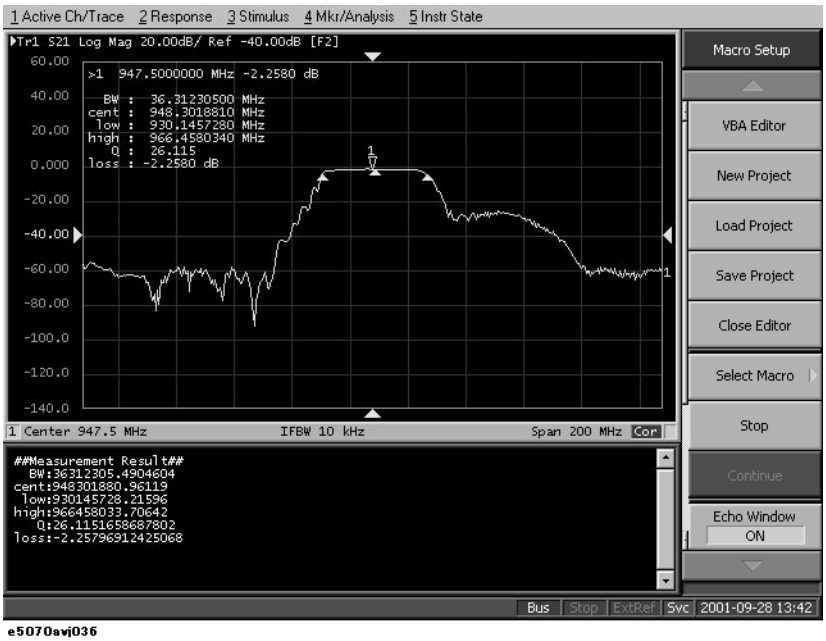
When you run this VBA program, reset is performed, the measurement conditions are automatically set, and a message “Perform the full 2-port calibration.” is displayed. To perform the full 2-port calibration, click the **Yes** button; to skip it, click the **No** button.

To perform the calibration, follow the onscreen messages to connect each standard of the Agilent 85032F calibration kit to the specified port, and click the **OK** button to measure the calibration data. Click the **Cancel** button to return to the beginning of the calibration. You cannot skip the isolation calibration. When the calibration data measurement for all standards is complete, a 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, a message “Connect DUT, and then press [Macro Setup]-Continue button.” 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 in Figure 6-1. If no bandwidth search target is found, only the result of the insertion loss obtained with the marker is displayed.

**Figure 6-1** **Example of the display after executing the program in Example 6-1**




e5070avj036

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.

Lines 120 to 160 — 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), the power level (-10 dBm) into the variables Center, Span, Nop, IfBw, and Pow, respectively.

Lines 170 to 210 — Stores 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.

Line 250 — Returns the E5061A/E5062A to the preset state.

Lines 290 to 300 — For channel 1, turns on the continuous trigger startup mode to on and sets the trigger source to the bus trigger.

Lines 320 to 360 — For channel 1, sets 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.

Lines 380 to 410 — For channel 1, sets the number of traces to the NumTrac variable, the measurement parameter to the Par variable, and the data format to the Fmt variable, respectively.

Line 450 — Stores the calibration kit number for channel 1 to the CalKit variable.

Line 460 — Stores 1 and 2 to the Port variable that indicates ports used for the full 2-port calibration.

Line 480 — Calls the Calib_Solt procedure (lines 1200 to 2130). For information on the Calib_Solt procedure, see the description later.

Lines 520 to 530 Saves the instrument setting and the calibration coefficient into a file whose name is specified with the File variable.

Line 580 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.

Lines 620 to 630 Generates a trigger to start a single sweep and waits until the measurement finishes (1 is read out with the **SCPI.IEEE4882.OPC** object).

Line 650 For trace 1 of channel 1, executes the auto scale to set the optimum scale.

Lines 690 to 710 Displays marker 1, and moves it so that the stimulus value becomes equal to the value of the Center variable. Then, reads out the response value of marker 1 and stores it into the MkrVal variable.

Line 730 Enables the error handling routine starting from Bw_Err (lines 890 to 950). If a runtime error occurs, the program goes to the error handling routine.

Lines 750 to 770 Sets the bandwidth definition value to -3 dB and the bandwidth search result display to on, reads out the bandwidth search result (bandwidth, center frequency, Q value, and insertion loss), and stores it into the BwData variable.

Lines 790 to 840 Based on the bandwidth search result, stores 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, respectively. Then, goes to the processing starting from Skip_Bw_Err.

Lines 880 to 960 Defines a runtime error handler. Reads out and displays the error number and error message of the error that occurred, and stores 0 to the Bw, Cent, and Qfac variables and the response value of marker 1 (the MkrVal(0) variable) to the Loss variable. Then, finishes the error handling and proceeds to the next processing.

Lines 1000 to 1010 Calculates the 2 (higher and lower) cutoff frequencies from the values in the Bw and Cent variables and stores them into the CutLow and CutHigh variables.

Lines 1030 to 1110 Displays the search result (the values of the Bw, Cent, CutLow, CutHigh, Qfac, and Loss variables) in the echo window.

Lines 1130 to 1160 Displays the message asking you whether you want to perform measurement again. Click the **Yes** button to return to the DUT connection section. Click the **No** button to terminate the program.

Procedure: Calib_Solt (lines 1200 to 2130).

Lines 1260 to 1300 Displays the message that prompts for the execution of the full n-port calibration (specified with the SoltType variable). Click the **Cancel** button to cancel the calibration.

Lines 1320 to 1410 Sets the calibration type to the full n-port calibration for the port specified with the Port variable.

Lines 1450 to 1520 Displays the message that prompts for connecting the open standard to the specified port. Starts the measurement of the open calibration data

initiated by clicking the **OK** button after the connection and waits for the completion of the measurement. Click the **Cancel** button to return to the beginning of the calibration.

Lines 1540 to 1610 Displays the message that prompts for connecting the short standard to the specified port. Starts the measurement of the short calibration data initiated by clicking the **OK** button after the connection and waits for the completion of the measurement. Click the **Cancel** button to return to the beginning of the calibration.

Lines 1630 to 1700 Displays the message that prompts for connecting the load standard to the specified port. Starts the measurement of the load calibration data initiated by clicking the **OK** button after the connection and waits for the completion of the measurement. Click the **Cancel** button to return to the beginning of the calibration.

Lines 1750 to 1840 Displays the message that prompts for connecting the thru standard between the specified ports. Starts the measurement of the thru calibration data initiated by clicking the **OK** button after the connection and waits for the completion of the measurement. Click the **Cancel** button to return to the beginning of the calibration.

Lines 1880 to 2060 When the calibration type is not the full 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 the **Yes** button is clicked, displays the message that prompts for connecting the load standard to the specified 2 ports (specified with the Port(I-1) and Port(J-1) variables). Starts the measurement of the isolation calibration data initiated by clicking the **OK** button after the connection and waits for the completion of the measurement. Click the **Cancel** button to return to the beginning of the calibration.

Lines 2080 to 2090 Calculates the calibration coefficients from the measured calibration data and turns on the error correction function. Then, displays a calibration completion message.

**Example 6-1** **Measuring a band-pass filter (object name: mdlBscMeas)**

```
 10|  Sub Main()
  20|
  30|    Dim Par As String, Fmt As String, File As String
  40|    Dim Center As Double, Span As Double, IfBw As Double, Pow
As Double
  50|    Dim Bw As Double, Cent As Double
  60|    Dim CutLow As Double, CutHigh As Double
  70|    Dim Qfac As Double, Loss As Double
  80|    Dim MkrVal As Variant, BwData As Variant
  90|    Dim Nop As Long, NumTrac As Long, CalKit As Long, Buff As
Long
 100|    Dim Port As Variant, Error As Variant
 110|
 120|    Center = 947500000#       'Center freq        : 947.5 MHz
 130|    Span = 200000000#        'Span freq          : 200 MHz
 140|    Nop = 401                'Number of points   : 401
 150|    IfBw = 10000#            'IF bandwidth       : 10 kHz
 160|    Pow = -10                'Power level        : -10dBm
 170|    NumTrac = 1              'Number of traces   : 1
```

```
 180|    Par = "S21"                'Meas. parameter    : S21
 190|    Fmt = "MLOG"               'Data format        : Log Mag
 200|    CalKit = 4                 'Calibration kit    : 85032F
 210|    File = "State08.sta"       'Saved file name    : State08.sta
 220|
 230|  '''Presetting the E5061A/E5062A
 240|
 250|    SCPI.SYSTem.PRESet
 260|
 270|  '''Setting measurement conditions
 280|
 290|    SCPI.INITiate(1).CONTinuous = True
 300|    SCPI.TRIGger.SEQuence.Source = "BUS"
 310|
 320|    SCPI.SENSe(1).FREQuency.Center = Center
 330|    SCPI.SENSe(1).FREQuency.Span = Span
 340|    SCPI.SENSe(1).SWEep.POINts = Nop
 350|    SCPI.SENSe(1).BANDwidth.RESolution = IfBw
 360|    SCPI.Source(1).POWer.LEVel.IMMediate.AMPLitude = Pow
 370|
 380|    SCPI.CALCulate(1).PARameter.Count = NumTrac
 390|    SCPI.CALCulate(1).PARameter(1).DEFine = Par
 400|    SCPI.CALCulate(1).PARameter(1).Select
 410|    SCPI.CALCulate(1).SELected.Format = Fmt
 420|
 430|  '''Performing full 2-port calibration
 440|
 450|    SCPI.SENSe(1).CORRection.COLLect.CKIT.Select = CalKit
 460|    Port = Array(1, 2)
 470|
 480|    Calib_Solt 1, 2, Port
 490|
 500|  '''Saving state & cal data
 510|
 520|    SCPI.MMEMory.STORe.STYPe = "CST"
 530|    SCPI.MMEMory.STORe.STATe = File
 540|
 550|  '''Connecting DUT
 560|
 570|  Meas_Start:
 580|    Prompt ("Connect DUT, and then press [Macro Setup]-Continue
button.")
 590|
 600|  '''Performing single sweep
 610|
 620|    SCPI.TRIGger.SEQuence.SINGle
 630|    Dmy = SCPI.IEEE4882.OPC
 640|
 650|    SCPI.DISPlay.WINDow(1).TRACe(1).Y.SCALe.AUTO
 660|
 670|  '''Analyzing the results
 680|
 690|    SCPI.CALCulate(1).SELected.MARKer(1).STATe = True
 700|    SCPI.CALCulate(1).SELected.MARKer(1).X = Center
 710|    MkrVal = SCPI.CALCulate(1).SELected.MARKer(1).Y
 720|
 730|    On Error GoTo Bw_Err
 740|
```

```
 750|    SCPI.CALCulate(1).SELected.MARKer(1).BWIDth.THReshold = -3
 760|    SCPI.CALCulate(1).SELected.MARKer(1).BWIDth.STATe = True
 770|    BwData = SCPI.CALCulate(1).SELected.MARKer(1).BWIDth.DATA
 780|
 790|    Bw = BwData(0)
 800|    Cent = BwData(1)
 810|    Qfac = BwData(2)
 820|    Loss = BwData(3)
 830|
 840|    GoTo Skip_Bw_Err
 850|
 860|  Bw_Err:
 870|
 880|    Error = SCPI.SYSTem.Error
 890|    MsgBox "Error No:" & Error(0) & " , Description:" & Error(
1)
 900|
 910|    Bw = 0
 920|    Cent = 0
 930|    Qfac = 0
 940|    Loss = MkrVal(0)
 950|
 960|    Resume Skip_Bw_Err
 970|
 980|  Skip_Bw_Err:
 990|
1000|    CutLow = Cent - Bw / 2
1010|    CutHigh = Cent + Bw / 2
1020|
1030|    ECHO "##Measurement Result##"
1040|    ECHO "  BW:" & Bw
1050|    ECHO "cent:" & Cent
1060|    ECHO " low:" & CutLow
1070|    ECHO "high:" & CutHigh
1080|    ECHO "   Q:" & Qfac
1090|    ECHO "loss:" & Loss
1100|    SCPI.DISPlay.TABLe.TYPE = "ECHO"
1110|    SCPI.DISPlay.TABLe.STATe = True
1120|
1130|    Buff = MsgBox("Do you make another measurement?", vbYesNo,
"Bandpass fileter measurement")
1140|    If Buff = vbYes Then
1150|        GoTo Meas_Start
1160|    End If
1170|
1180|  End Sub
1190|
1200|  Private Sub Calib_Solt(Chan As Long, SoltType As Long, Port
As Variant)
1210|
1220|    Dim Dmy As Long, I As Long, J As Long, Buff As Long
1230|
1240|  Cal_Start:
1250|
1260|    Buff = MsgBox("Perform the full " & SoltType & "-port cali
bration.", vbOKCancel, "Full" & SoltType & "-port calibration")
1270|
1280|    If Buff = vbCancel Then
```

```
1290|        GoTo Cal_Skip
1300|    End If
1310|
1320|    Select Case SoltType
1330|        Case 1
1340|           SCPI.SENSe(Chan).CORRection.COLLect.METHod.SOLT1 =
Port(0)
1350|        Case 2
1360|           SCPI.SENSe(Chan).CORRection.COLLect.METHod.SOLT2 =
Port
1370|        Case 3
1380|           SCPI.SENSe(Chan).CORRection.COLLect.METHod.SOLT3 =
Port
1390|        Case 4
1400|           SCPI.SENSe(Chan).CORRection.COLLect.METHod.SOLT4 =
Port
1410|    End Select
1420|
1430|    For I = 1 To SoltType
1440|
1450|       Buff = MsgBox("Connect the Open standard to Port " & CS
tr(Port(I - 1)) & ".", _
1460|                       vbOKCancel, "Full" & SoltType & "-port
calibration")
1470|        If Buff = vbOK Then
1480|           SCPI.SENSe(Chan).CORRection.COLLect.ACQuire.OPEN =
Port(I - 1)
1490|            Dmy = SCPI.IEEE4882.OPC
1500|        Else
1510|            GoTo Cal_Start
1520|        End If
1530|
1540|        Buff = MsgBox("Connect the Short standard to Port " &
CStr(Port(I - 1)) & ".", _
1550|                       vbOKCancel, "Full" & SoltType & "-port
calibration")
1560|        If Buff = vbOK Then
1570|          SCPI.SENSe(Chan).CORRection.COLLect.ACQuire.Short =
Port(I - 1)
1580|            Dmy = SCPI.IEEE4882.OPC
1590|        Else
1600|            GoTo Cal_Start
1610|        End If
1620|
1630|        Buff = MsgBox("Connect the Load standard to Port " &
CStr(Port(I - 1)) & ".", _
1640|                       vbOKCancel, "Full" & SoltType & "-port
calibration")
1650|        If Buff = vbOK Then
1660|           SCPI.SENSe(Chan).CORRection.COLLect.ACQuire.Load =
Port(I - 1)
1670|            Dmy = SCPI.IEEE4882.OPC
1680|        Else
1690|            GoTo Cal_Start
1700|        End If
1710|    Next I
1720|
1730|    For I = 1 To SoltType - 1
```

```
1740|        For J = I + 1 To SoltType
1750|          Buff = MsgBox("Connect the Thru standard between Por
t " & CStr(Port(I - 1)) & _
1760|                                " and Port " & CStr(Port(J - 1))
& ".", vbOKCancel, "Full" & SoltType & "-port calibration")
1770|            If Buff = vbOK Then
1780|
SCPI.SENSe(Chan).CORRection.COLLect.ACQuire.THRU = Array(Port(I - 1
), Port(J - 1))
1790|                Dmy = SCPI.IEEE4882.OPC
1800|
SCPI.SENSe(Chan).CORRection.COLLect.ACQuire.THRU = Array(Port(J - 1
), Port(I - 1))
1810|                Dmy = SCPI.IEEE4882.OPC
1820|            Else
1830|                GoTo Cal_Start
1840|            End If
1850|        Next J
1860|    Next I
1870|
1880|    If SoltType <> 1 Then
1890|        Buff = MsgBox("Do you measure the Isolation (Optional)
?", vbYesNo, "Full" & SoltType & "-port calibration")
1900|        If Buff = vbYes Then
1910|            For I = 1 To SoltType - 1
1920|                For J = I + 1 To SoltType
1930|                   Buff = MsgBox("Connect the Load standard to
Port " & Port(I - 1) & " and Port " & Port(J - 1) & ".", _
1940|                                        vbOKCancel, "Full" & Solt
Type & "-port calibration")
1950|                    If Buff = vbOK Then
1960|
SCPI.SENSe(Chan).CORRection.COLLect.ACQuire.ISOLation = Array(Port(
I - 1), Port(J - 1))
1970|                        Dmy = SCPI.IEEE4882.OPC
1980|
SCPI.SENSe(Chan).CORRection.COLLect.ACQuire.ISOLation = Array(Port(
J - 1), Port(I - 1))
1990|                        Dmy = SCPI.IEEE4882.OPC
2000|                    Else
2010|                        GoTo Cal_Start
2020|                    End If
2030|                Next J
2040|            Next I
2050|        End If
2060|    End If
2070|
2080|    SCPI.SENSe(1).CORRection.COLLect.SAVE
2090|    MsgBox "All calibration data completion."
2100|
2110|  Cal_Skip:
2120|
2130|    End Sub
```

# Connecting Hard Disk (Shared Folder) of External PC

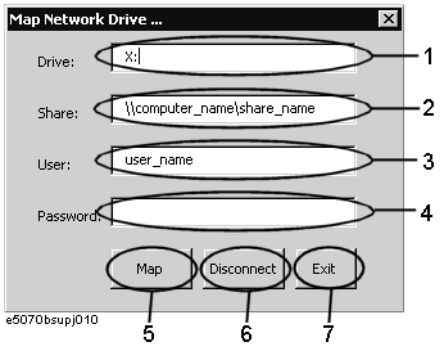
Example 6-2 shows a sample program (VBA program) that demonstrates how to connect a hard disk (a shared folder) of an external PC to the E5061A/E5062A. You can find the source file of this program, named “map_drive.vba”, on the sample program disk. This VBA program consists of the following modules:

| Object name | Module type | Description |
|---|---|---|
| frmMapDrive | User form | Connects or disconnects a hard disk. |
| Module1 | Standard module | Displays frmMapDrive. |

## Using VBA program

**Step 1.** Load the map_drive.vba and press **[Macro Run]**. The following macro appears.

**Figure 6-2 Shared folder connection macro**




### Step 2. Connecting (Mapping)

Enter the drive letter for the shared folder (1 in Figure 6-2), the share name of the shared folder (2 in Figure 6-2), the user name (3 in Figure 6-2) and the password (4 in Figure 6-2) in the external PC. And then click the **Map** button (5 in Figure 6-2).

**NOTE** Consult your network administrator and enter the settings in the same way as the Windows 2000® PC. If you enter an incorrect setting, an error occurs and the program is interrupted.

### Disconnecting

Enter the drive letter for the shared folder (1 in Figure 6-2), and then click the **Disconnect** button (6 in Figure 6-2).

**Step 3.** Click the **Exit** button (7 in Figure 6-2) to exit from the program.

## Description of operation in VBA program

The program (object name: frmMapDrive) is described in detail below:

### Sub CommandButton1_Click

This procedure is called when the user clicks the **Map** button. This procedure checks if the drive letter is used using the IsDriveNameInUse procedure. And then this procedure connects the shared folder using the MapDrive procedure if the drive letter is not used, or displays a message to show the drive letter is used if the drive letter is used.

### Sub CommandButton2_Click

This procedure is called when the user clicks the **Disconnect** button. This procedure disconnects the shared folder using the DisconnectDrive procedure.

### Function IsDriveNameInUse

This procedure checks if the txtDrive.Text (the drive letter specified by 1 in Figure 6-2) is used.

### Sub MapDrive

This procedure connects the shared folder as the txtDrive.Text (the drive letter specified by 1 in Figure 6-2) drive using the parameters: txtShare.Text (the share name specified by 2 in Figure 6-2), txtUser.Text (the user name specified by 3 in Figure 6-2), and txtPasswd.Text (the password specified by 4 in Figure 6-2).

### Sub DisconnectDrive

This procedure disconnects the txtDrive.Text (the drive letter specified by 1 in Figure 6-2) drive.

### Sub CommandButton3_Click

This procedure is called when the user clicks the **Exit** button. This procedure ends the program.

**Example 6-2** **Connecting a hard disk of external PC (Object name: frmMapDrive)**

```
Private Sub CommandButton1_Click()
    If Not IsDriveNameInUse Then
        Call MapDrive
    Else
       MsgBox "Drive """ & txtDrive.Text & """ is Already used", vb
Critical
    End If
End Sub

Private Sub CommandButton2_Click()
    Call DisconnectDrive
End Sub

Private Function IsDriveNameInUse() As Boolean
    Set fso = CreateObject("Scripting.FileSystemObject")
    IsDriveNameInUse = fso.DriveExists(txtDrive.Text)
End Function

Private Sub MapDrive()
    Set network = CreateObject("wscript.network")
   Call network.MapNetworkDrive(txtDrive.Text, txtShare.Text, vbFal
se, txtUser. Text, txtPasswd.Text)
End Sub

Private Sub DisconnectDrive()
    Set network = CreateObject("wscript.network")
    network.RemoveNetworkDrive txtDrive.Text
End Sub

Private Sub CommandButton3_Click()
    Unload Me
End Sub
```

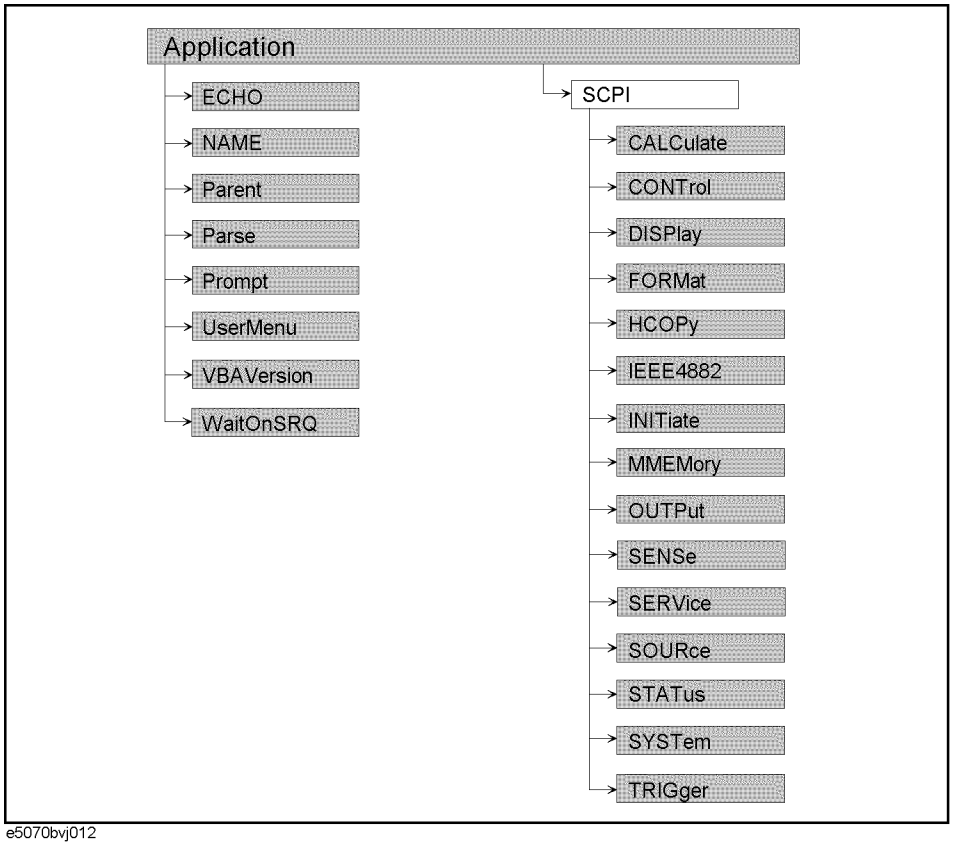
# 7 COM Object Reference

This chapter describes the COM object model of the Agilent E5061A/E5062A and the COM object reference in alphabetical order. If you want to look up COM objects by corresponding front panel keys, see “COM object list by front panel key.”

# COM Object Model

The COM objects provided for the E5061A/E5062A are structured hierarchically as shown in Figure 7-1.

**Figure 7-1** **E5061A/E5062A COM object model**




# Application Objects

The Application objects are at the top of the hierarchy of the E5061A/E5062A COM object model. They consist of 7 objects dedicated to the COM interface and SCPI objects corresponding to SCPI commands. For information on the basic use of the 7 objects dedicated to the COM interface, see “Application Objects” on page 100.

# SCPI Objects

The SCPI objects are created to realize the SCPI commands of the E5061A/E5062A with the COM interface. For information on the basic use of the SCPI objects, see “SCPI Objects” on page 101.

The conversion rules from the SCPI commands when writing SCPI object messages 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?"<br>ENTER 717;A$ | → | A = SCPI.SENSe(1).CORRection.COLLect.METHod:TYPE |
| OUTPUT 717;"*CLS" | → | SCPI.IEEE4882.CLS |

# COM Object List

## List by Front Panel Key

Table 7-1 shows the COM objects that correspond to the front panel keys (in alphabetical order).

**Table 7-1 Front panel key tree vs. COM objects correspondence table**

| Front panel key (Operation) | | | | Corresponding COM object |
|---|---|---|---|---|
| [Analysis] | Bandwidth Limit | BW Display | | SCPI.CALCulate(Ch).SELected.BLIMit.DISPlay.VALue on page 131 |
| | | BW Marker | | SCPI.CALCulate(Ch).SELected.BLIMit.DISPlay.MARKer on page 130 |
| | | BW Test | | SCPI.CALCulate(Ch).SELected.BLIMit.STATe on page 136 |
| | | Fail Sign | | SCPI.DISPlay.FSIGn on page 242 |
| | | Max Bandwidth | | SCPI.CALCulate(Ch).SELected.BLIMit.MAXimum on page 133 |
| | | Min Bandwidth | | SCPI.CALCulate(Ch).SELected.BLIMit.MINimum on page 134 |
| | | N dB Points | | SCPI.CALCulate(Ch).SELected.BLIMit.DB on page 129 |
| | Conversion | Conversion | | SCPI.CALCulate(Ch).SELected.CONVersion.STATe on page 138 |
| | | Function | | SCPI.CALCulate(Ch).SELected.CONVersion.FUNCtion on page 137 |
| | Limit Test | Clip Lines | | SCPI.CALCulate(Ch).SELected.LIMit.DISPlay.CLIP on page 160 |
| | | Edit Limit Line | Add / Delete / Clear Limit Table | SCPI.CALCulate(Ch).SELected.LIMit.DATA on page 158 |
| | | | Export to CSV File | SCPI.MMEMory.STORe.LIMit on page 296 |
| | | | Import from CSV File | SCPI.MMEMory.LOAD.LIMit on page 287 |
| | | Fail Sign | | SCPI.DISPlay.FSIGn on page 242 |
| | | Limit Line | | SCPI.CALCulate(Ch).SELected.LIMit.DISPlay.STATe on page 161 |
| | | Limit Line Offset | Ampletude Offset | SCPI.CALCulate(Ch).SELected.LIMit.OFFSet.AMPLitude on page 163 |
| | | | Marker -> Amplitude Offset | SCPI.CALCulate(Ch).SELected.LIMit.OFFSet.MARKer on page 164 |
| | | | Stimulus Offset | SCPI.CALCulate(Ch).SELected.LIMit.OFFSet.STIMulus on page 165 |
| | | Limit Test | | SCPI.CALCulate(Ch).SELected.LIMit.STATe on page 169 |
| | Ripple Limit | Edit Limit Line | Add / Delete / Clear Limit Table | SCPI.CALCulate(Ch).SELected.RLIMit.DATA on page 211 |
| | | | Export to CSV File | SCPI.MMEMory.STORe.RLIMit on page 297 |
| | | | Import from CSV File | SCPI.MMEMory.LOAD.RLIMit on page 288 |
| | | Fail Sign | | SCPI.DISPlay.FSIGn on page 242 |
| | | Ripple Limit | | SCPI.CALCulate(Ch).SELected.RLIMit.DISPlay.LINE on page 213 |
| | | Ripple Limit Test | | SCPI.CALCulate(Ch).SELected.RLIMit.STATe on page 218 |
| | | Ripple Value | | SCPI.CALCulate(Ch).SELected.RLIMit.DISPlay.VALue on page 215 |
| | | Ripple Band | | SCPI.CALCulate(Ch).SELected.RLIMit.DISPlay.SELect on page 214 |

**Table 7-1 Front panel key tree vs. COM objects correspondence table**

| Front panel key (Operation) | | | | | Corresponding COM object |
|---|---|---|---|---|---|
| [Avg] | Averaging | | | | SCPI.SENSe(Ch).AVERage.STATe on page 304 |
| | Averaging Restart | | | | SCPI.SENSe(Ch).AVERage.CLEar on page 303 |
| | Avg Factor | | | | SCPI.SENSe(Ch).AVERage.COUNt on page 303 |
| | Smo Aperture | | | | SCPI.CALCulate(Ch).SELected.SMOothing.APERture on page 219 |
| | Smoothing | | | | SCPI.CALCulate(Ch).SELected.SMOothing.STATe on page 220 |
| | IF Bandwidth | | | | SCPI.SENSe(Ch).BANDwidth.RESolution on page 305<br>SCPI.SENSe(Ch).BWIDth.RESolution on page 306 |
| [Cal] | Cal Kit | | | | SCPI.SENSe(Ch).CORRection.COLLect.CKIT.SELect on page 319 |
| | Calibrate | 1-Port Cal | Done | | SCPI.SENSe(Ch).CORRection.COLLect.SAVE on page 345 |
| | | | Load | | SCPI.SENSe(Ch).CORRection.COLLect.ACQuire.LOAD on page 311 |
| | | | Open | | SCPI.SENSe(Ch).CORRection.COLLect.ACQuire.OPEN on page 312 |
| | | | Port | | SCPI.SENSe(Ch).CORRection.COLLect.METHod. SOLT1 on page 342 |
| | | | Short | | SCPI.SENSe(Ch).CORRection.COLLect.ACQuire. SHORt on page 312 |
| | | 2-Port Cal | Done | | SCPI.SENSe(Ch).CORRection.COLLect.SAVE on page 345 |
| | | | Isolation (Optional) | Port 1-2 Isol | SCPI.SENSe(Ch).CORRection.COLLect.ACQuire. ISOLation on page 310 |
| | | | Reflection | Port n Load | SCPI.SENSe(Ch).CORRection.COLLect.ACQuire.LOAD on page 311 |
| | | | | Port n Open | SCPI.SENSe(Ch).CORRection.COLLect.ACQuire.OPEN on page 312 |
| | | | | Port n Short | SCPI.SENSe(Ch).CORRection.COLLect.ACQuire. SHORt on page 312 |
| | | | Transmission | Port 1-2 Thru | SCPI.SENSe(Ch).CORRection.COLLect.ACQuire.THRU on page 313 |

**Table 7-1 Front panel key tree vs. COM objects correspondence table**

| Front panel key (Operation) | | | | | Corresponding COM object |
|---|---|---|---|---|---|
| [Cal] (Continued) | Calibrate (Continued) | Enhanced Response | Done | | SCPI.SENSe(Ch).CORRection.COLLect.SAVE on page 345 |
| | | | Isolation (Optional) | | SCPI.SENSe(Ch).CORRection.COLLect.ACQuire. ISOLation on page 310 |
| | | | Open | | SCPI.SENSe(Ch).CORRection.COLLect.ACQuire.OPEN on page 312 |
| | | | Ports | | SCPI.SENSe(Ch).CORRection.COLLect.METHod. ERESponse on page 340 |
| | | | Short | | SCPI.SENSe(Ch).CORRection.COLLect.ACQuire. SHORt on page 312 |
| | | | Thru | | SCPI.SENSe(Ch).CORRection.COLLect.ACQuire.THRU on page 313 |
| | | Response (Open) | Done | | SCPI.SENSe(Ch).CORRection.COLLect.SAVE on page 345 |
| | | | Load (Optional) | | SCPI.SENSe(Ch).CORRection.COLLect.ACQuire.LOAD on page 311 |
| | | | Open | | SCPI.SENSe(Ch).CORRection.COLLect.ACQuire.OPEN on page 312 |
| | | | Port | | SCPI.SENSe(Ch).CORRection.COLLect.METHod. RESPonse.OPEN on page 341 |
| | | Response (Short) | Done | | SCPI.SENSe(Ch).CORRection.COLLect.SAVE on page 345 |
| | | | Load (Optional) | | SCPI.SENSe(Ch).CORRection.COLLect.ACQuire.LOAD on page 311 |
| | | | Port | | SCPI.SENSe(Ch).CORRection.COLLect.METHod. RESPonse.SHORt on page 341 |
| | | | Short | | SCPI.SENSe(Ch).CORRection.COLLect.ACQuire. SHORt on page 312 |
| | | Response (Thru) | Done | | SCPI.SENSe(Ch).CORRection.COLLect.SAVE on page 345 |
| | | | Isolation (Optional) | | SCPI.SENSe(Ch).CORRection.COLLect.ACQuire. ISOLation on page 310 |
| | | | Ports | | SCPI.SENSe(Ch).CORRection.COLLect.METHod. RESPonse.THRU on page 342 |
| | | | Thru | | SCPI.SENSe(Ch).CORRection.COLLect.ACQuire.THRU on page 313 |
| | Clear All | OK | | | SCPI.SENSe(Ch).CORRection.CLEar on page 307 |
| | | Cancel | | | N/A |
| | Correction | | | | SCPI.SENSe(Ch).CORRection.STATe on page 351 |
| | ECal | 1-Port Cal | | | SCPI.SENSe(Ch).CORRection.COLLect.ECAL.SOLT1 on page 337 |
| | | 2-Port Cal | | | SCPI.SENSe(Ch).CORRection.COLLect.ECAL.SOLT2 on page 338 |
| | | Ecal | | | SCPI.SENSe(Ch).CORRection.COLLect.ECAL.ERESponse on page 334 |
| | | Isolation | | | SCPI.SENSe(Ch).CORRection.COLLect.ECAL.ISOLation.STATe on page 335 |
| | | Thru Cal | | | SCPI.SENSe(Ch).CORRection.COLLect.ECAL.THRU on page 339 |
| | Modify Cal Kit | Define STDs | 1. XXXX to 21. XXXX | Arb. Impedance | SCPI.SENSe(Ch).CORRection.COLLect.CKIT.STAN(Std).ARBitrary on page 320 |
| | | | | C0 | SCPI.SENSe(Ch).CORRection.COLLect.CKIT.STAN(Std).C0 on page 321 |
| | | | | C1 | SCPI.SENSe(Ch).CORRection.COLLect.CKIT.STAN(Std).C1 on page 322 |
| | | | | C2 | SCPI.SENSe(Ch).CORRection.COLLect.CKIT.STAN(Std).C2 on page 323 |
| | | | | C3 | SCPI.SENSe(Ch).CORRection.COLLect.CKIT.STAN(Std).C3 on page 324 |
| | | | | L0 | SCPI.SENSe(Ch).CORRection.COLLect.CKIT.STAN(Std).L0 on page 326 |
| | | | | L1 | SCPI.SENSe(Ch).CORRection.COLLect.CKIT.STAN(Std).L1 on page 327 |

**Table 7-1 Front panel key tree vs. COM objects correspondence table**

| Front panel key (Operation) | | | | | Corresponding COM object |
|---|---|---|---|---|---|
| **[Cal]** (Continued) | **Modify Cal Kit** (Continued) | **Define STDs** (Continued) | **1. XXXX to 21. XXXX** (Continued) | **L2** | SCPI.SENSe(Ch).CORRection.COLLect.CKIT.STAN(Std).L2 on page 328 |
| | | | | **L3** | SCPI.SENSe(Ch).CORRection.COLLect.CKIT.STAN(Std).L3 on page 329 |
| | | | | **Label** | SCPI.SENSe(Ch).CORRection.COLLect.CKIT.STAN(Std).LABel on page 330 |
| | | | | **Offset Delay** | SCPI.SENSe(Ch).CORRection.COLLect.CKIT.STAN(Std).DELay on page 325 |
| | | | | **Offset Loss** | SCPI.SENSe(Ch).CORRection.COLLect.CKIT.STAN(Std).LOSS on page 331 |
| | | | | **Offset Z0** | SCPI.SENSe(Ch).CORRection.COLLect.CKIT.STAN(Std).Z0 on page 333 |
| | | | | **STD Type** | SCPI.SENSe(Ch).CORRection.COLLect.CKIT.STAN(Std).TYPE on page 332 |
| | | **Label Kit** | | | SCPI.SENSe(Ch).CORRection.COLLect.CKIT.LABel on page 314 |
| | | **Specify CLSs** | **Load** | | SCPI.SENSe(Ch).CORRection.COLLect.CKIT.ORDer. LOAD(Cpt) on page 315 |
| | | | **Open** | | SCPI.SENSe(Ch).CORRection.COLLect.CKIT.ORDer. OPEN(Cpt) on page 316 |
| | | | **Short** | | SCPI.SENSe(Ch).CORRection.COLLect.CKIT.ORDer. SHORt(Cpt) on page 317 |
| | | | **Thru** | | SCPI.SENSe(Ch).CORRection.COLLect.CKIT.ORDer. THRU(Cpt_m,Cpt_n) on page 318 |
| | **Port Extensions** | **Extension Port 1** | | | SCPI.SENSe(Ch).CORRection.EXTension.PORT(Pt).TIME on page 346 |
| | | **Extension Port 2** | | | |
| | | **Extensions** | | | SCPI.SENSe(Ch).CORRection.EXTension.STATe on page 347 |
| | **Property** | | | | SCPI.SENSe(Ch).CORRection.PROPerty on page 349 |
| | **Velocity Factor** | | | | SCPI.SENSe(Ch).CORRection.RVELocity.COAX on page 350 |
| **[Center]** | | | | | SCPI.SENSe(Ch).FREQuency.CENTer on page 353<br>SCPI.SOURce(Ch).POWer.CENTer on page 373 |
| **[Channel Prev]** | | | | | SCPI.DISPlay.WINDow(Ch).ACTivate on page 250 |
| **[Channel Max]** | | | | | SCPI.DISPlay.MAXimize on page 244 |
| **[Channel Next]** | | | | | SCPI.DISPlay.WINDow(Ch).ACTivate on page 250 |
| **[Display]** | **Allocate Channels** | | | | SCPI.DISPlay.SPLit on page 246 |
| | **Allocate Traces** | | | | SCPI.DISPlay.WINDow(Ch).SPLit on page 255 |
| | **Data - > Mem** | | | | SCPI.CALCulate(Ch).SELected.MATH.MEMorize on page 209 |
| | **Data Math** | | | | SCPI.CALCulate(Ch).SELected.MATH.FUNCtion on page 208 |
| | **Display** | | | | SCPI.DISPlay.WINDow(Ch).TRACe(Tr).STATe on page 262<br>SCPI.DISPlay.WINDow(Ch).TRACe(Tr).MEMory. STATe on page 261 |
| | **Edit Title Label** | | | | SCPI.DISPlay.WINDow(Ch).TITLe.DATA on page 256 |
| | **Frequency** | | | | SCPI.DISPlay.ANNotation.FREQuency.STATe on page 232 |
| | **Graticule Label** | | | | SCPI.DISPlay.WINDow(Ch).LABel on page 253 |
| | **Graticule Label** | | | | SCPI.DISPlay.WINDow(Ch).TRACe(Tr).ANNotation. YAXis.MODE on page 260 |
| | **Invert Color** | | | | SCPI.DISPlay.IMAGe on page 243 |
| | **Num of Traces** | | | | SCPI.CALCulate(Ch).PARameter.COUNt on page 126 |
| | **Title Label** | | | | SCPI.DISPlay.WINDow(Ch).TITLe.STATe on page 257 |
| | **Update** | | | | SCPI.DISPlay.ENABle on page 241 |

**Table 7-1 Front panel key tree vs. COM objects correspondence table**

| Front panel key (Operation) | | | Corresponding COM object |
|---|---|---|---|
| [Format] | | | SCPI.CALCulate(Ch).SELected.FORMat on page 145 |
| [Macro Break] | | | N/A |
| [Macro Run] | | | N/A |
| [Macro Setup] | Clear Echo | | SCPI.DISPlay.ECHO.CLEar on page 240 |
| | Close Editor | | N/A |
| | Continue | | N/A |
| | Echo Window | | SCPI.DISPlay.TABLe.STATe on page 248<br>SCPI.DISPlay.TABLe.TYPE on page 249 |
| | Load & Run | | N/A |
| | Load Project | | N/A |
| | New Project | | N/A |
| | Preset User Menu | | UserMenu.PRESet on page 121 |
| | Save Project | | N/A |
| | Select Macro | | N/A |
| | Stop | | N/A |
| | User Menu | | UserMenu.Press(Key_id) on page 122 |
| | VBA Editor | | N/A |
| [Marker] | Clear Marker Menu | | SCPI.CALCulate(Ch).SELected.MARKer(Mk).STATe on page 205 |
| | Marker 1 to Marker 4 | | SCPI.CALCulate(Ch).SELected.MARKer(Mk).STATe on page 205<br>SCPI.CALCulate(Ch).SELected.MARKer(Mk).ACTivate on page 170<br>SCPI.CALCulate(Ch).SELected.MARKer(Mk).X on page 206 |
| | Marker - > Ref Marker | | N/A |
| | More Markers | Marker 5 to Marker 9 | SCPI.CALCulate(Ch).SELected.MARKer(Mk).STATe on page 205<br>SCPI.CALCulate(Ch).SELected.MARKer(Mk).ACTivate on page 170<br>SCPI.CALCulate(Ch).SELected.MARKer(Mk).X on page 206 |
| | Ref Marker | | SCPI.CALCulate(Ch).SELected.MARKer(Mk).STATe on page 205<br>SCPI.CALCulate(Ch).SELected.MARKer(Mk).ACTivate on page 170<br>SCPI.CALCulate(Ch).SELected.MARKer(Mk).X on page 206<br>SCPI.CALCulate(Ch).SELected.MARKer.REFerence. STATe on page 203 |
| | Ref Marker Mode | | SCPI.CALCulate(Ch).SELected.MARKer.REFerence. STATe on page 203 |

**Table 7-1 Front panel key tree vs. COM objects correspondence table**

| Front panel key (Operation) | | | Corresponding COM object |
|---|---|---|---|
| [Marker Func] | Annotation Options | Active Only | SCPI.DISPlay.WINDow(Ch).ANNotation.MARKer.SINGle.STATe on page 252 |
| | | Align | SCPI.DISPlay.WINDow(Ch).ANNotation.MARKer.ALIGn.STATe on page 251 |
| | | Marker Info X Pos | SCPI.DISPlay.WINDow(Ch).TRACe(Tr).ANNotation. MARKer. POSition.X on page 258 |
| | | Marker Info Y Pos | SCPI.DISPlay.WINDow(Ch).TRACe(Tr).ANNotation. MARKer. POSition.Y on page 259 |
| | Couple | | SCPI.CALCulate(Ch).SELected.MARKer.COUPle on page 174 |
| | Discrete | | SCPI.CALCulate(Ch).SELected.MARKer(Mk).DISCrete on page 175 |
| | Flatness | | SCPI.CALCulate(Ch).SELected.MARKer.MATH.FLATness.STATe on page 195<br>SCPI.CALCulate(Ch).SELected.MARKer.MATH.FLATness.DATA on page 194 |
| | Marker Table | | SCPI.DISPlay.TABLe.STATe on page 248<br>SCPI.DISPlay.TABLe.TYPE on page 249 |
| | Marker - > Center | | SCPI.CALCulate(Ch).SELected.MARKer(Mk).SET on page 204 |
| | Marker - > Delay | | |
| | Marker - > Reference | | |
| | Marker - > Start | | |
| | Marker - > Stop | | |
| | RF Filter Stats | | SCPI.CALCulate(Ch).SELected.MARKer.MATH.FSTatistics.STATe on page 197<br>SCPI.CALCulate(Ch).SELected.MARKer.MATH.FSTatistics.DATA on page 196 |
| | Statistics | | SCPI.CALCulate(Ch).SELected.MSTatistics.STATe on page 210<br>SCPI.CALCulate(Ch).SELected.MSTatistics.DATA on page 209 |

**Table 7-1 Front panel key tree vs. COM objects correspondence table**

| Front panel key (Operation) | | | Corresponding COM object |
|---|---|---|---|
| [Marker Search] | Bandwidth | | SCPI.CALCulate(Ch).SELected.MARKer.BWIDth.STATe on page 172<br>SCPI.CALCulate(Ch).SELected.MARKer(Mk).BWIDth. DATA on page 171 |
| | Bandwidth Value | | SCPI.CALCulate(Ch).SELected.MARKer(Mk).BWIDth. THReshold on page 173 |
| | Max | | SCPI.CALCulate(Ch).SELected.MARKer(Mk).FUNCtion. TYPE on page 192<br>SCPI.CALCulate(Ch).SELected.MARKer(Mk).FUNCtion. EXECute on page 180 |
| | Min | | |
| | Multi Peak | Peak Excursion | SCPI.CALCulate(Ch).SELected.MARKer.FUNCtion.MULTi.PEXCursion on page 181 |
| | | Peak Polarity | SCPI.CALCulate(Ch).SELected.MARKer.FUNCtion.MULTi.PPOLarity on page 182 |
| | | Search Multi Peak | SCPI.CALCulate(Ch).SELected.MARKer.FUNCtion.MULTi.TYPE on page 186 |
| | Multi Target | Search Multi Target | SCPI.CALCulate(Ch).SELected.MARKer.FUNCtion.MULTi.TYPE on page 186 |
| | | Target Transition | SCPI.CALCulate(Ch).SELected.MARKer.FUNCtion.MULTi.TTRansition on page 185 |
| | | Target Value | SCPI.CALCulate(Ch).SELected.MARKer.FUNCtion.MULTi.TARGet on page 183 |
| | Notch | | SCPI.CALCulate(Ch).SELected.MARKer.NOTCh.STATe on page 201 |
| | Notch Value | | SCPI.CALCulate(Ch).SELected.MARKer(Mk).NOTCh. THReshold on page 202 |
| | Peak | Peak Excursion | SCPI.CALCulate(Ch).SELected.MARKer(Mk).FUNCtion. PEXCursion on page 187 |
| | | Peak Polarity | SCPI.CALCulate(Ch).SELected.MARKer(Mk).FUNCtion. PPOLarity on page 188 |
| | | Search Left | SCPI.CALCulate(Ch).SELected.MARKer(Mk).FUNCtion. TYPE on page 192<br>SCPI.CALCulate(Ch).SELected.MARKer(Mk).FUNCtion. EXECute on page 180 |
| | | Search Peak | |
| | | Search Right | |
| | Search Range | Couple | SCPI.CALCulate(Ch).SELected.MARKer.FUNCtion. DOMain.COUPle on page 176 |
| | | Search Range | SCPI.CALCulate(Ch).SELected.MARKer.FUNCtion. DOMain.STATe on page 178 |
| | | Start | SCPI.CALCulate(Ch).SELected.MARKer.FUNCtion. DOMain.STARt on page 177 |
| | | Stop | SCPI.CALCulate(Ch).SELected.MARKer.FUNCtion. DOMain.STOP on page 179 |
| | Target | Search Left | SCPI.CALCulate(Ch).SELected.MARKer(Mk).FUNCtion. TYPE on page 192<br>SCPI.CALCulate(Ch).SELected.MARKer(Mk).FUNCtion. EXECute on page 180 |
| | | Search Right | |
| | | Search Target | |
| | | Target Transition | SCPI.CALCulate(Ch).SELected.MARKer(Mk).FUNCtion. TTRansition on page 191 |
| | | Target Value | SCPI.CALCulate(Ch).SELected.MARKer(Mk).FUNCtion. TARGet on page 189 |
| | Tracking | | SCPI.CALCulate(Ch).SELected.MARKer(Mk).FUNCtion. TRACking on page 190<br>SCPI.CALCulate(Ch).SELected.MARKer.FUNCtion.MULTi.TRACking on page 184 |

**Table 7-1 Front panel key tree vs. COM objects correspondence table**

| Front panel key (Operation) | | | Corresponding COM object |
|---|---|---|---|
| [Meas] | | | SCPI.CALCulate(Ch).PARameter(Tr).DEFine on page 127 |
| [Preset] | OK | | SCPI.SYSTem.PRESet on page 422<br>SCPI.SYSTem.UPReset on page 426 |
| [Save/ Recall] | Channel/Trace | | SCPI.MMEMory.STORe.SALL on page 298 |
| | Explorer | | N/A |
| | Recall Channel | Cal Only A - Cal Only D | SCPI.MMEMory.LOAD.CHANnel.COEFficient on page 285 |
| | | State A - State D | SCPI.MMEMory.LOAD.CHANnel.STATe on page 286 |
| | Recall State | | SCPI.MMEMory.LOAD.STATe on page 290 |
| | Save Channel | Cal Only A - Cal Only D | SCPI.MMEMory.STORe.CHANnel.COEFficient on page 292 |
| | | Clear States | SCPI.MMEMory.STORe.CHANnel.CLEar on page 292 |
| | | State A - State D | SCPI.MMEMory.STORe.CHANnel.STATe on page 293 |
| | Save State | | SCPI.MMEMory.STORe.STATe on page 300 |
| | Save Trace Data | | SCPI.MMEMory.STORe.FDATa on page 294 |
| | Save Type | | SCPI.MMEMory.STORe.STYPe on page 301 |
| [Scale] | Auto Scale | | SCPI.DISPlay.WINDow(Ch).TRACe(Tr).Y.SCALe.AUTO on page 262 |
| | Auto Scale All | | N/A |
| | Divisions | | SCPI.DISPlay.WINDow(Ch).Y.SCALe.DIVisions on page 269 |
| | Electrical Delay | | SCPI.CALCulate(Ch).SELected.CORRection.EDELay. TIME on page 139 |
| | Marker - > Reference | | SCPI.CALCulate(Ch).SELected.MARKer(Mk).SET on page 204 |
| | Phase Offset | | SCPI.CALCulate(Ch).SELected.CORRection.OFFSet. PHASe on page 140 |
| | Reference Position | | SCPI.DISPlay.WINDow(Ch).TRACe(Tr).Y.SCALe. RPOSition on page 265 |
| | Reference Tracking | Tracking | SCPI.DISPlay.WINDow(Ch).TRACe(Tr).Y. TRACk. MODE on page 267 |
| | | Track Frequency | SCPI.DISPlay.WINDow(Ch).TRACe(Tr).Y. TRACk. FREQuency on page 266 |
| | Reference Value | | SCPI.DISPlay.WINDow(Ch).TRACe(Tr).Y.SCALe.RLEVel on page 264 |
| | Scale/Div | | SCPI.DISPlay.WINDow(Ch).TRACe(Tr).Y.SCALe. PDIVision on page 263 |
| [Softkey On/Off] | | | SCPI.DISPlay.SKEY.STATe on page 245 |
| [Span] | | | SCPI.SENSe(Ch).FREQuency.SPAN on page 357<br>SCPI.SOURce(Ch).POWer.SPAN on page 379 |
| [Start] | | | SCPI.SENSe(Ch).FREQuency.STARt on page 358<br>SCPI.SOURce(Ch).POWer.STARt on page 380 |
| [Stop] | | | SCPI.SENSe(Ch).FREQuency.STOP on page 359<br>SCPI.SOURce(Ch).POWer.STOP on page 381 |
| [Sweep Setup] | Edit Segment Table | | SCPI.SENSe(Ch).SEGMent.DATA on page 361 |
| | Edit Segment Table | Export to CSV File | SCPI.MMEMory.STORe.SEGMent on page 299 |
| | | Import from CSV File | SCPI.MMEMory.LOAD.SEGMent on page 289 |
| | Points | | SCPI.SENSe(Ch).SWEep.POINts on page 365 |
| | Power | CW Freq | SCPI.SENSe(Ch).FREQuency.CW on page 354<br>SCPI.SENSe(Ch).FREQuency.FIXed on page 356 |
| | | Port Couple | SCPI.SOURce(Ch).POWer.PORT.COUPle on page 377 |
| | | Port Power | SCPI.SOURce(Ch).POWer.PORT(Pt).LEVel.IMMediate. AMPLitude on page 378 |
| | | Power | SCPI.SOURce(Ch).POWer.LEVel.IMMediate. AMPLitude on page 374 |

**Table 7-1 Front panel key tree vs. COM objects correspondence table**

| Front panel key (Operation) | | | Corresponding COM object |
|---|---|---|---|
| **[Sweep Setup]** (Continued) | **Power** (Continued) | **Power Ranges** | SCPI.SOURce(Ch).POWer.ATTenuation.DATA on page 372 |
| | | **RF Out** | SCPI.OUTPut.STATe on page 302 |
| | | **Slope [ON/OFF]** | SCPI.SOURce(Ch).POWer.LEVel.SLOPe.STATe on page 376 |
| | | **Slope [xx dB/GHz]** | SCPI.SOURce(Ch).POWer.LEVel.SLOPe.DATA on page 375 |
| | **Segment Display** | | SCPI.DISPlay.WINDow(Ch).X.SPACing on page 268 |
| | **Sweep Delay** | | SCPI.SENSe(Ch).SWEep.DELay on page 364 |
| | **Sweep Time** | | SCPI.SENSe(Ch).SWEep.TIME.DATA on page 367<br>SCPI.SENSe(Ch).SWEep.TIME.AUTO on page 366 |
| | **Sweep Type** | | SCPI.SENSe(Ch).SWEep.TYPE on page 368 |

**Table 7-1 Front panel key tree vs. COM objects correspondence table**

| Front panel key (Operation) | | | | Corresponding COM object |
|---|---|---|---|---|
| [System] | 87050/75 Setup | 87050/75 Control | | SCPI.SENSe.MULTiport.STATe |
| | | Property | | SCPI.SENSe.MULTiport.PROPerty |
| | | Reflection | | SCPI.SENSe.MULTiport.PORT1 |
| | | Transmission | | SCPI.SENSe.MULTiport.PORT2 |
| | Abort Printing | | | SCPI.HCOPy.ABORt on page 272 |
| | Backlight | | | SCPI.SYSTem.BACKlight on page 415 |
| | Dump Screen Image | | | SCPI.MMEMory.STORe.IMAGe on page 295 |
| | Firmware Revision | | | SCPI.IEEE4882.IDN on page 275 |
| | Invert Image | | | SCPI.HCOPy.IMAGe on page 272 |
| | Misc Setup | Beeper | Beep Complete | SCPI.SYSTem.BEEPer.COMPlete.STATe on page 416 |
| | | | Beep Warning | SCPI.SYSTem.BEEPer.WARNing.STATe on page 417 |
| | | | Test Beep Complete | SCPI.SYSTem.BEEPer.COMPlete.IMMediate on page 416 |
| | | | Test Beep Warning | SCPI.SYSTem.BEEPer.WARNing.IMMediate on page 417 |
| | | Clock Setup | Set Date and Time | SCPI.SYSTem.DATE on page 418<br>SCPI.SYSTem.TIME on page 425 |
| | | | Show Clock | SCPI.DISPlay.CLOCk on page 233 |
| | | Color Setup | | SCPI.DISPlay.COLor(Dnum).TRACe(Tr).DATA on page 238<br>SCPI.DISPlay.COLor(Dnum).TRACe(Tr).MEMory on page 239<br>SCPI.DISPlay.COLor(Dnum).GRATicule(Gnum) on page 235<br>SCPI.DISPlay.COLor(Dnum).LIMit(Lnum) on page 236<br>SCPI.DISPlay.COLor(Dnum).BACK on page 234<br>SCPI.DISPlay.COLor(Dnum).RESet on page 237 |
| | | Control Panel... | | N/A |
| | | GPIB Setup | System Controller Configuration | N/A |
| | | | Talker/Listener Address | N/A |
| | | Key Lock | Front Panel & Keyboard Lock | SCPI.SYSTem.KLOCk.KBD on page 420 |
| | | | Touch Screen & Mouse Lock | SCPI.SYSTem.KLOCk.MOUSe on page 421 |
| | | Network Setup | LAN Dialog... | N/A |
| | | | Network Configuration | N/A |
| | | | Network Identification | N/A |
| | | | SICL-LAN Address | N/A |
| | | | SICL-LAN Server | N/A |
| | | | Telnet Server | N/A |
| | | | VNC Server Configuration... | N/A |
| | | | Web Server | N/A |
| | | Preset Setup | Confirm | N/A |
| | | | State | N/A |

**Table 7-1 Front panel key tree vs. COM objects correspondence table**

| Front panel key (Operation) | | | Corresponding COM object |
|---|---|---|---|
| [System] (Continued) | Print | | SCPI.HCOPy.IMMediate on page 273 |
| | Printer Setup | | N/A |
| | Service Menu | Security Level | SCPI.SYSTem.SECurity.LEVel on page 423 |
| [Trace Prev] | | | SCPI.CALCulate(Ch).PARameter(Tr).SELect on page 128 |
| [Trace Max] | | | SCPI.DISPlay.WINDow(Ch).MAXimize on page 254 |
| [Trace Next] | | | SCPI.CALCulate(Ch).PARameter(Tr).SELect on page 128 |
| [Trigger] | Continuous | | SCPI.INITiate(Ch).CONTinuous on page 280 |
| | Continuous Disp Channels | | N/A |
| | Hold | | SCPI.ABORt on page 125<br>SCPI.INITiate(Ch).CONTinuous on page 280 |
| | Hold All Channels | | N/A |
| | Restart | | SCPI.ABORt on page 125 |
| | Single | | SCPI.ABORt on page 125<br>SCPI.INITiate(Ch).CONTinuous on page 280<br>SCPI.INITiate(Ch).IMMediate on page 281 |
| | Trigger Source | | SCPI.TRIGger.SEQuence.SOURce on page 429 |
| | Trigger | | SCPI.TRIGger.SEQuence.IMMediate on page 427 |

## Notational Rules of COM Objects

This section describes the rules for the description of the COM objects in this chapter.

## Object Type

Part with heading “Object type” describes the type of the E5061A/E5062A COM object. The E5061A/E5062A provides properties and methods as the types of COM objects. In the E5061A/E5062A COM objects, COM objects to set (send)/read (return) the state of the E5061A/E5062A using variables are defined as property and ones to prompt some kind of processing as method.

## Syntax

Part with heading “Syntax” describes the syntax to send a COM object from the E5061A/E5062A VBA to the E5061A/E5062A. The syntax consists of 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 the following 3 types of syntax for coding using objects.

"Object (property) = *variable*": to set the stat of the E5061A/E5062A.

*variable*=object (property): to read the stat of the E5061A/E5062A.

"Object (method)": to make the E5061A/E5062A perform some processing.

## Description

Part with heading “Description” describes how to use the COM object or the operation when executed. COM objects used only to read the state of the E5061A/E5062A are indicated with “Read only” and ones used only to set the state of the E5061A/E5062A “No read.”

# Variable

Part with heading “Variable” describes necessary variables when using the object. It gives the description, data type, allowable range, preset value, unit, resolution, and notes for *variable (italic)* shown in the syntax.

Variables declared as the string data type (String) are case insensitive. 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 E5061A/E5062A COM objects include 5 types as shown in Table 7-2. Before using variables, declare the data type of each variable. If you do not declare the data type of a variable, it is automatically dealt as the variant type.

**Table 7-2 Data type**

| Data type | Name | Consumed memory | Range |
|---|---|---|---|
| Long | Long integer type | 4 bytes | -2,147,483,648 to 2,147,483,647 |
| Double | Double precision floating point type | 8 bytes | For a negative value:<br>-1.79769313486232E+308 to -4.94065645841247E-324<br>For a positive value:<br>-1.79769313486232E+308 to -4.94065645841247E-324 |
| Boolean | Boolean type | 2 bytes | -1 (True) or 0 (False) |
| String | Character string type *1 | 1 byte/alphanumeric character | Up to approximately 2 billion characters |
| Variant | Variant type | 16 bytes | No limitation |

*1. For a fixed length string, declare the number of characters.

## Examples

Part with heading “Examples” describes a simple example of how to use the object for coding with E5061A/E5062A VBA.

## Related Objects

Part with heading “Related objects” describes related objects when using the object.

## Equivalent Key

Part with heading “Equivalent key” shows the operational procedure of the front panel keys that has the same effect as this object.

**[Key]** Indicates that you press the key named Key.

**[Key] - Item** Indicates a series of key operation in which you press the **[Key]** key, move the focus to the button called Item on the displayed menu using the **[←↓]** key and so on, and then press the **[Enter]** key.

## Application Objects

The Application objects are at the top of the hierarchy of the E5061A/E5062A COM object model. They consist of 7 objects dedicated to the E5061A/E5062A COM interface and SCPI objects corresponding to SCPI commands. This section describes the objects dedicated to the E5061A/E5062A COM interface.

# ECHO

Object type

Method

Syntax

ECHO *V1,V2,···,V10*

ECHO *SCPI object*

Description

Provides display in the echo window. (No read)

There is the following difference from the display with the SCPI.DISPlay.ECHO.DATA object.

- Up to 10 data items can be displayed.
- Data is displayed as the declared data type without a cast.

| | ***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 on page 240

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. “E5061A” or “E5062A” is always read out. (Read only)

Variable

| | ***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

Syntax

Parse(*Scpi*)

*Return* = Parse(*Scpi?*)

Description

Executes an SCPI command of the E5061A/E5062A. For information on the SCPI commands, see Chapter “SCPI Command Reference” in the *E5061A/E5062A Programmer’s Guide*.

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

| | ***Scpi*** |
|---|---|
| Description | SCPI command |
| Data type | Character string type (String) |

| | ***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

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 closed since you need to press the **[Macro Setup] - Continue**. For more information, see “Running a Program from the E5061A/E5062A Measurement Screen” on page 50. If you need to abort the program, see “Stopping with the Dialog Box Appeared” on page 51.

Variable

| | ***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

**Table 7-3 Variable (*Key_id*)**

| | ***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. |

| | ***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 enabled is displayed with the grey color and its softkey cannot be pressed.

Variable

| | ***Status*** |
|---|---|
| Description | Enabled/disabled for the user menu function softkey |
| Data type | Boolean type (Boolean) |
| Range | Select from the following.<br>•True or -1 Makes the softkey enabled.<br>•False or 0 Makes the softkey enabled. |
| Preset value | True or -1 |

For information on the variable (*Key_id*), see Table 7-3, “Variable (Key_id),” on page 119.

Examples

```
Dim KeyEna As Boolean
UserMenu.Item(10).Enabled = False
KeyEna = UserMenu.Item(10).Enabled
```

Related objects

UserMenu.Press(Key_id) on page 122

Equivalent key

No equivalent key is available on the front panel.

## 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*) is pressed. Write the processing in the "UserMenu" object. For more information on its use, see "Executing a Procedure with a Softkey (User Menu Function)" on page 76.

Variable — For information on the variable (*Key_id*), see Table 7-3, "Variable (Key_id)," on page 119.

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.

## UserMenu.PRESet

Object type — Method

Syntax — UserMenu.PRESet

Description — Presets the label name and enabled/disabled settings for the user menu softkeys. (No read)

Examples

```
UserMenu.PRESet
```

Related objects — UserMenu.Item(Key_id).Caption on page 119
UserMenu.Item(Key_id).Enabled on page 120

Equivalent key — **[Macro Setup] - Preset User Menu**

## UserMenu.Press(*Key_id*)

| | |
|---|---|
| Object type | Method |
| 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 Table 7-3, “Variable (Key_id),” on page 119. |
| Examples | `UserMenu.Press(1)` |
| Related objects | UserMenu.Item(Key_id).Enabled on page 120 |
| Equivalent key | **[Macro Setup] - User Menu - Button 1\| Button 2\| Button 3\| Button 4\| Button 5\| Button 6 \| Button 7 \| Button 8 \| Button 9 \| Button 10** |

## UserMenu.Show

| | |
|---|---|
| Object type | Method |
| 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** |

# VBAVersion

Object type — Property

Syntax — *Vers* = VBAVersion

Description — Reads out the version information of VBA installed in the E5061A/E5062A. (Read only)

Variable

| | ***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

Syntax

WaitOnSRQ *Status*, *Timeout*

Description

Suspends the program for specified time until the RQS/MSS bit (bit 6) of the status byte register changes to 1. For information on the structure of the status register, see Appendix "Status Reporting System" in the *E5061A/E5062A Programmer's Guide*. (No read)

Variable

| | ***Status*** |
|---|---|
| Description | State of the RQS/MSS bit (read only) |
| Data type | Boolean type (Boolean) |
| Range | One of the following is returned.<br>•True or -1 — 1 has been received within the specified time.<br>•False or 0 — 1 has not been received within the specified time due to timeout or abort. |

| | ***Timeout*** |
|---|---|
| Description | Timeout 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.

## SCPI Objects

SCPI objects are a collection of the COM interface having one-on-one correspondence with the SCPI commands. This section describes the SCPI objects provided for the E5061A/E5062A.

## SCPI.ABORt

Object type — Method

Syntax — SCPI.ABORt

Description — 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) change into the startup state immediately after the change to the idle state.

For details about the trigger system, see Section “Trigger System” in the *E5061A/E5062A Programmer’s Guide*. (No read)

Examples

```
SCPI.ABORt
```

Related objects — SCPI.INITiate(Ch).IMMediate on page 281

SCPI.INITiate(Ch).CONTinuous on page 280

Equivalent key — **[Trigger]** - **Restart**

# SCPI.CALCulate(*Ch*).PARameter.COUNt

Object type — Property

Syntax — SCPI.CALCulate(*Ch*).PARameter.COUNt = *Value*

*Value* = SCPI.CALCulate(*Ch*).PARameter.COUNt

Description — Sets the number of traces of channels 1 to 4 (*Ch*).

Variable

**Table 7-4 Variable (*Ch*)**

| | ***Ch*** |
|---|---|
| Description | Channel number |
| Data type | Long integer type (Long) |
| Range | 1 to 4 |
| Preset value | 1 |
| Note | If the specified variable is out of the allowable setup range, an error occurs when executed. |

| | ***Value*** |
|---|---|
| Description | Number of traces |
| Data type | Long integer type (Long) |
| Range | 1 to 4 |
| 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**

# SCPI.CALCulate(*Ch*).PARameter(*Tr*).DEFine

Object type — Property

Syntax — SCPI.CALCulate(*Ch*).PARameter(*Tr*).DEFine = *Param*

*Param* = SCPI.CALCulate(*Ch*).PARameter(*Tr*).DEFine

Description — For channels 1 to 4 (*Ch*), sets the measurement parameter of traces 1 to 4 (*Tr*).

Variable

| | ***Param*** |
|---|---|
| Description | Measurement parameter |
| Data type | Character string type (String) |
| Range | Select from the following.<br>•"S11" Specifies S11.<br>•"S21" Specifies S21.<br>•"S12" Specifies S12.<br>•"S22" Specifies S22. |
| Preset value | "S11" |

For information on the variable (*Ch*) and the variable (*Tr*), see Table 7-4, “Variable (Ch),” on page 126 and Table 7-5, “Variable (Tr),” on page 128, respectively.

Examples

```
Dim MeasPara As String
SCPI.CALCulate(1).PARameter(1).DEFine = "s21"
MeasPara = SCPI.CALCulate(1).PARameter(1).DEFine
```

Equivalent key — **[Meas]** - **S11**|**S21**|**S12**|**S22**

# SCPI.CALCulate(*Ch*).PARameter(*Tr*).SELect

Object type — Method

Syntax — SCPI.CALCulate(*Ch*).PARameter(*Tr*).SELect

Description — Sets traces 1 to 4 (*Tr*) of channels 1 to 4 (*Ch*) to the active trace.

You can set only a trace displayed to the active trace. If this object is used to set a trace not displayed to the active trace, an error occurs when executed and the object is ignored. (No read)

Variable

**Table 7-5 Variable (*Tr*)**

| | ***Tr*** |
|---|---|
| Description | Trace number |
| Data type | Long integer type (Long) |
| Range | 1 to 4 |
| Preset value | 1 |
| Note | If the specified variable is out of the allowable setup range, an error occurs when executed. |

For information on the variable (*Ch*), see Table 7-4, “Variable (Ch),” on page 126.

Examples

```
SCPI.CALCulate(2).PARameter(2).SELect
```

Related objects — SCPI.DISPlay.WINDow(Ch).ACTivate on page 250

Equivalent key — **[Trace Prev]** / **[Trace Next]**

# SCPI.CALCulate(*Ch*).SELected.BLIMit.DB

Object type

Property

Syntax

SCPI.CALCulate(*Ch*).SELected.BLIMit.DB = *Value*

*Value* = SCPI.CALCulate(*Ch*).SELected.BLIMit.DB

Description

For channel 1 to channel 4 (specified with the SCPI.CALCulate(Ch).PARameter(Tr).SELect command), sets the bandwidth threshold value (attenuation from the peak) of the bandwidth test.

Variable

| | ***Value*** |
|---|---|
| Description | Bandwidth N dB points. |
| Data type | Double precision floating point type (Double) |
| Range | 0 to 5E8 |
| Preset value | 3 |
| Unit | dB |

For information on the variable (*Ch*), see Table 7-4, “Variable (Ch),” on page 126.

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 on page 128

SCPI.CALCulate(Ch).SELected.BLIMit.STATe on page 136

Equivalent key

**[Analysis] - Bandwidth Limit - N dB Points**

# SCPI.CALCulate(*Ch*).SELected.BLIMit.DISPlay.MARKer

Object type

Property

Syntax

SCPI.CALCulate(*Ch*).SELected.BLIMit.DISPlay.MARKer = *Status*

*Status* = SCPI.CALCulate(*Ch*).SELected.BLIMit.DISPlay.MARKer

Description

For the active trace of channel 1 to channel 4 (specified with the SCPI.CALCulate(Ch).PARameter(Tr).SELect command), turns ON/OFF the marker display of the bandwidth test.

Variable

| | ***Status*** |
|---|---|
| Description | ON/OFF of the bandwidth marker. |
| Data type | Boolean type (Boolean) |
| Range | Select from the following.<br>•True or -1 Turns ON the bandwidth marker.<br>•False or 0 Turns OFF the bandwidth marker. |
| Preset value | False or 0 |

For information on the variable (*Ch*), see Table 7-4, “Variable (Ch),” on page 126.

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 on page 128

SCPI.CALCulate(Ch).SELected.BLIMit.STATe on page 136

SCPI.CALCulate(Ch).SELected.BLIMit.DISPlay.VALue on page 131

Equivalent key

**[Analysis]** - **Bandwidth Limit** - **BW Marker**

# SCPI.CALCulate(*Ch*).SELected.BLIMit.DISPlay.VALue

Object type

Property

Syntax

SCPI.CALCulate(*CH*).SELected.BLIMit.DISPlay.VALue = *Status*

*Status* = SCPI.CALCulate(*Ch*).SELected.BLIMit.DISPlay.VALue

Description

For the active trace of channel 1 to channel 4 (specified with the SCPI.CALCulate(Ch).PARameter(Tr).SELect command), turns ON/OFF the bandwidth value display of the bandwidth test.

Variable

| | ***Status*** |
|---|---|
| Description | ON/OFF of the bandwidth display of the bandwidth test. |
| Data type | Boolean type (Boolean) |
| Range | Select from the following.<br>•True or -1 Turns ON the bandwidth display.<br>•False or 0 Turns OFF the bandwidth display. |
| Preset value | False or 0 |

For information on the variable (*Ch*), see Table 7-4, “Variable (Ch),” on page 126.

Examples

```
Dim BLimVal As Boolean
SCPI.CALCulate(1).PARameter(1).SELectSCPI.CALCulate(1).SELected.BLI
Mit.DISPlay.VALue = True
BLimVal = SCPI.CALCulate(1).SELected.BLIMit.DISPlay.VALue
```

Related objects

SCPI.CALCulate(Ch).PARameter(Tr).SELect on page 128

SCPI.CALCulate(Ch).SELected.BLIMit.STATe on page 136

SCPI.CALCulate(Ch).SELected.BLIMit.DISPlay.MARKer on page 130

Equivalent key

**[Analysis]** - **Bandwidth Limit** - **BW Display**

# SCPI.CALCulate(*Ch*).SELected.BLIMit.FAIL

Object type — Property

Syntax — *Status* = SCPI.CALCulate(*Ch*).SELected.BLIMit.FAIL

Description — For the active trace of channel 1 to channel 4 (specified with the SCPI.CALCulate(Ch).PARameter(Tr).SELect command), reads out the bandwidth limit test result. (Read only)

Variable

| | ***Status*** |
|---|---|
| Description | The bandwidth limit test result |
| Data type | Boolean type (Boolean) |
| Range | Select from the following.<br>•True or -1 Turns ON the bandwidth limit test result is FAIL.<br>•False or 0 Turns OFF the bandwidth limit test result is PASS. |
| Note | When the bandwidth limit test if set to OFF, False or 0 is always read out. |

For information on the variable (*Ch*), see Table 7-4, “Variable (Ch),” on page 126.

Examples

```
Dim Result As Boolean
SCPI.CALCulate(1).PARameter(1).SELect
SCPI.CALCulate(1).SELected.BLIMit.STATe = True
Result = SCPI.CALCulate(1).SELected.BLIMit.FAIL
```

Related objects — SCPI.CALCulate(Ch).PARameter(Tr).SELect on page 128
SCPI.CALCulate(Ch).SELected.BLIMit.STATe on page 136

Equivalent key — No equivalent key is available on the front panel.

# SCPI.CALCulate(*Ch*).SELected.BLIMit.MAXimum

Object type

Property

Syntax

SCPI.CALCulate(*Ch*).SELected.BLIMit.MAXimum = *Value*

*Value* = SCPI.CALCulate(*Ch*).SELected.BLIMit.MAXimum

Description

For channel 1 to channel 4 (specified with the SCPI.CALCulate(Ch).PARameter(Tr).SELect command), sets the upper limit value of the bandwidth test.

Variable

| | ***Value*** |
|---|---|
| Description | Maximum bandwidth |
| Data type | Double precision floating point type (Double) |
| Range | 0 to 1E12 |
| Preset value | 10 k |
| Unit | Hz (hertz), dB or second |

For information on the variable (*Ch*), see Table 7-4, "Variable (Ch)," on page 126.

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 on page 128

SCPI.CALCulate(Ch).SELected.BLIMit.STATe on page 136

SCPI.CALCulate(Ch).SELected.BLIMit.MINimum on page 134

Equivalent key

**[Analysis]** - **Bandwidth Limit** - **Max Bandwidth**

# SCPI.CALCulate(*Ch*).SELected.BLIMit.MINimum

Object type

Property

Syntax

SCPI.CALCulate(*Ch*).SELected.BLIMit.MINimum = *Value*

*Value* = SCPI.CALCulate(*Ch*).SELected.BLIMit.MINimum

Description

For channel 1 to channel 4 (specified with the SCPI.CALCulate(Ch).PARameter(Tr).SELect command), sets the lower limit value of the bandwidth test.

Variable

| | ***Value*** |
|---|---|
| Description | Minimum bandwidth |
| Data type | Double precision floating point type (Double) |
| Range | 0 to 1E12 |
| Preset value | 300 k |
| Unit | Hz (hertz), dB or second |

For information on the variable (*Ch*), see Table 7-4, “Variable (Ch),” on page 126.

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 on page 128

SCPI.CALCulate(Ch).SELected.BLIMit.STATe on page 136

SCPI.CALCulate(Ch).SELected.BLIMit.MAXimum on page 133

Equivalent key

**[Analysis]** - **Bandwidth Limit** - **Min Bandwidth**

# SCPI.CALCulate(*Ch*).SELected.BLIMit.REPort.DATA

Object type

Property

Syntax

*Data* = SCPI.CALCulate(*Ch*).SELected.BLIMit.REPort.DATA

Description

For the active trace of channel 1 to channel 4 (specified with the SCPI.CALCulate(Ch).PARameter(Tr).SELect command), reads out the bandwidth value of the bandwidth test.

Variable

| | ***Data*** |
|---|---|
| Description | The bandwidth value of the bandwidth |
| Data type | Double precision floating point type (Double) |

For information on the variable (*Ch*), see Table 7-4, “Variable (Ch),” on page 126.

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 on page 128
SCPI.CALCulate(Ch).SELected.BLIMit.STATe on page 136

Equivalent key

No equivalent key is available on the front panel.

# SCPI.CALCulate(*Ch*).SELected.BLIMit.STATe

Object type — Property

Syntax — SCPI.CALCulate(*Ch*).SELected.BLIMit.STATe = *Status*

*Status* = SCPI.CALCulate(*Ch*).SELected.BLIMit.STATe

Description — For the active trace of channel 1 to channel 4 (specified with the SCPI.CALCulate(Ch).PARameter(Tr).SELect command), turns ON/OFF the bandwidth test function.

Variable

| | ***Status*** |
|---|---|
| Description | ON/OFF the bandwidth test function. |
| Data type | Boolean type (Boolean) |
| Range | Select from the following.<br>•True or -1 Turns ON the bandwidth test function.<br>•False or 0 Turns OFF the bandwidth test function. |
| Preset value | False or 0 |

For information on the variable (*Ch*), see Table 7-4, “Variable (Ch),” on page 126.

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 on page 128
SCPI.CALCulate(Ch).SELected.BLIMit.DB on page 129
SCPI.CALCulate(Ch).SELected.BLIMit.DISPlay.MARKer on page 130
SCPI.CALCulate(Ch).SELected.BLIMit.DISPlay.VALue on page 131
SCPI.CALCulate(Ch).SELected.BLIMit.FAIL on page 132
SCPI.CALCulate(Ch).SELected.BLIMit.MAXimum on page 133
SCPI.CALCulate(Ch).SELected.BLIMit.MINimum on page 134
SCPI.CALCulate(Ch).SELected.BLIMit.REPort.DATA on page 135

Equivalent key — **[Analysis]** - **Bandwidth Limit** - **BW Test**

# SCPI.CALCulate(*Ch*).SELected.CONVersion.FUNCtion

Object type

Property

Syntax

SCPI.CALCulate(*Ch*).SELected.CONVersion.FUNCtion = *Param*

*Param* = SCPI.CALCulate(*Ch*).SELected.CONVersion.FUNCtion

Description

For the active trace of channels 1 to 4 (*Ch*), select the parameter after conversion using the parameter conversion function.

Variable

| | ***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 in transmission measurement. |
| | •"YREFlection" | Specifies the equivalent admittance in reflection measurement. |
| | •"YTRansmit" | Specifies the equivalent admittance in transmission measurement. |
| | •"INVersion" | Specifies the inverse S-parameter. |
| Preset value | "ZREFlection" | |

For information on the variable (*Ch*), see Table 7-4, “Variable (Ch),” on page 126.

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 on page 138

SCPI.CALCulate(Ch).PARameter(Tr).SELect on page 128

Equivalent key

**[Analysis] - Conversion - Z:Reflection|Z:Transmission|Y:Reflection|Y:Transmission|1/S**

# SCPI.CALCulate(*Ch*).SELected.CONVersion.STATe

Object type

Property

Syntax

SCPI.CALCulate(*Ch*).SELected.CONVersion.STATe = *Status*

*Status* = SCPI.CALCulate(*Ch*).SELected.CONVersion.STATe

Description

For the active trace of channels 1 to 4 (*Ch*), turns ON/OFF the parameter conversion function.

Variable

| | ***Status*** |
|---|---|
| Description | ON/OFF of the parameter conversion function |
| Data type | Boolean type (Boolean) |
| Range | Select from the following.<br>•True or -1 Turns ON the parameter conversion function.<br>•False or 0 Turns OFF the parameter conversion function. |
| Preset value | False or 0 |

For information on the variable (*Ch*), see Table 7-4, “Variable (Ch),” on page 126.

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 on page 137

SCPI.CALCulate(Ch).PARameter(Tr).SELect on page 128

Equivalent key

**[Analysis]** - **Conversion** - **Conversion**

# SCPI.CALCulate(*Ch*).SELected.CORRection.EDELay. TIME

Object type — Property

Syntax — SCPI.CALCulate(*Ch*).SELected.CORRection.EDELay.TIME = *Value*

*Value* = SCPI.CALCulate(*Ch*).SELected.CORRection.EDELay.TIME

Description — Sets the electrical delay time of the active trace of channels 1 to 4 (*Ch*).

Variable

| | ***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. |

For information on the variable (*Ch*), see Table 7-4, “Variable (Ch),” on page 126.

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 on page 128

Equivalent key — **[Scale] - Electrical Delay**

# SCPI.CALCulate(*Ch*).SELected.CORRection.OFFSet. PHASe

Object type

Property

Syntax

SCPI.CALCulate(*Ch*).SELected.CORRection.OFFSet.PHASe = *Value*

*Value* = SCPI.CALCulate(*Ch*).SELected.CORRection.OFFSet.PHASe

Description

Sets the phase offset of the active trace of channels 1 to 4 (*Ch*).

Variable

| | ***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. |

For information on the variable (*Ch*), see Table 7-4, “Variable (Ch),” on page 126.

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 on page 128

Equivalent key

**[Scale] - Phase Offset**

# SCPI.CALCulate(*Ch*).SELected.DATA.FDATa

Object type — Property

Syntax — SCPI.CALCulate(*Ch*).SELected.DATA.FDATa = *Data*

*Data* = SCPI.CALCulate(*Ch*).SELected.DATA.FDATa

Description — For the active trace of channels 1 to 4 (*Ch*), sets/reads out the formatted data array. 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 Section "Internal Data Processing" in the *E5061A/E5062A Programmer's Guide*.

**NOTE** If valid data is not calculated because of the invalid measurement, "1.#QNB" is read out.

Variable

| | ***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.<br>• *Data(n×2-2)* Data (primary value) at the n-th measurement point.<br>• *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.<br>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. |

For information on the variable (*Ch*), see Table 7-4, "Variable (Ch)," on page 126.

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 on page 128

SCPI.SENSe(Ch).SWEep.POINts on page 365

SCPI.CALCulate(Ch).SELected.FORMat on page 145

SCPI.CALCulate(Ch).SELected.DATA.FMEMory on page 142

SCPI.CALCulate(Ch).SELected.DATA.SDATa on page 143

Equivalent key — No equivalent key is available on the front panel.

# SCPI.CALCulate(*Ch*).SELected.DATA.FMEMory

Object type — Property

Syntax — SCPI.CALCulate(*Ch*).SELected.DATA.FMEMory = *Data*

*Data* = SCPI.CALCulate(*Ch*).SELected.DATA.FMEMory

Description — For the active trace of channels 1 to 4 (*Ch*), sets/reads out the formatted memory array. 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 Section "Internal Data Processing" in the *E5061A/E5062A Programmer's Guide*.

**NOTE** If valid data is not calculated because of the invalid measurement, "1.#QNB" is read out.

Variable

| | ***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.<br>• *Data(n×2-2)* Data (primary value) at the n-th measurement point.<br>• *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.<br>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. |

For information on the variable (*Ch*), see Table 7-4, "Variable (Ch)," on page 126.

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 on page 128

SCPI.SENSe(Ch).SWEep.POINts on page 365

SCPI.CALCulate(Ch).SELected.FORMat on page 145

SCPI.CALCulate(Ch).SELected.DATA.FDATa on page 141

SCPI.CALCulate(Ch).SELected.DATA.SMEMory on page 144

Equivalent key — No equivalent key is available on the front panel.

# SCPI.CALCulate(*Ch*).SELected.DATA.SDATa

Object type — Property

Syntax — SCPI.CALCulate(*Ch*).SELected.DATA.SDATa = *Data*

*Data* = SCPI.CALCulate(*Ch*).SELected.DATA.SDATa

Description — For the active trace of channels 1 to 4 (*Ch*), sets/reads out the corrected data array. For more information on the corrected data array, see Section "Internal Data Processing" in the *E5061A/E5062A Programmer's Guide*.

**NOTE** If valid data is not calculated because of the invalid measurement, "1.#QNB" is read out.

Variable

| | ***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.<br>• *Data(n×2-2)* Real part of the data (complex number) at the n-th measurement point.<br>• *Data(n×2-1)* Imaginary part of the data (complex number) at the n-th measurement point.<br>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. |

For information on the variable (*Ch*), see Table 7-4, "Variable (Ch)," on page 126.

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 on page 128
SCPI.SENSe(Ch).SWEep.POINts on page 365
SCPI.CALCulate(Ch).SELected.DATA.SMEMory on page 144
SCPI.CALCulate(Ch).SELected.DATA.FDATa on page 141

Equivalent key — No equivalent key is available on the front panel.

# SCPI.CALCulate(*Ch*).SELected.DATA.SMEMory

Object type — Property

Syntax — SCPI.CALCulate(*Ch*).SELected.DATA.SMEMory = *Data*

*Data* = SCPI.CALCulate(*Ch*).SELected.DATA.SMEMory

Description — For the active trace of channels 1 to 4 (*Ch*), sets/reads out the corrected memory array. For more information on the corrected memory array, see Section "Internal Data Processing" in the *E5061A/E5062A Programmer's Guide*.

**NOTE** If valid data is not calculated because of the invalid measurement, "1.#QNB" is read out.

Variable

| | ***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.<br>• *Data(n×2-2)* Real part of the data (complex number) at the n-th measurement point.<br>• *Data(n×2-1)* Imaginary part of the data (complex number) at the n-th measurement point.<br>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. |

For information on the variable (*Ch*), see Table 7-4, "Variable (Ch)," on page 126.

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 on page 128

SCPI.SENSe(Ch).SWEep.POINts on page 365

SCPI.CALCulate(Ch).SELected.DATA.SDATa on page 143

SCPI.CALCulate(Ch).SELected.DATA.FMEMory on page 142

Equivalent key — No equivalent key is available on the front panel.

# SCPI.CALCulate(*Ch*).SELected.FORMat

Object type — Property

Syntax — SCPI.CALCulate(*Ch*).SELected.FORMat = *Param*

*Param* = SCPI.CALCulate(*Ch*).SELected.FORMat

Description — Selects the data format of the active trace of channels 1 to 4 (*Ch*).

Variable

| | ***Param*** |
|---|---|
| Description | Data format |
| Data type | Character string type (String) |
| Range | Select from the following.<br>•"MLOGarithmic" Specifies the log magnitude format.<br>•"PHASe" Specifies the phase format.<br>•"GDELay" Specifies the group delay format.<br>•"SLINear" Specifies the Smith chart format (Lin/Phase).<br>•"SLOGarithmic" Specifies the Smith chart format (Log/Phase).<br>•"SCOMplex" Specifies the Smith chart format (Re/Im).<br>•"SMITh" Specifies the Smith chart format (R+jX).<br>•"SADMittance" Specifies the Smith chart format (G+jB).<br>•"PLINear" Specifies the polar format (Lin/Phase).<br>•"PLOGarithmic" Specifies the polar format (Log/Phase).<br>•"POLar" Specifies the polar format (Re/Im).<br>•"MLINear" Specifies the linear magnitude format.<br>•"SWR" Specifies the SWR format.<br>•"REAL" Specifies the real format.<br>•"IMAGinary" Specifies the imaginary format.<br>•"UPHase" Specifies the expanded phase format.<br>•"PPHase" Specifies the positive phase format. |
| Preset value | "MLOGarithmic" |

For information on the variable (*Ch*), see Table 7-4, “Variable (Ch),” on page 126.

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 on page 128

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**

# SCPI.CALCulate(*Ch*).SELected.FUNCtion.DATA

Object type — Property

Syntax — *Data* = SCPI.CALCulate(*Ch*).SELected.FUNCtion.DATA

Description — For the active trace of channels 1 to 4 (*Ch*), reads out the analysis result of the SCPI.CALCulate(Ch).SELected.FUNCtion.EXECute object. (Read only)

Variable

| | ***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.<br>• *Data(n×2-2)* Response value or analysis result of the searched n-th measurement point.<br>• *Data(n×2-1)* Stimulus value of the searched n-th measurement point. Always 0 for the analysis of the mean value*1, the standard deviation*1, and the difference between the maximum value and the minimum value*1.<br>The index of the array starts from 0. |
| Data type | Variant type (Variant) |

*1. To specify the type of the analysis, use the SCPI.CALCulate(Ch).SELected.FUNCtion.TYPE object.

For information on the variable (*Ch*), see Table 7-4, “Variable (Ch),” on page 126.

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 on page 128
SCPI.CALCulate(Ch).SELected.FUNCtion.TYPE on page 157
SCPI.CALCulate(Ch).SELected.FUNCtion.EXECute on page 151
SCPI.CALCulate(Ch).SELected.FUNCtion.POINts on page 153

Equivalent key — No equivalent key is available on the front panel.

# SCPI.CALCulate(*Ch*).SELected.FUNCtion.DOMain.COUPle

Object type

Property

Syntax

SCPI.CALCulate(*Ch*).SELected.FUNCtion.DOMain.COUPle = *Status*

*Status* = SCPI.CALCulate(*Ch*).SELected.FUNCtion.DOMain.COUPle

Description

For channels 1 to 4 (*Ch*), specifies whether to set the coupling of the analysis range of the SCPI.CALCulate(Ch).SELected.FUNCtion.EXECute object for all traces.

Variable

| | ***Status*** | |
|---|---|---|
| Description | On/off of the trace coupling of the analysis range. | |
| Data type | Boolean type (Boolean) | |
| Range | Select from the following. | |
| | •True or -1 | Specifies the analysis range with the trace coupling. |
| | •False or 0 | Specifies the analysis range for each trace. |
| Preset value | True or -1 | |

For information on the variable (*Ch*), refer to Table 7-4, “Variable (Ch),” on page 126.

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.EXECute on page 151

Equivalent key

No equivalent key is available on the front panel.

# SCPI.CALCulate(*Ch*).SELected.FUNCtion.DOMain.STARt

Object type

Property

Syntax

SCPI.CALCulate(*Ch*).SELected.FUNCtion.DOMain.STARt = *Value*

*Value* = SCPI.CALCulate(*Ch*).SELected.FUNCtion.DOMain.STARt

Description

For channels 1 to 4 (*Ch*), sets the start value of the analysis range of the SCPI.CALCulate(Ch).SELected.FUNCtion.EXECute object.

When the trace coupling is off, the active trace is the target to be set.

Variable

| | ***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) |

For information on the variable (*Ch*), see Table 7-4, “Variable (Ch),” on page 126.

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.DOMain.STOP on page 150

SCPI.CALCulate(Ch).SELected.FUNCtion.DOMain.STATe on page 149

SCPI.CALCulate(Ch).SELected.FUNCtion.DOMain.COUPle on page 147

SCPI.CALCulate(Ch).SELected.FUNCtion.EXECute on page 151

Equivalent key

No equivalent key is available on the front panel.

# SCPI.CALCulate(*Ch*).SELected.FUNCtion.DOMain.STATe

Object type

Property

Syntax

SCPI.CALCulate(*Ch*).SELected.FUNCtion.DOMain.STATe = *Status*

*Status* = SCPI.CALCulate(*Ch*).SELected.FUNCtion.DOMain.STATe

Description

For channels 1 to 4 (*Ch*), sets whether to use an arbitrary range when executing the analysis with the SCPI.CALCulate(Ch).SELected.FUNCtion.EXECute object.

When the trace coupling is off, the active trace is the target to be set.

Variable

| | ***Status*** |
|---|---|
| Description | Selection of the analysis range |
| Data type | Boolean type (Boolean) |
| Range | Select from the following.<br>•True or -1 Specifies an arbitrary range*1.<br>•False or 0 Specifies the entire sweep range. |
| Preset value | False or 0 |

*1. Specify with the SCPI.CALCulate(Ch).SELected.FUNCtion.DOMain.STARt object and the SCPI.CALCulate(Ch).SELected.FUNCtion.DOMain.STOP object.

For information on the variable (*Ch*), see Table 7-4, “Variable (Ch),” on page 126.

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.DOMain.STARt on page 148

SCPI.CALCulate(Ch).SELected.FUNCtion.DOMain.STOP on page 150

SCPI.CALCulate(Ch).SELected.FUNCtion.DOMain.COUPle on page 147

SCPI.CALCulate(Ch).SELected.FUNCtion.EXECute on page 151

Equivalent key

No equivalent key is available on the front panel.

# SCPI.CALCulate(*Ch*).SELected.FUNCtion.DOMain.STOP

Object type

Property

Syntax

SCPI.CALCulate(*Ch*).SELected.FUNCtion.DOMain.STOP = *Value*

*Value* = SCPI.CALCulate(*Ch*).SELected.FUNCtion.DOMain.STOP

Description

For channels 1 to 4 (*Ch*), sets the stop value of the analysis range of the SCPI.CALCulate(Ch).SELected.FUNCtion.EXECute object.

When the trace coupling is off, the active trace is the target to be set.

Variable

| | ***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) |

For information on the variable (*Ch*), see Table 7-4, “Variable (Ch),” on page 126.

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.DOMain.STARt on page 148

SCPI.CALCulate(Ch).SELected.FUNCtion.DOMain.STATe on page 149

SCPI.CALCulate(Ch).SELected.FUNCtion.DOMain.COUPle on page 147

SCPI.CALCulate(Ch).SELected.FUNCtion.EXECute on page 151

Equivalent key

No equivalent key is available on the front panel.

# SCPI.CALCulate(*Ch*).SELected.FUNCtion.EXECute

| | |
|---|---|
| Object type | Method |
| Syntax | SCPI.CALCulate(*Ch*).SELected.FUNCtion.EXECute |
| Description | For the active trace of channels 1 to 4 (*Ch*), executes the analysis specified with the SCPI.CALCulate(Ch).SELected.FUNCtion.TYPE object. (No read) |
| Variable | For information on the variable (*Ch*), see Table 7-4, "Variable (Ch)," on page 126. |
| Examples | `SCPI.CALCulate(1).PARameter(1).SELect`<br>`SCPI.CALCulate(1).SELected.FUNCtion.EXECute` |
| Related objects | SCPI.CALCulate(Ch).PARameter(Tr).SELect on page 128<br>SCPI.CALCulate(Ch).SELected.FUNCtion.TYPE on page 157<br>SCPI.CALCulate(Ch).SELected.FUNCtion.DOMain.STATe on page 149 |
| Equivalent key | No equivalent key is available on the front panel. |

# SCPI.CALCulate(*Ch*).SELected.FUNCtion.PEXCursion

Object type

Property

Syntax

SCPI.CALCulate(*Ch*).SELected.FUNCtion.PEXCursion = *Value*

*Value* = SCPI.CALCulate(*Ch*).SELected.FUNCtion.PEXCursion

Description

For the active trace of channels 1 to 4 (*Ch*), sets 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 information on the peak excursion value, see Section “Searching for the Peak” in the *E5061A/E5062A User’s Guide*.

Variable

| | ***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.<br>• Log magnitude (MLOG) : dB (decibel)<br>• Phase (PHAS), Expanded phase (UPH) or Positive phase (PPH) : ° (degree)<br>• Group delay (GDEL) : s (second)<br>• 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. |

For information on the variable (*Ch*), see Table 7-4, “Variable (Ch),” on page 126.

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).PARameter(Tr).SELect on page 128

SCPI.CALCulate(Ch).SELected.FUNCtion.TYPE on page 157

SCPI.CALCulate(Ch).SELected.FUNCtion.PPOLarity on page 154

SCPI.CALCulate(Ch).SELected.FUNCtion.EXECute on page 151

Equivalent key

No equivalent key is available on the front panel.

# SCPI.CALCulate(*Ch*).SELected.FUNCtion.POINts

Object type

Property

Syntax

*Value* = SCPI.CALCulate(*Ch*).SELected.FUNCtion.POINts

Description

For the active trace of channels 1 to 4 (*Ch*), reads out the number of data pairs of the analysis result of the SCPI.CALCulate(Ch).SELected.FUNCtion.EXECute object.

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. (Read only)

Variable

| | ***Value*** |
|---|---|
| Description | Number of analyzed data pairs |
| Data type | Long integer type (Long) |

For information on the variable (*Ch*), see Table 7-4, “Variable (Ch),” on page 126.

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).PARameter(Tr).SELect on page 128

SCPI.CALCulate(Ch).SELected.FUNCtion.EXECute on page 151

SCPI.CALCulate(Ch).SELected.FUNCtion.DATA on page 146

Equivalent key

No equivalent key is available on the front panel.

# SCPI.CALCulate(*Ch*).SELected.FUNCtion.PPOLarity

Object type

Property

Syntax

SCPI.CALCulate(*Ch*).SELected.FUNCtion.PPOLarity = *Param*

*Param* = SCPI.CALCulate(*Ch*).SELected.FUNCtion.PPOLarity

Description

For the active trace of channels 1 to 4 (*Ch*), selects the polarity when performing the peak search with the SCPI.CALCulate(Ch).SELected.FUNCtion.EXECute object.

Variable

| | ***Param*** |
|---|---|
| Description | Polarity for peak search |
| Data type | Character string type (String) |
| Range | Select from the following.<br>•"POSitive" Specifies the positive peak.<br>•"NEGative" Specifies the negative peak.<br>•"BOTH" Specifies both the positive peak and the negative peak. |
| Preset value | "POSitive" |

For information on the variable (*Ch*), see Table 7-4, “Variable (Ch),” on page 126.

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).PARameter(Tr).SELect on page 128

SCPI.CALCulate(Ch).SELected.FUNCtion.TYPE on page 157

SCPI.CALCulate(Ch).SELected.FUNCtion.PEXCursion on page 152

SCPI.CALCulate(Ch).SELected.FUNCtion.EXECute on page 151

Equivalent key

No equivalent key is available on the front panel.

# SCPI.CALCulate(*Ch*).SELected.FUNCtion.TARGet

Object type

Property

Syntax

SCPI.CALCulate(*Ch*).SELected.FUNCtion.TARGet = *Value*

*Value* = SCPI.CALCulate(*Ch*).SELected.FUNCtion.TARGet

Description

For the active trace of channels 1 to 4 (*Ch*), selects the target value when performing the target search with the SCPI.CALCulate(Ch).SELected.FUNCtion.EXECute object.

Variable

| | ***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.<br>• Log magnitude (MLOG) : dB (decibel)<br>• Phase (PHAS), Expanded phase (UPH) or Positive phase (PPH) : ° (degree)<br>• Group delay (GDEL) : s (second)<br>• 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. |

For information on the variable (*Ch*), see Table 7-4, “Variable (Ch),” on page 126.

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).PARameter(Tr).SELect on page 128

SCPI.CALCulate(Ch).SELected.FUNCtion.TYPE on page 157

SCPI.CALCulate(Ch).SELected.FUNCtion.TTRansition on page 156

SCPI.CALCulate(Ch).SELected.FUNCtion.EXECute on page 151

Equivalent key

No equivalent key is available on the front panel.

# SCPI.CALCulate(*Ch*).SELected.FUNCtion.TTRansition

Object type

Property

Syntax

SCPI.CALCulate(*Ch*).SELected.FUNCtion.TTRansition = *Param*

*Param* = SCPI.CALCulate(*Ch*).SELected.FUNCtion.TTRansition

Description

For the active trace of channels 1 to 4 (*Ch*), selects the transition type when performing the target search with the SCPI.CALCulate(Ch).SELected.FUNCtion.EXECute object. For more information on the transition type, see Section “Searching for the Target Value” in the *E5061A/E5062A User’s Guide*.

Variable

| | ***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" | |

For information on the variable (*Ch*), see Table 7-4, “Variable (Ch),” on page 126.

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).PARameter(Tr).SELect on page 128

SCPI.CALCulate(Ch).SELected.FUNCtion.TYPE on page 157

SCPI.CALCulate(Ch).SELected.FUNCtion.TARGet on page 155

SCPI.CALCulate(Ch).SELected.FUNCtion.EXECute on page 151

Equivalent key

No equivalent key is available on the front panel.

# SCPI.CALCulate(*Ch*).SELected.FUNCtion.TYPE

Object type — Property

Syntax — SCPI.CALCulate(*Ch*).SELected.FUNCtion.TYPE = *Param*

*Param* = SCPI.CALCulate(*Ch*).SELected.FUNCtion.TYPE

Description — For the active trace of channels 1 to 4 (*Ch*), selects the type of analysis.

Variable

| | ***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*1. |
| | •"APEak" | Specifies the search for all peaks*1. |
| | •"ATARget" | Specifies the search for all targets*2. |
| Preset value | "PTPeak" | |

*1. To specify the conditions of the peak, use the SCPI.CALCulate(Ch).SELected.FUNCtion.PEXCursion object and the SCPI.CALCulate(Ch).SELected.FUNCtion.PPOLarity object.

*2. To specify the conditions of the target, use the SCPI.CALCulate(Ch).SELected.FUNCtion.TARGet object and the SCPI.CALCulate(Ch).SELected.FUNCtion.TTRansition object.

For information on the variable (*Ch*), see Table 7-4, “Variable (Ch),” on page 126.

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).PARameter(Tr).SELect on page 128
SCPI.CALCulate(Ch).SELected.FUNCtion.PEXCursion on page 152
SCPI.CALCulate(Ch).SELected.FUNCtion.PPOLarity on page 154
SCPI.CALCulate(Ch).SELected.FUNCtion.TARGet on page 155
SCPI.CALCulate(Ch).SELected.FUNCtion.TTRansition on page 156
SCPI.CALCulate(Ch).SELected.FUNCtion.EXECute on page 151

Equivalent key — No equivalent key is available on the front panel.

# SCPI.CALCulate(*Ch*).SELected.LIMit.DATA

Object type

Property

Syntax

SCPI.CALCulate(*Ch*).SELected.LIMit.DATA = *Data*

*Data* = SCPI.CALCulate(*Ch*).SELected.LIMit.DATA

Description

For the active trace of channels 1 to 4 (*Ch*), sets the limit table for the limit test.

Variable

| | ***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.<br>• *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)*.<br>• *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<br>• *Data(n×5-3)* — The value on the horizontal axis (frequency/power/time) of the start point of the n-th line.<br>• *Data(n×5-2)* — The value on the horizontal axis (frequency/power/time) of the end point of the n-th line.<br>• *Data(n×5-1)* — The value on the vertical axis of the start point of the n-th line.<br>• *Data(n×5)* — The value on the vertical axis of the end point of the n-th line.<br>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))×5 when setting a formatted memory array, an error occurs when executed 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 when executed. 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. |

For information on the variable (*Ch*), see Table 7-4, “Variable (Ch),” on page 126.

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 on page 128

SCPI.CALCulate(Ch).SELected.LIMit.STATe on page 169

SCPI.CALCulate(Ch).SELected.LIMit.DISPlay.STATe on page 161

Equivalent key

**[Analysis]** - **Limit Test** - **Edit Limit Line**

# SCPI.CALCulate(*Ch*).SELected.LIMit.DISPlay.CLIP

Object type

Property

Syntax

SCPI.CALCulate(*Ch*).SELected.LIMit.DISPlay.CLIP = *Status*

*Status* = SCPI.CALCulate(*Ch*).SELected.LIMit.DISPlay.CLIP

Description

For the active trace of channels 1 to 4 (*Ch*), specifies whether to display the part of the limit line(s) that is not used for evaluation.

Variable

| | ***Status*** |
|---|---|
| Description | Displays the clipped limit lines |
| Data type | Boolean type (Boolean) |
| Range | Select from the following.<br>•True or -1 Displays the clipped limit lines.<br>•False or 0 Displays the entire limit lines. |
| Preset value | True or -1 |

For information on the variable (*Ch*), see Table 7-4, “Variable (Ch),” on page 126.

Examples

```
Dim LimClip As Boolean
SCPI.CALCulate(1).PARameter(1).SELect
SCPI.CALCulate(1).SELected.LIMit.DISPlay.CLIP = True
LimClip = SCPI.CALCulate(1).SELected.LIMit.DISPlay.CLIP
```

Related objects

SCPI.CALCulate(Ch).PARameter(Tr).SELect on page 128

SCPI.CALCulate(Ch).SELected.LIMit.STATe on page 169

Equivalent key

**[Analysis]** - **Limit Test** - **Clip Lines**

# SCPI.CALCulate(*Ch*).SELected.LIMit.DISPlay.STATe

Object type — Property

Syntax

SCPI.CALCulate(*Ch*).SELected.LIMit.DISPlay.STATe = *Status*

*Status* = SCPI.CALCulate(*Ch*).SELected.LIMit.DISPlay.STATe

Description — For the active trace of channels 1 to 4 (*Ch*), turns ON/OFF the limit line display.

Variable

| | ***Status*** |
|---|---|
| Description | Limit line display |
| Data type | Boolean type (Boolean) |
| Range | Select from the following.<br>•True or -1 Turns ON the limit line display.<br>•False or 0 Turns OFF the limit line display. |
| Preset value | False or 0 |

For information on the variable (*Ch*), see Table 7-4, “Variable (Ch),” on page 126.

Examples

```
Dim LimDisp As Boolean
SCPI.CALCulate(1).PARameter(1).SELect
SCPI.CALCulate(1).SELected.LIMit.DISPlay.STATe = True
LimDisp = SCPI.CALCulate(1).SELected.LIMit.DISPlay.STATe
```

Related objects

SCPI.CALCulate(Ch).PARameter(Tr).SELect on page 128
SCPI.CALCulate(Ch).SELected.LIMit.STATe on page 169

Equivalent key — **[Analysis]** - **Limit Test** - **Limit Line**

# SCPI.CALCulate(*Ch*).SELected.LIMit.FAIL

Object type — Property

Syntax — *Status* = SCPI.CALCulate(*Ch*).SELected.LIMit.FAIL

Description — For the active trace of channels 1 to 4 (*Ch*), reads out the limit test result. (Read only)

Variable

| | ***Status*** |
|---|---|
| Description | Limit test result |
| Data type | Boolean type (Boolean) |
| Range | Select from the following.<br>•True or -1 The limit test result is FAIL.<br>•False or 0 The limit test result is PASS. |
| Note | When the limit test is set to OFF, False or 0 is always read out. |

For information on the variable (*Ch*), see Table 7-4, “Variable (Ch),” on page 126.

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 on page 128

SCPI.CALCulate(Ch).SELected.LIMit.STATe on page 169

Equivalent key — No equivalent key is available on the front panel.

# SCPI.CALCulate(*Ch*).SELected.LIMit.OFFSet.AMPLitude

Object type

Property

Syntax

SCPI.CALCulate(*Ch*).SELected.LIMit.OFFSet.AMPLitude = *Value*

*Value* = SCPI.CALCulate(*Ch*).SELected.LIMit.OFFSet.AMPLitude

Description

For channel 1 to channel 4 (specified with the SCPI.CALCulate(Ch).PARameter(Tr).SELect command), sets the limit line amplitude offset.

The setting of the limit line doesn't change even if the offset value is changed.

Variable

| | ***Value*** |
|---|---|
| Description | The limit line amplitude offset |
| Data type | Double precision floating point type (Double) |
| Range | -5E8 to 5E8 |
| Preset value | 0 |
| Unit | dB |

For information on the variable (*Ch*), see Table 7-4, “Variable (Ch),” on page 126.

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 on page 169

SCPI.CALCulate(Ch).SELected.LIMit.OFFSet.MARKer on page 164

SCPI.CALCulate(Ch).SELected.LIMit.OFFSet.STIMulus on page 165

Equivalent key

**[Analysis]** - **Limit Test** - **Limit Line Offsets** - **Amplitude Offset**

# SCPI.CALCulate(*Ch*).SELected.LIMit.OFFSet.MARKer

Object type — Method

Syntax — SCPI.CALCulate(*Ch*).SELected.LIMit.OFFSet.MARKer

Description — For channel 1 to channel 4 (specified with the SCPI.CALCulate(Ch).PARameter(Tr).SELect command), sets the active marker value to amplitude offset using the limit line.

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 — For information on the variable (*Ch*), see Table 7-4, "Variable (Ch)," on page 126.

Examples

```
SCPI.CALCulate(1).PARameter(1).SELect
SCPI.CALCulate(1).SELected.LIMit.OFFSet.MARKer
```

Related objects — SCPI.CALCulate(Ch).SELected.LIMit.STATe on page 169

SCPI.CALCulate(Ch).SELected.LIMit.OFFSet.AMPLitude on page 163

SCPI.CALCulate(Ch).SELected.LIMit.OFFSet.STIMulus on page 165

Equivalent key — **[Analysis] - Limit Test - Limit Line Offsets - Marker -> Amplitude Offset**

# SCPI.CALCulate(*Ch*).SELected.LIMit.OFFSet.STIMulus

Object type

Property

Syntax

SCPI.CALCulate(*Ch*).SELected.LIMit.OFFSet.STIMulus = *Value*

*Value* = SCPI.CALCulate(*Ch*).SELected.LIMit.OFFSet.STIMulus

Description

For channel 1 to channel 4 (specified with the SCPI.CALCulate(Ch).PARameter(Tr).SELect command), sets the stimulus offset of the limit line.

The setting of the limit line doesn't change even if the offset value is changed.

Variable

| | ***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 |

For information on the variable (*Ch*), see Table 7-4, “Variable (Ch),” on page 126.

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 on page 169

SCPI.CALCulate(Ch).SELected.LIMit.OFFSet.AMPLitude on page 163

SCPI.CALCulate(Ch).SELected.LIMit.OFFSet.MARKer on page 164

Equivalent key

**[Analysis]** - **Limit Test** - **Limit Line Offsets** - **Stimulus Offset**

# SCPI.CALCulate(*Ch*).SELected.LIMit.REPort.ALL

Object type — Property

Syntax — Data = SCPI.CALCulate(*Ch*).SELected.LIMit.REPort.ALL

Description — For the active trace of channel 1 to channel 4 (specified with the SCPI.CALCulate(Ch).PARameter(Tr).SELect command), reads out the bandwidth test results (stimulus value, limit test result, upper limit value, lower limit value of all measurement points). (Read only)

Variable

| | ***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.<br>• *Data(n×4-3)* — The stimulus value for the measurement point.<br>• *Data(n×4-2)* — The limit test result.<br>Specify an integer -1 to 1 as follows.<br>-1: No limit<br>0: Fail<br>1: Pass<br>• *Data(n×4-1)* — The upper limit value at the measurement point. (If there is no limit at this point, reads out the 0.)<br>• *Data(n×4)* — The lower limit value at the measurement point. (If there is no limit at this point, reads out the 0.)<br>The index of the array starts from 0. |
| Data type | Variant type (Variant) |

For information on the variable (*Ch*), see Table 7-4, “Variable (Ch),” on page 126.

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 on page 128

SCPI.CALCulate(Ch).SELected.LIMit.STATe on page 169

SCPI.CALCulate(Ch).SELected.LIMit.REPort.DATA on page 167

SCPI.CALCulate(Ch).SELected.LIMit.REPort.POINts on page 168

Equivalent key — No equivalent key is available on the front panel.

# SCPI.CALCulate(*Ch*).SELected.LIMit.REPort.DATA

Object type

Property

Syntax

*Data* = SCPI.CALCulate(*Ch*).SELected.LIMit.REPort.DATA

Description

For the active trace of channels 1 to 4 (*Ch*), reads out the stimulus values (frequency, power level or time) at all the measurement points that failed the limit test. (Read only)

Variable

| | ***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) |

For information on the variable (*Ch*), see Table 7-4, “Variable (Ch),” on page 126.

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 on page 128

SCPI.CALCulate(Ch).SELected.LIMit.REPort.POINts on page 168

SCPI.CALCulate(Ch).SELected.LIMit.STATe on page 169

Equivalent key

No equivalent key is available on the front panel.

# SCPI.CALCulate(*Ch*).SELected.LIMit.REPort.POINts

Object type

Property

Syntax

*Value* = SCPI.CALCulate(*Ch*).SELected.LIMit.REPort.POINts

Description

For the active trace of channels 1 to 4 (*Ch*), reads out the number of the measurement points that failed the limit test. (Read only)

Variable

| | ***Value*** |
|---|---|
| Description | Number of measurement points that failed |
| Data type | Long integer type (Long) |

For information on the variable (*Ch*), see Table 7-4, “Variable (Ch),” on page 126.

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 on page 128

SCPI.CALCulate(Ch).SELected.LIMit.STATe on page 169

Equivalent key

No equivalent key is available on the front panel.

# SCPI.CALCulate(*Ch*).SELected.LIMit.STATe

Object type — Property

Syntax — SCPI.CALCulate(*Ch*).SELected.LIMit.STATe = *Status*

*Status* = SCPI.CALCulate(*Ch*).SELected.LIMit.STATe

Description — For the active trace of channels 1 to 4 (*Ch*), turns ON/OFF the limit line function.

Variable

| | ***Status*** |
|---|---|
| Description | ON/OFF of the limit test function |
| Data type | Boolean type (Boolean) |
| Range | Select from the following.<br>•True or -1 Turns ON the limit test function.<br>•False or 0 Turns OFF the limit test function. |
| Preset value | False or 0 |

For information on the variable (*Ch*), see Table 7-4, “Variable (Ch),” on page 126.

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 on page 128
SCPI.CALCulate(Ch).SELected.LIMit.DISPlay.STATe on page 161
SCPI.DISPlay.FSIGn on page 242

Equivalent key — **[Analysis]** - **Limit Test** - **Limit Test**

# SCPI.CALCulate(*Ch*).SELected.MARKer(*Mk*).ACTivate

Object type — Method

Syntax — SCPI.CALCulate(*Ch*).SELected.MARKer(*Mk*).ACTivate

Description — For the active trace of channels 1 to 4 (*Ch*), sets marker 1 to 9 (*Mk*) and reference marker (*Mk*:10) to the active marker. (No read)

**NOTE** — If you set a marker not displayed to the active marker, the marker display is automatically set to ON.

Variable

**Table 7-6 Variable (*Mk*)**

| | ***Mk*** |
|---|---|
| Description | Marker number |
| Data type | Long integer type (Long) |
| Range | 1 to 10<br>Notice that 10 is for the reference marker. |
| Preset value | 1 |
| Note | If the specified variable is out of the allowable setup range, an error occurs when executed. |

For information on the variable (*Ch*), see Table 7-4, "Variable (Ch)," on page 126.

Examples

```
SCPI.CALCulate(1).PARameter(1).SELect
SCPI.CALCulate(1).SELected.MARKer(1).ACTivate
```

Related objects

SCPI.CALCulate(Ch).PARameter(Tr).SELect on page 128

SCPI.DISPlay.WINDow(Ch).ACTivate on page 250

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**

# SCPI.CALCulate(*Ch*).SELected.MARKer(*Mk*).BWIDth. DATA

Object type

Property

Syntax

*Data* = SCPI.CALCulate(*Ch*).SELected.MARKer(*Mk*).BWIDth.DATA

Description

For the active trace of channels 1 to 4 (*Ch*), reads out the bandwidth search result of marker 1 to 9 (*Mk*) and reference marker (*Mk*:10).

If the bandwidth search is impossible, an error occurs when executed and the object is ignored. (Read only)

Variable

| | ***Data*** |
|---|---|
| Description | Indicates 4-element array data (bandwidth search result).<br>• *Data(0)* The bandwidth.<br>• *Data(1)* Center point frequency of the 2 cutoff frequency points.<br>• *Data(2)* The Q value.<br>• *Data(3)* Insertion loss<br>The index of the array starts from 0. |
| Data type | Variant type (Variant) |

For information on the variable (*Ch*) and the variable (*Mk*), see Table 7-4, "Variable (Ch)," on page 126 and Table 7-6, "Variable (Mk)," on page 170, respectively.

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 on page 128

SCPI.CALCulate(Ch).SELected.MARKer.BWIDth.STATe on page 172

SCPI.CALCulate(Ch).SELected.MARKer(Mk).BWIDth. THReshold on page 173

Equivalent key

No equivalent key is available on the front panel.

# SCPI.CALCulate(*Ch*).SELected.MARKer.BWIDth.STATe

Object type

Property

Syntax

SCPI.CALCulate(*Ch*).SELected.MARKer.BWIDth.STATe = *Status*

*Status* = SCPI.CALCulate(*Ch*).SELected.MARKer.BWIDth.STATe

Description

For the active trace of channels 1 to 4 (*Ch*), turns ON/OFF the bandwidth search result display.

Variable

| | ***Status*** |
|---|---|
| Description | ON/OFF of the bandwidth search result display |
| Data type | Boolean type (Boolean) |
| Range | Select from the following.<br>•True or -1 Turns ON the bandwidth search result display.<br>•False or 0 Turns OFF the bandwidth search result display. |
| Preset value | False or 0 |

For information on the variable (*Ch*), see Table 7-4, “Variable (Ch),” on page 126.

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 on page 128

SCPI.CALCulate(Ch).SELected.MARKer(Mk).BWIDth. DATA on page 171

SCPI.CALCulate(Ch).SELected.MARKer(Mk).BWIDth. THReshold on page 173

Equivalent key

**[Marker Search]** - **Bandwidth**

# SCPI.CALCulate(*Ch*).SELected.MARKer(*Mk*).BWIDth. THReshold

Object type

Property

Syntax

SCPI.CALCulate(*Ch*).SELected.MARKer(*Mk*).BWIDth.THReshold = *Value*

*Value* = SCPI.CALCulate(*Ch*).SELected.MARKer(*Mk*).BWIDth.THReshold

Description

For the active trace of channels 1 to 4 (*Ch*), sets 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).

Variable

| | ***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.<br>• Log magnitude (MLOG): dB (decibel)<br>• Phase (PHAS), Expanded phase (UPH) or Positive phase (PPH): ° (degree)<br>• Group delay (GDEL): s (second)<br>• 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. |

For information on the variable (*Ch*) and the variable (*Mk*), see Table 7-4, “Variable (Ch),” on page 126 and Table 7-6, “Variable (Mk),” on page 170, respectively.

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 on page 128

SCPI.CALCulate(Ch).SELected.MARKer.BWIDth.STATe on page 172

Equivalent key

**[Marker Search] - Bandwidth Value**

# SCPI.CALCulate(*Ch*).SELected.MARKer.COUPle

Object type

Property

Syntax

SCPI.CALCulate(*Ch*).SELected.MARKer.COUPle = *Status*

*Status* = SCPI.CALCulate(*Ch*).SELected.MARKer.COUPle

Description

For channels 1 to 4 (*Ch*), turns ON/OFF the marker coupling between traces.

Variable

| | ***Status*** |
|---|---|
| Description | ON/OFF of the marker coupling between traces |
| Data type | Boolean type (Boolean) |
| Range | Select from the following.<br>•True or -1 Turns ON the marker coupling.<br>•False or 0 Turns OFF the marker coupling. |
| Preset value | True or -1 |

For information on the variable (*Ch*), see Table 7-4, “Variable (Ch),” on page 126.

Examples

```
Dim MkrCpl As Boolean
SCPI.CALCulate(1).SELected.MARKer.COUPle = False
MkrCpl = SCPI.CALCulate(1).SELected.MARKer.COUPle
```

Equivalent key

**[Marker Fctn] - Couple**

# SCPI.CALCulate(*Ch*).SELected.MARKer(*Mk*).DISCrete

Object type

Property

Syntax

SCPI.CALCulate(*Ch*).SELected.MARKer(*Mk*).DISCrete = *Status*

*Status* = SCPI.CALCulate(*Ch*).SELected.MARKer(*Mk*).DISCrete

Description

For the active trace of channels 1 to 4 (*Ch*), 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).

Variable

| | ***Status*** |
|---|---|
| Description | ON/OFF of the marker discrete mode |
| Data type | Boolean type (Boolean) |
| Range | Select from the following.<br>•True or -1 Turns ON the discrete mode.<br>•False or 0 Turns OFF the discrete mode. |
| Preset value | False or 0 |

For information on the variable (*Ch*), see Table 7-4, “Variable (Ch),” on page 126.

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 on page 128

Equivalent key

**[Marker Fctn] - Discrete**

# SCPI.CALCulate(*Ch*).SELected.MARKer.FUNCtion. DOMain.COUPle

Object type

Property

Syntax

SCPI.CALCulate(*Ch*).SELected.MARKer.FUNCtion.DOMain.COUPle = *Status*

*Status* = SCPI.CALCulate(*Ch*).SELected.MARKer.FUNCtion.DOMain.COUPle

Description

For channels 1 to 4 (*Ch*), specifies whether to set the coupling of the marker search range for all traces.

Variable

| | ***Status*** |
|---|---|
| Description | On/off of the trace coupling of the marker search range. |
| Data type | Boolean type (Boolean) |
| Range | Select from the following.<br>•True or -1 Specifies the search range with the trace coupling.<br>•False or 0 Specifies the search range for each trace. |
| Preset value | True or -1 |

For information on the variable (*Ch*), refer to Table 7-4, “Variable (Ch),” on page 126.

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 on page 180

Equivalent key

**[Marker Search] - Search Range - Couple**

# SCPI.CALCulate(*Ch*).SELected.MARKer.FUNCtion. DOMain.STARt

Object type

Property

Syntax

SCPI.CALCulate(*Ch*).SELected.MARKer.FUNCtion.DOMain.STARt = *Value*

*Value* = SCPI.CALCulate(*Ch*).SELected.MARKer.FUNCtion.DOMain.STARt

Description

For channels 1 to 4 (*Ch*), sets the start value of the marker search range.

When the trace coupling is off, the active trace is the target to be set.

Variable

| | ***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) |

For information on the variable (*Ch*), refer to Table 7-4, “Variable (Ch),” on page 126.

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 on page 179
SCPI.CALCulate(Ch).SELected.MARKer.FUNCtion. DOMain.STATe on page 178
SCPI.CALCulate(Ch).SELected.MARKer(Mk).FUNCtion. EXECute on page 180

Equivalent key

**[Marker Search]** - **Search Range** - **Start**

# SCPI.CALCulate(*Ch*).SELected.MARKer.FUNCtion. DOMain.STATe

Object type

Property

Syntax

SCPI.CALCulate(*Ch*).SELected.MARKer.FUNCtion.DOMain.STATe = *Status*

*Status* = SCPI.CALCulate(*Ch*).SELected.MARKer.FUNCtion.DOMain.STATe

Description

For channels 1 to 4 (*Ch*), sets whether to use an arbitrary range when executing the marker search.

When the trace coupling is off, the active trace is the target to be set.

Variable

| | ***Status*** | |
|---|---|---|
| Description | Selects the search range. | |
| Data type | Boolean type (Boolean) | |
| Range | Select from the following. | |
| | •True or -1 | Specifies an arbitrary range*1. |
| | •False or 0 | Specifies the entire sweep range. |
| Preset value | False or 0 | |

*1. Specify with the SCPI.CALCulate(Ch).SELected.MARKer.FUNCtion. DOMain.STARt object and the SCPI.CALCulate(Ch).SELected.MARKer.FUNCtion. DOMain.STOP object.

For information on the variable (*Ch*), refer to Table 7-4, “Variable (Ch),” on page 126.

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 on page 177

SCPI.CALCulate(Ch).SELected.MARKer.FUNCtion. DOMain.STOP on page 179

SCPI.CALCulate(Ch).SELected.MARKer(Mk).FUNCtion. EXECute on page 180

Equivalent key

**[Marker Search]** - **Search Range** - **Search Range [ON/OFF]**

# SCPI.CALCulate(*Ch*).SELected.MARKer.FUNCtion. DOMain.STOP

Object type

Property

Syntax

SCPI.CALCulate(*Ch*).SELected.MARKer.FUNCtion.DOMain.STOP = *Value*

*Value* = SCPI.CALCulate(*Ch*).SELected.MARKer.FUNCtion.DOMain.STOP

Description

For channels 1 to 4 (*Ch*), sets the stop value of the marker search range.

When the trace coupling is off, the active trace is the target to be set.

Variable

| | ***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) |

For information on the variable (*Ch*), refer to Table 7-4, “Variable (Ch),” on page 126.

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 on page 177

SCPI.CALCulate(Ch).SELected.MARKer.FUNCtion. DOMain.STATe on page 178

SCPI.CALCulate(Ch).SELected.MARKer(Mk).FUNCtion. EXECute on page 180

Equivalent key

**[Marker Search]** - **Search Range** - **Stop**

# SCPI.CALCulate(*Ch*).SELected.MARKer(*Mk*).FUNCtion. EXECute

Object type — Method

Syntax — SCPI.CALCulate(*Ch*).SELected.MARKer(*Mk*).FUNCtion.EXECute

Description — For the active trace of channels 1 to 4 (*Ch*), executes search with marker 1 to 9 (*Mk*) and reference marker (*Mk*:10).

To specify the type of the search, use the SCPI.CALCulate(Ch).SELected.MARKer(Mk).FUNCtion. TYPE object. (No read)

Variable — For information on the variable (*Ch*) and the variable (*Mk*), see Table 7-4, "Variable (Ch)," on page 126 and Table 7-6, "Variable (Mk)," on page 170, respectively.

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 on page 128

SCPI.CALCulate(Ch).SELected.MARKer(Mk).FUNCtion. TYPE on page 192

SCPI.CALCulate(Ch).SELected.MARKer.FUNCtion. DOMain.STATe on page 178

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.

# SCPI.CALCulate(*Ch*).SELected.MARKer.FUNCtion.MULTi.PEXCursion

Object type

Property

Syntax

SCPI.CALCulate(*Ch*).SELected.MARKer(*Mk*).FUNCtion.PEXCursion = *Value*

*Value* = SCPI.CALCulate(*Ch*).SELected.MARKer(*Mk*).FUNCtion.PEXCursion

Description

For the active trace of channels 1 to 4 (*Ch*), sets the lower limit of peak excursion value when executing the multi peak search. For information on the peak excursion value, see Section “Searching for the Peak” in the *E5061A/E5062A User’s Guide*.

Variable

| | ***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.<br>• Log magnitude (MLOG): dB (decibel)<br>• Phase (PHAS), Expanded phase (UPH) or Positive phase (PPH): ° (degree)<br>• Group delay (GDEL): s (second)<br>• 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. |

For information on the variable (*Ch*) and the variable (*Mk*), see Table 7-4, “Variable (Ch),” on page 126 and Table 7-6, “Variable (Mk),” on page 170, respectively.

Examples

```
Dim PeakExc As Double
SCPI.CALCulate(1).PARameter(1).SELect
SCPI.CALCulate(1).SELected.MARKer.FUNCtion.MULTi.TYPE = "peak"
SCPI.CALCulate(1).SELected.MARKer.FUNCtion.MULTi.PEXCursion = 0.2
PeakExc =
SCPI.CALCulate(1).SELected.MARKer.FUNCtion.MULTi.PEXCursion
```

Related objects

Equivalent key

**[Marker Search] - Multi Peak - Peak Excursion**

# SCPI.CALCulate(*Ch*).SELected.MARKer.FUNCtion.MULTi.PPOLarity

Object type

Property

Syntax

SCPI.CALCulate(*Ch*).SELected.MARKer(*Mk*).FUNCtion.MULTi.PPOLarity = *Param*

*Param* = SCPI.CALCulate(*Ch*).SELected.MARKer(*Mk*).FUNCtion.MULTi.PPOLarity

Description

For the active trace of channels 1 to 4 (*Ch*), selects the polarity of the multi peak search.

Variable

| | ***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" | |

For information on the variable (*Ch*) and the variable (*Mk*), see Table 7-4, “Variable (Ch),” on page 126 and Table 7-6, “Variable (Mk),” on page 170, respectively.

Examples

```
Dim PeakPol As String
SCPI.CALCulate(1).PARameter(1).SELect
SCPI.CALCulate(1).SELected.MARKer.FUNCtion.MULTi.TYPE = "peak"
SCPI.CALCulate(1).SELected.MARKer.FUNCtion.MULTi.PPOLarity = "both"
PeakPol =
SCPI.CALCulate(1).SELected.MARKer.FUNCtion.MULTi.PPOLarity
```

Related objects

SCPI.CALCulate(Ch).PARameter(Tr).SELect on page 128

SCPI.CALCulate(Ch).SELected.MARKer(Mk).FUNCtion. TYPE on page 192

SCPI.CALCulate(Ch).SELected.MARKer(Mk).FUNCtion. PEXCursion on page 187

Equivalent key

**[Marker Search]** - **Multi Peak** - **Peak Polarity**

# SCPI.CALCulate(*Ch*).SELected.MARKer.FUNCtion.MULTi.TARGet

Object type

Property

Syntax

SCPI.CALCulate(*Ch*).SELected.MARKer(*Mk*).FUNCtion.MULTi.TARGet = *Value*

*Value* = SCPI.CALCulate(*Ch*).SELected.MARKer(*Mk*).FUNCtion.MULTi.TARGet

Description

For the active trace of channels 1 to 4 (*Ch*), sets the target value to be searched with the multi target search function.

Variable

| | ***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.<br>• Log magnitude (MLOG): dB (decibel)<br>• Phase (PHAS), Expanded phase (UPH) or Positive phase (PPH): ° (degree)<br>• Group delay (GDEL): s (second)<br>• 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. |

For information on the variable (*Ch*) and the variable (*Mk*), see Table 7-4, “Variable (Ch),” on page 126 and Table 7-6, “Variable (Mk),” on page 170, respectively.

Examples

```
Dim TargVal As Double
SCPI.CALCulate(1).PARameter(1).SELect
SCPI.CALCulate(1).SELected.MARKer.FUNCtion.MULTi.TARGet = -12.5
TargVal = SCPI.CALCulate(1).SELected.MARKer.FUNCtion.MULTi.TARGet
```

Related objects

SCPI.CALCulate(Ch).PARameter(Tr).SELect on page 128

SCPI.CALCulate(Ch).SELected.MARKer(Mk).FUNCtion. TYPE on page 192

SCPI.CALCulate(Ch).SELected.MARKer(Mk).FUNCtion. TTRansition on page 191

Equivalent key

**[Marker Search]** - **Multi Target** - **Target Value**

# SCPI.CALCulate(*Ch*).SELected.MARKer.FUNCtion.MULTi.TRACking

Object type

Property

Syntax

SCPI.CALCulate(*Ch*).SELected.MARKer(*Mk*).FUNCtion.MULTi.TRACking = *Status*

*Status* = SCPI.CALCulate(*Ch*).SELected.MARKer(*Mk*).FUNCtion.MULTi.TRACking

Description

For the active trace of channels 1 to 4 (*Ch*), turns ON/OFF the search tracking (function to repeat search for each sweep) of the multi search.

Variable

| | ***Status*** |
|---|---|
| Description | ON/OFF of the marker search tracing |
| Data type | Boolean type (Boolean) |
| Range | Select from the following.<br>•True or -1 Turns ON the search tracking.<br>•False or 0 Turns OFF the search tracking. |
| Preset value | False or 0 |

For information on the variable (*Ch*) and the variable (*Mk*), see Table 7-4, “Variable (Ch),” on page 126 and Table 7-6, “Variable (Mk),” on page 170, respectively.

Examples

```
Dim SrchTrac As Boolean
SCPI.CALCulate(1).PARameter(1).SELect
SCPI.CALCulate(1).SELected.MARKer.FUNCtion.MULTi.TYPE = "targ"
SCPI.CALCulate(1).SELected.MARKer.FUNCtion.MULTi.TRACking = True
SrchTrac =
SCPI.CALCulate(1).SELected.MARKer.FUNCtion.MULTi.TRACking
```

Related objects

Equivalent key

**[Marker Search] - Tracking**

# SCPI.CALCulate(*Ch*).SELected.MARKer.FUNCtion.MULTi.TTRansition

Object type

Property

Syntax

SCPI.CALCulate(*Ch*).SELected.MARKer(*Mk*).FUNCtion.MULTi.TTRansition = *Param*

*Param* = SCPI.CALCulate(*Ch*).SELected.MARKer(*Mk*).FUNCtion.MULTi.TTRansition

Description

For the active trace of channels 1 to 4 (*Ch*), selects the transition type of the multi target search. For more information on the transition type, see Section “Searching for the Target Value” in the *E5061A/E5062A User’s Guide*.

Variable

| | ***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" | |

For information on the variable (*Ch*) and the variable (*Mk*), see Table 7-4, “Variable (Ch),” on page 126 and Table 7-6, “Variable (Mk),” on page 170, respectively.

Examples

```
Dim TargTran As String
SCPI.CALCulate(1).PARameter(1).SELect
SCPI.CALCulate(1).SELected.MARKer.FUNCtion.MULTi.TYPE = "targ"
SCPI.CALCulate(1).SELected.MARKer.FUNCtion.MULTi.TTRansition =
"neg"
TargTran =
SCPI.CALCulate(1).SELected.MARKer.FUNCtion.MULTi.TTRansition
```

Related objects

Equivalent key

**[Marker Search] - Multi Target - Target Transition**

# SCPI.CALCulate(*Ch*).SELected.MARKer.FUNCtion.MULTi.TYPE

Object type

Property

Syntax

SCPI.CALCulate(*Ch*).SELected.MARKer(*Mk*).FUNCtion.MULTi.TYPE = *Param*

*Param* = SCPI.CALCulate(*Ch*).SELected.MARKer(*Mk*).FUNCtion.MULTi.TYPE

Description

For the active trace of channels 1 to 4 (*Ch*), selects the search type for marker 1 to 9 (*Mk*) and reference marker (*Mk*:10).

Variable

| | ***Param*** |
|---|---|
| Description | Search type of marker |
| Data type | Character string type (String) |
| Range | Select from the following.<br>•"OFF" Turn off the multi search function.<br>•"PEAK" Sets the search type to the peak search<br>•"TARGet" Sets the search type to the target search. |
| Preset value | "MAXimum" |

For information on the variable (*Ch*) and the variable (*Mk*), see Table 7-4, “Variable (Ch),” on page 126 and Table 7-6, “Variable (Mk),” on page 170, respectively.

Examples

```
Dim SrchType As String
SCPI.CALCulate(1).PARameter(1).SELect
SCPI.CALCulate(1).SELected.MARKer.FUNCtion.MULTi.TYPE = "targ"
SrchType = SCPI.CALCulate(1).SELected.MARKer.FUNCtion.MULTi.TYPE
```

Related objects

Equivalent key

**[Marker Search]** - **Max**|**Min**

**[Marker Search]** - **Multi Peak** - **Search Multi Peak**

**[Marker Search]** - **Multi Target** - **Search Multi Target**

# SCPI.CALCulate(*Ch*).SELected.MARKer(*Mk*).FUNCtion. PEXCursion

Object type

Property

Syntax

SCPI.CALCulate(*Ch*).SELected.MARKer(*Mk*).FUNCtion.PEXCursion = *Value*

*Value* = SCPI.CALCulate(*Ch*).SELected.MARKer(*Mk*).FUNCtion.PEXCursion

Description

For the active trace of channels 1 to 4 (*Ch*), sets 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 marker 1 to 9 (*Mk*) and reference marker (*Mk*:10). For information on the peak excursion value, see Section "Searching for the Peak" in the *E5061A/E5062A User's Guide*.

Variable

| | ***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.<br>• Log magnitude (MLOG): dB (decibel)<br>• Phase (PHAS), Expanded phase (UPH) or Positive phase (PPH): ° (degree)<br>• Group delay (GDEL): s (second)<br>• 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. |

For information on the variable (*Ch*) and the variable (*Mk*), see Table 7-4, "Variable (Ch)," on page 126 and Table 7-6, "Variable (Mk)," on page 170, respectively.

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 on page 128

SCPI.CALCulate(Ch).SELected.MARKer(Mk).FUNCtion. TYPE on page 192

SCPI.CALCulate(Ch).SELected.MARKer(Mk).FUNCtion. PPOLarity on page 188

Equivalent key

**[Marker Search]** - **Peak** - **Peak Excursion**

# SCPI.CALCulate(*Ch*).SELected.MARKer(*Mk*).FUNCtion. PPOLarity

Object type

Property

Syntax

SCPI.CALCulate(*Ch*).SELected.MARKer(*Mk*).FUNCtion.PPOLarity = *Param*

*Param* = SCPI.CALCulate(*Ch*).SELected.MARKer(*Mk*).FUNCtion.PPOLarity

Description

For the active trace of channels 1 to 4 (*Ch*), selects the polarity of the peak search with marker 1 to 9 (*Mk*) and reference marker (*Mk*:10).

Variable

| | ***Param*** |
|---|---|
| Description | Polarity for peak search |
| Data type | Character string type (String) |
| Range | Select from the following.<br>•"POSitive" Specifies the positive peak.<br>•"NEGative" Specifies the negative peak.<br>•"BOTH" Specifies both the positive peak and the negative peak. |
| Preset value | "POSitive" |

For information on the variable (*Ch*) and the variable (*Mk*), see Table 7-4, “Variable (Ch),” on page 126 and Table 7-6, “Variable (Mk),” on page 170, respectively.

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 on page 128

SCPI.CALCulate(Ch).SELected.MARKer(Mk).FUNCtion. TYPE on page 192

SCPI.CALCulate(Ch).SELected.MARKer(Mk).FUNCtion. PEXCursion on page 187

Equivalent key

**[Marker Search]** - **Peak** - **Peak Polarity**

# SCPI.CALCulate(*Ch*).SELected.MARKer(*Mk*).FUNCtion. TARGet

Object type

Property

Syntax

SCPI.CALCulate(*Ch*).SELected.MARKer(*Mk*).FUNCtion.TARGet = *Value*

*Value* = SCPI.CALCulate(*Ch*).SELected.MARKer(*Mk*).FUNCtion.TARGet

Description

For the active trace of channels 1 to 4 (*Ch*), sets the target value to be searched with marker 1 to 9 (*Mk*) and reference marker (*Mk*:10).

Variable

| | ***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.<br>• Log magnitude (MLOG): dB (decibel)<br>• Phase (PHAS), Expanded phase (UPH) or Positive phase (PPH): ° (degree)<br>• Group delay (GDEL): s (second)<br>• 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. |

For information on the variable (*Ch*) and the variable (*Mk*), see Table 7-4, “Variable (Ch),” on page 126 and Table 7-6, “Variable (Mk),” on page 170, respectively.

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 on page 128

SCPI.CALCulate(Ch).SELected.MARKer(Mk).FUNCtion. TYPE on page 192

SCPI.CALCulate(Ch).SELected.MARKer(Mk).FUNCtion. TTRansition on page 191

Equivalent key

**[Marker Search]** - **Target** - **Target Value**

# SCPI.CALCulate(*Ch*).SELected.MARKer(*Mk*).FUNCtion. TRACking

Object type

Property

Syntax

SCPI.CALCulate(*Ch*).SELected.MARKer(*Mk*).FUNCtion.TRACking = *Status*

*Status* = SCPI.CALCulate(*Ch*).SELected.MARKer(*Mk*).FUNCtion.TRACking

Description

For the active trace of channels 1 to 4 (*Ch*), turns ON/OFF the search tracking (function to repeat search for each sweep) for marker 1 to 9 (*Mk*) and reference marker (*Mk*:10).

Variable

| | ***Status*** |
|---|---|
| Description | ON/OFF of the marker search tracing |
| Data type | Boolean type (Boolean) |
| Range | Select from the following.<br>•True or -1 Turns ON the search tracking.<br>•False or 0 Turns OFF the search tracking. |
| Preset value | False or 0 |

For information on the variable (*Ch*) and the variable (*Mk*), see Table 7-4, “Variable (Ch),” on page 126 and Table 7-6, “Variable (Mk),” on page 170, respectively.

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 on page 128

SCPI.CALCulate(Ch).SELected.MARKer(Mk).FUNCtion. TYPE on page 192

SCPI.CALCulate(Ch).SELected.MARKer(Mk).FUNCtion. EXECute on page 180

Equivalent key

**[Marker Search]** - **Tracking**

# SCPI.CALCulate(*Ch*).SELected.MARKer(*Mk*).FUNCtion. TTRansition

Object type

Property

Syntax

SCPI.CALCulate(*Ch*).SELected.MARKer(*Mk*).FUNCtion.TTRansition = *Param*

*Param* = SCPI.CALCulate(*Ch*).SELected.MARKer(*Mk*).FUNCtion.TTRansition

Description

For marker 1 to 9 (*Mk*) and reference marker (*Mk*:10) of the active trace of channels 1 to 4 (*Ch*), selects the transition type of the target search. For more information on the transition type, see Section “Searching for the Target Value” in the *E5061A/E5062A User’s Guide*.

Variable

| | ***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" | |

For information on the variable (*Ch*) and the variable (*Mk*), see Table 7-4, “Variable (Ch),” on page 126 and Table 7-6, “Variable (Mk),” on page 170, respectively.

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 on page 128

SCPI.CALCulate(Ch).SELected.MARKer(Mk).FUNCtion. TYPE on page 192

SCPI.CALCulate(Ch).SELected.MARKer(Mk).FUNCtion. TARGet on page 189

Equivalent key

**[Marker Search]** - **Target** - **Target Transition**

# SCPI.CALCulate(*Ch*).SELected.MARKer(*Mk*).FUNCtion. TYPE

Object type

Property

Syntax

SCPI.CALCulate(*Ch*).SELected.MARKer(*Mk*).FUNCtion.TYPE = *Param*

*Param* = SCPI.CALCulate(*Ch*).SELected.MARKer(*Mk*).FUNCtion.TYPE

Description

For the active trace of channels 1 to 4 (*Ch*), selects the search type for marker 1 to 9 (*Mk*) and reference marker (*Mk*:10).

Variable

| | ***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*1. |
| | •"LPEak" | Sets the search type to the peak search*1 to the left from the marker position. |
| | •"RPEak" | Sets the search type to the peak search*1 to the right from the marker position. |
| | •"TARGet" | Sets the search type to the target search*2. |
| | •"LTARget" | Sets the search type to the target search*2 to the left from the marker position. |
| | •"RTARget" | Sets the search type to the target search*2 to the right from the marker position. |
| Preset value | "MAXimum" | |

*1. To specify the conditions of the peak, use the SCPI.CALCulate(Ch).SELected.MARKer(Mk).FUNCtion. PEXCursion object and the SCPI.CALCulate(Ch).SELected.MARKer(Mk).FUNCtion. PPOLarity object.

*2. To specify the conditions of the target, use the SCPI.CALCulate(Ch).SELected.MARKer(Mk).FUNCtion. TARGet object and the SCPI.CALCulate(Ch).SELected.MARKer(Mk).FUNCtion. TTRansition object.

For information on the variable (*Ch*) and the variable (*Mk*), see Table 7-4, “Variable (Ch),” on page 126 and Table 7-6, “Variable (Mk),” on page 170, respectively.

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 on page 128

SCPI.CALCulate(Ch).SELected.MARKer(Mk).FUNCtion. PEXCursion on page 187

SCPI.CALCulate(Ch).SELected.MARKer(Mk).FUNCtion. PPOLarity on page 188

SCPI.CALCulate(Ch).SELected.MARKer(Mk).FUNCtion. TARGet on page 189

SCPI.CALCulate(Ch).SELected.MARKer(Mk).FUNCtion. TTRansition on page 191

SCPI.CALCulate(Ch).SELected.MARKer(Mk).FUNCtion. EXECute on page 180

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.

# SCPI.CALCulate(*Ch*).SELected.MARKer.MATH.FLATness.DATA

**Object type** Property

Syntax *Data* = SCPI.CALCulate(*Ch*).SELected.MARKer.MATH.FLATness.DATA

Description Reads out the marker flatness values of the active trace of channels 1 to 4 (*Ch*). (Read only)

**Variable**

| | ***Data*** |
|---|---|
| Description | Indicates 4-element array data (statistics value).<br>• *Data(0)* Span<br>• *Data(1)* Gain<br>• *Data(2)* Slope<br>• *Data(3)* Flatness<br>The index of the array starts from 0. |
| Data type | Variant type (Variant) |

For information on the variable (*Ch*), see Table 7-4, “Variable (Ch),” on page 126.

**Examples**

```
Dim FlatData As Variant
SCPI.CALCulate(1).PARameter(1).SELect
SCPI.CALCulate(1).SELected.MARKer.MATH.FLATness.STATe = True
FlatData = SCPI.CALCulate(1).SELected.MARKer.MATH.FLATness.DATA
```

Related objects SCPI.CALCulate(Ch).SELected.MARKer.MATH.FLATness.STATe on page 195

Equivalent key No equivalent key is available on the front panel.

# SCPI.CALCulate(*Ch*).SELected.MARKer.MATH.FLATness.STATe

Object type

Property

Syntax

SCPI.CALCulate(*Ch*).SELected.MARKer.MATH.FLATness.STATe = *Status*

*Status* = SCPI.CALCulate(*Ch*).SELected.MARKer.MATH.FLATness.STATe

Description

For the active trace of channels 1 to 4 (*Ch*), turns ON/OFF the marker flatness values display.

Variable

| | ***Status*** | |
|---|---|---|
| Description | ON/OFF of the flatness value display | |
| Data type | Boolean type (Boolean) | |
| Range | Select from the following. | |
| | •True or -1 | Turns ON the flatness value display. |
| | •False or 0 | Turns OFF the flatness value display. |
| Preset value | False or 0 | |

For information on the variable (*Ch*), see Table 7-4, “Variable (Ch),” on page 126.

Examples

```
Dim FlatMode As Boolean
SCPI.CALCulate(1).PARameter(1).SELect
SCPI.CALCulate(1).SELected.MARKer.MATH.FLATness.STATe = True
FlatMode = SCPI.CALCulate(1).SELected.MARKer.MATH.FLATness.STATe
```

Related objects

SCPI.CALCulate(Ch).SELected.MARKer.MATH.FLATness.DATA on page 194

Equivalent key

**[Marker Fctn]** - **Flatness**

# SCPI.CALCulate(*Ch*).SELected.MARKer.MATH.FSTatistics.DATA

**Object type** Property

Syntax *Data* = SCPI.CALCulate(*Ch*).SELected.MARKer.MATH.FSTatistics.DATA

Description Reads out the filter statistics values of the active trace of channels 1 to 4 (*Ch*). (Read only)

**Variable**

| | ***Data*** |
|---|---|
| Description | Indicates 3-element array data (statistics value).<br>• *Data(0)* Loss<br>• *Data(1)* Ripple<br>• *Data(2)* Attenuation<br>The index of the array starts from 0. |
| Data type | Variant type (Variant) |

For information on the variable (*Ch*), see Table 7-4, “Variable (Ch),” on page 126.

**Examples**

```
Dim FSTData As Variant
SCPI.CALCulate(1).PARameter(1).SELect
SCPI.CALCulate(1).SELected.MARKer.MATH.FSTatistics.STATe = True
FSTData = SCPI.CALCulate(1).SELected.MARKer.MATH.FSTatistics.DATA
```

Related objects SCPI.CALCulate(Ch).SELected.MARKer.MATH.FSTatistics.STATe on page 197

Equivalent key No equivalent key is available on the front panel.

# SCPI.CALCulate(*Ch*).SELected.MARKer.MATH.FSTatistics.STATe

Object type

Property

Syntax

SCPI.CALCulate(*Ch*).SELected.MARKer.MATH.FSTatistics.STATe = *Status*

*Status* = SCPI.CALCulate(*Ch*).SELected.MARKer.MATH.FSTatistics.STATe

Description

For the active trace of channels 1 to 4 (*Ch*), turns ON/OFF the filter statistics values display.

Variable

| | ***Status*** |
|---|---|
| Description | ON/OFF of the statistics value display |
| Data type | Boolean type (Boolean) |
| Range | Select from the following.<br>•True or -1 Turns ON the statistics value display.<br>•False or 0 Turns OFF the statistics value display. |
| Preset value | False or 0 |

For information on the variable (*Ch*), see Table 7-4, “Variable (Ch),” on page 126.

Examples

```
Dim FSTMode As Boolean
SCPI.CALCulate(1).PARameter(1).SELect
SCPI.CALCulate(1).SELected.MARKer.MATH.FSTatistics.STATe = True
FSTMode = SCPI.CALCulate(1).SELected.MARKer.MATH.FSTatistics.STATe
```

Related objects

SCPI.CALCulate(Ch).SELected.MARKer.MATH.FSTatistics.DATA on page 196

Equivalent key

**[Marker Fctn]** - **RF Filter Stats**

# SCPI.CALCulate(*Ch*).SELected.MARKer.MATH.STATistics.DATA

**Object type** Property

Syntax *Data* = SCPI.CALCulate(*Ch*).SELected.MARKer.MATH.STATistics.DATA

Description Reads out the statistics values of the active trace of channels 1 to 4 (*Ch*). (Read only)

**Variable**

| | ***Data*** |
|---|---|
| Description | Indicates 4-element array data (statistics value).<br>• *Data(0)* Span<br>• *Data(1)* Mean value<br>• *Data(2)* Standard deviation<br>• *Data(3)* Difference between the maximum value and the minimum value (Peak to Peak)<br>The index of the array starts from 0. |
| Data type | Variant type (Variant) |

For information on the variable (*Ch*), see Table 7-4, “Variable (Ch),” on page 126.

**Examples**

```
Dim StatData As Variant
SCPI.CALCulate(1).PARameter(1).SELect
SCPI.CALCulate(1).SELected.MARKer.MATH.STATistics.STATe = True
StatData = SCPI.CALCulate(1).SELected.MARKer.MATH.STATistics.DATA
```

Related objects SCPI.CALCulate(Ch).SELected.MARKer.MATH.STATistics.STATe on page 199

Equivalent key No equivalent key is available on the front panel.

# SCPI.CALCulate(*Ch*).SELected.MARKer.MATH.STATistics.STATe

Object type — Property

Syntax — SCPI.CALCulate(*Ch*).SELected.MARKer.MATH.STATistics.STATe = *Status*

*Status* = SCPI.CALCulate(*Ch*).SELected.MARKer.MATH.STATistics.STATe

Description — For the active trace of channels 1 to 4 (*Ch*), turns ON/OFF the statistics values display.

Variable

| | ***Status*** |
|---|---|
| Description | ON/OFF of the statistics value display |
| Data type | Boolean type (Boolean) |
| Range | Select from the following.<br>•True or -1 Turns ON the statistics value display.<br>•False or 0 Turns OFF the statistics value display. |
| Preset value | False or 0 |

For information on the variable (*Ch*), see Table 7-4, “Variable (Ch),” on page 126.

Examples

```
Dim STATMode As Boolean
SCPI.CALCulate(1).PARameter(1).SELect
SCPI.CALCulate(1).SELected.MARKer.MATH.STATistics.STATe = True
STATMode = SCPI.CALCulate(1).SELected.MARKer.MATH.STATistics.STATe
```

Related objects — SCPI.CALCulate(Ch).SELected.MARKer.MATH.STATistics.DATA on page 198

Equivalent key — **[Marker Fctn]** - **Statistics**

# SCPI.CALCulate(*Ch*).SELected.MARKer(*Mk*).NOTCh. DATA

Object type

Property

Syntax

*Data* = SCPI.CALCulate(*Ch*).SELected.MARKer(*Mk*).NOTCh.DATA

Description

For the active trace of channels 1 to 4 (*Ch*), reads out the notch search result of marker 1 to 9 (*Mk*) and reference marker (*Mk*:10).

If the notch search is impossible, an error occurs when executed and the object is ignored. (Read only)

Variable

| | ***Data*** |
|---|---|
| Description | Indicates 4-element array data (notch search result).<br>• *Data(0)* The bandwidth.<br>• *Data(1)* Center point frequency of the 2 cutoff frequency points.<br>• *Data(2)* The Q value.<br>• *Data(3)* Insertion loss<br>The index of the array starts from 0. |
| Data type | Variant type (Variant) |

For information on the variable (*Ch*) and the variable (*Mk*), see Table 7-4, “Variable (Ch),” on page 126 and Table 7-6, “Variable (Mk),” on page 170, respectively.

Examples

```
Dim NotchData As Variant
SCPI.CALCulate(1).PARameter(1).SELect
NotchData = SCPI.CALCulate(1).SELected.MARKer(1).NOTCh.DATA
```

Related objects

SCPI.CALCulate(Ch).PARameter(Tr).SELect on page 128

SCPI.CALCulate(Ch).SELected.MARKer.NOTCh.STATe on page 201

SCPI.CALCulate(Ch).SELected.MARKer(Mk).NOTCh. THReshold on page 202

Equivalent key

No equivalent key is available on the front panel.

# SCPI.CALCulate(*Ch*).SELected.MARKer.NOTCh.STATe

Object type

Property

Syntax

SCPI.CALCulate(*Ch*).SELected.MARKer.NOTCh.STATe = *Status*

*Status* = SCPI.CALCulate(*Ch*).SELected.MARKer.NOTCh.STATe

Description

For the active trace of channels 1 to 4 (*Ch*), turns ON/OFF the notch search result display.

Variable

| | ***Status*** |
|---|---|
| Description | ON/OFF of the notch search result display |
| Data type | Boolean type (Boolean) |
| Range | Select from the following.<br>•True or -1 Turns ON the notch search result display.<br>•False or 0 Turns OFF the notch search result display. |
| Preset value | False or 0 |

For information on the variable (*Ch*), see Table 7-4, “Variable (Ch),” on page 126 .

Examples

```
Dim NotchMode As Boolean
SCPI.CALCulate(1).PARameter(1).SELect
SCPI.CALCulate(1).SELected.MARKer.NOTCh.STATe = True
NotchMode = SCPI.CALCulate(1).SELected.MARKer.NOTCh.STATe
```

Related objects

SCPI.CALCulate(Ch).PARameter(Tr).SELect on page 128

SCPI.CALCulate(Ch).SELected.MARKer(Mk).NOTCh. DATA on page 200

SCPI.CALCulate(Ch).SELected.MARKer(Mk).NOTCh. THReshold on page 202

Equivalent key

**[Marker Search]** - **Notch**

# SCPI.CALCulate(*Ch*).SELected.MARKer(*Mk*).NOTCh. THReshold

Object type

Property

Syntax

SCPI.CALCulate(*Ch*).SELected.MARKer(*Mk*).NOTCh.THReshold = *Value*

*Value* = SCPI.CALCulate(*Ch*).SELected.MARKer(*Mk*).NOTCh.THReshold

Description

For the active trace of channels 1 to 4 (*Ch*), sets the notch definition value (the value to define the pass-band of the filter) of marker 1 to 9 (*Mk*) and reference marker (*Mk*:10).

Variable

| | ***Value*** |
|---|---|
| Description | Notch 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.<br>• Log magnitude (MLOG): dB (decibel)<br>• Phase (PHAS), Expanded phase (UPH) or Positive phase (PPH): ° (degree)<br>• Group delay (GDEL): s (second)<br>• 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. |

For information on the variable (*Ch*) and the variable (*Mk*), see Table 7-4, “Variable (Ch),” on page 126 and Table 7-6, “Variable (Mk),” on page 170, respectively.

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).PARameter(Tr).SELect on page 128

SCPI.CALCulate(Ch).SELected.MARKer(Mk).NOTCh. DATA on page 200

SCPI.CALCulate(Ch).SELected.MARKer.NOTCh.STATe on page 201

Equivalent key

**[Marker Search]** - **Notch Value**

# SCPI.CALCulate(*Ch*).SELected.MARKer.REFerence. STATe

Object type — Property

Syntax — SCPI.CALCulate(*Ch*).SELected.MARKer.REFerence.STATe = *Status*

*Status* = SCPI.CALCulate(*Ch*).SELected.MARKer.REFerence.STATe

Description — For the active trace of channels 1 to 4 (*Ch*), turns ON/OFF the reference marker mode.

Variable

| | ***Status*** |
|---|---|
| Description | ON/OFF of the reference marker mode |
| Data type | Boolean type (Boolean) |
| Range | Select from the following.<br>•True or -1 Turns ON the reference marker mode.<br>•False or 0 Turns OFF the reference marker mode. |
| Preset value | False or 0 |

For information on the variable (*Ch*), see Table 7-4, “Variable (Ch),” on page 126.

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 on page 128

Equivalent key — **[Marker] - Ref Marker Mode**

# SCPI.CALCulate(*Ch*).SELected.MARKer(*Mk*).SET

Object type

Property

Syntax

SCPI.CALCulate(*Ch*).SELected.MARKer(*Mk*).SET = *Param*

Description

For the active trace of channels 1 to 4 (*Ch*), 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*).

Variable

| | ***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%). |

For information on the variable (*Ch*) and the variable (*Mk*), see Table 7-4, “Variable (Ch),” on page 126 and Table 7-6, “Variable (Mk),” on page 170, respectively.

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 on page 128

SCPI.CALCulate(Ch).SELected.MARKer.REFerence. STATe on page 203

Equivalent key

**[Marker Fctn] - Marker -> Start|Marker -> Stop|Marker -> Center|Marker -> Reference | Marker -> Delay**

# SCPI.CALCulate(*Ch*).SELected.MARKer(*Mk*).STATe

Object type

Property

Syntax

SCPI.CALCulate(*Ch*).SELected.MARKer(*Mk*).STATe = *Status*

*Status* = SCPI.CALCulate(*Ch*).SELected.MARKer(*Mk*).STATe

Description

For the active trace of channels 1 to 4 (*Ch*), turns ON/OFF the display of marker 1 to 9 (*Mk*) and reference marker (*Mk*:10).

Variable

| | ***Status*** |
|---|---|
| Description | ON/OFF of display of markers 1 to 9 and reference marker |
| Data type | Boolean type (Boolean) |
| Range | Select from the following.<br>•True or -1 Turns ON the display of the marker.<br>•False or 0 Turns OFF the display of the marker. |
| Preset value | False or 0 |

For information on the variable (*Ch*) and the variable (*Mk*), see Table 7-4, “Variable (Ch),” on page 126 and Table 7-6, “Variable (Mk),” on page 170, respectively.

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 on page 128

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**

**NOTE**

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**

# SCPI.CALCulate(*Ch*).SELected.MARKer(*Mk*).X

Object type — Property

Syntax — SCPI.CALCulate(*Ch*).SELected.MARKer(*Mk*).X = *Value*

*Value* = SCPI.CALCulate(*Ch*).SELected.MARKer(*Mk*).X

Description — For the active trace of channels 1 to 4 (*Ch*), sets the stimulus value for marker 1 to 9 (*Ch*) and reference marker (*Ch*:10).

Variable

| | ***Value*** |
|---|---|
| Description | Stimulus value of the marker*1 |
| Data type | Double precision floating point type (Double) |
| Range | Sweep start value to sweep stop value*2 |
| Preset value | Sweep start value*3 |
| 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. |

*1. When the reference marker mode is ON ("True" is specified with the SCPI.CALCulate(Ch).SELected.MARKer.REFerence. STATe object), it is the value relative to the reference marker.
*2. When the span value of the sweep range is 0, the range is from 0 to sweep time value.
*3. When the span value of the sweep range is 0, the preset value is 0.

For information on the variable (*Ch*) and the variable (*Mk*), see Table 7-4, "Variable (Ch)," on page 126 and Table 7-6, "Variable (Mk)," on page 170, respectively.

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 on page 128

SCPI.CALCulate(Ch).SELected.MARKer.REFerence. STATe on page 203

SCPI.CALCulate(Ch).SELected.MARKer(Mk).Y on page 207

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**

**NOTE** When performing the operation from the front panel, you turn ON the marker and set the stimulus value at the same time.

# SCPI.CALCulate(*Ch*).SELected.MARKer(*Mk*).Y

**Object type** Property

**Syntax** *Data* = SCPI.CALCulate(*Ch*).SELected.MARKer(*Mk*).Y

**Description** For the active trace of channels 1 to 4 (*Ch*), reads out the response value of marker 1 to 9 (*Mk*) and reference marker (*Mk*:10).

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. (Read only)

**Variable**

| | ***Data*** |
|---|---|
| Description | Indicates 2-element array data (response value of marker).<br>• *Data(0)* Response value (primary value) at the marker position.<br>• *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.<br>The index of the array starts from 0. |
| Data type | Variant type (Variant) |

For information on the variable (*Ch*) and the variable (*Mk*), see Table 7-4, "Variable (Ch)," on page 126 and Table 7-6, "Variable (Mk)," on page 170, respectively.

**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 on page 128

SCPI.CALCulate(Ch).SELected.MARKer.REFerence. STATe on page 203

SCPI.CALCulate(Ch).SELected.MARKer(Mk).X on page 206

**Equivalent key** No equivalent key is available on the front panel.

# SCPI.CALCulate(*Ch*).SELected.MATH.FUNCtion

Object type

Property

Syntax

SCPI.CALCulate(*Ch*).SELected.MATH.FUNCtion = *Param*

*Param* = SCPI.CALCulate(*Ch*).SELected.MATH.FUNCtion

Description

For the active trace of channels 1 to 4 (*Ch*), selects the data trace display method (math method between measurement data and memory trace data).

The math result according to this setting is displayed on the data trace.

Variable

| | ***Param*** |
|---|---|
| Description | Math method between measurement data and memory trace data |
| Data type | Character string type (String) |
| Range | Select from the following.<br>•"NORMal" Specifies *Data* (no math).<br>•"DIVide" Specifies *Data / Mem*.<br>•"MULTiply" Specifies *Data × Mem*.<br>•"SUBTract" Specifies *Data - Mem*.<br>•"ADD" Specifies *Data + Mem*.<br>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" |

For information on the variable (*Ch*), see Table 7-4, “Variable (Ch),” on page 126.

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 on page 128

Equivalent key

**[Display] - Data Math - OFF|Data / Mem|Data * Mem|Data – Mem|Data + Mem**

## SCPI.CALCulate(*Ch*).SELected.MATH.MEMorize

Object type — Method

Syntax — SCPI.CALCulate(*Ch*).SELected.MATH.MEMorize

Description — For the active trace of channels 1 to 4 (*Ch*), copies the measurement data at the execution of this object to the memory trace. (No read)

Variable — For information on the variable (*Ch*), see Table 7-4, "Variable (Ch)," on page 126.

Examples

```
SCPI.CALCulate(1).PARameter(1).SELect
SCPI.CALCulate(1).SELected.MATH.MEMorize
```

Related objects — SCPI.CALCulate(Ch).PARameter(Tr).SELect on page 128

Equivalent key — **[Display] - Data –> Mem**

## SCPI.CALCulate(*Ch*).SELected.MSTatistics.DATA

**Object type** — Property

Syntax — *Data* = SCPI.CALCulate(*Ch*).SELected.MSTatistics.DATA

Description — Reads out the statistics values (the mean vale, the standard deviation, and the difference between the maximum value and the minimum value) of the active trace of channels 1 to 4 (*Ch*). (Read only)

**Variable**

| | ***Data*** |
|---|---|
| Description | Indicates 3-element array data (statistics value).<br>• *Data(0)* Mean value<br>• *Data(1)* Standard deviation<br>• *Data(2)* Difference between the maximum value and the minimum value (Peak to Peak)<br>The index of the array starts from 0. |
| Data type | Variant type (Variant) |

For information on the variable (*Ch*), see Table 7-4, "Variable (Ch)," on page 126.

**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 on page 128

SCPI.CALCulate(Ch).SELected.MSTatistics.STATe on page 210

Equivalent key — No equivalent key is available on the front panel.

# SCPI.CALCulate(*Ch*).SELected.MSTatistics.STATe

Object type

Property

Syntax

SCPI.CALCulate(*Ch*).SELected.MSTatistics.STATe = *Status*

*Status* = SCPI.CALCulate(*Ch*).SELected.MSTatistics.STATe

Description

For the active trace of channels 1 to 4 (*Ch*), turns ON/OFF the statistics values (the mean vale, the standard deviation, and the difference between the maximum value and the minimum value) display.

Variable

| | ***Status*** |
|---|---|
| Description | ON/OFF of the statistics value display |
| Data type | Boolean type (Boolean) |
| Range | Select from the following.<br>•True or -1 Turns ON the statistics value display.<br>•False or 0 Turns OFF the statistics value display. |
| Preset value | False or 0 |

For information on the variable (*Ch*), see Table 7-4, “Variable (Ch),” on page 126.

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 on page 128

SCPI.CALCulate(Ch).SELected.MSTatistics.DATA on page 209

Equivalent key

No equivalent key is available on the front panel.

# SCPI.CALCulate(*Ch*).SELected.RLIMit.DATA

Object type

Property

Syntax

SCPI.CALCulate(*Ch*).SELected.RLIMit.DATA = *Data*

*Data* = SCPI.CALCulate(*Ch*).SELected.RLIMit.DATA

Description

For the active trace of channel 1 to channel 4 (specified with the SCPI.CALCulate(Ch).PARameter(Tr).SELect command), sets the ripple limit table.

The data transfer format when this command is executed depends on the setting with the SCPI.FORMat.DATA command.

Variable

| | ***Status*** |
|---|---|
| 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.<br>• *Data(0)* — The number of limit lines you want to set. Specify an integer ranging 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)*.<br>• *Data(n×4-3)* — The type of the n-th line. Specify an integer 0 to 1 as follows. 0: OFF 1: ON<br>• *Data(n×4-2)* — The value on the horizontal axis (frequency/power/time) of the start point of the n-th line.<br>• *Data(n×4-1)* — The value on the horizontal axis (frequency/power/time) of the end point of the n-th line.<br>• *Data(n×4)* — The ripple line value (dB) of the n-th line.<br>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. |

For information on the variable (*Ch*), see Table 7-4, “Variable (Ch),” on page 126.

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 on page 128

SCPI.CALCulate(Ch).SELected.RLIMit.STATe on page 218

Equivalent key

**[Analysis]** - **Ripple Limit** - **Edit Ripple Limit** - **Add**

# SCPI.CALCulate(*Ch*).SELected.RLIMit.DISPlay.LINE

Object type

Property

Syntax

SCPI.CALCulate(Ch).SELected.RLIMit.DISPlay.LINE = Status

Status = SCPI.CALCulate(Ch).SELected.RLIMit.DISPlay.LINE

Description

For the active trace of channel 1 to channel 4 (specified with the SCPI.CALCulate(Ch).PARameter(Tr).SELect command), turns ON/OFF the ripple limit line display.

Variable

| | ***Status*** |
|---|---|
| Description | ON/OFF the ripple limit line display. |
| Data type | Boolean type (Boolean) |
| Range | Select from the following.<br>•True or -1 Turns ON the ripple limit line display.<br>•False or 0 Turns OFF the ripple limit line display. |
| Preset value | False or 0 |

For information on the variable (*Ch*), see Table 7-4, “Variable (Ch),” on page 126.

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 on page 128

SCPI.CALCulate(Ch).SELected.RLIMit.STATe on page 218

SCPI.CALCulate(Ch).SELected.RLIMit.DISPlay.SELect on page 214

SCPI.CALCulate(Ch).SELected.RLIMit.DISPlay.VALue on page 215

Equivalent key

**[Analysis]** - **Ripple Limit** - **Ripple Limit**

# SCPI.CALCulate(*Ch*).SELected.RLIMit.DISPlay.SELect

Object type

Property

Syntax

SCPI.CALCulate(Ch).SELected.RLIMit.DISPlay.SELect = Value

Value = SCPI.CALCulate(Ch).SELected.RLIMit.DISPlay.SELect

Description

For channel 1 to channel 4 (specified with the SCPI.CALCulate(Ch).PARameter(Tr).SELect command), sets the ripple limit band for ripple value display.

Variable

| | ***Value*** |
|---|---|
| Description | The ripple limit band. |
| Data type | Long integer type (Long) |
| Range | 1 to 12 |
| Preset value | 1 |

For information on the variable (*Ch*), see Table 7-4, “Variable (Ch),” on page 126.

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 on page 128

SCPI.CALCulate(Ch).SELected.RLIMit.STATe on page 218

SCPI.CALCulate(Ch).SELected.RLIMit.DISPlay.LINE on page 213

SCPI.CALCulate(Ch).SELected.RLIMit.DISPlay.VALue on page 215

Equivalent key

**[Analysis]** - **Ripple Limit** - **Ripple Band**

# SCPI.CALCulate(*Ch*).SELected.RLIMit.DISPlay.VALue

Object type — Property

Syntax

SCPI.CALCulate(Ch).SELected.RLIMit.DISPlay.VALue = Param

Param = SCPI.CALCulate(Ch).SELected.RLIMit.DISPlay.VALue

Description — For the active trace of channel 1 to channel 4 (specified with the SCPI.CALCulate(Ch).PARameter(Tr).SELect command), selects the display type of ripple value.

Variable

| | ***Param*** |
|---|---|
| Description | The displaying type of ripple value. |
| Data type | Character string type (String) |
| Range | Select from the following.<br>•"OFF" Specifies the display off.<br>•"ABSolute" Specifies the absolute value for display type.<br>•"MARgin" Specifies the margin for display type. |
| Preset value | "OFF" |

For information on the variable (*Ch*), see Table 7-4, “Variable (Ch),” on page 126.

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 on page 128

SCPI.CALCulate(Ch).SELected.RLIMit.STATe on page 218

SCPI.CALCulate(Ch).SELected.RLIMit.DISPlay.LINE on page 213

SCPI.CALCulate(Ch).SELected.RLIMit.DISPlay.SELect on page 214

Equivalent key — **[Analysis]** - **Ripple Limit** - **Ripple Value - OFF**|**Absolute**|**Margin**

# SCPI.CALCulate(*Ch*).SELected.RLIMit.FAIL

Object type

Property

Syntax

*Status* = SCPI.CALCulate(*Ch*).SELected.RLIMit.FAIL

Description

For the active trace of channel 1 to channel 4 (specified with the SCPI.CALCulate(Ch).PARameter(Tr).SELect command), reads out the ripple test result. (Read only)

Variable

| | ***Status*** |
|---|---|
| Description | The ripple test result |
| Data type | Boolean type (Boolean) |
| Range | Select from the following.<br>•True or -1 Turns ON the ripple test result is FAIL.<br>•False or 0 Turns OFF the ripple test result is FAIL. |
| Note | When the ripple test if set to OFF, False or 0 is always read out. |

For information on the variable (*Ch*), see Table 7-4, “Variable (Ch),” on page 126.

Examples

```
Dim Result As Boolean
Result = SCPI.CALCulate(1).SELected.RLIMit.FAIL
```

Related objects

SCPI.CALCulate(Ch).PARameter(Tr).SELect on page 128

SCPI.CALCulate(Ch).SELected.RLIMit.STATe on page 218

Equivalent key

No equivalent key is available on the front panel.

# SCPI.CALCulate(*Ch*).SELected.RLIMit.REPort.DATA

Object type

Property

Syntax

*Data* = SCPI.CALCulate(*Ch*).SELected.RLIMit.REPort.DATA

Description

For the active trace of channel 1 to channel 4 (specified with the SCPI.CALCulate(Ch).PARameter(Tr).SELect command), reads out the ripple value of the ripple test.

The data transfer format when this command is executed depends on the setting with the SCPI.FORMat.DATA command.(Read only)

Variable

| | ***Status*** |
|---|---|
| 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.<br>• *Data(0)* Number of ripple limit line.<br>• *Data(n×3-2)* Number of ripple limit bands.<br>• *Data(n×3-1)* Ripple value.<br>• *Data(n×3)* Rresults of ripple test.<br>Select from the following.<br>0:PASS<br>1:FAIL.<br>The index of the array starts from 0. |
| Data type | Variant type (Variant) |

For information on the variable (*Ch*), see Table 7-4, “Variable (Ch),” on page 126.

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 on page 128

SCPI.CALCulate(Ch).SELected.RLIMit.STATe on page 218

Equivalent key

No equivalent key is available on the front panel.

# SCPI.CALCulate(*Ch*).SELected.RLIMit.STATe

Object type

Property

Syntax

SCPI.CALCulate(*Ch*).SELected.RLIMit.STATe = *Status*

*Status* = SCPI.CALCulate(*Ch*).SELected.RLIMit.STATe

Description

For the active trace of channel 1 to channel 4 (specified with the SCPI.CALCulate(Ch).PARameter(Tr).SELect command), turns ON/OFF the ripple test function.

Variable

| | ***Status*** |
|---|---|
| Description | ON/OFF the ripple test function |
| Data type | Boolean type (Boolean) |
| Range | Select from the following.<br>•True or -1 Turns ON the ripple test function.<br>•False or 0 Turns OFF the ripple test function. |
| Preset value | False or 0 |

For information on the variable (*Ch*), see Table 7-4, "Variable (Ch)," on page 126.

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 on page 128

SCPI.CALCulate(Ch).SELected.RLIMit.DATA on page 211

SCPI.CALCulate(Ch).SELected.RLIMit.DISPlay.LINE on page 213

SCPI.CALCulate(Ch).SELected.RLIMit.DISPlay.SELect on page 214

SCPI.CALCulate(Ch).SELected.RLIMit.DISPlay.VALue on page 215

SCPI.CALCulate(Ch).SELected.RLIMit.FAIL on page 216

SCPI.CALCulate(Ch).SELected.RLIMit.REPort.DATA on page 217

Equivalent key

**[Analysis]** - **Ripple Limit** - **Ripple Limit Test**

# SCPI.CALCulate(*Ch*).SELected.SMOothing.APERture

Object type

Property

Syntax

SCPI.CALCulate(*Ch*).SELected.SMOothing.APERture = *Value*

*Value* = SCPI.CALCulate(*Ch*).SELected.SMOothing.APERture

Description

Sets the smoothing aperture (percentage to the sweep span value) of the active trace of channels 1 to 4 (*Ch*).

Variable

| | ***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. |

For information on the variable (*Ch*), see Table 7-4, “Variable (Ch),” on page 126.

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 on page 128

SCPI.CALCulate(Ch).SELected.SMOothing.STATe on page 220

Equivalent key

**[Avg] - Smo Aperture**

# SCPI.CALCulate(*Ch*).SELected.SMOothing.STATe

Object type

Property

Syntax

SCPI.CALCulate(*Ch*).SELected.SMOothing.STATe = *Status*

*Status* = SCPI.CALCulate(*Ch*).SELected.SMOothing.STATe

Description

For the active trace of channels 1 to 4 (*Ch*), turns ON/OFF the smoothing.

Variable

| | ***Status*** |
|---|---|
| Description | ON/OFF of the smoothing |
| Data type | Boolean type (Boolean) |
| Range | Select from the following.<br>•True or -1 Turns ON the smoothing.<br>•False or 0 Turns OFF the smoothing. |
| Preset value | False or 0 |

For information on the variable (*Ch*), see Table 7-4, “Variable (Ch),” on page 126.

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 on page 128

SCPI.CALCulate(Ch).SELected.SMOothing.APERture on page 219

Equivalent key

**[Avg]** - **Smoothing**

# SCPI.CONTrol.HANDler.A.DATA

Object type

Property

Syntax

SCPI.CONTrol.HANDler.A.DATA = *Value*

Description

Outputs port information to output port A (A0 to A7) of the handler I/O. Port information is outputted as 8-bit binary data using A0 as LSB and A7 as MSB. (No read)

For more information on the handler I/O, see Chapter "Communication with External Instruments Using Handler I/O Port" in the *E5061A/E5062A Programmer's Guide*.

Variable

| | ***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.A.DATA = 15
```

Equivalent key

No equivalent key is available on the front panel.

# SCPI.CONTrol.HANDler.B.DATA

Object type

Property

Syntax

SCPI.CONTrol.HANDler.B.DATA = *Value*

Description

Outputs port information to output port B (B0 to B7) of the handler I/O. Port information is outputted as 8-bit binary data using B0 as LSB and B7 as MSB. (No read)

**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).

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).

For more information on the handler I/O, see Chapter "Communication with External Instruments Using Handler I/O Port" in the *E5061A/E5062A Programmer's Guide*.

Variable

| | ***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.

# SCPI.CONTrol.HANDler.C.DATA

Object type

Property

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, outputs port information to output port C (C0 to C3).

When input/output port C of the handler I/O is set to the input port, reads out port information inputted to port C (C0 to C3).

Port information is inputted/outputted as 4-bit binary data using C0 as LSB and C3 as MSB.

For more information on the handler I/O, see Chapter "Communication with External Instruments Using Handler I/O Port" in the *E5061A/E5062A Programmer's Guide*.

Variable

| | ***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 on page 224

Equivalent key

No equivalent key is available on the front panel.

# SCPI.CONTrol.HANDler.C.MODE

Object type

Property

Syntax

SCPI.CONTrol.HANDler.C.MODE = *Param*

*Param* = SCPI.CONTrol.HANDler.C.MODE

Description

Sets the input/output direction of port C of the handler I/O.

For more information on the handler I/O, see Chapter “Communication with External Instruments Using Handler I/O Port” in the *E5061A/E5062A Programmer's Guide*.

Variable

| | ***Param*** |
|---|---|
| Description | Input/output direction of port C |
| Data type | Character string type (String) |
| Range | Select from the following.<br>•"INPut" Sets the port C to input.<br>•"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 on page 223

Equivalent key

No equivalent key is available on the front panel.

# SCPI.CONTrol.HANDler.D.DATA

Object type

Property

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, 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, reads out port information inputted to port D (D0 to D3).

Port information is outputted as 4-bit binary data using D0 as LSB and D3 as MSB.

For more information on the handler I/O, see Chapter "Communication with External Instruments Using Handler I/O Port" in the *E5061A/E5062A Programmer's Guide*.

Variable

| | ***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 on page 226

Equivalent key

No equivalent key is available on the front panel.

# SCPI.CONTrol.HANDler.D.MODE

Object type

Property

Syntax

SCPI.CONTrol.HANDler.D.MODE = *Param*

*Param* = SCPI.CONTrol.HANDler.D.MODE

Description

Sets the input/output direction of port D of the handler I/O.

For more information on the handler I/O, see Chapter “Communication with External Instruments Using Handler I/O Port” in the *E5061A/E5062A Programmer's Guide*.

Variable

| | ***Param*** |
|---|---|
| Description | Input/output direction of port D |
| Data type | Character string type (String) |
| Range | Select from the following.<br>•"INPut" Sets the port D to input.<br>•"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 on page 225

Equivalent key

No equivalent key is available on the front panel.

# SCPI.CONTrol.HANDler.E.DATA

Object type

Property

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, 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, reads out port information inputted to port E (C0 to D3).

Port information is outputted as 8-bit binary data using C0 as LSB and D3 as MSB.

For more information on the handler I/O, see Chapter "Communication with External Instruments Using Handler I/O Port" in the *E5061A/E5062A Programmer's Guide*.

Variable

| | ***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 on page 224

SCPI.CONTrol.HANDler.D.MODE on page 226

SCPI.CONTrol.HANDler.C.DATA on page 223

SCPI.CONTrol.HANDler.D.DATA on page 225

Equivalent key

No equivalent key is available on the front panel.

# SCPI.CONTrol.HANDler.EXTension.INDex.STATe

Object type

Property

Syntax

SCPI.CONTrol.HANDler.EXTension.INDex.STATe = *Status*

*Status* = SCPI.CONTrol.HANDler.EXTension.INDex.STATe

Description

Turns ON/OFF outputting the INDEX signal to B6 of the handler I/O.

For more information on the handler I/O and the INDEX signal, see Chapter "Communication with External Instruments Using Handler I/O Port" in the *E5061A/E5062A Programmer's Guide*.

**NOTE**

When you use port B6 as the output port, turn OFF the INDEX signal output. When outputting the INDEX signal is turned ON, the bit 6 of the data outputted 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

| | ***Status*** |
|---|---|
| Description | ON/OFF of the INDEX signal output |
| Data type | Boolean type (Boolean) |
| Range | Select from the following.<br>•True or -1 Turns ON the INDEX signal output.<br>•False or 0 Turns OFF the INDEX signal output. |
| Preset value | False or 0 |

Examples

```
Dim Indx As Boolean
SCPI.CONTrol.HANDler.EXTension.INDex.STATe = True
Indx = SCPI.CONTrol.HANDler.EXTension.INDex.STATe
```

Related objects

SCPI.CONTrol.HANDler.EXTension.RTRigger.STATe on page 229

Equivalent key

No equivalent key is available on the front panel.

# SCPI.CONTrol.HANDler.EXTension.RTRigger.STATe

Object type

Property

Syntax

SCPI.CONTrol.HANDler.EXTension.RTRigger.STATe = *Status*

*Status* = SCPI.CONTrol.HANDler.EXTension.RTRigger.STATe

Description

Turns ON/OFF outputting the READY FOR TRIGGER signal to B7 of the handler I/O.

For more information on the handler I/O and the INDEX signal, see Chapter "Communication with External Instruments Using Handler I/O Port" in the *E5061A/E5062A Programmer's Guide*.

**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 outputted by the SCPI.CONTrol.HANDler.B.DATA object (the bit 15 of the data outputted by the SCPI.CONTrol.HANDler.F.DATA object) is ignored.

Variable

| | ***Status*** |
|---|---|
| Description | ON/OFF of the READY FOR TRIGGER signal output |
| Data type | Boolean type (Boolean) |
| Range | Select from the following.<br>•True or -1 Turns ON the READY FOR TRIGGER signal output.<br>•False or 0 Turns OFF the READY FOR TRIGGER signal output. |
| Preset value | False or 0 |

Examples

```
Dim RdyTrig As Boolean
SCPI.CONTrol.HANDler.EXTension.RTRigger.STATe = True
RdyTrig = SCPI.CONTrol.HANDler.EXTension.RTRigger.STATe
```

Related objects

SCPI.CONTrol.HANDler.EXTension.INDex.STATe on page 228

Equivalent key

No equivalent key is available on the front panel.

# SCPI.CONTrol.HANDler.F.DATA

Object type — Property

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 outputted as 16-bit binary using A0 as LSB and B7 as MSB. (No read)

**NOTE**

The bit 14 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).

The bit 15 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).

For more information on the handler I/O, see Chapter "Communication with External Instruments Using Handler I/O Port" in the *E5061A/E5062A Programmer's Guide*.

Variable

| | ***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 on page 221

SCPI.CONTrol.HANDler.B.DATA on page 222

Equivalent key — No equivalent key is available on the front panel.

# SCPI.CONTrol.HANDler.OUTPut(*Num*).DATA

Object type

Property

Syntax

SCPI.CONTrol.HANDler.OUTPut(*Num*) = *Value*

*Value* = SCPI.CONTrol.HANDler.OUTPut(*Num*)

Description

Sets HIGH / LOW of OUTPUT1 (*Num*:1) or OUTPUT2 (*Num*:2) of the handler I/O.

For more information on the handler I/O, see Chapter "Communication with External Instruments Using Handler I/O Port" in the *E5061A/E5062A Programmer's Guide*.

Variable

| | ***Num*** |
|---|---|
| Description | Number of the OUTPUT terminal |
| 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. |

| | ***Value*** |
|---|---|
| Description | Polarity (High/Low) |
| Data type | Long integer type (Long) |
| Range | Select from the following.<br>•1 Specifies LOW.<br>•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.

# SCPI.DISPlay.ANNotation.FREQuency.STATe

Object type — Property

Syntax — SCPI.DISPlay.ANNotation.FREQuency.STATe = *Status*

*Status* = SCPI.DISPlay.ANNotation.FREQuency.STATe

Description — Turns ON/OFF the frequency display on the LCD display.

Variable

| | ***Status*** |
|---|---|
| Description | ON/OFF of the frequency display |
| Data type | Boolean type (Boolean) |
| Range | Select from the following.<br>•True or -1 Turns ON the frequency display.<br>•False or 0 Turns OFF the frequency display. |
| Preset value | True or -1 |

Examples

```
Dim DispFreq As Boolean
SCPI.DISPlay.ANNotation.FREQuency.STATe = False
DispFreq = SCPI.DISPlay.ANNotation.FREQuency.STATe
```

Equivalent key — **[Display]** - **Frequency**

# SCPI.DISPlay.CCLear

Object type — Method

Syntax — SCPI.DISPlay.CCLear

Description — Clears the error message display on the instrument status bar (at the bottom of the LCD display). (No read)

Examples

```
SCPI.DISPlay.CCLear
```

Equivalent key — No equivalent key is available on the front panel.

# SCPI.DISPlay.CLOCk

Object type

Property

Syntax

SCPI.DISPlay.CLOCk = *Status*

*Status* = SCPI.DISPlay.CLOCk

Description

Turns ON/OFF the clock display at the right edge of the instrument status bar (at the bottom of the LCD display).

Variable

| | ***Status*** |
|---|---|
| Description | ON/OFF of the clock display |
| Data type | Boolean type (Boolean) |
| Range | Select from the following.<br>•True or -1 Turns ON the clock display.<br>•False or 0 Turns OFF the clock display. |
| Preset value | True or -1 |

Examples

```
Dim DispTime As Boolean
SCPI.DISPlay.CLOCk = False
DispTime = SCPI.DISPlay.CLOCk
```

Equivalent key

**[System] - Misc Setup - Clock Setup - Show Clock**

# SCPI.DISPlay.COLor(*Dnum*).BACK

Object type — Property

Syntax — SCPI.DISPlay.COLor(*Dnum*).BACK = *Data*

*Data* = SCPI.DISPlay.COLor(*Dnum*).BACK

Description — Sets the background color for normal display (*Dnum*: 1) and inverted display (*Dnum*: 2).

Variable

**Table 7-7 Variable(*Dnum*)**

| | ***Dnum*** |
|---|---|
| Description | The number of display mode<br>1: normal display<br>2: inverted display |
| 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. |

| | ***Data*** |
|---|---|
| Description | Indicates 3-element array data.<br>• *Data(0)* Sets amount of red.<br>• *Data(1)* Sets amount of green.<br>• *Data(2)* Sets amount of blue.<br>The index of the array starts from 0. |
| Data type | Variant type (Variant) |
| Range | • *Data(0)* 0 to 5<br>• *Data(1)* 0 to 5<br>• *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 on page 237

Equivalent key — **[System] - Misc Setup - Color Setup - Normal|Invert - Background**

# SCPI.DISPlay.COLor(*Dnum*).GRATicule(*Gnum*)

**Object type** Property

**Syntax** SCPI.DISPlay.COLor(*Dnum*).GRATicule(*Gnum*) = *Data*

*Data* = SCPI.DISPlay.COLor(*Dnum*).GRATicule(*Gnum*)

**Description** Sets the color of the graticule label and the outer frame line of the graph (*Gnum*: 1) and the color of the grid line of the graph (*Gnum*: 2) for normal display (*Dnum*: 1) and inverted display (*Dnum*: 2).

**Variable**

| | ***Gnum*** |
|---|---|
| Description | The number of item<br>1: The outer frame line of the graph<br>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. |

| | ***Data*** |
|---|---|
| Description | Indicates 3-element array data.<br>• *Data(0)* Sets amount of red.<br>• *Data(1)* Sets amount of green.<br>• *Data(2)* Sets amount of blue.<br>The index of the array starts from 0. |
| Data type | Variant type (Variant) |
| Range | • *Data(0)* 0 to 5<br>• *Data(1)* 0 to 5<br>• *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. |

For information on the variable (*Dnum*), see Table 7-7, “Variable(Dnum),” on page 234.

**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 on page 237

**Equivalent key** **[System]** - **Misc Setup** - **Color Setup** - **Normal**|**Invert** - **Graticule Main**|**Graticule Sub**

# SCPI.DISPlay.COLor(*Dnum*).LIMit(*Lnum*)

Object type

Property

Syntax

SCPI.DISPlay.COLor(*Dnum*).LIMit(*Lnum*) = *Data*

*Data* = SCPI.DISPlay.COLor(*Dnum*).LIMit(*Lnum*)

Description

Sets the fail display color used for the limit test result (*Lnum*: 1) and the color of the limit line (*Lnum*: 2) for normal display (*Dnum*: 1) and inverted display (*Dnum*: 2).

Variable

| | ***Lnum*** |
|---|---|
| Description | The number of item<br>1: The limit test result<br>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. |

| | ***Data*** |
|---|---|
| Description | Indicates 3-element array data.<br>• *Data(0)* Sets amount of red.<br>• *Data(1)* Sets amount of green.<br>• *Data(2)* Sets amount of blue.<br>The index of the array starts from 0. |
| Data type | Variant type (Variant) |
| Range | • *Data(0)* 0 to 5<br>• *Data(1)* 0 to 5<br>• *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. |

For information on the variable (*Dnum*), see Table 7-7, “Variable(Dnum),” on page 234.

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 on page 237

Equivalent key

**[System] - Misc Setup - Color Setup - Normal|Invert - Limit Fail|Limit Line**

# SCPI.DISPlay.COLor(*Dnum*).RESet

| | |
|---|---|
| Object type | Method |
| Syntax | SCPI.DISPlay.COLor(*Dnum*).RESet |
| Description | Resets the display color settings for all the items to the factory preset state for normal display (*Dnum*: 1) and inverted display (*Dnum*: 2). (No read) |
| Variable | For information on the variable (*Dnum*), see Table 7-7, "Variable(Dnum)," on page 234. |
| Examples | `SCPI.DISPlay.COLor(1).RESet` |
| Related objects | SCPI.DISPlay.COLor(Dnum).BACK on page 234<br>SCPI.DISPlay.COLor(Dnum).GRATicule(Gnum) on page 235<br>SCPI.DISPlay.COLor(Dnum).LIMit(Lnum) on page 236<br>SCPI.DISPlay.COLor(Dnum).TRACe(Tr).DATA on page 238<br>SCPI.DISPlay.COLor(Dnum).TRACe(Tr).MEMory on page 239 |
| Equivalent key | **[System]** - **Misc Setup** - **Color Setup** - **Normal**\|**Invert** - **Reset Color** - **OK** |

# SCPI.DISPlay.COLor(*Dnum*).TRACe(*Tr*).DATA

Object type

Property

Syntax

SCPI.DISPlay.COLor(*Dnum*).TRACe(*Tr*).DATA = *Data*

*Data* = SCPI.DISPlay.COLor(*Dnum*).TRACe(*Tr*).DATA

Description

Sets the color of the data trace of traces 1 to 4 (*Tr*) for normal display (*Dnum*: 1) and inverted display (*Dnum*: 2).

Variable

| | ***Data*** |
|---|---|
| Description | Indicates 3-element array data.<br>• *Data(0)* Sets amount of red.<br>• *Data(1)* Sets amount of green.<br>• *Data(2)* Sets amount of blue.<br>The index of the array starts from 0. |
| Data type | Variant type (Variant) |
| Range | • *Data(0)* 0 to 5<br>• *Data(1)* 0 to 5<br>• *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. |

For information on the variable (*Dnum*) and the variable (*Tr*), see Table 7-7, “Variable(Dnum),” on page 234 and Table 7-5, “Variable (Tr),” on page 128, respectively.

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 on page 237

Equivalent key

**[System] - Misc Setup - Color Setup - Normal|Invert - Data Trace 1|Data Trace 2|Data Trace 3|Data Trace 4**

# SCPI.DISPlay.COLor(*Dnum*).TRACe(*Tr*).MEMory

Object type

Property

Syntax

SCPI.DISPlay.COLor(*Dnum*).TRACe(*Tr*).MEMory = *Data*

*Data* = SCPI.DISPlay.COLor(*Dnum*).TRACe(*Tr*).MEMory

Description

Sets the color of the memory trace of traces 1 to 4 (*Tr*) for normal display (*Dnum*: 1) and inverted display (*Dnum*: 2).

Variable

| | ***Data*** |
|---|---|
| Description | Indicates 3-element array data.<br>• *Data(0)* Sets amount of red.<br>• *Data(1)* Sets amount of green.<br>• *Data(2)* Sets amount of blue.<br>The index of the array starts from 0. |
| Data type | Variant type (Variant) |
| Range | • *Data(0)* 0 to 5<br>• *Data(1)* 0 to 5<br>• *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. |

For information on the variable (*Dnum*) and the variable (*Tr*), see Table 7-7, “Variable(Dnum),” on page 234 and Table 7-5, “Variable (Tr),” on page 128, respectively.

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 on page 237

Equivalent key

**[System] - Misc Setup - Color Setup - Normal|Invert - Mem Trace 1|Mem Trace 2|Mem Trace 3|Mem Trace 4**

# SCPI.DISPlay.ECHO.CLEar

Object type — Method

Syntax — SCPI.DISPlay.ECHO.CLEar

Description — Clears all character strings displayed in the echo window. (No read)

Examples

```
SCPI.DISPlay.ECHO.CLEar
```

Related objects — ECHO on page 115

SCPI.DISPlay.ECHO.DATA on page 240

Equivalent key — **[Macro Setup]** - **Clear Echo**

# SCPI.DISPlay.ECHO.DATA

Object type — Property

Syntax — SCPI.DISPlay.ECHO.DATA = *Cont*

Description — Displays a character string in the echo window. (No read)

There is the following difference from the display with the ECHO object.

- Displays a single character string.

Variable

| | ***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 on page 115

SCPI.DISPlay.TABLe.TYPE on page 249

SCPI.DISPlay.TABLe.STATe on page 248

SCPI.DISPlay.ECHO.CLEar on page 240

Equivalent key — No equivalent key is available on the front panel.

# SCPI.DISPlay.ENABle

Object type — Property

Syntax — SCPI.DISPlay.ENABle = *Status*

*Status* = SCPI.DISPlay.ENABle

Description — Turns ON/OFF the display update on the E5061A/E5062A measurement screen.

Variable

| | ***Status*** |
|---|---|
| Description | ON/OFF of the display update of the E5061A/E5062A measurement screen |
| Data type | Boolean type (Boolean) |
| Range | Select from the following.<br>•True or -1 Turns ON the display update.<br>•False or 0 Turns OFF the display update. |
| Preset value | True or -1 |

Examples

```
Dim DispUpdt As Boolean
SCPI.DISPlay.ENABle = False
DispUpdt = SCPI.DISPlay.ENABle
```

Equivalent key — **[Display]** - **Update**

# SCPI.DISPlay.FSIGn

Object type

Property

Syntax

SCPI.DISPlay.FSIGn = *Status*

*Status* = SCPI.DISPlay.FSIGn

Description

Turns ON/OFF the “Fail” display on the LCD screen when the limit test fails.

Variable

| | ***Status*** |
|---|---|
| Description | ON/OFF of the “Fail” display when the limit test fails |
| Data type | Boolean type (Boolean) |
| Range | Select from the following.<br>•True or -1 Turns ON the “Fail” display.<br>•False or 0 Turns OFF the “Fail” display. |
| Preset value | True or -1 |

Examples

```
Dim DispFail As Boolean
SCPI.DISPlay.FSIGn = False
DispFail = SCPI.DISPlay.FSIGn
```

Related objects

SCPI.CALCulate(Ch).SELected.LIMit.STATe on page 169

Equivalent key

**[Analysis] - Limit Test - Fail Sign**

# SCPI.DISPlay.IMAGe

Object type

Property

Syntax

SCPI.DISPlay.IMAGe = *Param*

*Param* = SCPI.DISPlay.IMAGe

Description

Selects the display type of the LCD display.

Variable

| | ***Param*** |
|---|---|
| Description | Display type of the LCD display |
| Data type | Character string type (String) |
| Range | Select from the following.<br>•"NORMal" Specifies the normal display (background color: black).<br>•"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**

# SCPI.DISPlay.MAXimize

Object type

Property

Syntax

SCPI.DISPlay.MAXimize = *Status*

*Status* = SCPI.DISPlay.MAXimize

Description

Turns ON/OFF the window maximization of the active channel.

If you turned ON the maximization, only the window of the active channel is maximized on the LCD display and the windows of the other channels are not displayed.

Variable

| | ***Status*** |
|---|---|
| Description | ON/OFF of the window maximization |
| Data type | Boolean type (Boolean) |
| Range | Select from the following.<br>•True or -1 Turns ON the window maximization.<br>•False or 0 Turns OFF the window maximization. |
| Preset value | False or 0 |

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 on page 250

Equivalent key

**[Channel Max]**

# SCPI.DISPlay.SKEY.STATe

Object type

Property

Syntax

SCPI.DISPlay.SKEY.STATe = *Status*

*Status* = SCPI.DISPlay.SKEY.STATe

Description

Turns ON/OFF the display of the softkey menu bar.

Variable

| | ***Status*** |
|---|---|
| Description | ON/OFF of the softkey menu bar display |
| Data type | Boolean type (Boolean) |
| Range | Select from the following.<br>•True or -1 Turns ON the softkey menu bar display.<br>•False or 0 Turns OFF the softkey menu bar display. |
| Preset value | True or -1 |

Examples

```
Dim DispSkey As Boolean
SCPI.DISPlay.SKEY.STATe = False
DispSkey = SCPI.DISPlay.SKEY.STATe
```

Equivalent key

**[Entry Off]**

# SCPI.DISPlay.SPLit

Object type — Property

Syntax — SCPI.DISPlay.SPLit = *Param*

*Param* = SCPI.DISPlay.SPLit

Description — Sets the layout of the channel windows on the LCD display.

Variable

| | ***Param*** |
|---|---|
| Description | Layout of channel windows |
| Data type | Character string type (String) |
| Range | Select from the following.<br>•"D1" See Figure 7-2 on page 247.<br>•"D12" See Figure 7-2.<br>•"D1_2" See Figure 7-2.<br>•"D112" See Figure 7-2.<br>•"D1_1_2" See Figure 7-2.<br>•"D123" See Figure 7-2.<br>•"D1_2_3" See Figure 7-2.<br>•"D12_33" See Figure 7-2.<br>•"D11_23" See Figure 7-2.<br>•"D13_23" See Figure 7-2.<br>•"D12_13" See Figure 7-2.<br>•"D1234" See Figure 7-2.<br>•"D1_2_3_4" See Figure 7-2.<br>•"D12_34" See Figure 7-2. |
| Preset value | "D1" |

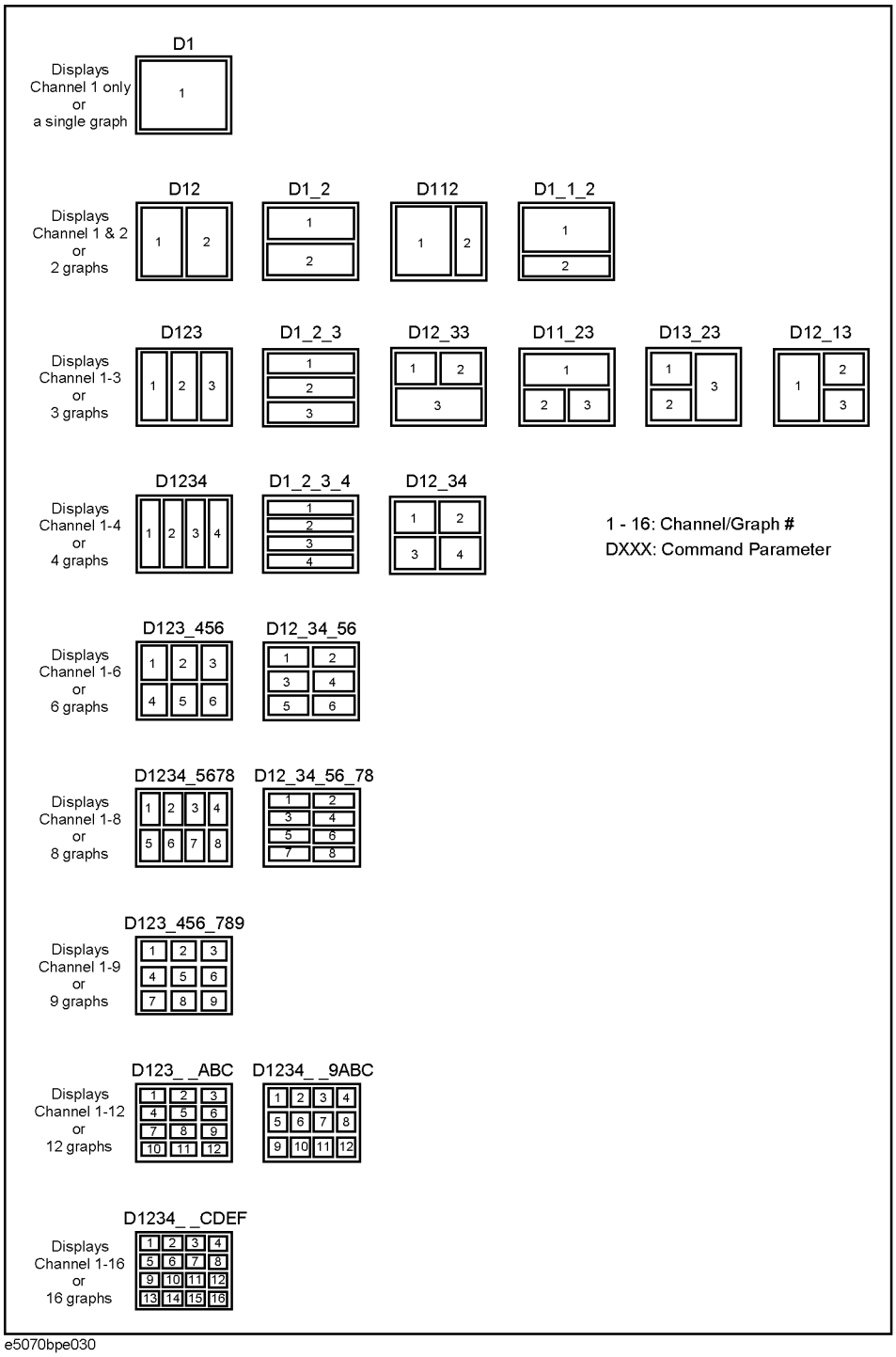
Examples

```
Dim ChanAloc As String
SCPI.DISPlay.SPLit = "d12_34"
ChanAloc = SCPI.DISPlay.SPLit
```

Related objects — SCPI.DISPlay.WINDow(Ch).SPLit on page 255

Equivalent key — **[Display]** - **Allocate Channels**

**Figure 7-2** **Channel/graph window layouts**




e5070bpe030

# SCPI.DISPlay.TABLe.STATe

Object type

Property

Syntax

SCPI.DISPlay.TABLe.STATe = *Status*

*Status* = SCPI.DISPlay.TABLe.STATe

Description

Turns ON/OFF the display of the window that appears in the lower part of the LCD display (specified with the SCPI.DISPlay.TABLe.TYPE object).

Variable

| | ***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.<br>•True or -1 Turns ON the display.<br>•False or 0 Turns OFF the display. |
| Preset value | False or 0 |

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 on page 249

Equivalent key

**[Sweep Setup]** - **Edit Segment Table**

**[Marker Fctn]** - **Marker Table**

**[Analysis]** - **Limit Test** - **Edit Limit Line**

**[Macro Setup]** - **Echo Window**

**NOTE**

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.

# SCPI.DISPlay.TABLe.TYPE

Object type — Property

Syntax — SCPI.DISPlay.TABLe.TYPE = *Param*

*Param* = SCPI.DISPlay.TABLe.TYPE

Description — Selects the type of the window that appears in the lower part of the LCD display.

Variable

| | ***Param*** |
|---|---|
| Description | Window type |
| Data type | Character string type (String) |
| Range | Select from the following.<br>•"MARKer" Specifies the marker table window.<br>•"LIMit" Specifies the limit test table window.<br>•"SEGMent" Specifies the segment table window.<br>•"ECHO" Specifies the echo 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 on page 248

Equivalent key — **[Sweep Setup]** - **Edit Segment Table**

**[Marker Fctn]** - **Marker Table**

**[Analysis]** - **Limit Test** - **Edit Limit Line**

**[Macro Setup]** - **Echo Window**

**NOTE** 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.

## SCPI.DISPlay.UPDate.IMMediate

Object type — Method

Syntax — SCPI.DISPlay.UPDate.IMMediate

Description — When the display update of the LCD screen is set to OFF (specifying False with the SCPI.DISPlay.ENABle object), executes the display update once. (No read)

Examples

```
SCPI.DISPlay.ENABle = False
SCPI.DISPlay.UPDate.IMMediate
```

Related objects — SCPI.DISPlay.ENABle on page 241

Equivalent key — No equivalent key is available on the front panel.

## SCPI.DISPlay.WINDow(*Ch*).ACTivate

Object type — Method

Syntax — SCPI.DISPlay.WINDow(*Ch*).ACTivate

Description — Specifies channels 1 to 4 (*Ch*) to the active channel.

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. (No read)

Variable — For information on the variable (*Ch*), see Table 7-4, “Variable (Ch),” on page 126.

Examples

```
SCPI.DISPlay.SPLit = "d1_2"
SCPI.DISPlay.WINDow(2).ACTivate
```

Related objects — SCPI.CALCulate(Ch).PARameter(Tr).SELect on page 128

Equivalent key — **[Channel Prev]** / **[Channel Next]**

# SCPI.DISPlay.WINDow(*Ch*).ANNotation.MARKer.ALIGn.STATe

Object type

Property

Syntax

SCPI.DISPlay.WINDow(*Ch*).ANNotation.MARKer.ALIGn.STATe = *Status*

*Status* = SCPI.DISPlay.WINDow(*Ch*).ANNotation.MARKer.ALIGn.STATe

Description

For the active trace of channels 1 to 4 (*Ch*), turns ON/OFF the alignment mode in which the display positions of the marker values for each trace are aligned relative to trace 1.

Variable

| | ***Status*** |
|---|---|
| Description | ON/OFF of the alignment mode |
| Data type | Boolean type (Boolean) |
| Range | Select from the following.<br>•True or -1 Turns ON the alignment mode using trace 1 as the reference.<br>•False or 0 Turn OFF the alignment mode. |
| Preset value | True or -1 |

For information on the variable (*Ch*), see Table 7-4, “Variable (Ch),” on page 126.

Examples

```
Dim MkAlign As Boolean
SCPI.DISPlay.WINDow(1).ANNotation.MARKer.ALIGn.STATe = True
MkAlign = SCPI.DISPlay.WINDow(1).ANNotation.MARKer.ALIGn.STATe
```

Equivalent key

**[Marker Fctn]** - **Annotation Options** - **Align**

# SCPI.DISPlay.WINDow(*Ch*).ANNotation.MARKer.SINGle.STATe

Object type

Property

Syntax

SCPI.DISPlay.WINDow(*Ch*).ANNotation.MARKer.SINGle.STATe = *Status*

*Status* = SCPI.DISPlay.WINDow(*Ch*).ANNotation.MARKer.SINGle.STATe

Description

For the active trace of channels 1 to 4 (*Ch*), turns ON/OFF the active trace marker value display.

When set to OFF, the marker values of all displayed traces (markers) are displayed.

Variable

| | ***Status*** |
|---|---|
| Description | ON/OFF of the active trace marker value display |
| Data type | Boolean type (Boolean) |
| Range | Select from the following.<br>•True or -1 Displays the marker value of the active trace only.<br>•False or 0 Displays the marker values of all traces. |
| Preset value | True or -1 |

For information on the variable (*Ch*), see Table 7-4, “Variable (Ch),” on page 126.

Examples

```
Dim MkSingle As Boolean
SCPI.DISPlay.WINDow(1).ANNotation.MARKer.SINGle.STATe = True
MkSingle = SCPI.DISPlay.WINDow(1).ANNotation.MARKer.SINGle.STATe
```

Equivalent key

**[Marker Fctn]** - **Annotation Options** - **Active Only**

# SCPI.DISPlay.WINDow(*Ch*).LABel

Object type — Property

Syntax — SCPI.DISPlay.WINDow(*Ch*).LABel = *Status*

*Status* = SCPI.DISPlay.WINDow(*Ch*).LABel

Description — Turns ON/OFF the graticule label display of the graph of channels 1 to 4 (*Ch*).

Variable

| | ***Status*** |
|---|---|
| Description | ON/OFF of the graticule label display of the graph |
| Data type | Boolean type (Boolean) |
| Range | Select from the following.<br>•True or -1 Turns ON the graticule label display.<br>•False or 0 Turns OFF the graticule label display. |
| Preset value | True or -1 |

For information on the variable (*Ch*), see Table 7-4, “Variable (Ch),” on page 126.

Examples

```
Dim DispGrat As Boolean
SCPI.DISPlay.WINDow(1).LABel = False
DispGrat = SCPI.DISPlay.WINDow(1).LABel
```

Equivalent key — **[Display] - Graticule Label**

# SCPI.DISPlay.WINDow(*Ch*).MAXimize

Object type

Property

Syntax

SCPI.DISPlay.WINDow(*Ch*).MAXimize = *Status*

*Status* = SCPI.DISPlay.WINDow(*Ch*).MAXimize

Description

Turns ON/OFF the maximization of the active trace of channels 1 to 4 (*Ch*).

If you turned ON the maximization, only the maximized active trace is displayed in the window and the other traces are not displayed.

Variable

| | ***Status*** |
|---|---|
| Description | ON/OFF of the maximization of the active trace |
| Data type | Boolean type (Boolean) |
| Range | Select from the following.<br>•True or -1 Turns ON the maxim display.<br>•False or 0 Turns OFF the maxim display. |
| Preset value | False or 0 |

For information on the variable (*Ch*), see Table 7-4, “Variable (Ch),” on page 126.

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 on page 128

SCPI.DISPlay.MAXimize on page 244

Equivalent key

**[Trace Max]**

# SCPI.DISPlay.WINDow(*Ch*).SPLit

Object type — Property

Syntax — SCPI.DISPlay.WINDow(*Ch*).SPLit = *Param*
*Param* = SCPI.DISPlay.WINDow(*Ch*).SPLit

Description — Sets the graph layout of channels 1 to 4 (*Ch*).

Variable

| | ***Param*** |
|---|---|
| Description | Graph layout |
| Data type | Character string type (String) |
| Range | Select from the following.<br>•"D1" See Figure 7-2 on page 247.<br>•"D12" See Figure 7-2.<br>•"D1_2" See Figure 7-2.<br>•"D112" See Figure 7-2.<br>•"D1_1_2" See Figure 7-2.<br>•"D123" See Figure 7-2.<br>•"D1_2_3" See Figure 7-2.<br>•"D12_33" See Figure 7-2.<br>•"D11_23" See Figure 7-2.<br>•"D13_23" See Figure 7-2.<br>•"D12_13" See Figure 7-2.<br>•"D1234" See Figure 7-2.<br>•"D1_2_3_4" See Figure 7-2.<br>•"D12_34" See Figure 7-2. |
| Preset value | "D1" |

For information on the variable (*Ch*), see Table 7-4, “Variable (Ch),” on page 126.

Examples

```
Dim TracAloc As String
SCPI.DISPlay.WINDow(1).SPLit = "d1_2"
TracAloc = SCPI.DISPlay.WINDow(1).SPLit
```

Related objects — SCPI.DISPlay.SPLit on page 246

Equivalent key — **[Display]** - **Allocate Traces**

# SCPI.DISPlay.WINDow(*Ch*).TITLe.DATA

Object type — Property

Syntax — SCPI.DISPlay.WINDow(*Ch*).TITLe.DATA = *Lbl*

*Lbl* = SCPI.DISPlay.WINDow(*Ch*).TITLe.DATA

Description — Sets the title label displayed in the title area of channels 1 to 4 (*Ch*).

Variable

| | ***Lbl*** |
|---|---|
| Description | Title label |
| Data type | Character string type (String) |
| Range | 254 characters or less |
| Preset value | "" |

For information on the variable (*Ch*), see Table 7-4, “Variable (Ch),” on page 126.

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 on page 257

Equivalent key — **[Display] - Edit Title Label**

# SCPI.DISPlay.WINDow(*Ch*).TITLe.STATe

Object type — Property

Syntax — SCPI.DISPlay.WINDow(*Ch*).TITLe.STATe = *Status*

*Status* = SCPI.DISPlay.WINDow(*Ch*).TITLe.STATe

Description — Turns ON/OFF the title label display in the title area of channels 1 to 4 (*Ch*).

Variable

| | ***Status*** |
|---|---|
| Description | ON/OFF of the title label display |
| Data type | Boolean type (Boolean) |
| Range | Select from the following.<br>•True or -1 Turns ON the title label display.<br>•False or 0 Turns ON the title label display. |
| Preset value | False or 0 |

For information on the variable (*Ch*), see Table 7-4, “Variable (Ch),” on page 126.

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 on page 256

Equivalent key — **[Display] - Title Label**

# SCPI.DISPlay.WINDow(*Ch*).TRACe(*Tr*).ANNotation. MARKer. POSition.X

Object type

Property

Syntax

SCPI.DISPlay.WINDow(*Ch*).TRACe(*Tr*).ANNotation.MARKer.POSition.X = *Value*

*Value* = SCPI.DISPlay.WINDow(*Ch*).TRACe(*Tr*).ANNotation.MARKer.POSition.X

Description

For traces 1 to 4 (*Tr*) of channels 1 to 4 (*Ch*), sets the X-axis position where the marker value is displayed as the percentage of the display area width.

Variable

| | ***Value*** |
|---|---|
| Description | The X-axis position where the marker value is displayed. |
| Data type | Long integer type (Long) |
| Range | -15 to 100 |
| Preset value | 1 |
| Unit | % (percent) |

For information on the variable (*Ch*) and the variable (*Tr*), see Table 7-4, “Variable (Ch),” on page 126 and Table 7-5, “Variable (Tr),” on page 128, respectively.

Examples

```
Dim PosX As Long
SCPI.CALCulate(1).PARameter(2).SELect
SCPI.DISPlay.WINDow(1).TRACe(2).ANNotation.MARKer.POSition.X = 20
PosX = SCPI.DISPlay.WINDow(1).TRACe(2).ANNotation.MARKer.POSition.X
```

Related objects

SCPI.CALCulate(Ch).SELected.FORMat on page 145

SCPI.DISPlay.WINDow(Ch).TRACe(Tr).ANNotation. MARKer. POSition.Y on page 259

Equivalent key

**[Marker Fctn]** - **Annotation Options** - **Marker Info X Pos**

# SCPI.DISPlay.WINDow(*Ch*).TRACe(*Tr*).ANNotation. MARKer. POSition.Y

Object type

Property

Syntax

SCPI.DISPlay.WINDow(*Ch*).TRACe(*Tr*).ANNotation.MARKer.POSition.Y = *Value*

*Value* = SCPI.DISPlay.WINDow(*Ch*).TRACe(*Tr*).ANNotation.MARKer.POSition.Y

Description

For traces 1 to 4 (*Tr*) of channels 1 to 4 (*Ch*), sets the Y-axis position where the marker value is displayed as the percentage of the display area height.

Variable

| | ***Value*** |
|---|---|
| Description | Y-axis position where the marker value is displayed |
| Data type | Long integer type (Long) |
| Range | -15 to 100 |
| Preset value | 1 |
| Unit | % (percent) |

For information on the variable (*Ch*) and the variable (*Tr*), see Table 7-4, “Variable (Ch),” on page 126 and Table 7-5, “Variable (Tr),” on page 128, respectively.

Examples

```
Dim PosY As Long
SCPI.CALCulate(1).PARameter(2).SELect
SCPI.DISPlay.WINDow(1).TRACe(2).ANNotation.MARKer.POSition.Y = 40
PosY = SCPI.DISPlay.WINDow(1).TRACe(2).ANNotation.MARKer.POSition.Y
```

Related objects

SCPI.CALCulate(Ch).SELected.FORMat on page 145

SCPI.DISPlay.WINDow(Ch).TRACe(Tr).ANNotation. MARKer. POSition.X on page 258

Equivalent key

**[Marker Fctn]** - **Annotation Options** - **Marker Info Y Pos**

# SCPI.DISPlay.WINDow(*Ch*).TRACe(*Tr*).ANNotation. YAXis.MODE

Object type

Property

Syntax

SCPI.DISPlay.WINDow(*Ch*).TRACe(*Tr*).ANNotation.YAXis.MODE = *Param*

*Param* = SCPI.DISPlay.WINDow(*Ch*).TRACe(*Tr*).ANNotation.YAXis.MODE

Description

For traces 1 to 4 (*Tr*) of channels 1 to 4 (*Ch*), sets the graticule label display format on the left side of the Y axis in the rectangular display format.

Variable

| | ***Param*** | |
|---|---|---|
| Description | sets the graticule label display format | |
| Data type | Character string type (String) | |
| Range | Select from the following. | |
| | •"AUTO" | Specifies the normal display format. |
| | •"RELative" | Specifies the relative display, based on the reference value. |
| Preset value | "AUTO" | |

For information on the variable (*Ch*) and the variable (*Tr*), see Table 7-4, “Variable (Ch),” on page 126 and Table 7-5, “Variable (Tr),” on page 128, respectively.

Examples

```
Dim YaxMode As String
SCPI.CALCulate(1).PARameter(2).SELect
SCPI.DISPlay.WINDow(1).TRACe(2).ANNotation.YAXis.MODE = "rel"
YaxMode = SCPI.DISPlay.WINDow(1).TRACe(2).ANNotation.YAXis.MODE
```

Related objects

SCPI.DISPlay.WINDow(Ch).Y.SCALe.DIVisions on page 269

SCPI.DISPlay.WINDow(Ch).TRACe(Tr).Y.SCALe.RLEVel on page 264

SCPI.DISPlay.WINDow(Ch).TRACe(Tr).Y.SCALe. RPOSition on page 265

Equivalent key

**[Display] - Graticule Label**

# SCPI.DISPlay.WINDow(*Ch*).TRACe(*Tr*).MEMory. STATe

Object type — Property

Syntax — SCPI.DISPlay.WINDow(*Ch*).TRACe(*Tr*).MEMory.STATe = *Status*

*Status* = SCPI.DISPlay.WINDow(*Ch*).TRACe(*Tr*).MEMory.STATe

Description — For traces 1 to 4 (*Tr*) of channels 1 to 4 (*Ch*), turns ON/OFF the memory trace display.

Variable

| | ***Status*** |
|---|---|
| Description | ON/OFF of the memory trace display |
| Data type | Boolean type (Boolean) |
| Range | Select from the following.<br>•True or -1 Turns ON the memory trace display.<br>•False or 0 Turns OFF the memory trace display. |
| Preset value | False or 0 |

For information on the variable (*Ch*) and the variable (*Tr*), see Table 7-4, “Variable (Ch),” on page 126 and Table 7-5, “Variable (Tr),” on page 128, respectively.

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 on page 209

SCPI.DISPlay.WINDow(Ch).TRACe(Tr).STATe on page 262

Equivalent key

**[Display]** - **Display** - **Mem** (when the data trace display is OFF)

**[Display]** - **Display** - **Data & Mem** (when the data trace display is ON)

## SCPI.DISPlay.WINDow(*Ch*).TRACe(*Tr*).STATe

Object type: Property

Syntax: SCPI.DISPlay.WINDow(*Ch*).TRACe(*Tr*).STATe = *Status*

*Status* = SCPI.DISPlay.WINDow(*Ch*).TRACe(*Tr*).STATe

Description: For traces 1 to 4 (*Tr*) of channels 1 to 4 (*Ch*), turns ON/OFF the data trace display.

Variable

| | ***Status*** |
|---|---|
| Description | ON/OFF of the data trace display |
| Data type | Boolean type (Boolean) |
| Range | Select from the following.<br>•True or -1 Turns ON the data trace display.<br>•False or 0 Turns OFF the data trace display. |
| Preset value | True or -1 |

For information on the variable (*Ch*) and the variable (*Tr*), see Table 7-4, "Variable (Ch)," on page 126 and Table 7-5, "Variable (Tr)," on page 128, respectively.

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 on page 261

Equivalent key: **[Display]** - **Display** - **Data** (when the memory trace display is OFF)

**[Display]** - **Display** - **Data & Mem** (when the memory trace display is ON)

## SCPI.DISPlay.WINDow(*Ch*).TRACe(*Tr*).Y.SCALe.AUTO

Object type: Method

Syntax: SCPI.DISPlay.WINDow(*Ch*).TRACe(*Tr*).Y.SCALe.AUTO

Description: For traces 1 to 4 (*Tr*) of channels 1 to 4 (*Ch*), executes the auto scale (function to automatically adjust the value of the reference division line and the scale per division to display the trace appropriately). (No read)

Variable: For information on the variable (*Ch*) and the variable (*Tr*), see Table 7-4, "Variable (Ch)," on page 126 and Table 7-5, "Variable (Tr)," on page 128, respectively.

Examples

```
SCPI.DISPlay.WINDow(1).TRACe(2).Y.SCALe.AUTO
```

Related objects: SCPI.DISPlay.WINDow(Ch).TRACe(Tr).Y.SCALe. PDIVision on page 263

SCPI.DISPlay.WINDow(Ch).TRACe(Tr).Y.SCALe.RLEVel on page 264

Equivalent key: **[Scale]** - **Auto Scale**

# SCPI.DISPlay.WINDow(*Ch*).TRACe(*Tr*).Y.SCALe. PDIVision

Object type

Property

Syntax

SCPI.DISPlay.WINDow(*Ch*).TRACe(*Tr*).Y.SCALe.PDIVision = *Value*

*Value* = SCPI.DISPlay.WINDow(*Ch*).TRACe(*Tr*).Y.SCALe.PDIVision

Description

For traces 1 to 4 (*Tr*) of channels 1 to 4 (*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

| | ***Value*** |
|---|---|
| Description | Scale value |
| Data type | Double precision floating point type (Double) |
| Range | 1E-18 to 1E8 |
| Preset value | Varies depending the data format.<br>• Log magnitude: 10<br>• Phase, Expanded phase or Positive phase: 90<br>• Group delay: 1E-8<br>• Smith chart or Polar or SWR: 1<br>• Linear magnitude: 0.1<br>• Real or Imaginary: 0.2 |
| Unit | Varies depending on the data format.<br>• Log magnitude: dB (decibel)<br>• Phase, Expanded phase or Positive phase: ° (degree)<br>• Group delay: s (second)<br>• 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. |

For information on the variable (*Ch*) and the variable (*Tr*), see Table 7-4, “Variable (Ch),” on page 126 and Table 7-5, “Variable (Tr),” on page 128, respectively.

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 on page 145

SCPI.DISPlay.WINDow(Ch).Y.SCALe.DIVisions on page 269

SCPI.DISPlay.WINDow(Ch).TRACe(Tr).Y.SCALe.RLEVel on page 264

SCPI.DISPlay.WINDow(Ch).TRACe(Tr).Y.SCALe. RPOSition on page 265

Equivalent key

**[Scale]** - **Scale/Div**

# SCPI.DISPlay.WINDow(*Ch*).TRACe(*Tr*).Y.SCALe.RLEVel

Object type — Property

Syntax — SCPI.DISPlay.WINDow(*Ch*).TRACe(*Tr*).Y.SCALe.RLEVel = *Value*

*Value* = SCPI.DISPlay.WINDow(*Ch*).TRACe(*Tr*).Y.SCALe.RLEVel

Description — For traces 1 to 4 (*Tr*) of channels 1 to 4 (*Ch*), sets the value of the reference division line.

Variable

| | ***Value*** |
|---|---|
| Description | Value of reference division line |
| Data type | Double precision floating point type (Double) |
| Range | -5E8 to 5E8 |
| Preset value | 0*1 |
| Unit | Varies depending on the data format.<br>• Log magnitude (MLOG): dB (decibel)<br>• Phase (PHAS), Expanded phase (UPH) or Positive phase (PPH): ° (degree)<br>• Group delay (GDEL): s (second)<br>• 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. |

*1. The preset value is 1 when the data format is SWR.

For information on the variable (*Ch*) and the variable (*Tr*), see Table 7-4, “Variable (Ch),” on page 126 and Table 7-5, “Variable (Tr),” on page 128, respectively.

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 on page 145

SCPI.DISPlay.WINDow(Ch).Y.SCALe.DIVisions on page 269

SCPI.DISPlay.WINDow(Ch).TRACe(Tr).Y.SCALe. PDIVision on page 263

SCPI.DISPlay.WINDow(Ch).TRACe(Tr).Y.SCALe. RPOSition on page 265

Equivalent key — **[Scale] - Reference Value**

# SCPI.DISPlay.WINDow(*Ch*).TRACe(*Tr*).Y.SCALe. RPOSition

Object type — Property

Syntax — SCPI.DISPlay.WINDow(*Ch*).TRACe(*Tr*).Y.SCALe.RPOSition = *Value*

*Value* = SCPI.DISPlay.WINDow(*Ch*).TRACe(*Tr*).Y.SCALe.RPOSition

Description — For traces 1 to 4 (*Tr*) of channels 1 to 4 (*Ch*), specifies the position of a reference division line with its number (an integer assigned starting from 0 from the lowest division).

Variable

| | ***Value*** |
|---|---|
| Description | Position of reference division line |
| Data type | Long integer type (Long) |
| Range | 0 to the number of divisions*1 |
| Preset value | 5*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. |

*1. Set with the SCPI.DISPlay.WINDow(Ch).Y.SCALe.DIVisions object.
*2. The preset value is 0 when the data format is linear magnitude or SWR.

For information on the variable (*Ch*) and the variable (*Tr*), see Table 7-4, “Variable (Ch),” on page 126 and Table 7-5, “Variable (Tr),” on page 128, respectively.

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 on page 145
SCPI.DISPlay.WINDow(Ch).Y.SCALe.DIVisions on page 269
SCPI.DISPlay.WINDow(Ch).TRACe(Tr).Y.SCALe. PDIVision on page 263
SCPI.DISPlay.WINDow(Ch).TRACe(Tr).Y.SCALe.RLEVel on page 264

Equivalent key — **[Scale] - Reference Position**

# SCPI.DISPlay.WINDow(*Ch*).TRACe(*Tr*).Y. TRACk. FREQuency

Object type

Property

Syntax

SCPI.DISPlay.WINDow(*Ch*).TRACe(*Tr*).Y.TRACk.FREQuency = *Value*

*Value* = SCPI.DISPlay.WINDow(*Ch*).TRACe(*Tr*).Y.TRACk.FREQuency

Description

For traces 1 to 4 (*Tr*) of channels 1 to 4 (*Ch*), sets a frequency when you want to specify a frequency on trace data as the reference value. 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

| | ***Value*** |
|---|---|
| Description | Frequency for tracking |
| Data type | Double precision floating point type (Double) |
| Range | -1E12 to 1E12 |
| Preset value | 0 |
| Unit | Hz (hertz), dBm or s (second) |

For information on the variable (*Ch*) and the variable (*Tr*), see Table 7-4, “Variable (Ch),” on page 126 and Table 7-5, “Variable (Tr),” on page 128, respectively.

Examples

```
Dim TrackFreq As Double
SCPI.CALCulate(1).PARameter(2).SELect
SCPI.DISPlay.WINDow(1).TRACe(2).Y.TRACk.FREQuency = 1E9
TrackFreq = SCPI.DISPlay.WINDow(1).TRACe(2).Y.TRACk.FREQuency
```

Related objects

SCPI.CALCulate(Ch).SELected.FORMat on page 145

SCPI.DISPlay.WINDow(Ch).Y.SCALe.DIVisions on page 269

SCPI.DISPlay.WINDow(Ch).TRACe(Tr).Y. TRACk. MODE on page 267

Equivalent key

**[Scale]** - **Reference Tracking** - **Track Frequency**

# SCPI.DISPlay.WINDow(*Ch*).TRACe(*Tr*).Y. TRACk. MODE

Object type

Property

Syntax

SCPI.DISPlay.WINDow(*Ch*).TRACe(*Tr*).Y.TRACk.MODE = *Param*

*Param* = SCPI.DISPlay.WINDow(*Ch*).TRACe(*Tr*).Y.TRACk.MODE

Description

For traces 1 to 4 (*Tr*) of channels 1 to 4 (*Ch*), sets the tracking method to offset the trace data after sweep.

Variable

| | ***Param*** | |
|---|---|---|
| Description | sets the tracking method | |
| 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" | |

For information on the variable (*Ch*) and the variable (*Tr*), see Table 7-4, “Variable (Ch),” on page 126 and Table 7-5, “Variable (Tr),” on page 128, respectively.

Examples

```
Dim TrackMode As String
SCPI.CALCulate(1).PARameter(2).SELect
SCPI.DISPlay.WINDow(1).TRACe(2).Y.TRACk.MODE = "peak"
TrackMode = SCPI.DISPlay.WINDow(1).TRACe(2).Y.TRACk.MODE
```

Related objects

SCPI.DISPlay.WINDow(Ch).Y.SCALe.DIVisions on page 269

SCPI.DISPlay.WINDow(Ch).TRACe(Tr).Y. TRACk. FREQuency on page 266

Equivalent key

**[Scale]** - **Reference Tracking** - **Tracking**

# SCPI.DISPlay.WINDow(*Ch*).X.SPACing

Object type

Property

Syntax

SCPI.DISPlay.WINDow(*Ch*).X.SPACing = *Param*

*Param* = SCPI.DISPlay.WINDow(*Ch*).X.SPACing

Description

Selects the display type of the graph horizontal axis of channels 1 to 4 (*Ch*) for segment sweep.

Variable

| | ***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" | |

For information on the variable (*Ch*), see Table 7-4, “Variable (Ch),” on page 126.

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 on page 368

Equivalent key

**[Sweep Setup]** - **Segment Display**

# SCPI.DISPlay.WINDow(*Ch*).Y.SCALe.DIVisions

Object type — Property

Syntax — SCPI.DISPlay.WINDow(*Ch*).Y.SCALe.DIVisions = *Value*

*Value* = SCPI.DISPlay.WINDow(*Ch*).Y.SCALe.DIVisions

Description — For channels 1 to 4 (*Ch*), sets the number of divisions in all the graphs.

Variable

| | ***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. |

For information on the variable (*Ch*), see Table 7-4, “Variable (Ch),” on page 126.

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 on page 263
SCPI.DISPlay.WINDow(Ch).TRACe(Tr).Y.SCALe.RLEVel on page 264
SCPI.DISPlay.WINDow(Ch).TRACe(Tr).Y.SCALe. RPOSition on page 265

Equivalent key — **[Scale] - Divisions**

# SCPI.FORMat.BORDer

Object type — Property

Syntax — SCPI.FORMat.BORDer = *Param*

*Param* = SCPI.FORMat.BORDer

Description — When the data transfer format is set to the binary transfer format (specify “REAL” with SCPI.FORMat.DATA object), sets the transfer order of each byte in data (byte order).

**NOTE** — This object is NOT used when controlling the E5061A/E5062A using COM objects in the E5061A/E5062A VBA.

Variable

| | ***Param*** |
|---|---|
| Description | Byte order |
| Data type | Character string type (String) |
| Range | Select from the following.<br>•"NORMal" Specifies the byte order in which transfer starts from the byte including MSB (Most Significant Bit).<br>•"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 on page 271

Equivalent key — No equivalent key is available on the front panel.

# SCPI.FORMat.DATA

Object type

Property

Syntax

SCPI.FORMat.DATA = *Param*

*Param* = SCPI.FORMat.DATA

Description

Use the following SCPI commands to set the format to read the data.

- :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?
- :SENS{1-4}:FREQ:DATA?
- :SENS{1-4}:SEGM:DATA

**NOTE**

ASCII transfer format must be specified when controlling the E5061A/E5062A using SCPI commands with the Parse object in the E5061A/E5062A VBA.

Variable

| | ***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 | "NORMal" | |

Examples

```
Dim Fmt As String
SCPI.FORMat.DATA = "asc"
Fmt = SCPI.FORMat.DATA
```

Related objects

SCPI.FORMat.BORDer on page 270

Parse on page 117

Equivalent key

No equivalent key is available on the front panel.

# SCPI.HCOPy.ABORt

Object type Method

Syntax SCPI.HCOPy.ABORt

Description Aborts the print output. (No read)

Examples
```
SCPI.HCOPy.ABORt
```

Related objects SCPI.HCOPy.IMMediate on page 273

Equivalent key **[System] - Abort Printing**

# SCPI.HCOPy.IMAGe

Object type Property

Syntax SCPI.HCOPy.IMAGe = *Param*

*Param* = SCPI.HCOPy.IMAGe

Description Selects the print color for output to the printer.

Variable

| | ***Param*** |
|---|---|
| Description | Print color for output to the printer. |
| Data type | Character string type (String) |
| Range | Select from the following.<br>•"NORMal" Specifies printing in close color to the display color.<br>•"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 on page 273

Equivalent key **[System] - Invert Image**

# SCPI.HCOPy.IMMediate

Object type — Method

Syntax — SCPI.HCOPy.IMMediate

Description — Outputs the display image on the LCD display to the printer connected to the E5061A/E5062A. (No read)

**NOTE** — When printing the E5061A/E5062A measurement screen, execute the VBA program with the Visual Basic editor closed. For the method, see “Running a Program from the E5061A/E5062A Measurement Screen” on page 50.

Examples

```
SCPI.HCOPy.IMMediate
```

Related objects — SCPI.HCOPy.ABORt on page 272

SCPI.HCOPy.IMAGe on page 272

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. Notice that, 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.

## SCPI.IEEE4882.CLS

Object type Method

Syntax SCPI.IEEE4882.CLS

Description Clears the followings. (No read)

- 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

Examples

```
SCPI.IEEE4882.CLS
```

Equivalent key No equivalent key is available on the front panel.

## SCPI.IEEE4882.ESE

Object type Property

Syntax SCPI.IEEE4882.ESE = *Value*

*Value* = SCPI.IEEE4882.ESE

Description Sets the value of the Standard Event Status Enable Register.

Variable

| | ***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 on page 278

Equivalent key No equivalent key is available on the front panel.

## SCPI.IEEE4882.ESR

Object type — Property

Syntax — *Value* = SCPI.IEEE4882.ESR

Description — Reads out the value of the Standard Event Status Register. Executing this object clears the register value. (Read only)

Variable

| | ***Value*** |
|---|---|
| Description | Value of the Standard Event Status Register |
| Data type | Long integer type (Long) |

Examples

```
Dim Stat As Long
Stat = SCPI.IEEE4882.ESR
```

Equivalent key — No equivalent key is available on the front panel.

## SCPI.IEEE4882.IDN

Object type — Property

Syntax — *Cont* = SCPI.IEEE4882.IDN

Description — Reads out the product information (manufacturer, model number, serial number, and firmware version number) of the E5061A/E5062A. (Read only)

Variable

| | ***Cont*** |
|---|---|
| Description | Product information ("{string 1},{string 2},{string 3},{string 4}")<br>• {string 1} Manufacturer. Agilent Technologies is always read out.<br>• {string 2} Model number (example: E5061A).<br>• {string 3} Serial number (example: JP1KI00101).<br>• {string 4} Firmware version number (example: 03.00). |
| Data type | Character string type (String) |

Examples

```
Dim Who As String
Who = SCPI.IEEE4882.IDN
```

Equivalent key — **[System]** - **Firmware Revision**

**[System]** - **Service Menu** - **Enable Options** - **Serial Number**

# SCPI.IEEE4882.OPC

Object type

Property

Syntax

(1) SCPI.IEEE4882.OPC = *Dummy*

(2) *Value* = SCPI.IEEE4882.OPC

Description

Case (1):
Specifies so that 1 is set to OPC bit (bit 0) of the Standard Event Status Register is et to 1 when all of pending operations complete. For information on the structure of the status register, see Appendix “Status Reporting System” in the *E5061A/E5062A Programmer’s Guide*.

Case (2):
Specifies so that 1 is read when all of pending operations complete.

Variable

Case (2):

| | ***Value*** |
|---|---|
| Description | 1 returned when all pending operations are complete |
| Data type | Long integer type (Long) |

Examples

```
Case(1):
SCPI.IEEE4882.OPC = 1


Case(2):
Dim Dmy As Long
Dmy = SCPI.IEEE4882.OPC
```

Related objects

SCPI.SENSe(Ch).CORRection.COLLect.ACQuire. ISOLation on page 310

SCPI.SENSe(Ch).CORRection.COLLect.ACQuire.LOAD on page 311

SCPI.SENSe(Ch).CORRection.COLLect.ACQuire.OPEN on page 312

SCPI.SENSe(Ch).CORRection.COLLect.ACQuire. SHORt on page 312

SCPI.SENSe(Ch).CORRection.COLLect.ACQuire.THRU on page 313

SCPI.TRIGger.SEQuence.SINGle on page 428

Equivalent key

No equivalent key is available on the front panel.

## SCPI.IEEE4882.OPT

Object type — Property

Syntax — *Cont* = SCPI.IEEE4882.OPT

Description — Reads out the identification numbers of options installed in the E5061A/E5062A. (Read only)

Variable

| | ***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.

## SCPI.IEEE4882.RST

Object type — Method

Syntax — SCPI.IEEE4882.RST

Description — Presets the setting state of the E5061A/E5062A. There is the following difference from the setting state preset with the SCPI.SYSTem.PRESet object. For details, see Appendix "List of Default Values" in the *E5061A/E5062A User's Guide*. (No read)

- The continuous initiation mode (see the SCPI.INITiate(Ch).CONTinuous object) of channel 1 is set to OFF.

Examples

```
SCPI.IEEE4882.RST
```

Related objects — SCPI.SYSTem.PRESet on page 422

SCPI.INITiate(Ch).CONTinuous on page 280

Equivalent key — No equivalent key is available on the front panel.

# SCPI.IEEE4882.SRE

Object type

Property

Syntax

SCPI.IEEE4882.SRE = *Value*

*Value* = SCPI.IEEE4882.SRE

Description

Sets the value of the Service Request Enable Register.

Variable

| | ***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 on page 274

SCPI.STATus.OPERation.ENABle on page 382

SCPI.STATus.QUEStionable.ENABle on page 394

Equivalent key

No equivalent key is available on the front panel.

## SCPI.IEEE4882.STB

Object type — Property

Syntax — *Value* = SCPI.IEEE4882.STB

Description — Reads out the value of the Status Byte Register. (Read only)

Variable

| | ***Value*** |
|---|---|
| Description | Value of the Status Byte Register |
| Data type | Long integer type (Long) |

Examples

```
Dim Stat As Long
Stat = SCPI.IEEE4882.STB
```

Equivalent key — No equivalent key is available on the front panel.

## SCPI.IEEE4882.TRG

Object type — Method

Syntax — SCPI.IEEE4882.TRG

Description — If the trigger source is set to GPIB/LAN (set to BUS with the SCPI.TRIGger.SEQuence.SOURce object), triggers the E5061A/E5062A waiting for trigger. For information on the waiting for trigger state, see Section “Trigger System” in the *E5061A/E5062A Programmer’s Guide*. (No read)

Examples

```
SCPI.TRIGger.SEQuence.SOURce = "bus"
SCPI.IEEE4882.TRG
```

Related objects — SCPI.TRIGger.SEQuence.SOURce on page 429

Equivalent key — No equivalent key is available on the front panel.

## SCPI.IEEE4882.WAI

Object type — Method

Syntax — SCPI.IEEE4882.WAI

Description — Waits for the execution of all objects sent before this object to be completed. (No read)

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.

# SCPI.INITiate(*Ch*).CONTinuous

Object type

Property

Syntax

SCPI.INITiate(*Ch*).CONTinuous = *Status*

*Status* = SCPI.INITiate(*Ch*).CONTinuous

Description

Turns ON/OFF of the continuous initiation mode (setting by which the trigger system initiates continuously) of channels 1 to 4 (*Ch*) in the trigger system.

For more information on the trigger system, see Section “Trigger System” in the *E5061A/E5062A Programmer’s Guide*.

Variable

| | ***Status*** |
|---|---|
| Description | ON/OFF of the continuous initiation mode |
| Data type | Boolean type (Boolean) |
| Range | Select from the following.<br>•True or -1 Turns ON the continuous initiation mode.<br>•False or 0 Turns OFF the continuous initiation mode. |
| Preset value | Varies depending on [variable (*Ch*)]*1 |

*1. Only channel 1 is initialized to ON at the execution of the SCPI.SYSTem.PRESet object; all the channels are initialized to OFF at the execution of the SCPI.IEEE4882.RST object.

For information on the variable (*Ch*), see Table 7-4, “Variable (Ch),” on page 126.

Examples

```
Dim ContMode As Boolean
SCPI.INITiate(2).CONTinuous = True
ContMode = SCPI.INITiate(2).CONTinuous
```

Related objects

SCPI.INITiate(Ch).IMMediate on page 281

Equivalent key

**[Trigger]** - **Continuous** (continuous initiation mode ON)

**[Trigger]** - **Hold** (continuous initiation mode OFF)

# SCPI.INITiate(*Ch*).IMMediate

Object type — Method

Syntax — SCPI.INITiate(*Ch*).IMMediate

Description — Changes the state of each channel of channels 1 to 4 (*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 Section “Trigger System” in the *E5061A/E5062A Programmer’s Guide*. (No read)

Variable — For information on the variable (*Ch*), see Table 7-4, “Variable (Ch),” on page 126.

Examples —

```
SCPI.INITiate(1).CONTinuous = False
SCPI.INITiate(1).IMMediate
```

Related objects — SCPI.INITiate(Ch).CONTinuous on page 280

Equivalent key — **[Trigger]** - **Single**

# SCPI.MMEMory.CATalog(*Dir*)

Object type

Property

Syntax

*Cont* = SCPI.MMEMory.CATalog(*Dir*)

Description

Reads out the following information on the built-in storage device of the E5061A/E5062A.

- Space in use
- Available space
- Name and size of all files (including directories) in the specified directory.

To read out the information in the root directory (folder), specify "\" (backslash). If you want to specify a directory on the floppy disk drive, you need to add "A:" at the beginning of the file name. Separate between directory names (file name) with "\" (back slash), or "/" (slash). (Read only)

Variable

| | ***Cont*** |
|---|---|
| Description | Directory information ("{A},{B},{Name 1},,{Size 1},{Name 2},,{Size 2},···,{Name N},,{Size N}")<br>Where N is the number of all files in the specified directory and n is an integer between 1 and N.<br>• {A} Space in use of the built-in storage device (byte)*1.<br>• {B} Available space of the built-in storage device (byte)*1.<br>• {Name n} Name of the n-th file (directory).<br>• {Size n} Size (byte) of the n-th file (directory). Always 0 for directories. |
| Data type | Character string type (String) |

*1. If you specify a directory on the floppy disk drive, it is the capacity of the floppy disk in the drive.

| | ***Dir*** |
|---|---|
| Description | Directory name whose 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("a:\")
```

Equivalent key

No equivalent key is available on the front panel.

# SCPI.MMEMory.COPY

**Object type** Property

**Syntax** SCPI.MMEMory.COPY = *File*

**Description** Copies a file.

Specify the file name with the extension. If you want to specify a file on the floppy disk drive, you need to add "A:" at the beginning of the file name. When you use directory names (folder names) and file name, separate them with "\" (back slash), or "/" (slash). (No read)

**Variable**

| | ***File*** |
|---|---|
| Description | Indicates 2 file names (copy source and copy destination).<br>• *File(0)* Copy source file name<br>• *File(1)* Copy destination file name<br>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","a:test01.sta")

Dim File(1) As Variant
File(0) = "test/state01.sta"
File(1) = "a:test01.sta"
SCPI.MMEMory.COPY = File
```

**Equivalent key** Practical front key operation is not available.

# SCPI.MMEMory.DELete

Object type

Property

Syntax

SCPI.MMEMory.DELete = *File*

Description

Deletes an existing file or directory (folder).

When you delete a directory, all the files and directories in it are deleted.

Specify the file name with the extension. If you want to specify a file or directory on the floppy disk drive, you need to add "A:" at the beginning of its name. When you specify a file (directory) under an existing directory, separate them with "\" (back slash), or "/" (slash).

To delete all files in the directory (folder), specify "\" (backslash). (No read)

Variable

| | ***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 = "a:\"

SCPI.MMEMory.DELete = "test/state01.sta"
```

Equivalent key

Practical front key operation is not available.

# SCPI.MMEMory.LOAD.CHANnel.COEFficient

Object type

Property

Syntax

SCPI.MMEMory.LOAD.CHANnel.STATe = *Register*

Description

Recalls the calibration coefficient for an individual channel from the specified register as the setting of the active channel.

It is possible to recall the register from a different channel where it was saved.

(No read)

Variable

| | ***Register*** |
|---|---|
| Description | Register |
| Data type | Character string type (String) |
| Range | Select from the following.<br>•"A" Specifies register A.<br>•"B" Specifies register B.<br>•"C" Specifies register C.<br>•"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.COEFficient = "a"
```

Equivalent key

**[Save/Recall] - Recall Channel - Cal Only A|B|C|D**

# SCPI.MMEMory.LOAD.CHANnel.STATe

Object type — Property

Syntax — SCPI.MMEMory.LOAD.CHANnel.STATe = *Register*

Description — 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.

It is possible to recall the register from a different channel where it was saved. (No read)

Variable

| | ***Register*** |
|---|---|
| Description | Register |
| Data type | Character string type (String) |
| Range | Select from the following.<br>•"A" Specifies register A.<br>•"B" Specifies register B.<br>•"C" Specifies register C.<br>•"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 on page 293
SCPI.DISPlay.WINDow(Ch).ACTivate on page 250

Equivalent key — **[Save/Recall]** - **Recall Channel** - **A**|**B**|**C**|**D**

# SCPI.MMEMory.LOAD.LIMit

Object type

Property

Syntax

SCPI.MMEMory.LOAD.LIMit = *File*

Description

As the limit table for the active trace of the active channel, recalls the specified limit table file (file with the .csv extension saved with the SCPI.MMEMory.STORe.LIMit object).

Specify the file name with the extension. If you want to specify a file on the floppy disk drive, you need to add "A:" at the beginning of the file name. When you use directory names and file name, separate them with "\" (back slash), or "/" (slash). (No read)

Variable

| | ***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 = "a:\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 on page 250

SCPI.CALCulate(Ch).PARameter(Tr).SELect on page 128

SCPI.MMEMory.STORe.LIMit on page 296

Equivalent key

**[Analysis]** - **Limit Test** - **Edit Limit Line** - **Import from CSV File**

# SCPI.MMEMory.LOAD.RLIMit

Object type

Property

Syntax

SCPI.MMEMory.LOAD.RLIMit = *File*

Description

As the ripple limit table for 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), recalls the specified ripple limit table file (file with the .csv extension saved with the SCPI.MMEMory.STORe.RLIMit command).

Specify the file name with the extension. If you want to specify a file on the floppy disk drive, you need to add "A:" at the beginning of the file name. 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. (Read only)

Variable

| | ***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. |

For information on the variable (*Ch*), see Table 7-4, “Variable (Ch),” on page 126.

Examples (1)

```
SCPI.DISPlay.WINDow(1).ACTive
SCPI.CALCulate(1).PARameter(1).SELect
SCPI.MMEMory.LOAD.RIMit = "A:\Rlimit01.csv"
```

Examples (2)

```
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 on page 250

SCPI.CALCulate(Ch).PARameter(Tr).SELect on page 128

SCPI.MMEMory.STORe.RLIMit on page 297

Equivalent key

**[Analysis]** - **Ripple Limit** - **Edit Ripple Line** - **Import from CSV File**

# SCPI.MMEMory.LOAD.SEGMent

Object type

Property

Syntax

SCPI.MMEMory.LOAD.SEGMent = *File*

Description

As the segment sweep table of the active channel, recalls the specified segment sweep table file (file with the .csv extension saved with the SCPI.MMEMory.STORe.SEGMent object).

Specify the file name with the extension. If you want to specify a file on the floppy disk drive, you need to add "A:" at the beginning of the file name. When you use directory names and file name, separate them with "\" (back slash), or "/" (slash). (No read)

Variable

| | ***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 = "a:\segm01.csv"

SCPI.DISPlay.WINDow(1).ACTivate
SCPI.MMEMory.LOAD.SEGMent = "test/segm01.csv"
```

Related objects

SCPI.DISPlay.WINDow(Ch).ACTivate on page 250

SCPI.MMEMory.STORe.SEGMent on page 299

Equivalent key

**[Sweep Setup] - Edit Segment Table - Import from CSV File**

# SCPI.MMEMory.LOAD.STATe

Object type — Property

Syntax — SCPI.MMEMory.LOAD.STATe = *File*

Description — Recalls the specified instrument state file (file with the .sta extension saved with the SCPI.MMEMory.STORe.STATe object).

Specify the file name with the extension. If you want to specify a file on the floppy disk drive, you need to add "A:" at the beginning of the file name. When you use directory names and file name, separate them with "\" (back slash), or "/" (slash). (No read)

Variable

| | ***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 = "a:\state01.sta"

SCPI.MMEMory.LOAD.STATe = "test/state01.sta"
```

Related objects — SCPI.MMEMory.STORe.STATe on page 300

Equivalent key — **[Save/Recall]** - **Recall State**

# SCPI.MMEMory.MDIRectory

Object type

Property

Syntax

SCPI.MMEMory.MDIRectory = *File*

Description

Creates a new directory (folder).

If you want to create a directory on the floppy disk drive, you need to add "A:" at the beginning of the directory name. When you create a directory under an existing directory, separate between the directory names with "\ " (back slash), or "/" (slash). (No read)

Variable

| | ***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 = "a:\test"

SCPI.MMEMory.MDIRectory = "test"
```

Equivalent key

Practical front key operation is not available.

# SCPI.MMEMory.STORe.CHANnel.CLEar

Object type — Method

Syntax — SCPI.MMEMory.STORe.CHANnel.CLEar

Description — Deletes the instrument state and calibration coefficient for each channel in all the registers. (No read)

Examples

```
SCPI.MMEMory.STORe.CHANnel.CLEar
```

Related objects — SCPI.MMEMory.STORe.CHANnel.STATe on page 293

Equivalent key — **[Save/Recall]** - **Save Channel** - **Clear States** - **OK**

# SCPI.MMEMory.STORe.CHANnel.COEFficient

Object type — Property

Syntax — SCPI.MMEMory.STORe.CHANnel.COEFficient = *Register*

Description — Saves the instrument calibration coefficient for the active channel into the specified register (volatile memory). (No read)

Variable

| | ***Register*** |
|---|---|
| Description | Register |
| Data type | Character string type (String) |
| Range | Select from the following.<br>•"A" Specifies register A.<br>•"B" Specifies register B.<br>•"C" Specifies register C.<br>•"D" Specifies register D. |
| Note | If an instrument state has been saved already in the specified register, its contents are overwritten. |

Examples

```
SCPI.MMEMory.STORe.CHANnel.COEFficient = "a"
```

Related objects

Equivalent key — **[Save/Recall]** - **Save Channel** - **Cal Only A**|**B**|**C**|**D**

# SCPI.MMEMory.STORe.CHANnel.STATe

Object type — Property

Syntax — SCPI.MMEMory.STORe.CHANnel.STATe = *Register*

Description — Saves the instrument state of the items set for the active channel specific to that channel only into the specified register (volatile memory). (No read)

Variable

| | ***Register*** |
|---|---|
| Description | Register |
| Data type | Character string type (String) |
| Range | Select from the following.<br>•"A" Specifies register A.<br>•"B" Specifies register B.<br>•"C" Specifies register C.<br>•"D" Specifies register D. |
| Note | If an instrument state has been saved already in the specified register, its contents are overwritten. |

Examples

```
SCPI.MMEMory.STORe.CHANnel.STATe = "a"
```

Related objects

SCPI.MMEMory.LOAD.CHANnel.STATe on page 286
SCPI.DISPlay.WINDow(Ch).ACTivate on page 250

Equivalent key — **[Save/Recall]** - **Save Channel** - **A**|**B**|**C**|**D**

# SCPI.MMEMory.STORe.FDATa

Object type

Property

Syntax

SCPI.MMEMory.STORe.FDATa = *File*

Description

For the active trace of the active channel, saves the formatted data array into a file in the CSV format (extension ".csv").

Specify the file name with the extension. If you want to specify a file on the floppy disk drive, you need to add "A:" at the beginning of the file name. When you use directory names and file name, separate them with "\" (back slash), or "/" (slash). (No read)

Variable

| | ***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 = "a:\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 on page 250

SCPI.CALCulate(Ch).PARameter(Tr).SELect on page 128

Equivalent key

**[Save/Recall]** - **Save Trace Data**

# SCPI.MMEMory.STORe.IMAGe

Object type

Property

Syntax

SCPI.MMEMory.STORe.IMAGe = *File*

Description

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. When saving the E5061A/E5062A measurement screen, execute the VBA program with the Visual Basic editor closed. For more information, see “Running a Program from the E5061A/E5062A Measurement Screen” on page 50.

Specify the file name with the extension. If you want to specify a file on the floppy disk drive, you need to add "A:" at the beginning of the file name. When you use directory names and file name, separate them with "\" (back slash), or "/" (slash). (No read)

Variable

| | ***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 = "a:\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. Notice that, 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.

# SCPI.MMEMory.STORe.LIMit

Object type

Property

Syntax

SCPI.MMEMory.STORe.LIMit = *File*

Description

Saves the limit table of the active trace of the active channel into a file in the CSV format (extension ".csv").

Specify the file name with the extension. If you want to specify a file on the floppy disk drive, you need to add "A:" at the beginning of the file name. When you use directory names and file name, separate them with "\" (back slash), or "/" (slash). (No read)

Variable

| | ***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 = "a:\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 on page 250

SCPI.CALCulate(Ch).PARameter(Tr).SELect on page 128

SCPI.MMEMory.LOAD.LIMit on page 287

Equivalent key

**[Analysis]** - **Limit Test** - **Edit Limit Line** - **Export to CSV File**

# SCPI.MMEMory.STORe.RLIMit

Object type — Property

Syntax — SCPI.MMEMory.STORe.RLIMit = *File*

Description — 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.

Specify the file name with the .sta extension. If you want to specify a file on the floppy disk drive, you need to add "A:" at the beginning of the file name. When you write directory names and file name, separate them with "/" (slash) or "\" (backslash).

Notice that if a file with the specified file name already exists, its contents will be overwritten. (Read only)

Variable

| | ***File*** |
|---|---|
| Description | Fiel 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. |

For information on the variable (*Ch*), see Table 7-4, “Variable (Ch),” on page 126.

Examples (1)

```
SCPI.DISPlay.WINDow(1).ACTivate
SCPI.CALCulate(1).PARameter(1).SELect
SCPI.MMEMory.STORe.RLIMit = "A:\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 on page 250

SCPI.CALCulate(Ch).PARameter(Tr).SELect on page 128

SCPI.MMEMory.LOAD.RLIMit on page 288

Equivalent key — **[Analysis]** - **Ripple Limit** - **Edit Ripple Line** - **Export to CSV File**

# SCPI.MMEMory.STORe.SALL

Object type

Property

Syntax

SCPI.MMEMory.STORe.SALL = *Status*

*Status* = SCPI.MMEMory.STORe.SALL

Description

Selects whether to save the setting of all channels/traces or that of the displayed channels/traces only as the instrument state to be saved.

Variable

| | ***Status*** | |
|---|---|---|
| Description | Selecting content to be saved as the instrument state setting. | |
| Data type | Boolean type (Boolean) | |
| Range | Select from the following. | |
| | •True or -1 | Specifies the setting of all channels/traces as the target to be saved. |
| | •False or 0 | Specifies the setting of displayed channels/traces only as the target to be saved. |
| Preset value | False or 0 | |

Examples

```
Dim Obj As Boolean
SCPI.MMEMory.STORe.SALL = True
Obj = SCPI.MMEMory.STORe.SALL
```

Related objects

SCPI.MMEMory.STORe.STATe on page 300

Equivalent key

**[Save/Recall]** - **Channel/Trace**

# SCPI.MMEMory.STORe.SEGMent

Object type — Property

Syntax — SCPI.MMEMory.STORe.SEGMent = *File*

Description — Saves the segment sweep table of the active channel into a file in the CSV format (extension ".csv").

Specify the file name with the extension. If you want to specify a file on the floppy disk drive, you need to add "A:" at the beginning of the file name. When you use directory names and file name, separate them with "\" (back slash), or "/" (slash). (No read)

Variable

| | ***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 = "a:\segm01.csv"

SCPI.DISPlay.WINDow(1).ACTivate
SCPI.MMEMory.STORe.SEGMent = "test/segm01.csv"
```

Related objects

SCPI.DISPlay.WINDow(Ch).ACTivate on page 250

SCPI.MMEMory.LOAD.SEGMent on page 289

Equivalent key — **[Sweep Setup]** - **Edit Segment Table** - **Export to CSV File**

# SCPI.MMEMory.STORe.STATe

Object type — Property

Syntax — SCPI.MMEMory.STORe.STATe = *File*

Description — Saves the instrument state (contents to be saved specified with the SCPI.MMEMory.STORe.STYPe object) into a file (file with the .sta extension).

Specify the file name with the extension. If you want to specify a file on the floppy disk drive, you need to add "A:" at the beginning of the file name. When you use directory names and file name, separate them with "\" (back slash), or "/" (slash). (No read)

**NOTE** The instrument setting file saved with the “autorec.sta” file name is automatically recalled when turning on the E5061A/E5062A.

Variable

| | ***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 = "a:\state01.sta"

Dim StaType As String
SCPI.MMEMory.STORe.STYPe = "cdst"
SCPI.MMEMory.STORe.STATe = "test/state01.sta"
```

Related objects — SCPI.MMEMory.STORe.STYPe on page 301

SCPI.MMEMory.LOAD.STATe on page 290

Equivalent key — **[Save/Recall]** - **Save State**

# SCPI.MMEMory.STORe.STYPe

Object type — Property

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

| | ***Param*** |
|---|---|
| Description | Data of instrument state |
| Data type | Character string type (String) |
| Range | Select from the following.<br>•"STATe" Specifies the save of the measurement conditions*1 only.<br>•"CSTate" Specifies the save of the measurement conditions*1 and the calibration state.<br>•"DSTate" Specifies the save of the measurement conditions*1 and the formatted data array.<br>•"CDSTate" Specifies the save of the measurement conditions*1, the calibration state, and the formatted data array. |
| Preset value | "CSTate" |

*1. For information on the measurement conditions to be saved, see Appendix "List of Default Values" in the *E5061A/E5062A User's Guide or Programmer's Guide*.

Examples

```
Dim StaType As String
SCPI.MMEMory.STORe.STYPe = "cdst"
StaType = SCPI.MMEMory.STORe.STYPe
```

Related objects — SCPI.MMEMory.STORe.STATe on page 300

Equivalent key — **[Save/Recall] - Save Type - State Only|State & Cal|State & Trace|All**

# SCPI.OUTPut.STATe

Object type

Property

Syntax

SCPI.OUTPut.STATe = *Status*

*Status* = SCPI.OUTPut.STATe

Description

Turns on/off of the stimulus signal output. You cannot perform measurement until you turn on the stimulus signal output.

Variable

| | ***Status*** |
|---|---|
| Description | On/off of the stimulus signal output |
| Data type | Boolean type (Boolean) |
| Range | Select from the following.<br>•True or -1 Turns on the stimulus signal.<br>•False or 0 Turns off the stimulus signal. |
| Preset value | True or -1 |

Examples

```
Dim Outp As Boolean
SCPI.OUTPut.STATe = True
Outp = SCPI.OUTPut.STATe
```

Equivalent key

**[Sweep Setup]** - **Power** - **RF Out**

## SCPI.SENSe(*Ch*).AVERage.CLEar

Object type — Method

Syntax — SCPI.SENSe(*Ch*).AVERage.CLEar

Description — Resets the data count to 0 used for averaging of channels 1 to 4 (*Ch*). Measurement data before the execution of this object is not used for averaging. (No read)

Variable — For information on the variable (*Ch*), see Table 7-4, “Variable (Ch),” on page 126.

Examples

```
SCPI.SENSe(1).AVERage.CLEar
```

Related objects — SCPI.SENSe(Ch).AVERage.COUNt on page 303

SCPI.SENSe(Ch).AVERage.STATe on page 304

Equivalent key — **[Avg]** - **Averaging Restart**

## SCPI.SENSe(*Ch*).AVERage.COUNt

Object type — Property

Syntax — SCPI.SENSe(*Ch*).AVERage.COUNt = *Value*

*Value* = SCPI.SENSe(*Ch*).AVERage.COUNt

Description — Sets the averaging factor of channels 1 to 4 (*Ch*).

Variable

| | ***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. |

For information on the variable (*Ch*), see Table 7-4, “Variable (Ch),” on page 126.

Examples

```
Dim AvgCnt As Long
SCPI.SENSe(1).AVERage.COUNt = 4
AvgCnt = SCPI.SENSe(1).AVERage.COUNt
```

Related objects — SCPI.SENSe(Ch).AVERage.STATe on page 304

SCPI.SENSe(Ch).AVERage.CLEar on page 303

Equivalent key — **[Avg]** - **Avg Factor**

# SCPI.SENSe(*Ch*).AVERage.STATe

Object type — Property

Syntax — SCPI.SENSe(*Ch*).AVERage.STATe = *Status*

*Status* = SCPI.SENSe(*Ch*).AVERage.STATe

Description — Turns ON/OFF the averaging function of channels 1 to 4 (*Ch*).

Variable

| | ***Status*** |
|---|---|
| Description | ON/OFF of the averaging function |
| Data type | Boolean type (Boolean) |
| Range | Select from the following.<br>•True or -1 Turns ON the averaging function.<br>•False or 0 Turns OFF the averaging function. |
| Preset value | False or 0 |

For information on the variable (*Ch*), see Table 7-4, “Variable (Ch),” on page 126.

Examples

```
Dim Avg As Boolean
SCPI.SENSe(1).AVERage.STATe = True
Avg = SCPI.SENSe(1).AVERage.STATe
```

Related objects — SCPI.SENSe(Ch).AVERage.COUNt on page 303
SCPI.SENSe(Ch).AVERage.CLEar on page 303

Equivalent key — **[Avg]** - **Averaging**

# SCPI.SENSe(*Ch*).BANDwidth.RESolution

Object type

Property

Syntax

SCPI.SENSe(*Ch*).BANDwidth.RESolution = *Value*

*Value* = SCPI.SENSe(*Ch*).BANDwidth.RESolution

Description

Sets the IF bandwidth of channels 1 to 4 (*Ch*).

This object provides the same function as the SCPI.SENSe(Ch).BWIDth.RESolution object.

Variable

| | ***Value*** |
|---|---|
| Description | IF bandwidth |
| Data type | Double precision floating point type (Double) |
| Range | 10 to 30000 |
| Preset value | 30000 |
| Unit | Hz (hertz) |
| Resolution | In steps of 1 or 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. |

For information on the variable (*Ch*), see Table 7-4, “Variable (Ch),” on page 126.

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 on page 306

Equivalent key

**[Avg]** - **IF Bandwidth**

# SCPI.SENSe(*Ch*).BWIDth.RESolution

Object type

Property

Syntax

SCPI.SENSe(*Ch*).BWIDth.RESolution = *Value*

*Value* = SCPI.SENSe(*Ch*).BWIDth.RESolution

Description

Sets the IF bandwidth of channels 1 to 4 (*Ch*).

This object provides the same function as the SCPI.SENSe(Ch).BANDwidth.RESolution object.

Variable

| | ***Value*** |
|---|---|
| Description | IF bandwidth |
| Data type | Double precision floating point type (Double) |
| Range | 10 to 30000 |
| Preset value | 30000 |
| Unit | Hz (hertz) |
| Resolution | In steps of 1 or 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. |

For information on the variable (*Ch*), see Table 7-4, “Variable (Ch),” on page 126.

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 on page 305

Equivalent key

**[Avg] - IF Bandwidth**

# SCPI.SENSe(*Ch*).CORRection.CLEar

| | |
|---|---|
| Object type | Property |
| Syntax | SCPI.SENSe(*Ch*).CORRection.CLEar |
| Description | Clears all callibration coefficient and measured standard data for calibration in the specified channel. (No read) |
| Variable | For information on the variable (*Ch*), see Table 7-4, “Variable (Ch),” on page 126. |
| Examples | `SCPI.SENSe(1).CORRection.CLEar` |
| Equivalent key | **[Cal]** - **Clear All** - **OK** |

# SCPI.SENSe(*Ch*).CORRection.COEFficient.DATA

Object type Property

Syntax *Array* = SCPI.SENSe(*Ch*).CORRection.COEFficient.DATA(*Str*, *Int1*, *Int2*)

Description Reads out the calibration coefficient of the specified channel. (Read only)

Variable

| | ***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.<br>• *Data(n×2-2)* Real part of the data (complex number) at the n-th measurement point.<br>• *Data(n×2-1)* Imaginary part of the data (complex number) at the n-th measurement point.<br>The index of the array starts from 0. |
| Data type | Variant type (Variant) |

| | ***Param*** |
|---|---|
| Description | Calibration type |
| Data type | Character string type (String) |
| Range | Select from the following.<br>•"ES" Source match<br>•"ER" Reflection tracking<br>•"ED" Directivity<br>•"EL" Load match<br>•"ET" Transmission tracking<br>•"EX" Isolation |

| | ***Int1*** |
|---|---|
| Description | Indicates the response port |
| Data type | Long integer type (Long) |
| Range | 1 to 2 |
| Resolution | 1 |
| 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. |

| | ***Int2*** |
|---|---|
| Description | Indicates the stimulus port |

| | ***Int2*** |
|---|---|
| Data type | Long integer type (Long) |
| Range | 1 to 2 |
| Resolution | 1 |
| 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 Table 7-4, “Variable (Ch),” on page 126.

Examples

```
DIM Array(200) as Variant
Array = SCPI.SENSe(1).CORRection.COEFficient.DATA("EL", 1, 2)
```

Equivalent key

No equivalent key is available on the front panel.

# SCPI.SENSe(*Ch*).CORRection.COLLect.ACQuire. ISOLation

Object type — Property

Syntax — SCPI.SENSe(*Ch*).CORRection.COLLect.ACQuire.ISOLation = *Ports*

Description — For channels 1 to 4 (*Ch*), measures the calibration data of the isolation from the specified stimulus port to the specified response port. (No read)

Variable

**Table 7-8 Variable (*Ports*)**

| | ***Ports*** |
|---|---|
| Description | Indicates 2-element array data (port number).<br>• *Ports(0)* Specifies the response port number.<br>• *Ports(1)* Specifies the stimulus port number.<br>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.<br>If you specify the same port number to 2 port numbers, an error occurs when executed. |

For information on the variable (*Ch*), see Table 7-4, “Variable (Ch),” on page 126.

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 on page 276

Equivalent key — **[Cal] - Calibrate - Response (Thru) - Isolation (Optional)**

**[Cal] - Calibrate - n-Port Cal - Isolation (Optional) - Port m-n Isol**

# SCPI.SENSe(*Ch*).CORRection.COLLect.ACQuire.LOAD

Object type — Property

Syntax — SCPI.SENSe(*Ch*).CORRection.COLLect.ACQuire.LOAD = *Port*

Description — For channels 1 to 4 (*Ch*), measures the calibration data of the load standard for the specified port. (No read)

Variable

**Table 7-9 Variable (*Port*)**

| | ***Port*** |
|---|---|
| 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. |

For information on the variable (*Ch*), see Table 7-4, “Variable (Ch),” on page 126.

Examples

```
Dim Dmy As Long
SCPI.SENSe(1).CORRection.COLLect.ACQuire.LOAD = 1
Dmy = SCPI.IEEE4882.OPC
```

Related objects — SCPI.IEEE4882.OPC on page 276

Equivalent key — **[Cal] - Calibrate - Response (Open)|Response (Short) - Load (Optional)**

**[Cal] - Calibrate - 1-Port Cal - Load**

**[Cal] - Calibrate - n-Port Cal - Reflection - Port m Load**

## SCPI.SENSe(*Ch*).CORRection.COLLect.ACQuire.OPEN

Object type — Property

Syntax — SCPI.SENSe(*Ch*).CORRection.COLLect.ACQuire.OPEN = *Port*

Description — For channels 1 to 4 (*Ch*), measures the calibration data of the open standard for the specified port. (No read)

Variable — For information on the variable (*Ch*) and the variable (*Port*), see Table 7-4, "Variable (Ch)," on page 126 and Table 7-9, "Variable (Port)," on page 311, respectively.

Examples

```
Dim Dmy As Long
SCPI.SENSe(1).CORRection.COLLect.ACQuire.OPEN = 1
Dmy = SCPI.IEEE4882.OPC
```

Related objects — SCPI.IEEE4882.OPC on page 276

Equivalent key — **[Cal]** - **Calibrate** - **Response (Open)**|**1-Port Cal** - **Open**

**[Cal]** - **Calibrate** - **n-Port Cal** - **Reflection** - **Port m Open**

## SCPI.SENSe(*Ch*).CORRection.COLLect.ACQuire. SHORt

Object type — Property

Syntax — SCPI.SENSe(*Ch*).CORRection.COLLect.ACQuire.SHORt = *Port*

Description — For channels 1 to 4 (*Ch*), measures the calibration data of the short standard for the specified port. (No read)

Variable — For information on the variable (*Ch*) and the variable (*Port*), see Table 7-4, "Variable (Ch)," on page 126 and Table 7-9, "Variable (Port)," on page 311, respectively.

Examples

```
Dim Dmy As Long
SCPI.SENSe(1).CORRection.COLLect.ACQuire.SHORt = 1
Dmy = SCPI.IEEE4882.OPC
```

Related objects — SCPI.IEEE4882.OPC on page 276

Equivalent key — **[Cal]** - **Calibrate** - **Response (Short)**|**1-Port Cal** - **Short**

**[Cal]** - **Calibrate** - **n-Port Cal** - **Reflection** - **Port m Short**

# SCPI.SENSe(*Ch*).CORRection.COLLect.ACQuire.THRU

Object type — Property

Syntax — SCPI.SENSe(*Ch*).CORRection.COLLect.ACQuire.THRU = *Ports*

Description — For channels 1 to 4 (*Ch*), measures the calibration data of the thru standard from the specified stimulus port to the specified response port. (No read)

Variable — For information on the variable (*Ch*) and the variable (*Ports*), see Table 7-4, "Variable (Ch)," on page 126 and Table 7-8, "Variable (Ports)," on page 310, respectively.

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 on page 276

Equivalent key — **[Cal]** - **Calibrate** - **Response (Thru)** - **Thru**

**[Cal]** - **Calibrate** - **n-Port Cal** - **Transmission** - **Port m-n Thru**

# SCPI.SENSe(*Ch*).CORRection.COLLect.CKIT.LABel

Object type — Property

Syntax — SCPI.SENSe(*Ch*).CORRection.COLLect.CKIT.LABel = *Lbl*

*Lbl* = SCPI.SENSe(*Ch*).CORRection.COLLect.CKIT.LABel

Description — Sets a calibration kit name for the calibration kit selected for channels 1 to 4 (*Ch*).

Variable

| | ***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.<br>• 1: “85033E“<br>• 2: “85033D“<br>• 3: “85052D“<br>• 4: “85032F“<br>• 5: “85032B“<br>• 6: “85036B/E“<br>• 7 to 10: “User“ |

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 on page 319

Equivalent key — **[Cal]** - **Modify Cal Kit** - **Label Kit**

# SCPI.SENSe(*Ch*).CORRection.COLLect.CKIT.ORDer. LOAD(*Cpt*)

Object type — Property

Syntax — SCPI.SENSe(*Ch*).CORRection.COLLect.CKIT.ORDer.LOAD(*Cpt*) = *Value*

*Value* = SCPI.SENSe(*Ch*).CORRection.COLLect.CKIT.ORDer.LOAD(*Cpt*)

Description — For the calibration kit selected for channels 1 to 4 (*Ch*), selects the standard used for the load measurement of the specified port (*Cpt*).

Variable

**Table 7-10 Variable (*Cpt*)**

| | ***Cpt*** |
|---|---|
| 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. |

**NOTE** Since the variable (*Cpt*) has no preset value, you cannot omit it. If you omit the variable (*Cpt*), an error occurs when executed.

| | ***Value*** |
|---|---|
| Description | Standard number |
| Data type | Long integer type (Long) |
| Range | 1 to 21 |
| Note | If the specified variable is out of the allowable setup range, an error occurs when executed. |

For information on the variable (*Ch*), see Table 7-4, “Variable (Ch),” on page 126.

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 on page 319

Equivalent key — **[Cal] - Modify Cal Kit - Specify CLSs - Load - Port 1|Port 2**

# SCPI.SENSe(*Ch*).CORRection.COLLect.CKIT.ORDer. OPEN(*Cpt*)

Object type — Property

Syntax — SCPI.SENSe(*Ch*).CORRection.COLLect.CKIT.ORDer.OPEN(*Cpt*) = *Value*

*Value* = SCPI.SENSe(*Ch*).CORRection.COLLect.CKIT.ORDer.OPEN(*Cpt*)

Description — For the calibration kit selected for channels 1 to 4 (*Ch*), selects the standard used for the open measurement of the specified port (*Cpt*).

Variable

| | ***Value*** |
|---|---|
| Description | Standard number |
| Data type | Long integer type (Long) |
| Range | 1 to 21 |
| Note | If the specified variable is out of the allowable setup range, an error occurs when executed. |

For information on the variable (*Ch*) and the variable (*Cpt*), see Table 7-4, "Variable (Ch)," on page 126 and Table 7-10, "Variable (Cpt)," on page 315, respectively.

**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 on page 319

Equivalent key — **[Cal]** - **Modify Cal Kit** - **Specify CLSs** - **Open** - **Port 1**|**Port 2**

# SCPI.SENSe(*Ch*).CORRection.COLLect.CKIT.ORDer. SHORt(*Cpt*)

Object type — Property

Syntax — SCPI.SENSe(*Ch*).CORRection.COLLect.CKIT.ORDer.SHORt(*Cpt*) = *Value*

*Value* = SCPI.SENSe(*Ch*).CORRection.COLLect.CKIT.ORDer.SHORt(*Cpt*)

Description — For the calibration kit selected for channels 1 to 4 (*Ch*), selects the standard used for the short measurement of the specified port (*Cpt*).

Variable

| | ***Value*** |
|---|---|
| Description | Standard number |
| Data type | Long integer type (Long) |
| Range | 1 to 21 |
| Note | If the specified variable is out of the allowable setup range, an error occurs when executed. |

For information on the variable (*Ch*) and the variable (*Cpt*), see Table 7-4, "Variable (Ch)," on page 126 and Table 7-10, "Variable (Cpt)," on page 315, respectively.

**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 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 on page 319

Equivalent key — **[Cal]** - **Modify Cal Kit** - **Specify CLSs** - **Short** - **Port 1**|**Port 2**

# SCPI.SENSe(*Ch*).CORRection.COLLect.CKIT.ORDer. THRU(*Cpt_m*,*Cpt_n*)

Object type

Property

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

For the calibration kit selected for channels 1 to 4 (*Ch*), selects the standard used for the thru measurement between the specified 2 ports (*Cpt_m* and *Cpt_n*).

Variable

| | ***Cpt_m*, *Cpt_n*** |
|---|---|
| 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. |

**NOTE**

Since the variables (*Cpt_m* and *Cpt_n*) have no preset value, you cannot omit them. If you omit the variables (*Cpt_m* and *Cpt_n*) or if you specify the same port number to 2 port numbers, an error occurs when executed. Notice that when you specify 2 ports with the variables (*Cpt_m* and *Cpt_n*), the order of the 2 port numbers is arbitrary.

| | ***Value*** |
|---|---|
| Description | Standard number |
| Data type | Long integer type (Long) |
| Range | 1 to 21 |
| Note | If the specified variable is out of the allowable setup range, an error occurs when executed. |

For information on the variable (*Ch*), see Table 7-4, “Variable (Ch),” on page 126.

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 on page 319

Equivalent key

**[Cal]** - **Modify Cal Kit** - **Specify CLSs** - **Thru - Port 1-2**

## SCPI.SENSe(*Ch*).CORRection.COLLect.CKIT.RESet

Object type — Method

Syntax — SCPI.SENSe(*Ch*).CORRection.COLLect.CKIT.RESet

Description — Resets the calibration kit selected for channels 1 to 4 (*Ch*) to the factory setting state. (No read)

Variable — For information on the variable (*Ch*), see Table 7-4, "Variable (Ch)," on page 126.

Examples

```
SCPI.SENSe(1).CORRection.COLLect.CKIT.RESet
```

Related objects — SCPI.SENSe(Ch).CORRection.COLLect.CKIT.SELect on page 319

Equivalent key — No equivalent key is available on the front panel.

## SCPI.SENSe(*Ch*).CORRection.COLLect.CKIT.SELect

Object type — Property

Syntax — SCPI.SENSe(*Ch*).CORRection.COLLect.CKIT.SELect = *Value*

*Value* = SCPI.SENSe(*Ch*).CORRection.COLLect.CKIT.SELect

Description — Selects the calibration kit of channels 1 to 4 (*Ch*).

Variable

| | ***Value*** |
|---|---|
| Description | Number of calibration kit*1 |
| 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. |

*1. The numbers of 1 to 10 assigned from the top to the calibration kit names displayed on the softkey labels when performing **[Cal]** - **Cal Kit**.

For information on the variable (*Ch*), see Table 7-4, "Variable (Ch)," on page 126.

Examples

```
Dim CalKit As Long
SCPI.SENSe(1).CORRection.COLLect.CKIT.SELect = 3
CalKit = SCPI.SENSe(1).CORRection.COLLect.CKIT.SELect
```

Equivalent key — **[Cal]** - **Cal Kit**

# SCPI.SENSe(*Ch*).CORRection.COLLect.CKIT.STAN(*Std*).ARBitrary

Object type

Property

Syntax

SCPI.SENSe(*Ch*).CORRection.COLLect.CKIT.STAN(*Std*).ARBitrary = *Value*

*Value* = SCPI.SENSe(*Ch*).CORRection.COLLect.CKIT.STAN(*Std*).ARBitrary

Description

For the calibration kit selected for channels 1 to 4 (*Ch*), sets the value of the arbitrary impedance of the standards 1 to 21 (*Std*).

Variable

**Table 7-11 Variable (*Std*)**

| | ***Std*** |
|---|---|
| Description | Standard number |
| Data type | Long integer type (Long) |
| Range | 1 to 21 |
| Preset value | 1 |
| Note | If the specified variable is out of the allowable setup range, an error occurs when executed. |

| | ***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 (ohm) |
| 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. |

For information on the variable (*Ch*), see Table 7-4, “Variable (Ch),” on page 126.

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 on page 319

Equivalent key

**[Cal] - Modify Cal Kit - Define STDs - no. name**[*1] **- Arb. Impedance**

*1. no: standard number (1 to 21), name: standard name (variable)

# SCPI.SENSe(*Ch*).CORRection.COLLect.CKIT.STAN(*Std*).C0

Object type

Property

Syntax

SCPI.SENSe(*Ch*).CORRection.COLLect.CKIT.STAN(*Std*).C0 = *Value*

*Value* = SCPI.SENSe(*Ch*).CORRection.COLLect.CKIT.STAN(*Std*).C0

Description

For the calibration kit selected for channels 1 to 4 (*Ch*), sets the value of the CO value of the standards 1 to 21 (*Std*).

Variable

| | ***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. |

For information on the variable (*Ch*) and the variable (*Std*), see Table 7-4, “Variable (Ch),” on page 126 and Table 7-11, “Variable (Std),” on page 320, respectively.

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 on page 319

Equivalent key

**[Cal] - Modify Cal Kit - Define STDs - no. name**[*1] **- C0**

*1. no: standard number (1 to 21), name: standard name (variable)

# SCPI.SENSe(*Ch*).CORRection.COLLect.CKIT.STAN(*Std*).C1

Object type

Property

Syntax

SCPI.SENSe(*Ch*).CORRection.COLLect.CKIT.STAN(*Std*).C1 = *Value*

*Value* = SCPI.SENSe(*Ch*).CORRection.COLLect.CKIT.STAN(*Std*).C1

Description

For the calibration kit selected for channels 1 to 4 (*Ch*), sets the value of the C1 value of the standards 1 to 21 (*Std*).

Variable

| | ***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. |

For information on the variable (*Ch*) and the variable (*Std*), see Table 7-4, “Variable (Ch),” on page 126 and Table 7-11, “Variable (Std),” on page 320, respectively.

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 on page 319

Equivalent key

**[Cal] - Modify Cal Kit - Define STDs - no. name**[*1] **- C1**

*1. no: standard number (1 to 21), name: standard name (variable)

# SCPI.SENSe(*Ch*).CORRection.COLLect.CKIT.STAN(*Std*).C2

Object type

Property

Syntax

SCPI.SENSe(*Ch*).CORRection.COLLect.CKIT.STAN(*Std*).C2 = *Value*

*Value* = SCPI.SENSe(*Ch*).CORRection.COLLect.CKIT.STAN(*Std*).C2

Description

For the calibration kit selected for channels 1 to 4 (*Ch*), sets the value of the C2 value of the standards 1 to 21 (*Std*).

Variable

| | ***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. |

For information on the variable (*Ch*) and the variable (*Std*), see Table 7-4, “Variable (Ch),” on page 126 and Table 7-11, “Variable (Std),” on page 320, respectively.

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 on page 319

Equivalent key

**[Cal] - Modify Cal Kit - Define STDs - no. name**[*1] **- C2**

*1. no: standard number (1 to 21), name: standard name (variable)

# SCPI.SENSe(*Ch*).CORRection.COLLect.CKIT.STAN(*Std*).C3

Object type

Property

Syntax

SCPI.SENSe(*Ch*).CORRection.COLLect.CKIT.STAN(*Std*).C3 = *Value*

*Value* = SCPI.SENSe(*Ch*).CORRection.COLLect.CKIT.STAN(*Std*).C3

Description

For the calibration kit selected for channels 1 to 4 (*Ch*), sets the value of the C3 value of the standards 1 to 21 (*Std*).

Variable

| | ***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. |

For information on the variable (*Ch*) and the variable (*Std*), see Table 7-4, “Variable (Ch),” on page 126 and Table 7-11, “Variable (Std),” on page 320, respectively.

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 on page 319

Equivalent key

**[Cal] - Modify Cal Kit - Define STDs - no. name**[*1] **- C3**

*1. no: standard number (1 to 21), name: standard name (variable)

# SCPI.SENSe(*Ch*).CORRection.COLLect.CKIT.STAN(*Std*).DELay

Object type — Property

Syntax — SCPI.SENSe(*Ch*).CORRection.COLLect.CKIT.STAN(*Std*).DELay = *Value*

*Value* = SCPI.SENSe(*Ch*).CORRection.COLLect.CKIT.STAN(*Std*).DELay

Description — For the calibration kit selected for channels 1 to 4 (*Ch*), sets the value of the offset delay of the standards 1 to 21 (*Std*).

Variable

| | ***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. |

For information on the variable (*Ch*) and the variable (*Std*), see Table 7-4, “Variable (Ch),” on page 126 and Table 7-11, “Variable (Std),” on page 320, respectively.

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 on page 319

Equivalent key — **[Cal] - Modify Cal Kit - Define STDs - no. name**[*1] **- Offset Delay**

*1. no: standard number (1 to 21), name: standard name (variable)

# SCPI.SENSe(*Ch*).CORRection.COLLect.CKIT.STAN(*Std*).L0

Object type

Property

Syntax

SCPI.SENSe(*Ch*).CORRection.COLLect.CKIT.STAN(*Std*).L0 = *Value*

*Value* = SCPI.SENSe(*Ch*).CORRection.COLLect.CKIT.STAN(*Std*).L0

Description

For the calibration kit selected for channels 1 to 4 (*Ch*), sets the value of the L0 value of the standards 1 to 21 (*Std*).

Variable

| | ***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. |

For information on the variable (*Ch*) and the variable (*Std*), see Table 7-4, “Variable (Ch),” on page 126 and Table 7-11, “Variable (Std),” on page 320, respectively.

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 on page 319

Equivalent key

**[Cal] - Modify Cal Kit - Define STDs - no. name**[*1] **- L0**

*1. no: standard number (1 to 21), name: standard name (variable)

# SCPI.SENSe(*Ch*).CORRection.COLLect.CKIT.STAN(*Std*).L1

Object type

Property

Syntax

SCPI.SENSe(*Ch*).CORRection.COLLect.CKIT.STAN(*Std*).L1 = *Value*

*Value* = SCPI.SENSe(*Ch*).CORRection.COLLect.CKIT.STAN(*Std*).L1

Description

For the calibration kit selected for channels 1 to 4 (*Ch*), sets the value of the L1 value of the standards 1 to 21 (*Std*).

Variable

| | ***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. |

For information on the variable (*Ch*) and the variable (*Std*), see Table 7-4, “Variable (Ch),” on page 126 and Table 7-11, “Variable (Std),” on page 320, respectively.

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 on page 319

Equivalent key

**[Cal] - Modify Cal Kit - Define STDs - no. name**[*1] **- L1**

*1. no: standard number (1 to 21), name: standard name (variable)

# SCPI.SENSe(*Ch*).CORRection.COLLect.CKIT.STAN(*Std*).L2

Object type

Property

Syntax

SCPI.SENSe(*Ch*).CORRection.COLLect.CKIT.STAN(*Std*).L2 = *Value*

*Value* = SCPI.SENSe(*Ch*).CORRection.COLLect.CKIT.STAN(*Std*).L2

Description

For the calibration kit selected for channels 1 to 4 (*Ch*), sets the value of the L2 value of the standards 1 to 21 (*Std*).

Variable

| | ***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. |

For information on the variable (*Ch*) and the variable (*Std*), see Table 7-4, “Variable (Ch),” on page 126 and Table 7-11, “Variable (Std),” on page 320, respectively.

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 on page 319

Equivalent key

**[Cal] - Modify Cal Kit - Define STDs - no. name**[*1] **- L2**

*1. no: standard number (1 to 21), name: standard name (variable)

# SCPI.SENSe(*Ch*).CORRection.COLLect.CKIT.STAN(*Std*).L3

Object type

Property

Syntax

SCPI.SENSe(*Ch*).CORRection.COLLect.CKIT.STAN(*Std*).L3 = *Value*

*Value* = SCPI.SENSe(*Ch*).CORRection.COLLect.CKIT.STAN(*Std*).L3

Description

For the calibration kit selected for channels 1 to 4 (*Ch*), sets the value of the L3 value of the standards 1 to 21 (*Std*).

Variable

| | ***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. |

For information on the variable (*Ch*) and the variable (*Std*), see Table 7-4, “Variable (Ch),” on page 126 and Table 7-11, “Variable (Std),” on page 320, respectively.

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 on page 319

Equivalent key

**[Cal] - Modify Cal Kit - Define STDs - no. name**[*1] **- L3**

*1. no: standard number (1 to 21), name: standard name (variable)

# SCPI.SENSe(*Ch*).CORRection.COLLect.CKIT.STAN(*Std*).LABel

Object type

Property

Syntax

SCPI.SENSe(*Ch*).CORRection.COLLect.CKIT.STAN(*Std*).LABel = *Lbl*

*Lbl* = SCPI.SENSe(*Ch*).CORRection.COLLect.CKIT.STAN(*Std*).LABel

Description

For the calibration kit selected for channels 1 to 4 (*Ch*), sets the name of the standards 1 to 21 (*Std*).

Variable

| | ***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. |

For information on the variable (*Ch*) and the variable (*Std*), see Table 7-4, “Variable (Ch),” on page 126 and Table 7-11, “Variable (Std),” on page 320, respectively.

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 on page 319

Equivalent key

**[Cal] - Modify Cal Kit - Define STDs - no. name**[*1] **- Label**

*1. no: standard number (1 to 21), name: standard name (variable)

# SCPI.SENSe(*Ch*).CORRection.COLLect.CKIT.STAN(*Std*).LOSS

Object type

Property

Syntax

SCPI.SENSe(*Ch*).CORRection.COLLect.CKIT.STAN(*Std*).LOSS = *Value*

*Value* = SCPI.SENSe(*Ch*).CORRection.COLLect.CKIT.STAN(*Std*).LOSS

Description

For the calibration kit selected for channels 1 to 4 (*Ch*), sets the value of the offset loss of the standards 1 to 21 (*Std*).

Variable

| | ***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 (ohm/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. |

For information on the variable (*Ch*) and the variable (*Std*), see Table 7-4, “Variable (Ch),” on page 126 and Table 7-11, “Variable (Std),” on page 320, respectively.

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 on page 319

Equivalent key

**[Cal] - Modify Cal Kit - Define STDs - no. name**[*1] **- Offset Loss**

*1. no: standard number (1 to 21), name: standard name (variable)

# SCPI.SENSe(*Ch*).CORRection.COLLect.CKIT.STAN(*Std*).TYPE

Object type

Property

Syntax

SCPI.SENSe(*Ch*).CORRection.COLLect.CKIT.STAN(*Std*).TYPE = *Param*

*Param* = SCPI.SENSe(*Ch*).CORRection.COLLect.CKIT.STAN(*Std*).TYPE

Description

For the calibration kit selected for channels 1 to 4 (*Ch*), sets the standard type of the standards 1 to 21 (*Std*).

Variable

| | ***Param*** | |
|---|---|---|
| Description | Standard type | |
| Data type | Character string type (String) | |
| Range | Select from the following. | |
| | •"OPEN" | Specifies open. |
| | •"SHORt" | Specifies short. |
| | •"LOAD" | Specifies load. |
| | •"THRU" | Specifies 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. | |

For information on the variable (*Ch*) and the variable (*Std*), see Table 7-4, “Variable (Ch),” on page 126 and Table 7-11, “Variable (Std),” on page 320, respectively.

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 on page 319

Equivalent key

**[Cal] - Modify Cal Kit - Define STDs - no. name**[*1] **- STD Type**

*1. no: standard number (1 to 21), name: standard name (variable)

# SCPI.SENSe(*Ch*).CORRection.COLLect.CKIT.STAN(*Std*).Z0

Object type

Property

Syntax

SCPI.SENSe(*Ch*).CORRection.COLLect.CKIT.STAN(*Std*).Z0 = *Value*

*Value* = SCPI.SENSe(*Ch*).CORRection.COLLect.CKIT.STAN(*Std*).Z0

Description

For the calibration kit selected for channels 1 to 4 (*Ch*), sets the value of the offset Z0 of the standards 1 to 21 (*Std*).

Variable

| | ***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 (ohm) |
| 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. |

For information on the variable (*Ch*) and the variable (*Std*), see Table 7-4, “Variable (Ch),” on page 126 and Table 7-11, “Variable (Std),” on page 320, respectively.

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 on page 319

Equivalent key

**[Cal] - Modify Cal Kit - Define STDs - no. name**[*1] **- Offset Z0**

*1. no: standard number (1 to 21), name: standard name (variable)

# SCPI.SENSe(*Ch*).CORRection.COLLect.ECAL.ERESponse

Object type — Property

Syntax — SCPI.SENSe(*Ch*).CORRection.COLLect.ECAL.ERES = *Eports*

Description — Executes enhanced resopnse calibration of channels 1 to 4 (*Ch*) using the ECal (Electronic Calibration) module.

If you execute this object when the ECal module is not connected, an error occurs when executed and the object is ignored. (No read)

Variable

| | ***Eports*** |
|---|---|
| Description | Indicates 2-element array data (port number).<br>• *EPorts(0)* Specifies the response port.<br>• *EPorts(1)* Specifies the stimulus port.<br>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. |

For information on the variable (*Ch*), see Table 7-4, “Variable (Ch),” on page 126.

Examples

```
SCPI.SENSe(1).CORRection.COLLect.ECAL.SOLT2 = Array(1,2)

Dim ERESPort(1) As Variant
ERESPort(0) = 1
ERESPort(1) = 2
SCPI.SENSe(1).CORRection.COLLect.ECAL.ERESponse = ERESPort
```

Equivalent key — **[Cal]** - **ECal** - **Enhansed Response** - **2-1(S21)**|**1-2(S12)**

# SCPI.SENSe(*Ch*).CORRection.COLLect.ECAL.ISOLation.STATe

Object type

Property

Syntax

SCPI.SENSe(*Ch*).CORRection.COLLect.ECAL.ISOLation.STATe = *Status*

*Status* = SCPI.SENSe(*Ch*).CORRection.COLLect.ECAL.ISOLation.STATe

Description

For channels 1 to 4 (*Ch*), turns ON/OFF the isolation measurement when executing Ecal (Electronic Calibration).

Variable

| | ***Status*** |
|---|---|
| Description | ON/OFF of the isolation measurement when executing ECal |
| Data type | Boolean type (Boolean) |
| Range | Select from the following.<br>•True or -1 Turns ON the isolation measurement.<br>•False or 0 Turns OFF the isolation measurement. |
| Preset value | False or 0 |

For information on the variable (*Ch*), see Table 7-4, “Variable (Ch),” on page 126.

Examples

```
Dim EcalIso As Boolean
SCPI.SENSe(1).CORRection.COLLect.ECAL.ISOLation.STATe = True
EcalIso = SCPI.SENSe(1).CORRection.COLLect.ECAL.ISOLation.STATe
```

Related objects

SCPI.SENSe(Ch).CORRection.COLLect.ECAL.SOLT1 on page 337

SCPI.SENSe(Ch).CORRection.COLLect.ECAL.SOLT2 on page 338

Equivalent key

**[Cal]** - **ECal** - **Isolation**

# SCPI.SENSe.CORRection.COLLect.ECAL.PATH(*Cpt*)

Object type — Property

Syntax — *Ept* = SCPI.SENSe.CORRection.COLLect.ECAL.PATH(*Cpt*)

Description — Reads out which port of the ECal module is connected with the specified port of the E5061A/E5062A. (Read only)

Variable

| | ***Ept*** |
|---|---|
| Description | Port of ECal module. |
| Data type | Long integer type (Long) |
| Range | One of the following is read out.<br>• 0 Nothing is connected.<br>• 1 Port A is connected.<br>• 2 Port B is connected.<br>• 3 Port C is connected.<br>• 4 Port D is connected. |

For information on the variable (*Cpt*), see Table 7-10, “Variable (Cpt),” on page 315.

Examples

```
Dim ECalPort As Long
ECalPort = SCPI.SENSe.CORRection.COLLect.ECAL.PATH(1)
```

Equivalent key — No equivalent key is available on the front panel.

# SCPI.SENSe(*Ch*).CORRection.COLLect.ECAL.SOLT1

Object type

Property

Syntax

SCPI.SENSe(*Ch*).CORRection.COLLect.ECAL.SOLT1 = *Eport*

Description

Executes full 1-port calibration of the specified port of channels 1 to 4 (*Ch*) using the ECal (Electronic Calibration) module.

If you execute this object when the ECal module is not connected, an error occurs when executed and the object is ignored. (No read)

Variable

| | ***Eport*** |
|---|---|
| Description | Port number |
| Data type | Long integer type (Long) |
| Range | 1 to 4 |
| Note | If the specified variable is out of the allowable setup range, an error occurs when executed. |

For information on the variable (*Ch*), see Table 7-4, “Variable (Ch),” on page 126.

Examples

```
SCPI.SENSe(1).CORRection.COLLect.ECAL.SOLT1 = 1
```

Equivalent key

**[Cal]** - **ECal** - **1-Port Cal** - **Port 1**|**Port 2**

# SCPI.SENSe(*Ch*).CORRection.COLLect.ECAL.SOLT2

Object type

Property

Syntax

SCPI.SENSe(*Ch*).CORRection.COLLect.ECAL.SOLT2 = *Eports*

Description

Executes full 2-port calibration between the specified 2 ports of channels 1 to 4 (*Ch*) using the ECal (Electronic Calibration) module.

If you execute this object when the ECal module is not connected, an error occurs when executed and the object is ignored. (No read)

Variable

| | ***Eports*** |
|---|---|
| Description | Indicates 2-element array data (port number).<br>• *EPorts(0)*<br>*EPorts*(1) Specifies the port numbers for 2-port ECal.<br>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. |

For information on the variable (*Ch*), see Table 7-4, “Variable (Ch),” on page 126.

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**

# SCPI.SENSe(*Ch*).CORRection.COLLect.ECAL.THRU

Object type

Property

Syntax

SCPI.SENSe(*Ch*).CORRection.COLLect.ECAL.THRU = *Eports*

Description

Executes response calibration (thru) between the specified 2 ports of channels 1 to 4 (*Ch*) using the ECal (Electronic Calibration) module.

If you execute this object when the ECal module is not connected, an error occurs when executed and the object is ignored. (No read)

Variable

| | ***Eports*** |
|---|---|
| Description | Indicates 2-element array data (port number).<br>• *Ports(0)* Specifies the response port number.<br>• *Ports(1)* Specifies the stimulus port number.<br>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. |

For information on the variable (*Ch*), see Table 7-4, “Variable (Ch),” on page 126.

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 - 2-1 (S21)|3-1 (S31)|4-1 (S41)|1-2 (S12)|3-2 (S32)| 4-2 (S42)| 1-3 (S13)|2-3 (S23)|4-3 (S43)|1-4 (S14)|2-4 (S24)|3-4 (S34)**

# SCPI.SENSe(*Ch*).CORRection.COLLect.METHod. ERESponse

Object type — Property

Syntax — SCPI.SENSe(*Ch*).CORRection.COLLect.METHod.ERESponse = *Ports*

Description — For channels 1 to 4 (*Ch*), sets the calibration type to the enhanced response calibration. (No read)

Variable

| | ***Ports*** |
|---|---|
| Description | Indicates 2-element array data (port number).<br>• *Ports(0)* Specifies the response port.<br>• *Ports(1)* Specifies the stimulus port.<br>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. |

For information on the variable (*Ch*), see Table 7-4, “Variable (Ch),” on page 126.

Examples

```
SCPI.SENSe(1).CORRection.COLLect.METHod.SOLT2 = Array(1,2)

Dim ERESPort(1) As Variant
ERESPort(0) = 1
ERESPort(1) = 2
SCPI.SENSe(1).CORRection.COLLect.METHod.ERESponse = ERESPort
```

Related objects

Equivalent key — **[Cal]** - **Calibrate** - **Enhansed Response - Ports**

## SCPI.SENSe(*Ch*).CORRection.COLLect.METHod. RESPonse.OPEN

Object type — Property

Syntax — SCPI.SENSe(*Ch*).CORRection.COLLect.METHod.RESPonse.OPEN = *Port*

Description — For channels 1 to 4 (*Ch*), sets the calibration type to the response calibration (open) of the specified port. (No read)

Variable — For information on the variable (*Ch*) and the variable (*Port*), see Table 7-4, "Variable (Ch)," on page 126 and Table 7-9, "Variable (Port)," on page 311, respectively.

Examples

```
SCPI.SENSe(1).CORRection.COLLect.METHod.RESPonse.OPEN = 1
```

Related objects — SCPI.SENSe(Ch).CORRection.COLLect.METHod.TYPE on page 344

Equivalent key — **[Cal]** - **Calibrate** - **Response (Open) - Select Port**

## SCPI.SENSe(*Ch*).CORRection.COLLect.METHod. RESPonse.SHORt

Object type — Property

Syntax — SCPI.SENSe(*Ch*).CORRection.COLLect.METHod.RESPonse.SHORt = *Port*

Description — For channels 1 to 4 (*Ch*), sets the calibration type to the response calibration (short) of the specified port. (No read)

Variable — For information on the variable (*Ch*) and the variable (*Port*), see Table 7-4, "Variable (Ch)," on page 126 and Table 7-9, "Variable (Port)," on page 311, respectively.

Examples

```
SCPI.SENSe(1).CORRection.COLLect.METHod.RESPonse.SHORt = 1
```

Related objects — SCPI.SENSe(Ch).CORRection.COLLect.METHod.TYPE on page 344

Equivalent key — **[Cal]** - **Calibrate** - **Response (Short) - Select Port**

## SCPI.SENSe(*Ch*).CORRection.COLLect.METHod. RESPonse.THRU

Object type — Property

Syntax — SCPI.SENSe(*Ch*).CORRection.COLLect.METHod.RESPonse.THRU = *Ports*

Description — For channels 1 to 4 (*Ch*), sets the calibration type to the response calibration (thru) between the specified 2 ports. (No read)

Variable — For information on the variable (*Ch*) and the variable (*Ports*), see Table 7-4, "Variable (Ch)," on page 126 and Table 7-8, "Variable (Ports)," on page 310, respectively.

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 on page 344

Equivalent key — **[Cal] - Calibrate - Response (Thru) - Select Ports**

## SCPI.SENSe(*Ch*).CORRection.COLLect.METHod. SOLT1

Object type — Property

Syntax — SCPI.SENSe(*Ch*).CORRection.COLLect.METHod.SOLT1 = *Port*

Description — For channels 1 to 4 (*Ch*), sets the calibration type to the full 1-port calibration of the specified port. (No read)

Variable — For information on the variable (*Ch*) and the variable (*Port*), see Table 7-4, "Variable (Ch)," on page 126 and Table 7-9, "Variable (Port)," on page 311, respectively.

Examples

```
SCPI.SENSe(1).CORRection.COLLect.METHod.SOLT1 = 1
```

Related objects — SCPI.SENSe(Ch).CORRection.COLLect.METHod.TYPE on page 344

Equivalent key — **[Cal] - Calibrate - 1-Port Cal - Select Port**

# SCPI.SENSe(*Ch*).CORRection.COLLect.METHod. SOLT2

Object type — Property

Syntax — SCPI.SENSe(*Ch*).CORRection.COLLect.METHod.SOLT2 = *Ports*

Description — For channels 1 to 4 (*Ch*), sets the calibration type to the full 2-port calibration between the specified 2 ports. (No read)

Variable

| | ***Ports*** |
|---|---|
| Description | Indicates 2-element array data (port number).<br>• *Ports(0)* Specifies a port for full 2-port calibration.<br>• *Ports(1)* Specifies the other port for full 2-port calibration.<br>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. |

For information on the variable (*Ch*), see Table 7-4, “Variable (Ch),” on page 126.

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 on page 344

Equivalent key — **[Cal]** - **Calibrate** - **2-Port Cal**

# SCPI.SENSe(*Ch*).CORRection.COLLect.METHod.TYPE

Object type — Property

Syntax — *Param* = SCPI.SENSe(*Ch*).CORRection.COLLect.METHod.TYPE

Description — Reads out the selected calibration type of channels 1 to 4 (*Ch*). (Read only)

**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

| | ***Param*** | |
|---|---|---|
| Description | Calibration type | |
| Data type | Character string type (String) | |
| Range | Select from the following. | |
| | •"ERES" | The calibration type is the enhanced response calibration. |
| | •"NONE" | The calibration type is set to nothing. |
| | •"RESPO" | The calibration type is the response calibration (open). |
| | •"RESPS" | The calibration type is the response calibration (short). |
| | •"RESPT" | The calibration type is the response calibration (thru). |
| | •"SOLT1" | The calibration type is the full 1-port calibration. |
| | •"SOLT2" | The calibration type is the full 2-port calibration. |

For information on the variable (*Ch*), see Table 7-4, “Variable (Ch),” on page 126.

Examples

```
Dim CalType As String
CalType = SCPI.SENSe(1).CORRection.COLLect.METHod.TYPE
```

Related objects — SCPI.SENSe(Ch).CORRection.COLLect.SAVE on page 345

SCPI.SENSe(Ch).CORRection.TYPE(Tr) on page 352

Equivalent key — No equivalent key is available on the front panel.

# SCPI.SENSe(*Ch*).CORRection.COLLect.SAVE

Object type — Method

Syntax — SCPI.SENSe(*Ch*).CORRection.COLLect.SAVE

Description — From the measured calibration data, calculates the calibration coefficients depending on the calibration type selection.

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 you execute this object before all necessary calibration data for calculating the calibration coefficients is measured, an error occurs when executed. (No read)

Variable — For information on the variable (*Ch*), see Table 7-4, "Variable (Ch)," on page 126.

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 on page 341
SCPI.SENSe(Ch).CORRection.COLLect.METHod. RESPonse.SHORt on page 341
SCPI.SENSe(Ch).CORRection.COLLect.METHod. RESPonse.THRU on page 342
SCPI.SENSe(Ch).CORRection.COLLect.METHod. SOLT1 on page 342
SCPI.SENSe(Ch).CORRection.COLLect.METHod. SOLT2 on page 343

Equivalent key — **[Cal]** - **Calibrate** - **Response**|**n-Port Cal** - **Done**

# SCPI.SENSe(*Ch*).CORRection.EXTension.PORT(*Pt*).TIME

Object type — Property

Syntax — SCPI.SENSe(*Ch*).CORRection.EXTension.PORT(*Pt*).TIME = *Value*

*Value* = SCPI.SENSe(*Ch*).CORRection.EXTension.PORT(*Pt*).TIME

Description — For channels 1 to 4 (*Ch*), sets the delay time for the port extension of ports 1 and 2 (*Pt*).

Variable

| | ***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. |

For information on the variable (*Ch*) and the variable (*Pt*), see Table 7-4, “Variable (Ch),” on page 126 and Table 7-12, “Variable (Pt),” on page 378, respectively.

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 on page 347

Equivalent key — **[Cal]** - **Port Extensions** - **Extension Port N**

# SCPI.SENSe(*Ch*).CORRection.EXTension.STATe

Object type — Property

Syntax — SCPI.SENSe(*Ch*).CORRection.EXTension.STATe = *Status*

*Status* = SCPI.SENSe(*Ch*).CORRection.EXTension.STATe

Description — For channels 1 to 4 (*Ch*), turns ON/OFF the port extension.

Variable

| | ***Status*** |
|---|---|
| Description | ON/OFF of the port extension correction |
| Data type | Boolean type (Boolean) |
| Range | Select from the following.<br>•True or -1 Turns ON the port extension.<br>•False or 0 Turns OFF the port extension. |
| Preset value | False or 0 |

For information on the variable (*Ch*), see Table 7-4, “Variable (Ch),” on page 126Table 7-4, “Variable (Ch),” on page 126.

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 on page 346

Equivalent key — **[Cal]** - **Port Extensions** - **Extensions**

# SCPI.SENSe.CORRection.IMPedance.INPut.MAGNitude

Object type

Property

Syntax

SCPI.SENSe.CORRection.IMPedance.INPut.MAGNitude = *Value*

*Value* = SCPI.SENSe.CORRection.IMPedance.INPut.MAGNitude

Description

Sets the system characteristic impedance (Z0) value.

Variable

| | ***Value*** |
|---|---|
| Description | System Z0 value |
| Data type | Double precision floating point type (Double) |
| Range | 1E-3 to 1000 |
| Preset value | 50 or 75 |
| Unit | Ω (ohm) |
| 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**

# SCPI.SENSe(*Ch*).CORRection.PROPerty

Object type

Property

Syntax

SCPI.SENSe(*Ch*).CORRection.PROPerty = *Status*

*Status* = SCPI.SENSe(*Ch*).CORRection.PROPerty

Description

For the active trace of channels 1 to 4 (*Ch*), turns ON/OFF the display of the calibration property.

Variable

| | ***Status*** |
|---|---|
| Description | ON/OFF of the display of the calibration property |
| Data type | Boolean type (Boolean) |
| Range | Select from the following.<br>•True or -1 Turns ON the display of the calibration property.<br>•False or 0 Turns OFF the display of the calibration property. |
| Preset value | False or 0 |

For information on the variable (*Ch*), see Table 7-4, “Variable (Ch),” on page 126.

Examples

```
Dim CalProp As Boolean
SCPI.SENSe(1).CORRection.PROPerty = True
CalProp = SCPI.SENSe(1).CORRection.PROPerty
```

Equivalent key

**[Cal]** - **Property**

# SCPI.SENSe(*Ch*).CORRection.RVELocity.COAX

Object type

Property

Syntax

SCPI.SENSe(*Ch*).CORRection.RVELocity.COAX = *Value*

*Value* = SCPI.SENSe(*Ch*).CORRection.RVELocity.COAX

Description

For channels 1 to 4 (*Ch*), sets the velocity factor.

Variable

| | ***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. |

For information on the variable (*Ch*), see Table 7-4, “Variable (Ch),” on page 126.

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**

# SCPI.SENSe(*Ch*).CORRection.STATe

Object type

Property

Syntax

SCPI.SENSe(*Ch*).CORRection.STATe = *Status*

*Status* = SCPI.SENSe(*Ch*).CORRection.STATe

Description

For the active trace of channels 1 to 4 (*Ch*), turns ON/OFF the error correction.

Variable

| | ***Status*** |
|---|---|
| Description | ON/OFF of the error correction |
| Data type | Boolean type (Boolean) |
| Range | Select from the following.<br>•True or -1 Turns ON the error correction.<br>•False or 0 Turns OFF the error correction. |
| Preset value | False or 0 |

Examples

```
Dim Corr As Boolean
SCPI.SENSe(1).CORRection.STATe = True
Corr = SCPI.SENSe(1).CORRection.STATe
```

Equivalent key

**[Cal]** - **Correction**

# SCPI.SENSe(*Ch*).CORRection.TYPE(*Tr*)

Object type

Properties

Syntax

*Data* = SCPI.SENSe(*Ch*).CORRection.TYPE(*Tr*)

Description

For traces 1 to 4 (*Tr*) of channels 1 to 4 (*Ch*), reads out the information (calibration type, port numbers) of the applied calibration coefficients for the actual error correction. (Read only)

Variable

| | ***Data*** |
|---|---|
| Description | Indicates 3 array data items (the calibration type and the port information to which the calibration is applied).<br>• *Data(0)* — The calibration type applied. For detail, refer to the Range section.<br>• *Data(1)* — The port number to which the calibration is applied (0 when the calibration type is NONE).<br>• *Data(2)* — The port number to which the calibration is applied (0 when the calibration type is not SOLT2, or ERES).<br>The array index starts from 0. |
| Range | One of the following is read out as *Data(0)*.<br>•"ERES" — The enhanced response calibration is applied.<br>•"NONE" — Nothing is applied.<br>•"RESPO" — The response calibration (open) is applied.<br>•"RESPS" — The response calibration (short) is applied.<br>•"RESPT" — The response calibration (thru) is applied.<br>•"SOLT1" — The full 1-port calibration is applied.<br>•"SOLT2" — The full 2-port calibration is applied. |
| Data type | Variant type (Variant) |

For information on the variable (*Ch*) and the variable (*Tr*), see Table 7-4, “Variable (Ch),” on page 126 and Table 7-5, “Variable (Tr),” on page 128, respectively.

Examples

```
Dim CalType As Variant
CalType = SCPI.SENSe(1).CORRection.TYPE(1)
```

Equivalent key

No equivalent key is available on the front panel.

# SCPI.SENSe(*Ch*).FREQuency.CENTer

Object type — Property

Syntax — SCPI.SENSe(*Ch*).FREQuency.CENTer = *Value*

*Value* = SCPI.SENSe(*Ch*).FREQuency.CENTer

Description — Sets the center value of the sweep range of channels 1 to 4 (*Ch*).

Variable

| | ***Value*** |
|---|---|
| Description | Center value |
| Data type | Double precision floating point type (Double) |
| Range | 3E5 to 3E9 |
| Preset value | 4.25015E9 |
| Unit | Hz (hertz) |
| Resolution | 0.5 or 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. |

For information on the variable (*Ch*), see Table 7-4, “Variable (Ch),” on page 126.

Examples

```
Dim Cntr As Double
SCPI.SENSe(1).FREQuency.CENTer = 2E9
Cntr = SCPI.SENSe(1).FREQuency.CENTer
```

Related objects — SCPI.SENSe(Ch).FREQuency.SPAN on page 357

Equivalent key — **[Center]**

# SCPI.SENSe(*Ch*).FREQuency.CW

Object type

Property

Syntax

SCPI.SENSe(*Ch*).FREQuency.CW = *Value*

*Value* = SCPI.SENSe(*Ch*).FREQuency.CW

Description

Sets the fixed frequency (CW frequency) for the power sweep for channels 1 to 4 (*Ch*).

This object provides the same function as the SCPI.SENSe(Ch).FREQuency.FIXed object.

Variable

| | ***Value*** |
|---|---|
| Description | Fixed frequency |
| Data type | Double precision floating point type (Double) |
| Range | 3E5 to 3E9 |
| Preset value | 3E5 |
| Unit | Hz (hertz) |
| 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. |

For information on the variable (*Ch*), refer to Table 7-4, “Variable (Ch),” on page 126.

Examples

```
Dim CwFreq As Double
SCPI.SENSe(1).FREQuency.CW = 1E9
CwFreq = SCPI.SENSe(1).FREQuency.CW
```

Related objects

SCPI.SENSe(Ch).FREQuency.FIXed on page 356

SCPI.SENSe(Ch).SWEep.TYPE on page 368

Equivalent key

**[Sweep Setup]** - - **CW Freq**

# SCPI.SENSe(*Ch*).FREQuency.DATA

Object type — Property

Syntax — *Data* = SCPI.SENSe(Ch).FREQuency.DATA

Description — Reads out the frequencies at all measurement points of channels 1 to 4 (Ch). (Read only)

Variable

| | ***Data*** |
|---|---|
| Description | Indicates the array data (frequency) of NOP (number of measurement points). Where n is an integer between 1 and NOP.<br>• *Data(n-1)* Frequency at the n-th measurement point<br>The index of the array starts from 0. |
| Data type | Variant type (Variant) |

For information on the variable (*Ch*), see Table 7-4, “Variable (Ch),” on page 126.

Examples

```
Dim FreqData As Variant
SCPI.SENSe(1).SWEep.POINts = 201
FreqData = SCPI.SENSe(1).FREQuency.DATA
```

Related objects — SCPI.SENSe(Ch).SWEep.POINts on page 365

Equivalent key — No equivalent key is available on the front panel.

# SCPI.SENSe(*Ch*).FREQuency.FIXed

Object type

Property

Syntax

SCPI.SENSe(*Ch*).FREQuency.FIXed = *Value*

*Value* = SCPI.SENSe(*Ch*).FREQuency.FIXed

Description

Sets the fixed frequency (CW frequency) for the power sweep for channels 1 to 4 (*Ch*).

This object provides the same function as the SCPI.SENSe(Ch).FREQuency.CW object.

Variable

| | ***Value*** |
|---|---|
| Description | Fixed frequency |
| Data type | Double precision floating point type (Double) |
| Range | 3E5 to 3E9 |
| Preset value | 3E5 |
| Unit | Hz (hertz) |
| 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. |

For information on the variable (*Ch*), refer to Table 7-4, “Variable (Ch),” on page 126.

Examples

```
Dim CwFreq As Double
SCPI.SENSe(1).FREQuency.FIXed = 1E9
CwFreq = SCPI.SENSe(1).FREQuency.FIXed
```

Related objects

SCPI.SENSe(Ch).FREQuency.CW on page 354

SCPI.SENSe(Ch).SWEep.TYPE on page 368

Equivalent key

**[Sweep Setup]** - **Power** - **CW Freq**

# SCPI.SENSe(*Ch*).FREQuency.SPAN

Object type — Property

Syntax — SCPI.SENSe(*Ch*).FREQuency.SPAN = *Value*

*Value* = SCPI.SENSe(*Ch*).FREQuency.SPAN

Description — Sets the span value of the sweep range of channels 1 to 4 (*Ch*).

Variable

| | ***Value*** |
|---|---|
| Description | Span value |
| Data type | Double precision floating point type (Double) |
| Range | 0 to 2.9997E9 |
| Preset value | 2.9997E9 |
| Unit | Hz (hertz) |
| 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. |

For information on the variable (*Ch*), see Table 7-4, “Variable (Ch),” on page 126.

Examples

```
Dim Span As Double
SCPI.SENSe(1).FREQuency.SPAN = 1E9
Span = SCPI.SENSe(1).FREQuency.SPAN
```

Related objects — SCPI.SENSe(Ch).FREQuency.CENTer on page 353

Equivalent key — **[Span]**

# SCPI.SENSe(*Ch*).FREQuency.STARt

Object type — Property

Syntax — SCPI.SENSe(*Ch*).FREQuency.STARt = *Value*

*Value* = SCPI.SENSe(*Ch*).FREQuency.STARt

Description — Sets the start value of the sweep range of channels 1 to 4 (*Ch*).

Variable

| | ***Value*** |
|---|---|
| Description | Start value |
| Data type | Double precision floating point type (Double) |
| Range | 3E5 to 3E9 |
| Preset value | 3E5 |
| Unit | Hz (hertz) |
| 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. |

For information on the variable (*Ch*), see Table 7-4, “Variable (Ch),” on page 126.

Examples

```
Dim Start As Double
SCPI.SENSe(1).FREQuency.STARt = 100E6
Start = SCPI.SENSe(1).FREQuency.STARt
```

Related objects — SCPI.SENSe(Ch).FREQuency.STOP on page 359

Equivalent key — **[Start]**

# SCPI.SENSe(*Ch*).FREQuency.STOP

Object type — Property

Syntax — SCPI.SENSe(*Ch*).FREQuency.STOP = *Value*

*Value* = SCPI.SENSe(*Ch*).FREQuency.STOP

Description — Sets the stop value of the sweep range of channels 1 to 4 (*Ch*).

Variable

| | ***Value*** |
|---|---|
| Description | Stop value |
| Data type | Double precision floating point type (Double) |
| Range | 3E5 to 3E9 |
| Preset value | 3E9 |
| Unit | Hz (hertz) |
| 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. |

For information on the variable (*Ch*), see Table 7-4, “Variable (Ch),” on page 126.

Examples

```
Dim Stp As Double
SCPI.SENSe(1).FREQuency.STOP = 3E9
Stp = SCPI.SENSe(1).FREQuency.STOP
```

Related objects — SCPI.SENSe(Ch).FREQuency.STARt on page 358

Equivalent key — **[Stop]**

# SCPI.SENSe(*Ch*).ROSCillator.SOURce

Object type — Property

Syntax — *Param* = SCPI.SENSe(*Ch*).ROSCillator.SOURce

Description — Reads out whether the external reference signal is inputted to the Ref In connector on the rear panel. (Read only)

Variable

| | ***Param*** |
|---|---|
| Description | Whether the external reference signal is inputted or not. |
| Data type | Character string type (String) |
| Range | Select from the following.<br>•"INTernal" The external reference signal is not inputted.<br>•"EXTernal" The external reference signal is inputted. |

For information on the variable (*Ch*), see Table 7-4, “Variable (Ch),” on page 126.

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).

# SCPI.SENSe(*Ch*).SEGMent.DATA

Object type

Property

Syntax

SCPI.SENSe(*Ch*).SEGMent.DATA = *Data*

*Data* = SCPI.SENSe(*Ch*).SEGMent.DATA

Description

Creates the segment sweep table of channels 1 to 4 (*Ch*).

Variable

| | ***Data*** |
|---|---|
| Description | 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.<br><br>*Data* = {<buf>,<stim>,<ifbw>,<pow>,<del>,<swp>,<time>,<segm>,<br><star 1>,<stop 1>,<nop 1>,<ifbw 1>,<pow 1>,<del 1>,<time 1>,···,<br><star n>,<stop n>,<nop n>,<ifbw n>,<pow n>,<del n>,<time n>,···,<br><star N>,<stop N>,<nop N>,<ifbw N>,<pow N>,<del N>,<time N>}<br><br>Each parameter in the above array data is detailed below.<br><br>• <buf> Always specify 5 or 6. You have to specify 6 if you need to set up the sweep mode setting for each segment.<br>• <stim> Stimulus setting mode<br>0: Specifies with start/stop values<br>1: Specifies with center/span values<br>• <ifbw> ON/OFF of the IF bandwidth setting for each segment<br>0: OFF, 1: ON<br>• <pow> ON/OFF of the power setting for each segment<br>0: OFF, 1: ON<br>• <del> ON/OFF of the sweep delay time setting for each segment<br>0: OFF, 1: ON<br>• <time> ON/OFF of the sweep time setting for each segment<br>0: OFF, 1: ON<br>• <segm> Number of segments<br>Specify an integer ranging 1 to 201.<br>• <star n> Start value/center value of the n-th segment<br>• <stop n> Stop value/span value of the n-th segment<br>• <nop n> Number of measurement points of the n-th segment<br>• <ifbw n> IF bandwidth of the n-th segment Not necessary when the IF bandwidth setting for each segment is OFF (<ifbw>:0).<br>• <pow n> Power of the n-th segment Not necessary when the power setting for each segment is OFF (<pow>:0).<br>• <del n> Sweep delay time of the n-th segment Not necessary when the sweep delay time setting for each segment is OFF (<del>:0). |
| Description | • <time n> Sweep time of the n-th segment Not necessary when the sweep time setting for each segment is OFF (<time>:0). |
| Data type | Variant type (Variant) |

| | ***Data*** |
|---|---|
| Note | If there is not the necessary amount of array data for the specified number of segments when setting the segment sweep table, an error occurs when executed and the object is ignored. For <stim>, <ifbw>, <pow>, <del>, <swp>, and <time>, if the specified value is not the allowable integer, an error occurs when 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. |

For information on the variable (*Ch*), see Table 7-4, “Variable (Ch),” on page 126.

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 on page 368

Equivalent key

**[Sweep Setup]** - **Edit Segment Table**

## SCPI.SENSe(*Ch*).SEGMent.SWEep.POINts

Object type — Property

Syntax — *Value* = SCPI.SENSe(*Ch*).SEGMent.SWEep.POINts

Description — For the segment sweep table of channels 1 to 4 (*Ch*), reads out the total number of the measurement points of all segments. (Read only)

Variable

| | ***Value*** |
|---|---|
| Description | Total number of measurement points of all segments |
| Data type | Long integer type (Long) |

For information on the variable (*Ch*), see Table 7-4, "Variable (Ch)," on page 126.

Examples

```
Dim SegmPoin As Long
SegmPoin = SCPI.SENSe(1).SEGMent.SWEep.POINts
```

Related objects — SCPI.SENSe(Ch).SEGMent.DATA on page 361

Equivalent key — No equivalent key is available on the front panel.

## SCPI.SENSe(*Ch*).SEGMent.SWEep.TIME.DATA

Object type — Property

Syntax — *Value* = SCPI.SENSe(*Ch*).SEGMent.SWEep.TIME.DATA

Description — For the segment sweep table of channels 1 to 4 (*Ch*), reads out the total sweep time (including sweep delay time) of all segments. (Read only)

Variable

| | ***Value*** |
|---|---|
| Description | Total sweep time of all segments |
| Data type | Double precision floating point type (Double) |

For information on the variable (*Ch*), see Table 7-4, "Variable (Ch)," on page 126.

Examples

```
Dim SegmTime As Double
SegmTime = SCPI.SENSe(1).SEGMent.SWEep.TIME.DATA
```

Related objects — SCPI.SENSe(Ch).SEGMent.DATA on page 361

Equivalent key — No equivalent key is available on the front panel.

# SCPI.SENSe(*Ch*).SWEep.DELay

Object type

Property

Syntax

SCPI.SENSe(*Ch*).SWEep.DELay = *Value*

*Value* = SCPI.SENSe(*Ch*).SWEep.DELay

Description

Sets the sweep delay time of channels 1 to 4 (*Ch*).

Variable

| | ***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. |

For information on the variable (*Ch*), see Table 7-4, “Variable (Ch),” on page 126.

Examples

```
Dim SweDel As Double
SCPI.SENSe(1).SWEep.DELay = 0.05
SweDel = SCPI.SENSe(1).SWEep.DELay
```

Equivalent key

**[Sweep Setup]** - **Sweep Delay**

# SCPI.SENSe(*Ch*).SWEep.POINts

Object type — Property

Syntax — SCPI.SENSe(*Ch*).SWEep.POINts = *Value*

*Value* = SCPI.SENSe(*Ch*).SWEep.POINts

Description — Sets the number of measurement points of channels 1 to 4 (*Ch*).

Variable

| | ***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. |

For information on the variable (*Ch*), see Table 7-4, “Variable (Ch),” on page 126.

Examples

```
Dim Nop As Long
SCPI.SENSe(1).SWEep.POINts = 801
Nop = SCPI.SENSe(1).SWEep.POINts
```

Equivalent key — **[Sweep Setup]** - **Points**

# SCPI.SENSe(*Ch*).SWEep.TIME.AUTO

Object type — Property

Syntax — SCPI.SENSe(*Ch*).SWEep.TIME.AUTO = *Status*

*Status* = SCPI.SENSe(*Ch*).SWEep.TIME.AUTO

Description — Sets whether to automatically set the sweep time of channels 1 to 4 (*Ch*).

Variable

| | ***Status*** |
|---|---|
| Description | ON/OFF of the auto setting of the sweep time |
| Data type | Boolean type (Boolean) |
| Range | Select from the following.<br>•True or -1 Turns ON the auto setting.<br>•False or 0 Turns OFF the auto setting. |
| Preset value | True or -1 |

For information on the variable (*Ch*), see Table 7-4, “Variable (Ch),” on page 126.

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 on page 367

Equivalent key — **[Sweep Setup]** - **Sweep Time**

**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.

# SCPI.SENSe(*Ch*).SWEep.TIME.DATA

Object type — Property

Syntax — SCPI.SENSe(*Ch*).SWEep.TIME.DATA = *Value*

*Value* = SCPI.SENSe(*Ch*).SWEep.TIME.DATA

Description — Sets the sweep time of channels 1 to 4 (*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

| | ***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. |

For information on the variable (*Ch*), see Table 7-4, “Variable (Ch),” on page 126.

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 on page 366

Equivalent key — **[Sweep Setup]** - **Sweep Time**

# SCPI.SENSe(*Ch*).SWEep.TYPE

Object type

Property

Syntax

SCPI.SENSe(*Ch*).SWEep.TYPE = *Param*

*Param* = SCPI.SENSe(*Ch*).SWEep.TYPE

Description

Sets the sweep type of channels 1 to 4 (*Ch*).

Variable

| | ***Param*** |
|---|---|
| Description | Sweep type |
| Data type | Character string type (String) |
| Range | Select from the following.<br>•"LINear" Sets the sweep type to the linear sweep.<br>•"LOGarithmic" Sets the sweep type to the log sweep. *1<br>•"SEGMent" Sets the sweep type to the segment sweep.<br>•"POWer" Sets the sweep type to the power sweep. |
| Preset value | "LINear" |

*1. If you execute this object to try to specify the log sweep when the frequency span condition necessary for the log sweep is not satisfied (the stop frequency is about 4 times or more the start frequency), an error occurs and the object is ignored.

For information on the variable (*Ch*), see Table 7-4, “Variable (Ch),” on page 126Table 7-4, “Variable (Ch),” on page 126.

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**

## SCPI.SERVice.CHANnel.ACTive

Object type — Property

Syntax — *Value* = SCPI.SERVice.CHANnel.ACTive

Description — Reads out the active channel number. (Read only)

Variable

| | ***Value*** |
|---|---|
| Description | Active channel number |
| Data type | Long integer type (Long) |

Examples

```
Dim ActChan As Long
ActChan = SCPI.SERVice.CHANnel.ACTive
```

Related objects — SCPI.DISPlay.WINDow(Ch).ACTivate on page 250

Equivalent key — No equivalent key is available on the front panel.

## SCPI.SERVice.CHANnel.COUNt

Object type — Property

Syntax — *Value* = SCPI.SERVice.CHANnel.COUNt

Description — Reads out the upper limit of the number of channels of the E5061A/E5062A. (Read only)

Variable

| | ***Value*** |
|---|---|
| Description | Upper limit of the number of channels. |
| Data type | Long integer type (Long) |

Examples

```
Dim MaxChan As Long
MaxChan = SCPI.SERVice.CHANnel.COUNt
```

Equivalent key — No equivalent key is available on the front panel.

## SCPI.SERVice.CHANnel(*Ch*).TRACe.ACTive

Object type Property

Syntax *Value* = SCPI.SERVice.CHANnel(*Ch*).TRACe.ACTive

Description Reads out the active trace number of channels 1 to 4 (*Ch*). (Read only)

Variable

| | ***Value*** |
|---|---|
| Description | Active trace number |
| Data type | Long integer type (Long) |

Examples

```
Dim ActTrac As Long
ActTrac = SCPI.SERVice.CHANnel(1).TRACe.ACTive
```

Related objects SCPI.CALCulate(Ch).PARameter(Tr).SELect on page 128

Equivalent key No equivalent key is available on the front panel.

## SCPI.SERVice.CHANnel.TRACe.COUNt

Object type Property

Syntax *Value* = SCPI.SERVice.CHANnel.TRACe.COUNt

Description Reads out the upper limit of the number of traces per channel. (Read only)

Variable

| | ***Value*** |
|---|---|
| Description | Upper limit of the number of traces. |
| Data type | Long integer type (Long) |

Examples

```
Dim MaxTrac As Long
MaxTrac = SCPI.SERVice.CHANnel.TRACe.COUNt
```

Equivalent key No equivalent key is available on the front panel.

# SCPI.SERVice.PORT.COUNt

Object type — Property

Syntax — *Value* = SCPI.SERVice.PORT.COUNt

Description — Reads out the number of ports of the E5061A/E5062A. (Read only)

Variable

| | ***Value*** |
|---|---|
| Description | Number of ports |
| Data type | Long integer type (Long) |

Examples

```
Dim MaxPort As Long
MaxPort = SCPI.SERVice.PORT.COUNt
```

Equivalent key — No equivalent key is available on the front panel.

# SCPI.SOURce(*Ch*).POWer.ATTenuation.DATA

Object type — Property

Syntax — SCPI.SOURce(*Ch*).POWer.ATTenuation.DATA = *Value*

*Value* = SCPI.SOURce(*Ch*).POWer.ATTenuation.DATA

Description — Selects the attenuator used for channels 1 to 4 (*Ch*). The power ranges are determined depending on the attenuator to be used.

**NOTE** This object is available only when extended power range function is installed.

Variable

| | ***Value*** | |
|---|---|---|
| Description | Power ranges | Setting |
| | -5 to +10[dB] | 0 |
| | -15 to 0 [dB] | 10 |
| | -25 to -10 [dB] | 20 |
| | -35 to -20 [dB] | 30 |
| | -45 to -30 [dB] | 40 |
| Data type | Long integer type (Long) | |
| Range | 0 to 40 | |
| Preset value | 0 | |
| Unit | dB | |
| Resolution | 10 | |
| 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. | |

For information on the variable (*Ch*), see Table 7-4, “Variable (Ch),” on page 126.

Examples

```
Dim Att As Long
SCPI.SOURce(1).POWer.ATTenuation.DATA = 10
Att = SCPI.SOURce(1).POWer.ATTenuation.DATA
```

Related objects — SCPI.SOURce(Ch).POWer.LEVel.IMMediate. AMPLitude on page 374

Equivalent key — **[Sweep Setup]** - **Power** - **Power Ranges**

# SCPI.SOURce(*Ch*).POWer.CENTer

Object type

Property

Syntax

SCPI.SOURce(*Ch*).POWer.CENTer = *Value*

*Value* = SCPI.SOURce(*Ch*).POWer.CENTer

Description

Sets the center value of the sweep range for the power sweep for channels 1 to 4 (*Ch*).

Variable

| | ***Value*** |
|---|---|
| Description | Center value |
| Data type | Double precision floating point type (Double) |
| Range | Varies depending on the power range. |
| Preset value | -7.5 |
| 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. |

For information on the variable (*Ch*), refer to Table 7-4, “Variable (Ch),” on page 126Table 7-4, “Variable (Ch),” on page 126.

Examples

```
Dim Pcntr As Double
SCPI.SOURce(1).POWer.CENTer = 0
Pcntr = SCPI.SOURce(1).POWer.CENTer
```

Related objects

SCPI.SENSe(Ch).SWEep.TYPE on page 368

SCPI.SOURce(Ch).POWer.ATTenuation.DATA on page 372

SCPI.SOURce(Ch).POWer.SPAN on page 379

Equivalent key

**[Center]**

# SCPI.SOURce(*Ch*).POWer.LEVel.IMMediate. AMPLitude

Object type — Property

Syntax — SCPI.SOURce(*Ch*).POWer.LEVel.IMMediate.AMPLitude = *Value*

*Value* = SCPI.SOURce(*Ch*).POWer.LEVel.IMMediate.AMPLitude

Description — Sets the power level of channels 1 to 4 (*Ch*).

Variable

| | ***Value*** |
|---|---|
| Description | Power level |
| Data type | Double precision floating point type (Double) |
| Range | Varies depending on the power range. |
| 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. |

For information on the variable (*Ch*), see Table 7-4, “Variable (Ch),” on page 126.

Examples

```
Dim PowLev As Double
SCPI.SOURce(1).POWer.LEVel.IMMediate.AMPLitude = -5
PowLev = SCPI.SOURce(1).POWer.LEVel.IMMediate.AMPLitude
```

Related objects — SCPI.SOURce(Ch).POWer.ATTenuation.DATA on page 372

Equivalent key — **[Sweep Setup]** - **Power**

# SCPI.SOURce(*Ch*).POWer.LEVel.SLOPe.DATA

Object type — Property

Syntax — SCPI.SOURce(*Ch*).POWer.LEVel.SLOPe.DATA = *Value*

*Value* = SCPI.SOURce(*Ch*).POWer.LEVel.SLOPe.DATA

Description — Sets the correction value of the power slope feature of channels 1 to 4 (*Ch*).

Variable

| | ***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. |

For information on the variable (*Ch*), refer to Table 7-4, “Variable (Ch),” on page 126.

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 on page 376

Equivalent key — **[Sweep Setup]** - **Power** - **Slop [xxx dB/GHz]**

# SCPI.SOURce(*Ch*).POWer.LEVel.SLOPe.STATe

Object type

Property

Syntax

SCPI.SOURce(*Ch*).POWer.LEVel.SLOPe.STATe = *Status*

*Status* = SCPI.SOURce(*Ch*).POWer.LEVel.SLOPe.STATe

Description

Turns on/off the power slope feature for channels 1 to 4 (*Ch*). This function is a function to correct the attenuation of simple power level proportional to the frequency (attenuation due to cables and so on).

Variable

| | ***Status*** |
|---|---|
| Description | On/off of the power slope feature |
| Data type | Boolean type (Boolean) |
| Range | Select from the following.<br>•True or -1 Turns on the power slop feature.<br>•False or 0 Turns off the power slop feature. |
| Preset value | False or 0 |

For information on the variable (*Ch*), refer to Table 7-4, “Variable (Ch),” on page 126.

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 on page 375

Equivalent key

**[Sweep Setup]** - **Power** - **Slop [ON/OFF]**

# SCPI.SOURce(*Ch*).POWer.PORT.COUPle

Object type

Property

Syntax

SCPI.SOURce(*Ch*).POWer.PORT.COUPle = *Status*

*Status* = SCPI.SOURce(*Ch*).POWer.PORT.COUPle

Description

Sets whether to output the same power level for each port of channels 1 to 4 (*Ch*). When the power slope feature is on, the same power level is always outputted to all ports regardless of this setting because different power levels cannot be outputted for each port.

Variable

| | ***Status*** |
|---|---|
| Description | Turning on/off the coupling between ports for the power level output |
| Data type | Boolean type (Boolean) |
| Range | Select from the following.<br>•True or -1 Outputs the same power level to individual ports.<br>•False or 0 Outputs different power levels to individual ports. |
| Preset value | True or -1 |

For information on the variable (*Ch*), refer to Table 7-4, “Variable (Ch),” on page 126.

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 on page 378

Equivalent key

**[Sweep Setup]** - **Power** - **Port Couple**

# SCPI.SOURce(*Ch*).POWer.PORT(*Pt*).LEVel.IMMediate. AMPLitude

Object type — Property

Syntax — SCPI.SOURce(*Ch*).POWer.PORT(*Pt*).LEVel.IMMediate.AMPLitude = *Value*

*Value* = SCPI.SOURce(*Ch*).POWer.PORT(*Pt*).LEVel.IMMediate.AMPLitude

Description — For ports 1 to 2 (*Pt*) of channels 1 to 4 (*Ch*), sets the power level.

Variable

**Table 7-12 Variable (*Pt*)**

| | ***Pt*** |
|---|---|
| Description | Port 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. |

| | ***Value*** |
|---|---|
| Description | Power level at the specified port. |
| Data type | Double precision floating point type (Double) |
| Range | Varies depending on the power range. |
| 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. |

For information on the variable (*Ch*) refer to Table 7-4, “Variable (Ch),” on page 126.

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 on page 377

SCPI.SOURce(Ch).POWer.ATTenuation.DATA on page 372

Equivalent key — **[Sweep Setup]** - **Power** - **Port Power** - **Port 1 Power | Port 2 Power**

# SCPI.SOURce(*Ch*).POWer.SPAN

Object type

Property

Syntax

SCPI.SOURce(*Ch*).POWer.SPAN = *Value*

*Value* = SCPI.SOURce(*Ch*).POWer.SPAN

Description

Sets the span value of the sweep range for the power sweep for channels 1 to 4 (*Ch*).

Variable

| | ***Value*** |
|---|---|
| Description | Span value |
| Data type | Double precision floating point type (Double) |
| Range | Varies depending on the power range. |
| Preset value | 5 |
| 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. |

For information on the variable (*Ch*), refer to Table 7-4, “Variable (Ch),” on page 126.

Examples

```
Dim Pspan As Double
SCPI.SOURce(1).POWer.SPAN = 10
Pspan = SCPI.SOURce(1).POWer.SPAN
```

Related objects

SCPI.SENSe(Ch).SWEep.TYPE on page 368

SCPI.SOURce(Ch).POWer.ATTenuation.DATA on page 372

SCPI.SOURce(Ch).POWer.CENTer on page 373

Equivalent key

**[Span]**

# SCPI.SOURce(*Ch*).POWer.STARt

Object type

Property

Syntax

SCPI.SOURce(*Ch*).POWer.STARt = *Value*

*Value* = SCPI.SOURce(*Ch*).POWer.STARt

Description

Sets the start value of the sweep range for the power sweep for channels 1 to 4 (*Ch*).

Variable

| | ***Value*** |
|---|---|
| Description | Start value |
| Data type | Double precision floating point type (Double) |
| Range | Varies depending on the power range. |
| Preset value | -5 |
| 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. |

For information on the variable (*Ch*), refer to Table 7-4, “Variable (Ch),” on page 126.

Examples

```
Dim Pstart As Double
SCPI.SOURce(1).POWer.STARt = -10
Pstart = SCPI.SOURce(1).POWer.STARt
```

Related objects

SCPI.SENSe(Ch).SWEep.TYPE on page 368

SCPI.SOURce(Ch).POWer.ATTenuation.DATA on page 372

SCPI.SOURce(Ch).POWer.STOP on page 381

Equivalent key

**[Start]**

# SCPI.SOURce(*Ch*).POWer.STOP

Object type

Property

Syntax

SCPI.SOURce(*Ch*).POWer.STOP = *Value*

*Value* = SCPI.SOURce(*Ch*).POWer.STOP

Description

Sets the stop value of the sweep range for the power sweep for channels 1 to 4 (*Ch*).

Variable

| | ***Value*** |
|---|---|
| Description | Stop value |
| Data type | Double precision floating point type (Double) |
| Range | Varies depending on the power range. |
| 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. |

For information on the variable (*Ch*), refer to Table 7-4, “Variable (Ch),” on page 126.

Examples

```
Dim Pstop As Double
SCPI.SOURce(1).POWer.STOP = 10
Pstop = SCPI.SOURce(1).POWer.STOP
```

Related objects

SCPI.SENSe(Ch).SWEep.TYPE on page 368

SCPI.SOURce(Ch).POWer.ATTenuation.DATA on page 372

SCPI.SOURce(Ch).POWer.STARt on page 380

Equivalent key

**[Stop]**

## SCPI.STATus.OPERation.CONDition

Object type Property

Syntax *Value* = SCPI.STATus.OPERation.CONDition

Description Reads out the value of the Operation Status Condition Register. (Read only)

Variable

| | ***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 on page 383

SCPI.STATus.OPERation.PTRansition on page 384

Equivalent key No equivalent key is available on the front panel.

## SCPI.STATus.OPERation.ENABle

Object type Property

Syntax SCPI.STATus.OPERation.ENABle = *Value*

*Value* = SCPI.STATus.OPERation.ENABle

Description Sets the value of the Operation Status Enable Register.

Variable

| | ***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 to13 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 on page 278

Equivalent key No equivalent key is available on the front panel.

## SCPI.STATus.OPERation.EVENt

Object type Property

Syntax *Value* = SCPI.STATus.OPERation.EVENt

Description Reads out the value of the Operation Status Event Register. (Read only)

Variable

| | ***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 on page 274

SCPI.STATus.OPERation.NTRansition on page 383

SCPI.STATus.OPERation.PTRansition on page 384

Equivalent key No equivalent key is available on the front panel.

## SCPI.STATus.OPERation.NTRansition

Object type Property

Syntax SCPI.STATus.OPERation.NTRansition = *Value*

*Value* = SCPI.STATus.OPERation.NTRansition

Description Sets the value of negative transition filter of the Operation Status Register.

Variable

| | ***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 to13 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 on page 383

SCPI.STATus.OPERation.PTRansition on page 384

Equivalent key No equivalent key is available on the front panel.

## SCPI.STATus.OPERation.PTRansition

Object type: Property

Syntax: SCPI.STATus.OPERation.PTRansition = *Value*

*Value* = SCPI.STATus.OPERation.PTRansition

Description: Sets the value of positive transition filter of the Operation Status Register.

Variable

| | ***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 to13 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 on page 383

SCPI.STATus.OPERation.NTRansition on page 383

Equivalent key: No equivalent key is available on the front panel.

## SCPI.STATus.PRESet

Object type: Method

Syntax: SCPI.STATus.PRESet

Description: Initialize the Operation Status Register, Questionable Status Register, Questionable Limit Status Register, and Questionable Limit Chnel{1-4} Status Register. (No read)

Examples

```
SCPI.STATus.PRESet
```

Equivalent key: No equivalent key is available on the front panel.

# SCPI.STATus.QUEStionable.BLIMit.CHANnel(*Ch*).CONDition

Object type

Property

Syntax

*Value* = SCPI.STATUS.QUEStionable.BLIMit.CHANnel(*Ch*).CONDition

Description

Reads out the value of the Questionable Bandwidth Limit Channel Status Condition Register of channel 1  to channel 4 . (Read only)

Variable

| | ***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 Table 7-4, “Variable (Ch),” on page 126.

Examples

```
Dim Stat As Long
Stat = SCPI.STATus.QUEStionable.BLIMit.CHANnel(1).CONDition
```

Related objects

SCPI.STATus.QUEStionable.BLIMit.CHANnel(Ch).NTRansition on page 388

SCPI.STATus.QUEStionable.BLIMit.CHANnel(Ch).PTRansition on page 389

Equivalent key

No equivalent key is available on the front panel.

# SCPI.STATus.QUEStionable.BLIMit.CHANnel(*Ch*).ENABle

Object type

Property

Syntax

SCPI.STATus.QUEStionable.BLIMit.CHANnel(*Ch*).ENABle = *Value*

*Value* = SCPI.STATus.QUEStionable.BLIMit.CHANnel(*Ch*).ENABle

Description

Sets the value of the Questionable Bandwidth Limit Channel Status Enable Register of channel 1 to channel 4 .

Variable

| | ***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 5 to 15 can not be set to 1. |

For information on the variable (*Ch*), see Table 7-4, “Variable (Ch),” on page 126.

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 on page 391

Equivalent key

No equivalent key is available on the front panel.

# SCPI.STATus.QUEStionable.BLIMit.CHANnel(Ch).EVENt

Object type — Property

Syntax — *Value* = SCPI.STATus.QUEStionable.BLIMit.CHANnel(*Ch*).EVENt

Description — Reads out the value of the Questionable Bandwidth Limit Channel Status Event Register of channel 1 to channel 4 . (Read only)

Variable

| | ***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 Table 7-4, “Variable (Ch),” on page 126.

Examples

```
Dim Stat As Long
Stat = SCPI.STATus.QUEStionable.BLIMit.CHANnel(1).EVENt
```

Related objects — SCPI.IEEE4882.CLS on page 274

Equivalent key — No equivalent key is available on the front panel.

# SCPI.STATus.QUEStionable.BLIMit.CHANnel(*Ch*).NTRansition

Object type

Property

Syntax

SCPI.STATus.QUEStionable.BLIMit.CHANnel(*Ch*).NTRansition = *Value*

*Value* = SCPI.STATus.QUEStionable.BLIMit.CHANnel(*Ch*).NTRansition

Description

Sets the value of the negative transition filter of the Questionable Bandwidth Limit Channel Status Register of channel 1 to channel 4 .

Variable

| | ***Value*** |
|---|---|
| Description | The value of the negative transition filter of the Questionable Bandwidth Limit Channel Status Register |
| Data type | Long integer type (Long) |
| Range | 0 to 65535 |
| Preset value | 0 |
| Note | The bit 5 to 15 can not be set to 1. |

For information on the variable (*Ch*), see Table 7-4, “Variable (Ch),” on page 126.

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 on page 387

SCPI.STATus.QUEStionable.BLIMit.CHANnel(Ch).PTRansition on page 389

Equivalent key

No equivalent key is available on the front panel.

# SCPI.STATus.QUEStionable.BLIMit.CHANnel(*Ch*).PTRansition

Object type

Property

Syntax

SCPI.STATus.QUEStionable.BLIMit.CHANnel(*Ch*).PTRansition = *Value*

*Value* = SCPI.STATus.QUEStionable.BLIMit.CHANnel(*Ch*).PTRansition

Description

Sets the value of the positive transition filter of the Questionable Bandwidth Limit Channel Status Register of channel 1 to channel 4 .

Variable

| | ***Value*** |
|---|---|
| Description | The value of the positive transition filter of the Questionable Bandwidth Limit Channel Status 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 5 to 15 can not be set to 1. |

For information on the variable (*Ch*), see Table 7-4, “Variable (Ch),” on page 126.

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 on page 387

SCPI.STATus.QUEStionable.BLIMit.CHANnel(Ch).NTRansition on page 388

Equivalent key

No equivalent key is available on the front panel.

# SCPI.STATus.QUEStionable.BLIMit.CONDition

Object type — Property

Syntax — *Value* = SCPI.STATus.QUEStionable.BLIMit.CONDition

Description — Reads out the value of the Questionable Bandwidth Limit Status Condition Register. (Read only)

Variable

| | ***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 on page 392

SCPI.STATus.QUEStionable.BLIMit.PTRansition on page 393

Equivalent key — No equivalent key is available on the front panel.

## SCPI.STATus.QUEStionable.BLIMit.ENABle

Object type: Property

Syntax: SCPI.STATus.QUEStionable.BLIMit.ENABle = *Value*

*Value* = SCPI.STATus.QUEStionable.BLIMit.ENABle

Description: Sets the value of the Questionable Bandwidth Limit Status Enable Register.

Variable

| | ***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 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 on page 394

Equivalent key: No equivalent key is available on the front panel.

## SCPI.STATus.QUEStionable.BLIMit.EVENt

Object type: Property

Syntax: *Value* = SCPI.STATus.QUEStionable.BLIMit.EVENt

Description: Reads out the value of the Questionable Bandwidth Limit Status Event Register. (Read only)

Variable

| | ***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 on page 274

Equivalent key: No equivalent key is available on the front panel.

# SCPI.STATus.QUEStionable.BLIMit.NTRansition

Object type

Property

Syntax

SCPI.STATus.QUEStionable.BLIMit.NTRansition = *Value*

*Value* = SCPI.STATus.QUEStionable.BLIMit.NTRansition

Description

Sets the value of the negative transition filter of the Questionable Bandwidth Limit Status Register.

Variable

| | ***Value*** |
|---|---|
| Description | The value of the negative transition filter of the Questionable Bandwidth Limit Status Register. |
| Data type | Long integer type (Long) |
| Range | 0 to 65535 |
| Preset value | 0 |
| Note | The 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 on page 391

SCPI.STATus.QUEStionable.BLIMit.PTRansition on page 393

Equivalent key

No equivalent key is available on the front panel.

# SCPI.STATus.QUEStionable.BLIMit.PTRansition

Object type

Property

Syntax

SCPI.STATus.QUEStionable.BLIMit.PTRansition = *Value*

*Value* = SCPI.STATus.QUEStionable.BLIMit.PTRansition

Description

Sets the value of the positive transition filter of the Questionable Bandwidth Limit Status Register.

Variable

| | ***Value*** |
|---|---|
| Description | The value of the positive transition filter of the Questionable Bandwidth Limit Status 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 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 on page 391

SCPI.STATus.QUEStionable.BLIMit.NTRansition on page 392

Equivalent key

No equivalent key is available on the front panel.

# SCPI.STATus.QUEStionable.CONDition

Object type — Property

Syntax — *Value* = SCPI.STATus.QUEStionable.CONDition

Description — Reads out the value of the Questionable Status Condition Register. (Read only)

Variable

| | ***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 on page 404

SCPI.STATus.QUEStionable.PTRansition on page 405

Equivalent key — No equivalent key is available on the front panel.

# SCPI.STATus.QUEStionable.ENABle

Object type — Property

Syntax — SCPI.STATus.QUEStionable.ENABle = *Value*

*Value* = SCPI.STATus.QUEStionable.ENABle

Description — Sets the value of the Questionable Status Enable Register.

Variable

| | ***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 9 and bit 11 to 15 can not be set to 1. |

Examples

```
Dim Stat As Long
SCPI.STATus.QUEStionable.ENABle = 6
Stat = SCPI.STATus.QUEStionable.ENABle
```

Related objects — SCPI.IEEE4882.SRE on page 278

Equivalent key — No equivalent key is available on the front panel.

## SCPI.STATus.QUEStionable.EVENt

Object type: Property

Syntax: *Value* = SCPI.STATus.QUEStionable.EVENt

Description: Reads out the value of the Questionable Status Event Register. (Read only)

Variable

| | ***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 on page 274
SCPI.STATus.QUEStionable.NTRansition on page 404
SCPI.STATus.QUEStionable.PTRansition on page 405

Equivalent key: No equivalent key is available on the front panel.

## SCPI.STATus.QUEStionable.LIMit.CHANnel(*Ch*). CONDition

Object type: Property

Syntax: *Value* = SCPI.STATus.QUEStionable.LIMit.CHANnel(*Ch*).CONDition

Description: Reads out the value of the Questionable Limit Channel Status Condition Register of channels 1 to 4 (*Ch*). (Read only)

Variable

| | ***Value*** |
|---|---|
| Description | Value of the Questionable Limit Channel Status Condition Register |
| Data type | Long integer type (Long) |

For information on the variable (*Ch*), see Table 7-4, “Variable (Ch),” on page 126.

Examples

```
Dim Stat As Long
Stat = SCPI.STATus.QUEStionable.LIMit.CHANnel(1).CONDition
```

Related objects

SCPI.STATus.QUEStionable.LIMit.CHANnel(Ch). NTRansition on page 398
SCPI.STATus.QUEStionable.LIMit.CHANnel(Ch). PTRansition on page 399

Equivalent key: No equivalent key is available on the front panel.

# SCPI.STATus.QUEStionable.LIMit.CHANnel(*Ch*).ENABle

Object type

Property

Syntax

SCPI.STATus.QUEStionable.LIMit.CHANnel(*Ch*).ENABle = *Value*

*Value* = SCPI.STATus.QUEStionable.LIMit.CHANnel(*Ch*).ENABle

Description

Sets the value of the Questionable Limit Channel Status Enable Register of channels 1 to 4 (*Ch*).

Variable

| | ***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 for the channel/trace number. |
| Note | The bit 5 to 15 can not be set to 1. |

For information on the variable (*Ch*), see Table 7-4, “Variable (Ch),” on page 126.

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 on page 401

Equivalent key

No equivalent key is available on the front panel.

# SCPI.STATus.QUEStionable.LIMit.CHANnel(*Ch*).EVENt

Object type — Property

Syntax — *Value* = SCPI.STATus.QUEStionable.LIMit.CHANnel(*Ch*).EVENt

Description — Reads out the value of the Questionable Limit Channel Status Event Register of channels 1 to 4 (*Ch*). (Read only)

Variable

| | ***Value*** |
|---|---|
| Description | Value of the Questionable Limit Channel Status Event Register of the specified channel |
| Data type | Long integer type (Long) |

For information on the variable (*Ch*), see Table 7-4, “Variable (Ch),” on page 126.

Examples

```
Dim Stat As Long
Stat = SCPI.STATus.QUEStionable.LIMit.CHANnel(1).EVENt
```

Related objects — SCPI.IEEE4882.CLS on page 274

Equivalent key — No equivalent key is available on the front panel.

# SCPI.STATus.QUEStionable.LIMit.CHANnel(*Ch*). NTRansition

Object type

Property

Syntax

SCPI.STATus.QUEStionable.LIMit.CHANnel(*Ch*).NTRansition = *Value*

*Value* = SCPI.STATus.QUEStionable.LIMit.CHANnel(*Ch*).NTRansition

Description

Sets the value of the negative transition filter of the Questionable Limit Channel Status Register of channels 1 to 4 (*Ch*).

Variable

| | ***Value*** |
|---|---|
| Description | Value of the negative transition filter |
| Data type | Long integer type (Long) |
| Range | 0 to 65535 |
| Preset value | 0 |
| Note | The bit 5 to 15 can not be set to 1. |

For information on the variable (*Ch*), see Table 7-4, “Variable (Ch),” on page 126.

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 on page 397

SCPI.STATus.QUEStionable.LIMit.CHANnel(Ch). PTRansition on page 399

Equivalent key

No equivalent key is available on the front panel.

# SCPI.STATus.QUEStionable.LIMit.CHANnel(*Ch*). PTRansition

Object type

Property

Syntax

SCPI.STATus.QUEStionable.LIMit.CHANnel(*Ch*).PTRansition = *Value*

*Value* = SCPI.STATus.QUEStionable.LIMit.CHANnel(*Ch*).PTRansition

Description

Sets the value of the positive transition filter of the Questionable Limit Channel Status Register of channels 1 to 4 (*Ch*).

Variable

| | ***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 for the channel/trace number. |
| Note | The bit 5 to 15 can not be set to 1. |

For information on the variable (*Ch*), see Table 7-4, “Variable (Ch),” on page 126.

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 on page 397

SCPI.STATus.QUEStionable.LIMit.CHANnel(Ch). NTRansition on page 398

Equivalent key

No equivalent key is available on the front panel.

# SCPI.STATus.QUEStionable.LIMit.CONDition

Object type — Property

Syntax — *Value* = SCPI.STATus.QUEStionable.LIMit.CONDition

Description — Reads out the value of the Questionable Limit Status Condition Register. (Read only)

Variable

| | ***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 on page 402

SCPI.STATus.QUEStionable.LIMit.PTRansition on page 403

Equivalent key — No equivalent key is available on the front panel.

## SCPI.STATus.QUEStionable.LIMit.ENABle

Object type: Property

Syntax: SCPI.STATus.QUEStionable.LIMit.ENABle = *Value*

*Value* = SCPI.STATus.QUEStionable.LIMit.ENABle

Description: Sets the value of the Questionable Limit Status Enable Register.

Variable

| | ***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 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 on page 394

Equivalent key: No equivalent key is available on the front panel.

## SCPI.STATus.QUEStionable.LIMit.EVENt

Object type: Property

Syntax: *Value* = SCPI.STATus.QUEStionable.LIMit.EVENt

Description: Reads out the value of the Questionable Limit Status Event Register. (Read only)

Variable

| | ***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 on page 274

Equivalent key: No equivalent key is available on the front panel.

# SCPI.STATus.QUEStionable.LIMit.NTRansition

Object type

Property

Syntax

SCPI.STATus.QUEStionable.LIMit.NTRansition = *Value*

*Value* = SCPI.STATus.QUEStionable.LIMit.NTRansition

Description

Sets the value of negative transition filter of the Questionable Limit Status Register.

Variable

| | ***Value*** |
|---|---|
| Description | Value of the negative transition filter |
| Data type | Long integer type (Long) |
| Range | 0 to 65535 |
| Preset value | 0 |
| Note | The 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 on page 401

SCPI.STATus.QUEStionable.LIMit.PTRansition on page 403

Equivalent key

No equivalent key is available on the front panel.

# SCPI.STATus.QUEStionable.LIMit.PTRansition

Object type

Property

Syntax

SCPI.STATus.QUEStionable.LIMit.PTRansition = *Value*

*Value* = SCPI.STATus.QUEStionable.LIMit.PTRansition

Description

Sets the value of positive transition filter of the Questionable Limit Status Register.

Variable

| | ***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 for the channel/trace number. |
| Note | The 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 on page 401

SCPI.STATus.QUEStionable.LIMit.NTRansition on page 402

Equivalent key

No equivalent key is available on the front panel.

# SCPI.STATus.QUEStionable.NTRansition

Object type

Property

Syntax

SCPI.STATus.QUEStionable.NTRansition = *Value*

*Value* = SCPI.STATus.QUEStionable.NTRansition

Description

Sets the value of negative transition filter of the Questionable Status Register.

Variable

| | ***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 9 and bit 11 to 15 can not be set to 1. |

Examples

```
Dim Stat As Long
SCPI.STATus.QUEStionable.NTRansition = 6
Stat = SCPI.STATus.QUEStionable.NTRansition
```

Related objects

SCPI.STATus.QUEStionable.EVENt on page 395

SCPI.STATus.QUEStionable.PTRansition on page 405

Equivalent key

No equivalent key is available on the front panel.

# SCPI.STATus.QUEStionable.PTRansition

Object type

Property

Syntax

SCPI.STATus.QUEStionable.PTRansition = *Value*

*Value* = SCPI.STATus.QUEStionable.PTRansition

Description

Sets the value of positive transition filter of the Questionable Status Register.

Variable

| | ***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 9 and bit 11 to 15 can not be set to 1. |

Examples

```
Dim Stat As Long
SCPI.STATus.QUEStionable.PTRansition = 6
Stat = SCPI.STATus.QUEStionable.PTRansition
```

Related objects

SCPI.STATus.QUEStionable.EVENt on page 395

SCPI.STATus.QUEStionable.NTRansition on page 404

Equivalent key

No equivalent key is available on the front panel.

# SCPI.STATus.QUEStionable.RLIMit.CHANnel(*Ch*).CONDition

Object type

Property

Syntax

*Value* = SCPI.STATus.QUEStionable.RLIMit.CHANnel(*Ch*).CONDition

Description

Reads out the value of the Questionable Ripple Limit Channel Status Condition Register of channel 1 to channel 4 . (Read only)

Variable

| | ***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 Table 7-4, “Variable (Ch),” on page 126.

Examples

```
Dim Stat As Long
Stat = SCPI.STATus.QUEStionable.RLIMit.CHANnel(1).CONDition
```

Related objects

SCPI.STATus.QUEStionable.RLIMit.CHANnel(Ch).NTRansition on page 409

SCPI.STATus.QUEStionable.RLIMit.CHANnel(Ch).PTRansition on page 410

Equivalent key

No equivalent key is available on the front panel.

# SCPI.STATus.QUEStionable.RLIMit.CHANnel(*Ch*).ENABle

Object type

Property

Syntax

SCPI.STATus.QUEStionable.RLIMit.CHANnel(*Ch*).ENABle = *Value*

*Value* = SCPI.STATus.QUEStionable.RLIMit.CHANnel(*Ch*).ENABle

Description

Sets the value of the Questionable Ripple Limit Channel Status Enable Register of channel 1 to channel 4 .

Variable

| | ***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 5 to 15 can not be set to 1. |

For information on the variable (*Ch*), see Table 7-4, “Variable (Ch),” on page 126.

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 on page 412

Equivalent key

No equivalent key is available on the front panel.

# SCPI.STATus.QUEStionable.RLIMit.CHANnel(*Ch*).EVENt

Object type — Property

Syntax — *Value* = SCPI.STATus.QUEStionable.RLIMit.CHANnel(*Ch*).EVENt

Description — Reads out the value of the Questionable Ripple Limit Channel Status Event Register of channel 1 to channel 4 . (Read only)

Variable

| | ***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 Table 7-4, “Variable (Ch),” on page 126.

Examples

```
Dim Stat As Long
Stat = SCPI.STATus.QUEStionable.RLIMit.CHANnel(1).EVENt
```

Related objects — SCPI.IEEE4882.CLS on page 274

Equivalent key — No equivalent key is available on the front panel.

# SCPI.STATus.QUEStionable.RLIMit.CHANnel(*Ch*).NTRansition

Object type

Property

Syntax

SCPI.STATus.QUEStionable.RLIMit.CHANnel(*Ch*).NTRansition = *Value*

*Value* = SCPI.STATus.QUEStionable.RLIMit.CHANnel(*Ch*).NTRansition

Description

Sets the value of the negative transition filter of the Questionable Ripple Limit Channel Status Register of channel 1 to channel 4 .

Variable

| | ***Value*** |
|---|---|
| Description | The value of the negative transition filter of the Questionable Ripple Limit Channel Status Register. |
| Data type | Long integer type (Long) |
| Range | 0 to 65535 |
| Preset value | 0 |
| Note | The bit 5 to 15 can not be set to 1. |

For information on the variable (*Ch*), see Table 7-4, “Variable (Ch),” on page 126.

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 on page 408

SCPI.STATus.QUEStionable.RLIMit.CHANnel(Ch).PTRansition on page 410

Equivalent key

No equivalent key is available on the front panel.

# SCPI.STATus.QUEStionable.RLIMit.CHANnel(*Ch*).PTRansition

Object type

Property

Syntax

SCPI.STATus.QUEStionable.RLIMit.CHANnel(*Ch*).PTRansition = *Value*

*Value* = SCPI.STATus.QUEStionable.RLIMit.CHANnel(*Ch*).PTRansition

Description

Sets the value of the positive transition filter of the Questionable Ripple Limit Channel Status Register of channel 1 to channel 4.

Variable

| | ***Value*** |
|---|---|
| Description | The value of the positive transition filter of the Questionable Ripple Limit Channel Status 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 | Bits 15 cannot be set to 1. |

For information on the variable (*Ch*), see Table 7-4, “Variable (Ch),” on page 126.

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 on page 408

SCPI.STATus.QUEStionable.RLIMit.CHANnel(Ch).NTRansition on page 409

Equivalent key

No equivalent key is available on the front panel.

# SCPI.STATus.QUEStionable.RLIMit.CONDition

Object type — Property

Syntax — *Value* = SCPI.STATus.QUEStionable.RLIMit.CONDition

Description — Reads out the value of the Questionable Ripple Limit Status Condition Register. (Read only)

Variable

| | ***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 on page 413

SCPI.STATus.QUEStionable.RLIMit.PTRansition on page 414

Equivalent key — No equivalent key is available on the front panel.

# SCPI.STATus.QUEStionable.RLIMit.ENABle

Object type Property

Syntax SCPI.STATus.QUEStionable.RLIMit.ENABle = *Value*

*Value* = SCPI.STATus.QUEStionable.RLIMit.ENABle

Description Sets the value of the Questionable Ripple Limit Status Enable Register.

Variable

| | ***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 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 on page 394

Equivalent key No equivalent key is available on the front panel.

# SCPI.STATus.QUEStionable.RLIMit.EVENt

Object type Property

Syntax *Value* = SCPI.STATus.QUEStionable.RLIMit.EVENt

Description Reads out the value of the Questionable Ripple Limit Status Event Register. (Read only)

Variable

| | ***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 on page 274

Equivalent key No equivalent key is available on the front panel.

# SCPI.STATus.QUEStionable.RLIMit.NTRansition

Object type

Property

Syntax

SCPI.STATus.QUEStionable.RLIMit.NTRansition = *Value*

*Value* = SCPI.STATus.QUEStionable.RLIMit.NTRansition

Description

Sets the value of the negative transition filter of the Questionable Ripple Limit Status Register.

Variable

| | ***Value*** |
|---|---|
| Description | The value of the negative transition filter of the Questionable Ripple Limit Status Register. |
| Data type | Long integer type (Long) |
| Range | 0 to 65535 |
| Preset value | 0 |
| Note | The 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 on page 412

SCPI.STATus.QUEStionable.RLIMit.PTRansition on page 414

Equivalent key

No equivalent key is available on the front panel.

# SCPI.STATus.QUEStionable.RLIMit.PTRansition

Object type

Property

Syntax

SCPI.STATus.QUEStionable.RLIMit.PTRansition = *Value*

*Value* = SCPI.STATus.QUEStionable.RLIMit.PTRansition

Description

Sets the value of the positive transition filter of the Questionable Ripple Limit Status Register.

Variable

| | ***Value*** |
|---|---|
| Description | The value of the positive transition filter of the Questionable Ripple Limit Status 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 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 on page 412

SCPI.STATus.QUEStionable.RLIMit.NTRansition on page 413

Equivalent key

No equivalent key is available on the front panel.

# SCPI.SYSTem.BACKlight

Object type

Property

Syntax

SCPI.SYSTem.BACKlight = *Status*

*Status* = SCPI.SYSTem.BACKlight

Description

Turns ON/OFF the backlight of the LCD display.

When the backlight is OFF, you cannot read the information on the display.

Variable

| | ***Status*** |
|---|---|
| Description | ON/OFF of the backlight |
| Data type | Boolean type (Boolean) |
| Range | Select from the following.<br>•True or -1 Turns ON the backlight.<br>•False or 0 Turns OFF the backlight. |
| Preset value | True or -1 |

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.

## SCPI.SYSTem.BEEPer.COMPlete.IMMediate

Object type Method

Syntax SCPI.SYSTem.BEEPer.COMPlete.IMMediate

Description Generates a beep for the notification of the completion of the operation. (No read)

Examples
```
SCPI.SYSTem.BEEPer.COMPlete.IMMediate
```

Related objects SCPI.SYSTem.BEEPer.COMPlete.STATe on page 416

SCPI.SYSTem.BEEPer.WARNing.IMMediate on page 417

Equivalent key **[System]** - **Misc Setup** - **Beeper** - **Test Beep Complete**

## SCPI.SYSTem.BEEPer.COMPlete.STATe

Object type Property

Syntax SCPI.SYSTem.BEEPer.COMPlete.STATe = *Status*

*Status* = SCPI.SYSTem.BEEPer.COMPlete.STATe

Description Turns ON/OFF the beeper for the notification of the completion of the operation.

Variable

| | ***Status*** |
|---|---|
| Description | ON/OFF of the beeper for the notification of the completion of the operation |
| Data type | Boolean type (Boolean) |
| Range | Select from the following.<br>•True or -1 — Turns ON the beeper for the notification of the completion of the operation.<br>•False or 0 — Turns OFF the beeper for the notification of the completion of the operation. |
| Preset value | True or -1 |

Examples
```
Dim BeepComp As Boolean
SCPI.SYSTem.BEEPer.COMPlete.STATe = False
BeepComp = SCPI.SYSTem.BEEPer.COMPlete.STATe
```

Related objects SCPI.SYSTem.BEEPer.COMPlete.IMMediate on page 416

SCPI.SYSTem.BEEPer.WARNing.STATe on page 417

Equivalent key **[System]** - **Misc Setup** - **Beeper** - **Beep Complete**

## SCPI.SYSTem.BEEPer.WARNing.IMMediate

Object type Method

Syntax SCPI.SYSTem.BEEPer.WARNing.IMMediate

Description Generates a beep for the notification of warning/limit test result. (No read)

Examples
```
SCPI.SYSTem.BEEPer.WARNing.IMMediate
```

Related objects SCPI.SYSTem.BEEPer.WARNing.STATe on page 417

SCPI.SYSTem.BEEPer.COMPlete.IMMediate on page 416

Equivalent key **[System]** - **Misc Setup** - **Beeper** - **Test Beep Warning**

## SCPI.SYSTem.BEEPer.WARNing.STATe

Object type Property

Syntax SCPI.SYSTem.BEEPer.WARNing.STATe = *Status*

*Status* = SCPI.SYSTem.BEEPer.WARNing.STATe

Description Turns ON/OFF the beeper for the notification of warning/limit test result.

Variable

| | ***Status*** |
|---|---|
| Description | ON/OFF of the beeper for the notification of warning/limit test result |
| Data type | Boolean type (Boolean) |
| Range | Select from the following.<br>•True or -1 Turns ON the beeper for the notification of warning/limit test result.<br>•False or 0 Turns OFF the beeper for the notification of warning/limit test result. |
| Preset value | True or -1 |

Examples
```
Dim BeepWarn As Boolean
SCPI.SYSTem.BEEPer.WARNing.STATe = False
BeepWarn = SCPI.SYSTem.BEEPer.WARNing.STATe
```

Related objects SCPI.SYSTem.BEEPer.WARNing.IMMediate on page 417

SCPI.SYSTem.BEEPer.COMPlete.STATe on page 416

Equivalent key **[System]** - **Misc Setup** - **Beeper** - **Beep Warning**

# SCPI.SYSTem.DATE

Object type

Property

Syntax

SCPI.SYSTem.DATE = *Data*

*Data* = SCPI.SYSTem.DATE

Description

Sets the date of the clock built in the E5061A/E5062A.

Variable

| | ***Data*** |
|---|---|
| Description | Indicates 3-element array data (date of the built-in clock).<br>• *Data(0)* Sets year.<br>• *Data(1)* Sets month.<br>• *Data(2)* Sets day.<br>The index of the array starts from 0. |
| Data type | Variant type (Variant) |
| Range | • *Data(0)* 1980 to 2099<br>• *Data(1)* 1 to 12<br>• *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 on page 425

SCPI.DISPlay.CLOCk on page 233

Equivalent key

**[System] - Misc Setup - Clock Setup - Set Date and Time**

# SCPI.SYSTem.ERRor

Object type — Property

Syntax — *Err* = SCPI.SYSTem.ERRor

Description — Reads out the oldest error of the errors stored in the error queue of the E5061A/E5062A. The read-out error is deleted from the error queue. The size of the error queue is 100.

Executing the SCPI.IEEE4882.CLS object clears the errors stored in the error queue. (Read only)

**NOTE** — This object can not return an error that occurs by the manual operation or the SCPI command used in controlling the E5061A/E5062A from the external controller.

Variable

| | ***Err*** |
|---|---|
| Description | Indicates 2-element array data (for error).<br>• *Err(0)* Error number<br>• *Err(1)* Error message<br>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 on page 274

Equivalent key — No equivalent key is available on the front panel.

# SCPI.SYSTem.KLOCk.KBD

Object type

Property

Syntax

SCPI.SYSTem.KLOCk.KBD = *Status*

*Status* = SCPI.SYSTem.KLOCk.KBD

Description

Sets whether to lock the operation of the front panel (key and rotary knob) and keyboard.

Variable

| | ***Status*** |
|---|---|
| Description | ON/OFF of lock |
| Data type | Boolean type (Boolean) |
| Range | Select from the following.<br>•True or -1 Specifies lock.<br>•False or 0 Specifies unlock. |
| Preset value | False or 0 |

Examples

```
Dim FKLock As Boolean
SCPI.SYSTem.KLOCk.KBD = True
FKLock = SCPI.SYSTem.KLOCk.KBD
```

Related objects

SCPI.SYSTem.KLOCk.MOUSe on page 421

Equivalent key

**[System] - Misc Setup - Front Panel & Keyboard Lock**

## SCPI.SYSTem.KLOCk.MOUSe

Object type — Property

Syntax — SCPI.SYSTem.KLOCk.MOUSe = *Status*

*Status* = SCPI.SYSTem.KLOCk.MOUSe

Description — Sets whether to lock the operation of the mouse and touch screen.

Variable

| | ***Status*** |
|---|---|
| Description | ON/OFF of lock |
| Data type | Boolean type (Boolean) |
| Range | Select from the following.<br>•True or -1 Specifies lock.<br>•False or 0 Specifies unlock. |
| Preset value | False or 0 |

Examples

```
Dim MTLock As Boolean
SCPI.SYSTem.KLOCk.MOUSe = True
MTLock = SCPI.SYSTem.KLOCk.MOUSe
```

Related objects — SCPI.SYSTem.KLOCk.KBD on page 420

Equivalent key — **[System]** - **Key Lock** - **Mouse Lock**

## SCPI.SYSTem.POFF

Object type — Method

Syntax — SCPI.SYSTem.POFF

Description — Turns OFF the E5061A/E5062A. (No read)

Examples

```
SCPI.SYSTem.POFF
```

Equivalent key — Standby switch

# SCPI.SYSTem.PRESet

| | |
|---|---|
| Object type | Method |
| Syntax | SCPI.SYSTem.PRESet |
| Description | Presets the setting state of the E5061A/E5062A. There is the following difference from the setting state preset with the SCPI.IEEE4882.RST object. For details, see Appendix “List of Default Values” in the *E5061A/E5062A User’s Guide*. (No read)<br>• The continuous startup mode (see the SCPI.INITiate(Ch).CONTinuous object) of channel 1 is set to ON. |
| Examples | `SCPI.SYSTem.PRESet` |
| Related objects | SCPI.IEEE4882.RST on page 277<br>SCPI.SYSTem.UPReset on page 426 |
| Equivalent key | **[Preset]** - **OK** |

# SCPI.SYSTem.SECurity.LEVel

Object type — Property

Syntax — SCPI.SYSTem.SECurity.LEVel = *Param*

*Param* = SCPI.SYSTem.SECurity.LEVel

Description — Sets/Reads the security level.

Variable

| | ***Param*** |
|---|---|
| Description | The security level. |
| Data type | Character string type (String) |
| Range | Select from the following.<br>•"NON" Specifies OFF to the security level.<br>•"LOW" Specifies LOW level to the security level.<br>•"HIGH" Specifies HIGH level to the security level. |
| Preset value | "NON" |
| Note | When the setting is LOW, it is able to change to NON or HIGH. But when this setting is HIGH, it is not able to change NON or LOW.<br>The setting can be turned NON by executing the preset or recalling when the setting of frequency blank function is HIGH.<br>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**

# SCPI.SYSTem.SERVice

Object type — Property

Syntax — *Status* = SCPI.SYSTem.SERVice

Description — Reads out whether to be in the service mode. (Read only)

Variable

| | ***Status*** |
|---|---|
| Description | Whether to be in the service mode |
| Data type | Boolean type (Boolean) |
| Range | Select from the following.<br>•True or -1 In the service mode.<br>•False or 0 Not in the service mode. |

Examples

```
Dim SvMode As Boolean
SvMode = SCPI.SYSTem.SERVice
```

Equivalent key — Displayed on the instrument status bar (at the bottom of the LCD display).

# SCPI.SYSTem.TIME

Object type — Property

Syntax — SCPI.SYSTem.TIME = *Data*

*Data* = SCPI.SYSTem.TIME

Description — Sets the time of the clock built in the E5061A/E5062A.

Variable

| | ***Data*** |
|---|---|
| Description | Indicates 3-element array data (time of the built-in clock).<br>• *Data(0)* Sets hour (24-hour basis)<br>• *Data(1)* Sets minute.<br>• *Data(2)* Sets second.<br>The index of the array starts from 0. |
| Data type | Variant type (Variant) |
| Range | • *Data(0)* 0 to 23<br>• *Data(1)* 0 to 59<br>• *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 on page 418

SCPI.DISPlay.CLOCk on page 233

Equivalent key — **[System]** - **Misc Setup** - **Clock Setup** - **Set Date and Time**

# SCPI.SYSTem.UPReset

| | |
|---|---|
| Object type | Method |
| Syntax | SCPI.SYSTem.UPReset |
| Description | Executes the user-specified preset.<br>The command is executed regardless of the preset operation mode.<br>If the user-specified preset file (D:\UserPreset.sta) does not exist, a warning message is displayed, and “SCPI.SYSTem.PRESet” is executed. (No read) |
| Examples | `SCPI.SYSTem.UPReset` |
| Related objects | SCPI.IEEE4882.RST on page 277<br>SCPI.SYSTem.PRESet on page 422 |
| Equivalent key | **[Preset]** - **OK** |

# SCPI.TRIGger.SEQuence.IMMediate

Object type — Method

Syntax — SCPI.TRIGger.SEQuence.IMMediate

Description — Regardless of the setting of the trigger mode, generates a trigger immediately and executes a measurement.

There is the following difference from the trigger with the SCPI.TRIGger.SEQuence.SINGle object.

- The execution of the object finishes at the time of a trigger.

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 Section “Trigger System” in the *E5061A/E5062A Programmer’s Guide*. (No read)

Examples

```
SCPI.TRIGger.SEQuence.SOURce = "bus"
SCPI.INITiate(1).CONTinuous = True
SCPI.TRIGger.SEQuence.IMMediate
```

Related objects — SCPI.TRIGger.SEQuence.IMMediate on page 427

Equivalent key — No equivalent key is available on the front panel.

# SCPI.TRIGger.SEQuence.SINGle

Object type — Method

Syntax — SCPI.TRIGger.SEQuence.SINGle

Description — Regardless of the setting of the trigger mode, generates a trigger immediately and executes a measurement.

There is the following difference from the trigger with the SCPI.TRIGger.SEQuence.IMMediate object.

- 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 Section “Trigger System” in the *E5061A/E5062A Programmer’s Guide*. (No read)

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 on page 427

SCPI.IEEE4882.OPC on page 276

Equivalent key — No equivalent key is available on the front panel.

# SCPI.TRIGger.SEQuence.SOURce

**Object type** Property

**Syntax** SCPI.TRIGger.SEQuence.SOURce = *Param*

*Param* = SCPI.TRIGger.SEQuence.SOURce

**Description** Selects the trigger source from the following 4 types.

| | |
|---|---|
| Internal trigger | Uses the internal trigger to generate continuous triggers automatically. |
| External trigger | Generates a trigger when the trigger signal is inputted externally via the Ext Trig connector or the 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. |

When you change the trigger source during sweep, the sweep is aborted.

**Variable**

| | ***Param*** | |
|---|---|---|
| Description | Trigger source | |
| Data type | Character string type (String) | |
| Range | Select from 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**

# 8 Waveform Analysis Library

This chapter describes how to use the ripple analysis library and the procedures in the ripple analysis library.

# Ripple Analysis Library

By combining the COM objects provided for the E5061A/E5062A and the ripple analysis library, you can easily perform the ripple analysis of waveforms.

## Flow of Programming Using the Ripple Analysis Library

Below table shows the flow of program development using the ripple analysis library. First, set up the analysis range and peak definition to use the procedures for ripple analysis.

| |
|---|
| **STEP 1. Condition setting before using the ripple analysis library**<br>❑ Specifying the analysis range<br>❑ Setting the peak definition |
| **STEP 2. Using the ripple analysis library** |

## Condition Setting Before Using the 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. For more information on each object, see Chapter 7, “COM Object Reference.”.

- SCPI.CALCulate(Ch).SELected.FUNCtion.DOMain.STARt on page 148
- SCPI.CALCulate(Ch).SELected.FUNCtion.DOMain.STOP on page 150
- SCPI.CALCulate(Ch).SELected.FUNCtion.DOMain.STATe on page 149
- SCPI.CALCulate(Ch).SELected.FUNCtion.DOMain.COUPle on page 147

### Setting the Peak Definition

Use the following COM objects to set up the peak definition for ripple analysis. For more information on each object, see Chapter 7, “COM Object Reference.”.

- SCPI.CALCulate(Ch).SELected.FUNCtion.PEXCursion on page 152
- SCPI.CALCulate(Ch).SELected.FUNCtion.PPOLarity on page 154

## List of the 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.

For more information on the E5061A/E5062A ripple analysis library, refer to **Procedure Reference** on page 435.

| List of ripple analysis library |
|---|
| • Returns the maximum value of the difference between a positive peak and a negative peak.<br>**MaxPeakToPeak(Chan)** on page 443 |
| • Returns the maximum value of the difference between a positive peak and its right adjacent negative peak.<br>**MaxRightGap(Chan)** on page 444 |
| • Returns the maximum value of the difference between a positive peak and its left adjacent negative peak.<br>**MaxLeftGap(Chan)** on page 442 |
| • Returns the maximum value of the difference between a positive peak and its adjacent negative peak.<br>**MaxGap(Chan)** on page 441 |
| • Returns the maximum value of the vertical distance between a line segment connecting 2 adjacent positive peaks and the negative peak between them.<br>**MaxEnvelopeGap(Chan)** on page 440 |
| • Returns the mean value of the differences between a negative peak and its right and left adjacent positive peaks.<br>**GapMean(Chan)** on page 439 |
| • Returns the maximum value of the total of the differences between a negative peak and its right and left adjacent positive peaks.<br>**MaxRippleValue(Chan)** on page 446 |
| • 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.<br>**MaxRipplePoint(Chan,Stim)** on page 445 |
| • 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.<br>**Pole(Chan,D,LeftStim,LeftValue,RightStim,RightValue)** on page 447 |
| • Returns the difference between the positive peak detected first when searched from the left edge toward the right edge and its right adjacent negative peak.<br>**FirstRightGap(Chan)** on page 437 |

| List of ripple analysis library |
|---|
| • Returns the difference between the positive peak detected first when searched from the right edge toward the left edge and its left adjacent negative peak.<br><br>**FirstLeftGap(Chan)** on page 435 |
| • Returns the difference of the stimulus value between the positive peak detected first when searched from the left edge toward the right edge and its right adjacent negative peak.<br><br>**FirstRightInterval(Chan)** on page 438 |
| • Returns the difference of the stimulus value between the positive peak detected first when searched from the left edge toward the right edge and its left adjacent negative peak.<br><br>**FirstLeftInterval(Chan)** on page 436 |

## Simple Use Example

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 for the peak search to 1.5 and both of positive peak and negative peak, respectively.
3. Sets the analysis range for channel 1 to 935 MHz to 960 MHz.
4. For channel 1, substitutes the return value from the MaxPeakToPeak function (procedure) in the ripple analysis library to the Val variable.

# Procedure Reference

This section describes the procedures in the ripple analysis library provided by the E5061A/E5062A in alphabetical order.

## FirstLeftGap(*Chan*)

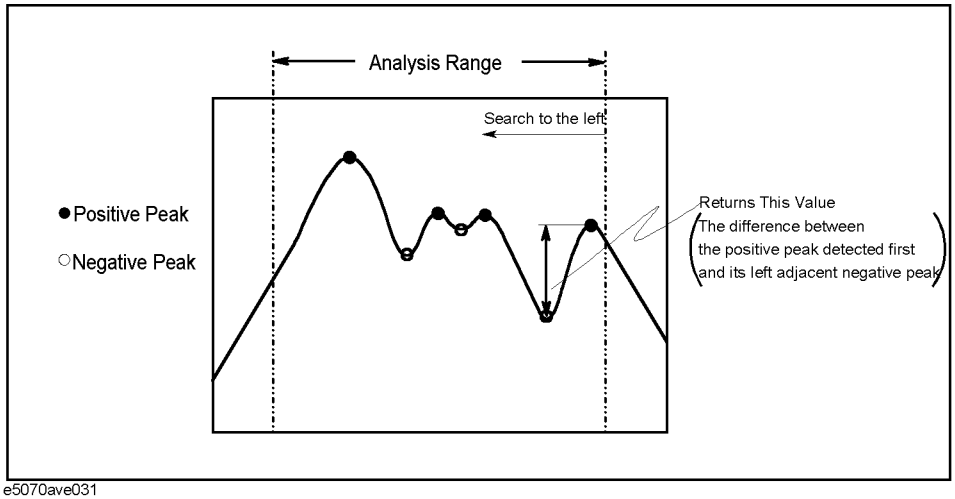
**Syntax**

*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.

**Figure 8-1 FirstLeftGap**




e5070ave031

**Variable**

| | *Chan* |
|---|---|
| Description | Specifies the channel number. |
| Data type | Integer type (Integer) |
| Range | 1 to 4 |
| Note | If the specified variable is out of the allowable setup range, an error occurs when executed. |

**Return value**

| | *Value* |
|---|---|
| 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. |
| 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*)

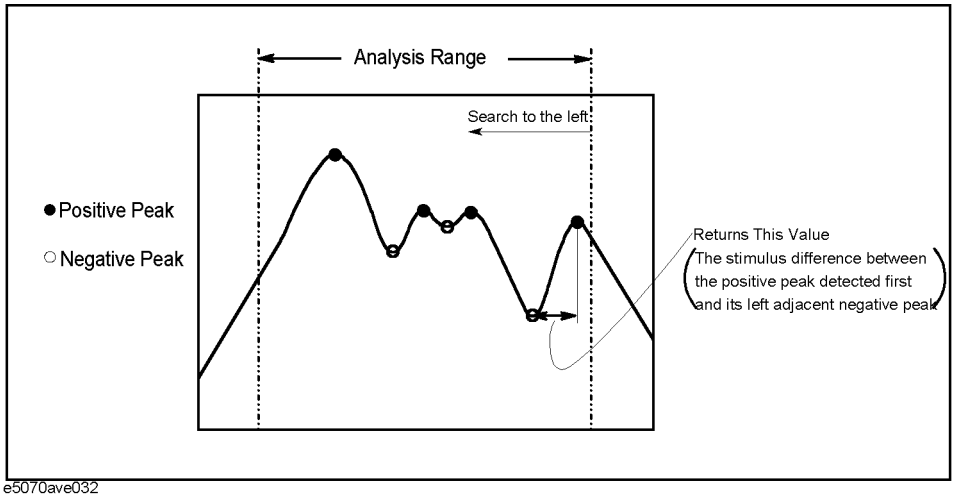
**Syntax**

*Value* = FirstLeftInterval(*Chan*)

**Description**

Returns the stimulus difference between the positive peak detected first when searched from the right edge toward the left edge within the analysis range and its left adjacent negative peak.

**Figure 8-2 FirstLeftInterval**




e5070ave032

**Variable**

| | ***Chan*** |
|---|---|
| Description | Specifies the channel number. |
| Data type | Integer type (Integer) |
| Range | 1 to 4 |
| Note | If the specified variable is out of the allowable setup range, an error occurs when executed. |

**Return value**

| | ***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*)

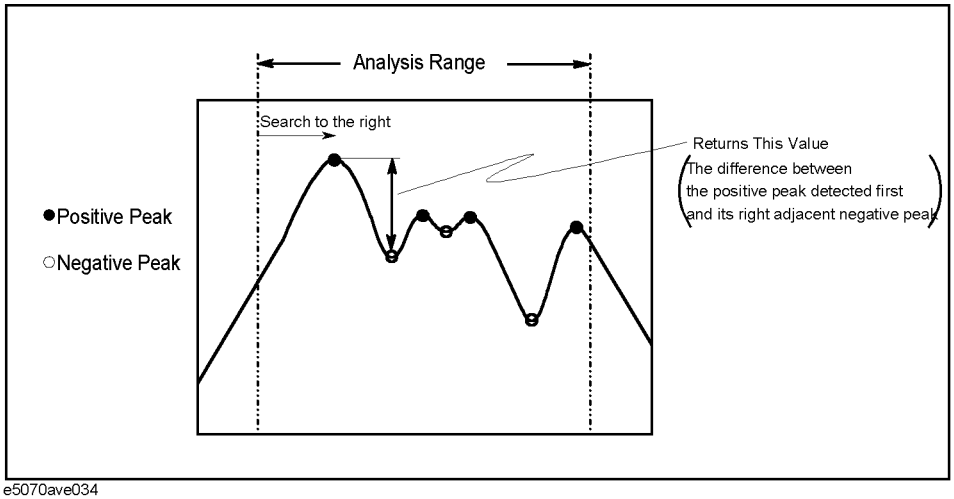
**Syntax**

*Value* = FirstRightGap(*Chan*)

**Description**

Returns the response difference between the positive peak detected first when searched from the left edge toward the right edge within the analysis range and its right adjacent negative peak.

**Figure 8-3 FirstRightGap**




e5070ave034

**Variable**

| | ***Chan*** |
|---|---|
| Description | Specifies the channel number. |
| Data type | Integer type (Integer) |
| Range | 1 to 4 |
| Note | If the specified variable is out of the allowable setup range, an error occurs when executed. |

**Return value**

| | ***Value*** |
|---|---|
| Description | Returns the response 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 = FirstRightGap(1)
MsgBox "First Right Gap =" & Value
```

## FirstRightInterval(*Chan*)

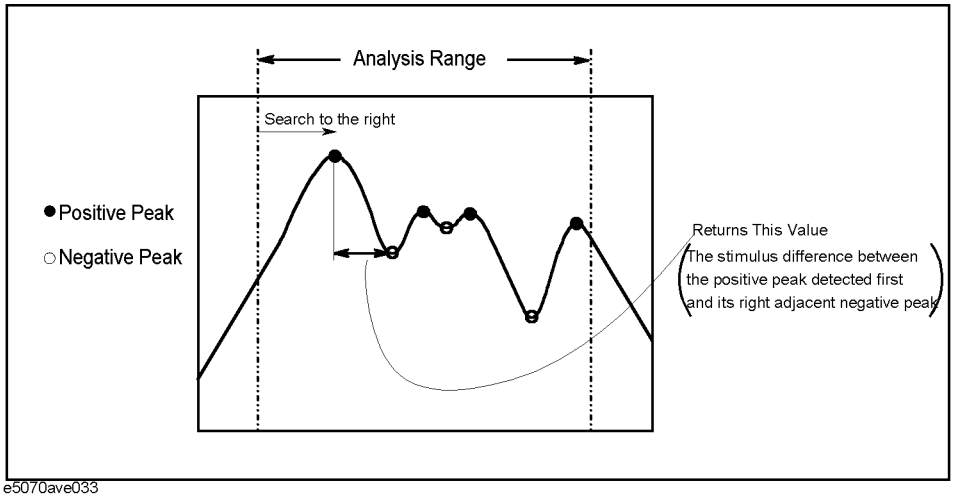
**Syntax**

*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.

**Figure 8-4 FirstRightInterval**




e5070ave033

**Variable**

| | ***Chan*** |
|---|---|
| Description | Specifies the channel number. |
| Data type | Integer type (Integer) |
| Range | 1 to 4 |
| Note | If the specified variable is out of the allowable setup range, an error occurs when executed. |

**Return value**

| | ***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*)

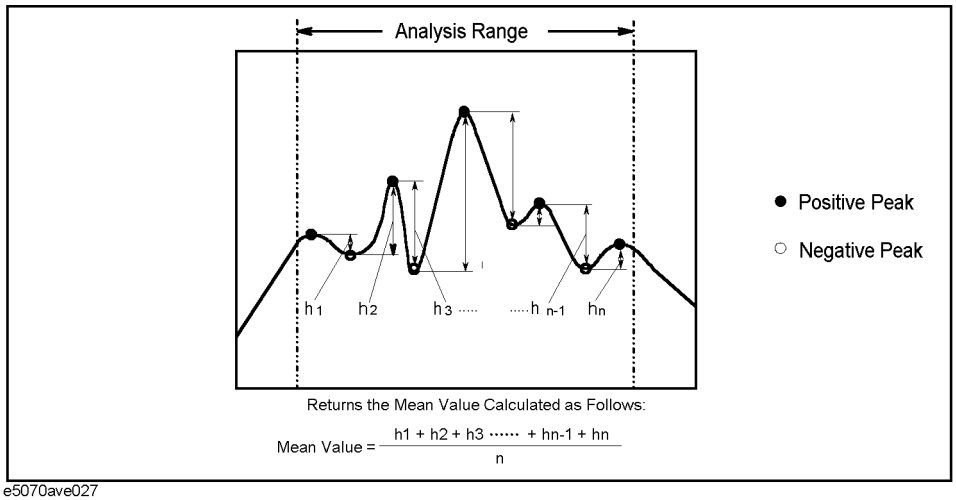
**Syntax**

*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.

**Figure 8-5** **GapMean**




e5070ave027

**Variable**

| | ***Chan*** |
|---|---|
| Description | Specifies the channel number. |
| Data type | Integer type (Integer) |
| Range | 1 to 4 |
| Note | If the specified variable is out of the allowable setup range, an error occurs when executed. |

**Return value**

| | ***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*)

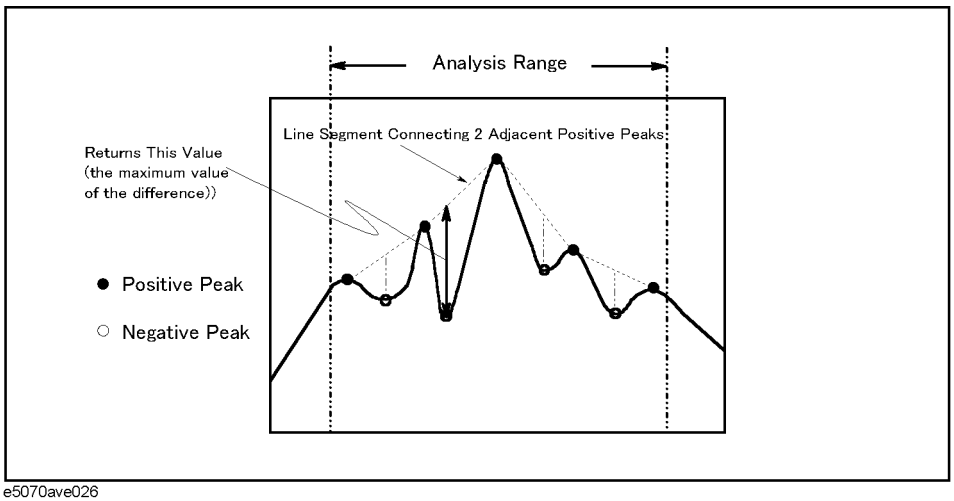
**Syntax**

*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.

**Figure 8-6 MaxEnvelopeGap**




e5070ave026

**Variable**

| | ***Chan*** |
|---|---|
| Description | Specifies the channel number. |
| Data type | Integer type (Integer) |
| Range | 1 to 4 |
| Note | If the specified variable is out of the allowable setup range, an error occurs when executed. |

**Return value**

| | ***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*)

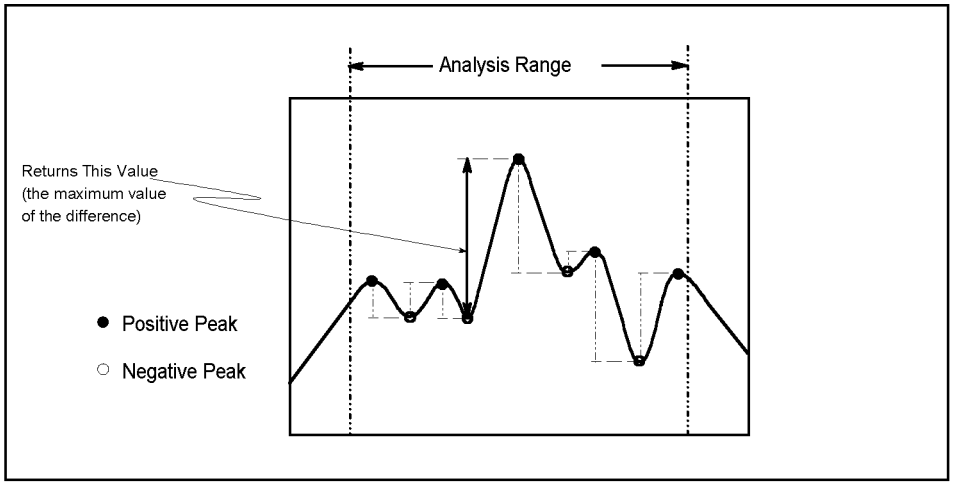
**Syntax**

*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.

**Figure 8-7** **MaxGap**




e5070ave025

**Variable**

| | ***Chan*** |
|---|---|
| Description | Specifies the channel number. |
| Data type | Integer type (Integer) |
| Range | 1 to 4 |
| Note | If the specified variable is out of the allowable setup range, an error occurs when executed. |

**Return value**

| | ***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*)

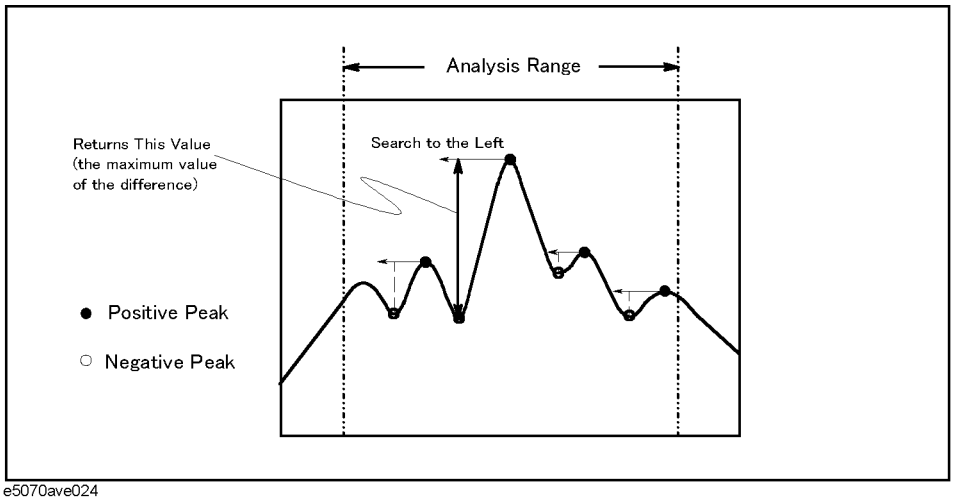
**Syntax**

*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.

**Figure 8-8 MaxLeftGap**




e5070ave024

**Variable**

| | ***Chan*** |
|---|---|
| Description | Specifies the channel number. |
| Data type | Integer type (Integer) |
| Range | 1 to 4 |
| Note | If the specified variable is out of the allowable setup range, an error occurs when executed. |

**Return value**

| | ***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*)

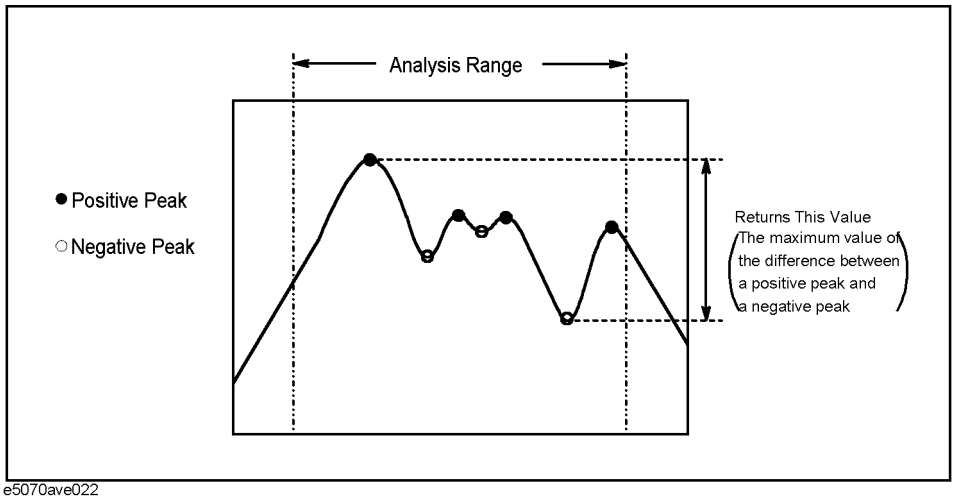
**Syntax**

*Value* = MaxPeakToPeak(*Chan*)

**Description**

Returns the maximum value of the response differences between the positive peaks and the negative peaks within the analysis range.

**Figure 8-9** **MaxPeakToPeak**




e5070ave022

**Variable**

| | ***Chan*** |
|---|---|
| Description | Specifies the channel number. |
| Data type | Integer type (Integer) |
| Range | 1 to 4 |
| Note | If the specified variable is out of the allowable setup range, an error occurs when executed. |

**Return value**

| | ***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*)

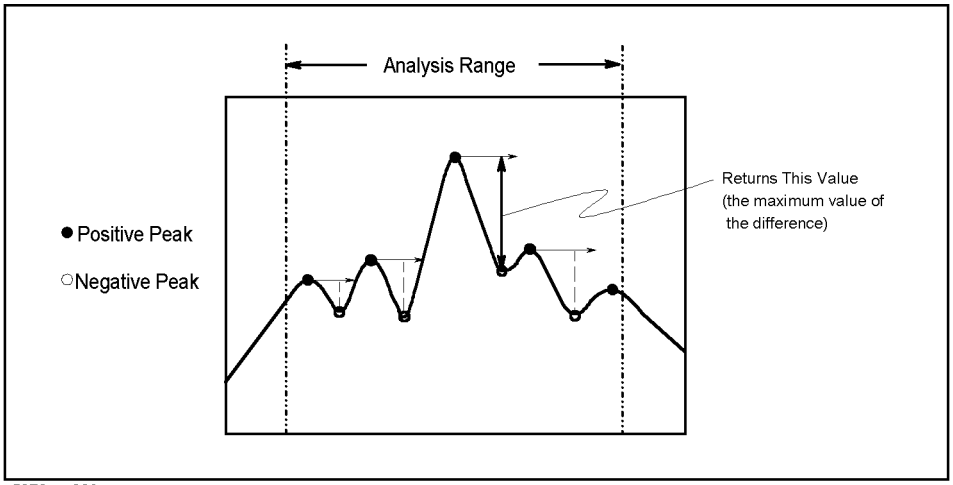
**Syntax**

*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.

**Figure 8-10 MaxRightGap**




e5070ave023

**Variable**

| | ***Chan*** |
|---|---|
| Description | Specifies the channel number. |
| Data type | Integer type (Integer) |
| Range | 1 to 4 |
| Note | If the specified variable is out of the allowable setup range, an error occurs when executed. |

**Return value**

| | ***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*)

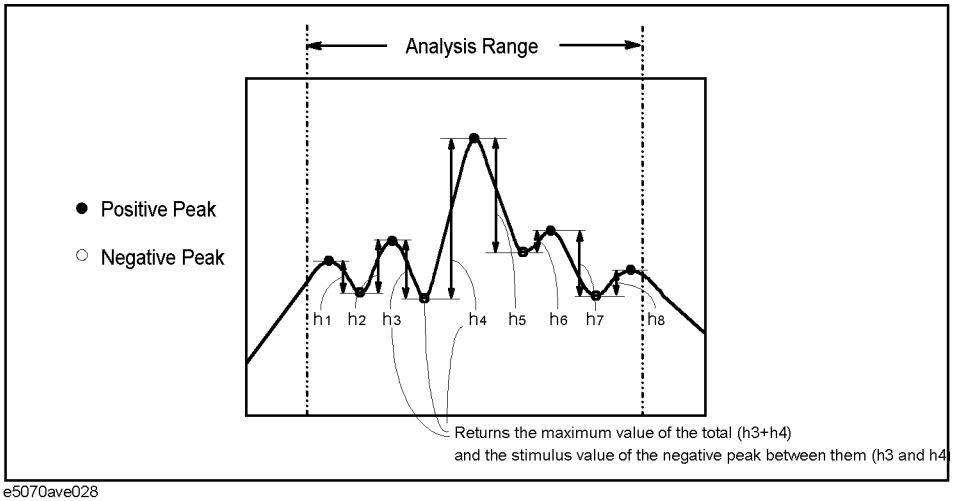
**Syntax**

*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.

**Figure 8-11 MaxRipplePoint**




e5070ave028

**Variable**

| | ***Chan*** |
|---|---|
| Description | Specifies the channel number. |
| Data type | Integer type (Integer) |
| Range | 1 to 4 |
| Note | If the specified variable is out of the allowable setup range, an error occurs when executed. |

**Return value**

| | ***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. |

| | ***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*)

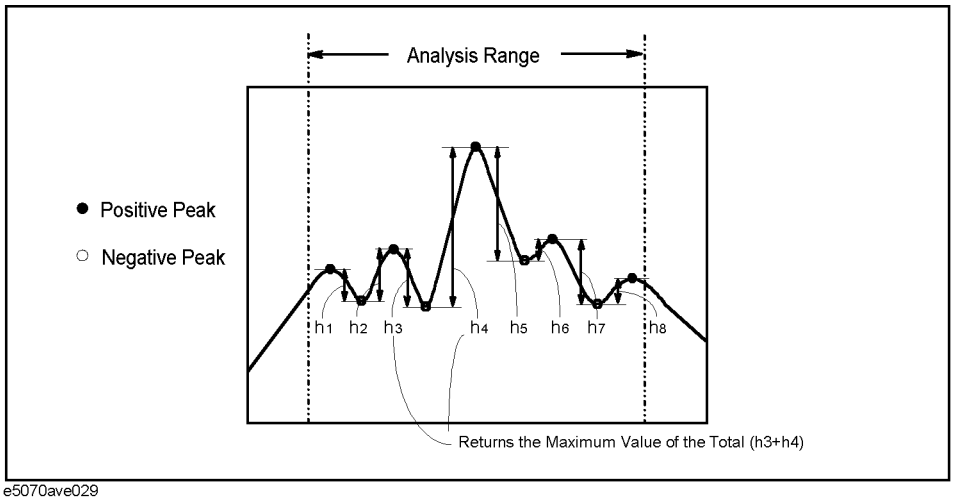
**Syntax**

*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.

**Figure 8-12** **MaxRippleValue**




e5070ave029

**Variable**

| | ***Chan*** |
|---|---|
| Description | Specifies the channel number. |
| Data type | Integer type (Integer) |
| Range | 1 to 4 |
| Note | If the specified variable is out of the allowable setup range, an error occurs when executed. |

**Return value**

| | ***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*)

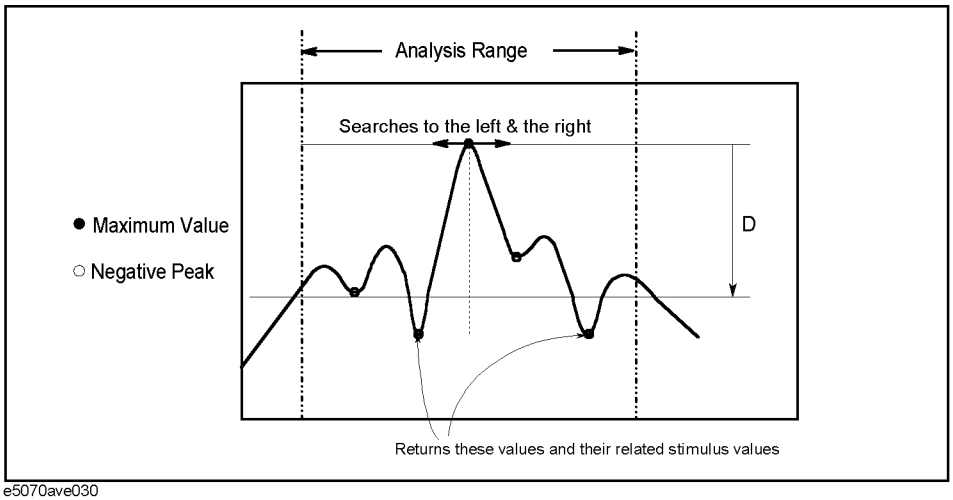
**Syntax**

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, returns the response value (*LeftValue*) and stimulus value (*LeftStimulus*) of the negative peak first detected when searched to the left from the maximum value of the positive peaks, and the response value (*RightValue*) and stimulus value (*RightStimulus*) of the negative peak first detected when searched to the right from the maximum value of the positive peaks.

**Figure 8-13 Pole**




e5070ave030

**Variable**

| | ***Chan*** |
|---|---|
| Description | Specifies the channel number. |
| Data type | Integer type (Integer) |
| Range | 1 to 4 |
| Note | If the specified variable is out of the allowable setup range, an error occurs when executed. |

| | ***D*** |
|---|---|
| Description | Specifies the difference from the maximum value. |
| Data type | Double precision floating point type (Double) |

## Return value (arguments)

| | *LeftStim* |
|---|---|
| Description | Returns the stimulus value of the negative peak first detected 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. |

| | *LeftValue* |
|---|---|
| Description | Returns the response value of the negative peak first detected 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. |

| | *RightStim* |
|---|---|
| Description | Returns the stimulus value of the negative peak first detected 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. |

| | *RightValue* |
|---|---|
| Description | Returns the response value of the negative peak first detected 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
```

# 9 Complex Operation Library

This chapter describes the 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)** on page 454 | Sets a complex number.<br>(Specify a real part and an imaginary part.) |
| **ComplexPolar(x,y)** on page 454 | Sets a complex number.<br>(Specify an absolute value and a phase angle.) |
| **ComplexSetArray(x)** on page 455 | Converts a variant type or double floating point type array to a complex type array. |
| **ComplexAdd(x,y)** on page 451 | Returns the result of the addition. |
| **ComplexSub(x,y)** on page 456 | Returns the result of the subtraction. |
| **ComplexMul(x,y)** on page 453 | Returns the result of the multiplication. |
| **ComplexDiv(x,y)** on page 452 | Returns the result of the division. |
| **ComplexAbs(x)** on page 451 | Returns the absolute value. |
| **ComplexArg(x)** on page 451 | Returns the phase angle. |
| **ComplexNorm(x)** on page 454 | Returns the square of the absolute value. |
| **ComplexConj(x)** on page 452 | Returns the conjugate complex number. |
| **ComplexCos(x)** on page 452 | Returns the cosine. |
| **ComplexCosh(x)** on page 452 | Returns the hyperbolic cosine. |
| **ComplexSin(x)** on page 455 | Returns the sine. |
| **ComplexSinh(x)** on page 455 | Returns the hyperbolic sine. |
| **ComplexExp(x)** on page 453 | Returns $e^x$. |
| **ComplexLog(x)** on page 453 | Returns the natural logarithm. |
| **ComplexLog10(x)** on page 453 | Returns the common logarithm. |
| **ComplexSqrt(x)** on page 456 | Returns the square root. |

# Procedure Reference

This section describes the procedures in the complex operation library in alphabetical order.

## ComplexAbs(*x*)

**Syntax** *Result* = ComplexAbs(*x*)

**Description** Returns the absolute value of a 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 of a 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 a 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 a 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 a 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 a 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 a 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 $e^x$.

**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 a 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 a 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 a 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 a 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*. Specify 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*. Specify 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 a 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 a 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 a 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 ( $\sqrt{x}$ ) of a 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 of a 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)
```

# Sample Program

```
                    :
                    :

Dim Dmy As Long
Dim s21_raw As Variant
Dim s11_raw As Variant
Dim s21_Comp As Complex
Dim s11_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 = "s11"
SCPI.SENSe(1).SWEep.POINts = 201

                    :
                    :
                    :

SCPI.TRIGger.SEQuence.Source = "bus"
SCPI.TRIGger.SEQuence.SINGle
Dmy = SCPI.IEEE4882.OPC

''''  Get corrected data array
SCPI.CALCulate(1).PARameter(1).SELect
s21_raw = SCPI.CALCulate(1).SELected.DATA.SDATa
SCPI.CALCulate(1).PARameter(2).SELect
s11_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))
  s11_Comp = ComplexSet(s11_raw(2 * i), s11_raw(2 * i + 1))

  ''''  Calculate the ratio of s11 and S21
  ''''  s11/S21
  trAce_ratio_comp = ComplexDiv(s11_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 "s11/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 "s11/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

                    :
                    :
```

# A Manual Changes

This appendix contains the information required to adapt this manual to versions or configurations of the E5061A/E5062A manufactured earlier than the current printing date of this manual.

# Manual Changes

To adapt this manual to your E5061A/E5062A, refer to Table A-1 and Table A-2.

**Table A-1 Manual Changes by Serial Number**

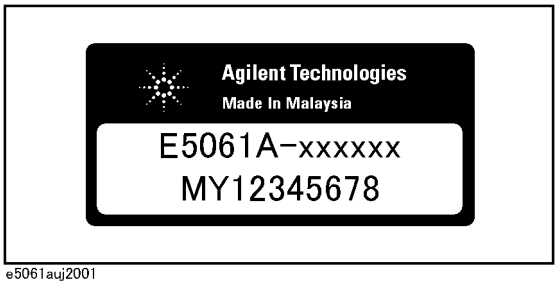
| Serial Prefix or Number | Make Manual Changes |
|---|---|
| | |

**Table A-2 Manual Changes by Firmware Version**

| Version | Make Manual Changes |
|---|---|
| A.02.00 or later | Change 1 |
| A.02.10 or later | Change 2 |
| A.03.00 or later | Change 3 |

Agilent Technologies uses a two-part, ten-character serial number that is stamped on the serial number plate (Figure A-1).

**Figure A-1 Example of Serial Number Plate**



## Change 3

The firmware revision A.02.10 or below does not support the following functions. Please disregard the descriptions of these functions in this manual.

❑ Compliant with LXI (Lan eXtensions for Instrumentation) standard Class C.*1

## Change 2

The firmware revision A.02.00 or below does not support the following functions. Please disregard the descriptions of these functions in this manual.

❑ Offset limit line function

❑ Ripple test function

❑ Bandwidth test function

The firmware revision A.02.00 and below does not support the following COM objects. Please delete their descriptions in this manual.

- SCPI.CALCulate(Ch).SELected.BLIMit.DB on page 129
- SCPI.CALCulate(Ch).SELected.BLIMit.DISPlay.MARKer on page 130
- SCPI.CALCulate(Ch).SELected.BLIMit.DISPlay.VALue on page 131
- SCPI.CALCulate(Ch).SELected.BLIMit.FAIL on page 132
- SCPI.CALCulate(Ch).SELected.BLIMit.MAXimum on page 133
- SCPI.CALCulate(Ch).SELected.BLIMit.MINimum on page 134
- SCPI.CALCulate(Ch).SELected.BLIMit.REPort.DATA on page 135
- SCPI.CALCulate(Ch).SELected.BLIMit.STATe on page 136
- SCPI.CALCulate(Ch).SELected.LIMit.OFFSet.AMPLitude on page 163
- SCPI.CALCulate(Ch).SELected.LIMit.OFFSet.MARKer on page 164
- SCPI.CALCulate(Ch).SELected.LIMit.OFFSet.STIMulus on page 165
- SCPI.CALCulate(Ch).SELected.LIMit.REPort.ALL on page 166
- SCPI.CALCulate(Ch).SELected.RLIMit.DATA on page 211
- SCPI.CALCulate(Ch).SELected.RLIMit.DISPlay.LINE on page 213
- SCPI.CALCulate(Ch).SELected.RLIMit.DISPlay.SELect on page 214
- SCPI.CALCulate(Ch).SELected.RLIMit.DISPlay.VALue on page 215
- SCPI.CALCulate(Ch).SELected.RLIMit.FAIL on page 216
- SCPI.CALCulate(Ch).SELected.RLIMit.REPort.DATA on page 217
- SCPI.CALCulate(Ch).SELected.RLIMit.STATe on page 218
- SCPI.MMEMory.LOAD.RLIMit on page 288
- SCPI.MMEMory.STORe.RLIMit on page 297
- SCPI.STATus.QUEStionable.BLIMit.CHANnel(Ch).CONDition on page 385

*1. This function is available when the volume label on the hard disk is AL300 or higher.

- SCPI.STATus.QUEStionable.BLIMit.CHANnel(Ch).ENABle on page 386
- SCPI.STATus.QUEStionable.BLIMit.CHANnel(Ch).EVENt on page 387
- SCPI.STATus.QUEStionable.BLIMit.CHANnel(Ch).NTRansition on page 388
- SCPI.STATus.QUEStionable.BLIMit.CHANnel(Ch).PTRansition on page 389
- SCPI.STATus.QUEStionable.BLIMit.CONDition on page 390
- SCPI.STATus.QUEStionable.BLIMit.ENABle on page 391
- SCPI.STATus.QUEStionable.BLIMit.EVENt on page 391
- SCPI.STATus.QUEStionable.BLIMit.NTRansition on page 392
- SCPI.STATus.QUEStionable.BLIMit.PTRansition on page 393
- SCPI.STATus.QUEStionable.RLIMit.CHANnel(Ch).CONDition on page 406
- SCPI.STATus.QUEStionable.RLIMit.CHANnel(Ch).ENABle on page 407
- SCPI.STATus.QUEStionable.RLIMit.CHANnel(Ch).EVENt on page 408
- SCPI.STATus.QUEStionable.RLIMit.CHANnel(Ch).NTRansition on page 409
- SCPI.STATus.QUEStionable.RLIMit.CHANnel(Ch).PTRansition on page 410
- SCPI.STATus.QUEStionable.RLIMit.CONDition on page 411
- SCPI.STATus.QUEStionable.RLIMit.ENABle on page 412
- SCPI.STATus.QUEStionable.RLIMit.EVENt on page 412
- SCPI.STATus.QUEStionable.RLIMit.NTRansition on page 413
- SCPI.STATus.QUEStionable.RLIMit.PTRansition on page 414


## Change 1

The following functions are newly integrated into the firmware version A.02.00 onward. They are not supported by firmware version A.01.0x.

- ❑ User preset function.
- ❑ Notch search function
- ❑ Function to all the marker value are displayed.
- ❑ Function to change display position where marker value are displayed.
- ❑ Function to align the marker value.
- ❑ Display the overlap limit lines.
- ❑ Reference tracking function.
- ❑ Function to change display value where Y axis are displayed.
- ❑ Frequency information appearing as asterisks.
- ❑ User recovery function.*1
- ❑ Remote control using HTTP. *1

*1. This function is available when the volume label of the hard disk is AL200 or higher.

The firmware version A.01.0x does not support the following COM objects. Please delete their descriptions in this manual.

- SCPI.CALCulate(Ch).SELected.LIMit.DISPlay.CLIP on page 160
- SCPI.CALCulate(Ch).SELected.MARKer(Mk).NOTCh. DATA on page 200
- SCPI.CALCulate(Ch).SELected.MARKer.NOTCh.STATe on page 201
- SCPI.CALCulate(Ch).SELected.MARKer(Mk).NOTCh. THReshold on page 202
- SCPI.DISPlay.WINDow(Ch).ANNotation.MARKer.ALIGn.STATe on page 251
- SCPI.DISPlay.WINDow(Ch).ANNotation.MARKer.SINGle.STATe on page 252
- SCPI.DISPlay.WINDow(Ch).TRACe(Tr).ANNotation. MARKer. POSition.X on page 258
- SCPI.DISPlay.WINDow(Ch).TRACe(Tr).ANNotation. MARKer. POSition.Y on page 259
- SCPI.DISPlay.WINDow(Ch).TRACe(Tr).ANNotation. YAXis.MODE on page 260
- SCPI.DISPlay.WINDow(Ch).TRACe(Tr).Y. TRACk. FREQuency on page 266
- SCPI.DISPlay.WINDow(Ch).TRACe(Tr).Y. TRACk. MODE on page 267
- SCPI.SYSTem.SECurity.LEVel on page 423
- SCPI.SYSTem.UPReset on page 426

# Index


**A**

analysis
  ripple analysis library, 432
Application objects, 100
array data
  storing array data, 71
Array function, 40
automatic Loading
  Loading a project at power-on, 46

**B**

break point, 55

**C**

Calib_Solt
  procedure for full n-port calibration, 90
calibration
  Calib_Solt procedure, 90
class module, 33
  inserting class module, 36
Clear
  Error queue, 274
  Status register, 274
Clear Echo button, 59
Close Editor button, 32
code window, 38
coding
  auto-complete function, 41
  coding a VBA program, 38
COM OBJECT
  NAME, 116
  OnPress Event, 121
  Parse, 117
  Prompt, 118
  SCPI..CALCulate(Ch).SELected.MARKer(Mk).FUNCtion .PEXCursion, 181
  SCPI.ABORt, 125
  SCPI.CALCulate(Ch).PARameter(Tr).COUNt, 126
  SCPI.CALCulate(Ch).PARameter(Tr).DEFine, 127
  SCPI.CALCulate(Ch).PARameter(Tr).SELect, 128
  SCPI.CALCulate(Ch).SELected.BLIMit.DB, 129
  SCPI.CALCulate(Ch).SELected.BLIMit.DISPlay.MARKe r, 130
  SCPI.CALCulate(Ch).SELected.BLIMit.DISPlay.VALue, 131
  SCPI.CALCulate(Ch).SELected.BLIMit.FAIL, 132
  SCPI.CALCulate(Ch).SELected.BLIMit.MAXimum, 133
  SCPI.CALCulate(Ch).SELected.BLIMit.MINimum, 134
  SCPI.CALCulate(Ch).SELected.BLIMit.REPort.DATA, 135
  SCPI.CALCulate(Ch).SELected.BLIMit.STATe, 136
  SCPI.CALCulate(Ch).SELected.CONVersion.FUNCtion, 137
  SCPI.CALCulate(Ch).SELected.CONVersion.STATe, 138
  SCPI.CALCulate(Ch).SELected.CORRection.EDELay.TI ME, 139
  SCPI.CALCulate(Ch).SELected.CORRection.OFFSet.PH ASe, 140
  SCPI.CALCulate(Ch).SELected.DATA.FDATa, 141
  SCPI.CALCulate(Ch).SELected.DATA.FMEMory, 142
  SCPI.CALCulate(Ch).SELected.DATA.SDATa, 143
  SCPI.CALCulate(Ch).SELected.DATA.SMEMory, 144
  SCPI.CALCulate(Ch).SELected.FLATness.STATe, 195
  SCPI.CALCulate(Ch).SELected.FORMat, 145
  SCPI.CALCulate(Ch).SELected.FUNCtion.DATA, 146
  SCPI.CALCulate(Ch).SELected.FUNCtion.DOMain.COU Ple, 147
  SCPI.CALCulate(Ch).SELected.FUNCtion.DOMain.STA Rt, 148
  SCPI.CALCulate(Ch).SELected.FUNCtion.DOMain.STA Te, 149
  SCPI.CALCulate(Ch).SELected.FUNCtion.DOMain.STO P, 150
  SCPI.CALCulate(Ch).SELected.FUNCtion.EXECute, 151
  SCPI.CALCulate(Ch).SELected.FUNCtion.PEXCursion, 152
  SCPI.CALCulate(Ch).SELected.FUNCtion.POINts, 153
  SCPI.CALCulate(Ch).SELected.FUNCtion.PPOLarity, 154
  SCPI.CALCulate(Ch).SELected.FUNCtion.TARGet, 155
  SCPI.CALCulate(Ch).SELected.FUNCtion.TTRansition, 156
  SCPI.CALCulate(Ch).SELected.FUNCtion.TYPE, 157
  SCPI.CALCulate(Ch).SELected.LIMit.DATA, 158
  SCPI.CALCulate(Ch).SELected.LIMit.DISPlay.CLIP, 160
  SCPI.CALCulate(Ch).SELected.LIMit.DISPlay.STATe, 161
  SCPI.CALCulate(Ch).SELected.LIMit.FAIL, 162
  SCPI.CALCulate(Ch).SELected.LIMit.OFFSet.AMPLitud e, 163
  SCPI.CALCulate(Ch).SELected.LIMit.OFFSet.MARKer, 164
  SCPI.CALCulate(Ch).SELected.LIMit.OFFSet.STIMulus, 165
  SCPI.CALCulate(Ch).SELected.LIMit.REPort.ALL, 166
  SCPI.CALCulate(Ch).SELected.LIMit.REPort.DATA, 167
  SCPI.CALCulate(Ch).SELected.LIMit.REPort.POINts, 168
  SCPI.CALCulate(Ch).SELected.LIMit.STATe, 169
  SCPI.CALCulate(Ch).SELected.MARKer(Mk).ACTivate, 170
  SCPI.CALCulate(Ch).SELected.MARKer(Mk).BWIDth.D ATA, 171
  SCPI.CALCulate(Ch).SELected.MARKer(Mk).BWIDth.S TATe, 172
  SCPI.CALCulate(Ch).SELected.MARKer(Mk).BWIDth.T HReshhold, 173
  SCPI.CALCulate(Ch).SELected.MARKer(Mk).COUPle, 174
  SCPI.CALCulate(Ch).SELected.MARKer(Mk).DISCrete, 175
  SCPI.CALCulate(Ch).SELected.MARKer(Mk).FUNCtion. DOMain.COUPle, 176

# Index


SCPI.CALCulate(Ch).SELected.MARKer(Mk).FUNCtion.DOMain.STARt, 177
SCPI.CALCulate(Ch).SELected.MARKer(Mk).FUNCtion.DOMain.STATe, 178
SCPI.CALCulate(Ch).SELected.MARKer(Mk).FUNCtion.DOMain.STOP, 179
SCPI.CALCulate(Ch).SELected.MARKer(Mk).FUNCtion.EXECute, 180
SCPI.CALCulate(Ch).SELected.MARKer(Mk).FUNCtion.MULTi.PPOLarity, 182
SCPI.CALCulate(Ch).SELected.MARKer(Mk).FUNCtion.MULTi.TARGet, 183
SCPI.CALCulate(Ch).SELected.MARKer(Mk).FUNCtion.MULTi.TRACking, 184
SCPI.CALCulate(Ch).SELected.MARKer(Mk).FUNCtion.MULTi.TTRansition, 185
SCPI.CALCulate(Ch).SELected.MARKer(Mk).FUNCtion.MULTi.TYPE, 186
SCPI.CALCulate(Ch).SELected.MARKer(Mk).FUNCtion.PPOLarity, 188
SCPI.CALCulate(Ch).SELected.MARKer(Mk).FUNCtion.TARGet, 189
SCPI.CALCulate(Ch).SELected.MARKer(Mk).FUNCtion.TRACking, 190
SCPI.CALCulate(Ch).SELected.MARKer(Mk).FUNCtion.TTRansition, 191
SCPI.CALCulate(Ch).SELected.MARKer(Mk).FUNCtion.TYPE, 192
SCPI.CALCulate(Ch).SELected.MARKer(Mk).NOTCh.DATA, 200
SCPI.CALCulate(Ch).SELected.MARKer(Mk).NOTCh.THReshhold, 202
SCPI.CALCulate(Ch).SELected.MARKer(Mk).REFerence.STATe, 203
SCPI.CALCulate(Ch).SELected.MARKer(Mk).SET, 204
SCPI.CALCulate(Ch).SELected.MARKer(Mk).STATe, 205
SCPI.CALCulate(Ch).SELected.MARKer(Mk).X, 206
SCPI.CALCulate(Ch).SELected.MARKer(Mk).Y, 207
SCPI.CALCulate(Ch).SELected.MARKer.MATH.FLATness.DATA, 194
SCPI.CALCulate(Ch).SELected.MARKer.MATH.FSTatistics.DATA, 196
SCPI.CALCulate(Ch).SELected.MARKer.MATH.FSTatistics.STATe, 197
SCPI.CALCulate(Ch).SELected.MARKer.MATH.STATistics.DATA, 198
SCPI.CALCulate(Ch).SELected.MARKer.MATH.STATistics.STATe, 199
SCPI.CALCulate(Ch).SELected.MARKer.NOTCÇà.STATe, 201
SCPI.CALCulate(Ch).SELected.MATH.FUNCtion, 208
SCPI.CALCulate(Ch).SELected.MATH.MEMorize, 209
SCPI.CALCulate(Ch).SELected.MSTatistics.DATA, 209
SCPI.CALCulate(Ch).SELected.MSTatistics.STATe, 210
SCPI.CALCulate(Ch).SELected.RLIMit.DATA, 211
SCPI.CALCulate(Ch).SELected.RLIMit.DISPlay.LINE, 213
SCPI.CALCulate(Ch).SELected.RLIMit.DISPlay.SELect, 214
SCPI.CALCulate(Ch).SELected.RLIMit.DISPlay.VALue, 215
SCPI.CALCulate(Ch).SELected.RLIMit.FAIL, 216
SCPI.CALCulate(Ch).SELected.RLIMit.REPort.DATA, 217
SCPI.CALCulate(Ch).SELected.RLIMit.STATe, 218
SCPI.CALCulate(Ch).SELected.SMOothing.APERture, 219
SCPI.CALCulate(Ch).SELected.SMOothing.STATe, 220
SCPI.CALCulate(Cn).SELected.MARKer(Mk).FUNCtion.PEXCursion, 187
SCPI.CONTrol.HANDler.A.DATA, 221
SCPI.CONTrol.HANDler.B.DATA, 222
SCPI.CONTrol.HANDler.C.DATA, 223
SCPI.CONTrol.HANDler.C.MODE, 224
SCPI.CONTrol.HANDler.D.DATA, 225
SCPI.CONTrol.HANDler.D.MODE, 226
SCPI.CONTrol.HANDler.E.DATA, 227
SCPI.CONTrol.HANDler.EXTension.INDex.STATe, 228
SCPI.CONTrol.HANDler.EXTension.RTRigger.STATe, 229
SCPI.CONTrol.HANDler.F.DATA, 230
SCPI.CONTrol.HANDler.OUTPut(Num).DATA, 231
SCPI.DISPlay.ANNotation.FREQuency.STATe, 232
SCPI.DISPlay.CCLear, 232
SCPI.DISPlay.CLOCk, 233
SCPI.DISPlay.COLor(Dnum).BACK, 234
SCPI.DISPlay.COLor(Dnum).GRATicule(Gnum), 235
SCPI.DISPlay.COLor(Dnum).LIMit(Lnum), 236
SCPI.DISPlay.COLor(Dnum).RESet, 237
SCPI.DISPlay.COLor(Dnum).TRACe(Tr).DATA, 238
SCPI.DISPlay.COLor(Dnum).TRACe(Tr).MEMory, 239
SCPI.DISPlay.ECHO.CLEar, 240
SCPI.DISPlay.ECHO.DATA, 240
SCPI.DISPlay.ENABle, 241
SCPI.DISPlay.FSIGn, 242
SCPI.DISPlay.IMAGe, 243
SCPI.DISPlay.MAXimize, 244
SCPI.DISPlay.SKEY.STATe, 245
SCPI.DISPlay.SPLit, 246
SCPI.DISPlay.TABLe.STATe, 248
SCPI.DISPlay.TABLe.TYPE, 249
SCPI.DISPlay.UPDate.IMMediate, 250
SCPI.DISPlay.WINDow(Ch).ACTivate, 250
SCPI.DISPlay.WINDow(Ch).ANNotation.MARKer.ALIGn.STATe, 251
SCPI.DISPlay.WINDow(Ch).ANNotation.MARKer.SINGle.STATe, 252
SCPI.DISPlay.WINDow(Ch).LABel, 253
SCPI.DISPlay.WINDow(Ch).MAXimize, 254
SCPI.DISPlay.WINDow(Ch).SPLit, 255
SCPI.DISPlay.WINDow(Ch).TITLe.DATA, 256
SCPI.DISPlay.WINDow(Ch).TITLe.STATe, 257

SCPI.DISPlay.WINDow(Ch).TRACe(Tr).ANNotation.MARKer.POSition.X, 258
SCPI.DISPlay.WINDow(Ch).TRACe(Tr).ANNotation.MARKer.POSition.Y, 259
SCPI.DISPlay.WINDow(Ch).TRACe(Tr).ANNotation.YAXis.MODE, 260
SCPI.DISPlay.WINDow(Ch).TRACe(Tr).MEMory.STATe, 261
SCPI.DISPlay.WINDow(Ch).TRACe(Tr).STATe, 262
SCPI.DISPlay.WINDow(Ch).TRACe(Tr).Y.SCALe.AUTO, 262
SCPI.DISPlay.WINDow(Ch).TRACe(Tr).Y.SCALe.PDIVision, 263
SCPI.DISPlay.WINDow(Ch).TRACe(Tr).Y.SCALe.RLEVel, 264
SCPI.DISPlay.WINDow(Ch).TRACe(Tr).Y.SCALe.RPOSition, 265
SCPI.DISPlay.WINDow(Ch).TRACe(Tr).Y.TRACk.FREQuency, 266
SCPI.DISPlay.WINDow(Ch).TRACe(Tr).Y.TRACk.MODE, 267
SCPI.DISPlay.WINDow(Ch).X.SPACing, 268
SCPI.DISPlay.WINDow(Ch).Y.SCALe.DIVisions, 269
SCPI.FORMat.BORDer, 270
SCPI.FORMat.DATA, 271
SCPI.HCOPy.ABORt, 272
SCPI.HCOPy.IMAGe, 272
SCPI.HCOPy.IMMediate, 273
SCPI.IEEE4882.CLS, 274
SCPI.IEEE4882.ESE, 274
SCPI.IEEE4882.ESR, 275
SCPI.IEEE4882.IDN, 275
SCPI.IEEE4882.OPC, 276
SCPI.IEEE4882.OPT, 277
SCPI.IEEE4882.RST, 277
SCPI.IEEE4882.SRE, 278
SCPI.IEEE4882.STB, 279
SCPI.IEEE4882.TRG, 279
SCPI.IEEE4882.WAI, 279
SCPI.INITiate(Ch).CONTinuous, 280
SCPI.INITiate(Ch).IMMediate, 281
SCPI.MMEMory.CATalog(Dir), 282
SCPI.MMEMory.COPY, 283
SCPI.MMEMory.DELete, 284
SCPI.MMEMory.LOAD.CHANnel.COEFficient, 285
SCPI.MMEMory.LOAD.CHANnel.STATe, 286
SCPI.MMEMory.LOAD.LIMit, 287
SCPI.MMEMory.LOAD.RLIMit, 288
SCPI.MMEMory.LOAD.SEGMent, 289
SCPI.MMEMory.LOAD.STATe, 290
SCPI.MMEMory.MDIRectory, 291
SCPI.MMEMory.STORe.CHANnel.CLEar, 292
SCPI.MMEMory.STORe.CHANnel.COEFficient = Register, 292
SCPI.MMEMory.STORe.CHANnel.STATe, 293
SCPI.MMEMory.STORe.FDATa, 294
SCPI.MMEMory.STORe.IMAGe, 295
SCPI.MMEMory.STORe.LIMit, 296
SCPI.MMEMory.STORe.RLIMit, 297
SCPI.MMEMory.STORe.SALL, 298
SCPI.MMEMory.STORe.SEGMent, 299
SCPI.MMEMory.STORe.STATe, 300
SCPI.MMEMory.STORe.STYPe, 301
SCPI.OUTPut.STATe, 302
SCPI.SENSe(Ch).AVERage.CLEar, 303
SCPI.SENSe(Ch).AVERage.COUNt, 303
SCPI.SENSe(Ch).AVERage.STATe, 304
SCPI.SENSe(Ch).BANDwidth.RESolution, 305
SCPI.SENSe(Ch).BWIDth.RESolution, 306
SCPI.SENSe(Ch).CORRection.CLEar, 307
SCPI.SENSe(Ch).CORRection.COEFficient.DATA(Str, Int1, Int2), 308
SCPI.SENSe(Ch).CORRection.COLLect.ACQuire.ISOLation, 310
SCPI.SENSe(Ch).CORRection.COLLect.ACQuire.LOAD, 311
SCPI.SENSe(Ch).CORRection.COLLect.ACQuire.OPEN, 312
SCPI.SENSe(Ch).CORRection.COLLect.ACQuire.SHORt, 312
SCPI.SENSe(Ch).CORRection.COLLect.ACQuire.THRU, 313
SCPI.SENSe(Ch).CORRection.COLLect.CKIT.LABel, 314
SCPI.SENSe(Ch).CORRection.COLLect.CKIT.ORDer.LOAD(Cpt), 315
SCPI.SENSe(Ch).CORRection.COLLect.CKIT.ORDer.OPEN(Cpt), 316
SCPI.SENSe(Ch).CORRection.COLLect.CKIT.ORDer.SHORt(Cpt), 317
SCPI.SENSe(Ch).CORRection.COLLect.CKIT.ORDer.THRU(Cpt_m,Cpt_n), 318
SCPI.SENSe(Ch).CORRection.COLLect.CKIT.RESet, 319
SCPI.SENSe(Ch).CORRection.COLLect.CKIT.SELect, 319
SCPI.SENSe(Ch).CORRection.COLLect.CKIT.STAN(Std).ARBitrary, 320
SCPI.SENSe(Ch).CORRection.COLLect.CKIT.STAN(Std).C0, 321
SCPI.SENSe(Ch).CORRection.COLLect.CKIT.STAN(Std).C1, 322
SCPI.SENSe(Ch).CORRection.COLLect.CKIT.STAN(Std).C2, 323
SCPI.SENSe(Ch).CORRection.COLLect.CKIT.STAN(Std).C3, 324
SCPI.SENSe(Ch).CORRection.COLLect.CKIT.STAN(Std).DELay, 325
SCPI.SENSe(Ch).CORRection.COLLect.CKIT.STAN(Std).L0, 326
SCPI.SENSe(Ch).CORRection.COLLect.CKIT.STAN(Std).L1, 327
SCPI.SENSe(Ch).CORRection.COLLect.CKIT.STAN(Std).L2, 328

# Index


SCPI.SENSe(Ch).CORRection.COLLect.CKIT.STAN(Std).L3, 329
SCPI.SENSe(Ch).CORRection.COLLect.CKIT.STAN(Std).LABel, 330
SCPI.SENSe(Ch).CORRection.COLLect.CKIT.STAN(Std).LOSS, 331
SCPI.SENSe(Ch).CORRection.COLLect.CKIT.STAN(Std).TYPE, 332
SCPI.SENSe(Ch).CORRection.COLLect.CKIT.STAN(Std).Z0, 333
SCPI.SENSe(Ch).CORRection.COLLect.ECAL.ERES, 334
SCPI.SENSe(Ch).CORRection.COLLect.ECAL.ISOLation.STATe, 335
SCPI.SENSe(Ch).CORRection.COLLect.ECAL.PATH(Cpt), 336
SCPI.SENSe(Ch).CORRection.COLLect.ECAL.SOLT1, 337
SCPI.SENSe(Ch).CORRection.COLLect.ECAL.SOLT2, 338
SCPI.SENSe(Ch).CORRection.COLLect.ECAL.THRU, 339
SCPI.SENSe(Ch).CORRection.COLLect.METHod.RESPonse.OPEN, 341
SCPI.SENSe(Ch).CORRection.COLLect.METHod.RESPonse.SHORt, 341
SCPI.SENSe(Ch).CORRection.COLLect.METHod.RESPonse.THRU, 342
SCPI.SENSe(Ch).CORRection.COLLect.METHod.SOLT1, 342
SCPI.SENSe(Ch).CORRection.COLLect.METHod.SOLT2, 343
SCPI.SENSe(Ch).CORRection.COLLect.METHod.TYPE, 344
SCPI.SENSe(Ch).CORRection.COLLect.SAVE, 345
SCPI.SENSe(Ch).CORRection.EXTension.PORT(Pt).TIME, 346
SCPI.SENSe(Ch).CORRection.EXTension.STATe, 347
SCPI.SENSe(Ch).CORRection.IMPedance.INPut.MAGNitude, 348
SCPI.SENSe(Ch).CORRection.PROPerty, 349
SCPI.SENSe(Ch).CORRection.RVELocity.COAX, 350
SCPI.SENSe(Ch).CORRection.STATe, 351
SCPI.SENSe(Ch).CORRection.TYPE, 352
SCPI.SENSe(Ch).FREQuency.CENTer, 353
SCPI.SENSe(Ch).FREQuency.CW, 354
SCPI.SENSe(Ch).FREQuency.DATA, 355
SCPI.SENSe(Ch).FREQuency.FIXed, 356
SCPI.SENSe(Ch).FREQuency.SPAN, 357
SCPI.SENSe(Ch).FREQuency.STARt, 358
SCPI.SENSe(Ch).FREQuency.STOP, 359
SCPI.SENSe(Ch).ROSCillator.SOURce, 360
SCPI.SENSe(Ch).SEGMent.DATA, 361
SCPI.SENSe(Ch).SEGMent.SWEep.POINts, 363
SCPI.SENSe(Ch).SEGMent.SWEep.TIME.DATA, 363
SCPI.SENSe(Ch).SWEep.DELay, 364
SCPI.SENSe(Ch).SWEep.POINts, 365
SCPI.SENSe(Ch).SWEep.TIME.AUTO, 366
SCPI.SENSe(Ch).SWEep.TIME.DATA, 367
SCPI.SENSe(Ch).SWEep.TYPE, 368
SCPI.SERVice.CHANnel(Ch).TRACe.ACTive, 370
SCPI.SERVice.CHANnel.ACTive, 369
SCPI.SERVice.CHANnel.COUNt, 369
SCPI.SERVice.CHANnel.TRACe.COUNt, 370
SCPI.SERVice.PORT.COUNt, 371
SCPI.SOURce(Ch).POWer.ATTenuation.DATA, 372
SCPI.SOURce(Ch).POWer.CENTer, 373
SCPI.SOURce(Ch).POWer.LEVel.IMMediate.AMPLitude, 374
SCPI.SOURce(Ch).POWer.LEVel.SLOPe.DATA, 375
SCPI.SOURce(Ch).POWer.LEVel.SLOPe.STATe, 376
SCPI.SOURce(Ch).POWer.PORT(Pt).LEVel.IMMediate.AMPLitude, 378
SCPI.SOURce(Ch).POWer.PORT.COUPle, 377
SCPI.SOURce(Ch).POWer.SPAN, 379
SCPI.SOURce(Ch).POWer.STARt, 380
SCPI.SOURce(Ch).POWer.STOP, 381
SCPI.STATus.OPERation.CONDition, 382
SCPI.STATus.OPERation.ENABle, 382
SCPI.STATus.OPERation.EVENt, 383
SCPI.STATus.OPERation.NTRansition, 383
SCPI.STATus.OPERation.PTRansition, 384
SCPI.STATus.PRESet, 384
SCPI.STATus.QUEStionable.BLIMit.CHANnel(Ch).CONDition, 385
SCPI.STATus.QUEStionable.BLIMit.CHANnel(Ch).ENABle, 386
SCPI.STATus.QUEStionable.BLIMit.CHANnel(Ch).EVENt, 387
SCPI.STATus.QUEStionable.BLIMit.CHANnel(Ch).NTRansition, 388
SCPI.STATus.QUEStionable.BLIMit.CHANnel(Ch).PTRansition, 389
SCPI.STATus.QUEStionable.BLIMit.CONDition, 390
SCPI.STATus.QUEStionable.BLIMit.ENABle, 391
SCPI.STATus.QUEStionable.BLIMit.EVENt, 391
SCPI.STATus.QUEStionable.BLIMit.NTRansition, 392
SCPI.STATus.QUEStionable.BLIMit.PTRansition, 393
SCPI.STATus.QUEStionable.CONDition, 394
SCPI.STATus.QUEStionable.ENABle, 394
SCPI.STATus.QUEStionable.EVENt, 395
SCPI.STATus.QUEStionable.LIMit.CHANnel(Ch).CONDition, 395
SCPI.STATus.QUEStionable.LIMit.CHANnel(Ch).ENABle, 396
SCPI.STATus.QUEStionable.LIMit.CHANnel(Ch).EVENt, 397
SCPI.STATus.QUEStionable.LIMit.CHANnel(Ch).NTRansition, 398
SCPI.STATus.QUEStionable.LIMit.CHANnel(Ch).PTRansition, 399
SCPI.STATus.QUEStionable.LIMit.CONDition, 400
SCPI.STATus.QUEStionable.LIMit.ENABle, 401
SCPI.STATus.QUEStionable.LIMit.EVENt, 401

SCPI.STATus.QUEStionable.LIMit.NTRansition, 402
SCPI.STATus.QUEStionable.LIMit.PTRansition, 403
SCPI.STATus.QUEStionable.NTRansition, 404
SCPI.STATus.QUEStionable.PTRansition, 405
SCPI.STATus.QUEStionable.RLIMit.CHANnel(Ch).CONDition, 406
SCPI.STATus.QUEStionable.RLIMit.CHANnel(Ch).ENABle, 407
SCPI.STATus.QUEStionable.RLIMit.CHANnel(Ch).EVENt, 408
SCPI.STATus.QUEStionable.RLIMit.CHANnel(Ch).NTRansition, 409
SCPI.STATus.QUEStionable.RLIMit.CHANnel(Ch).PTRansition, 410
SCPI.STATus.QUEStionable.RLIMit.CONDition, 411
SCPI.STATus.QUEStionable.RLIMit.ENABle, 412
SCPI.STATus.QUEStionable.RLIMit.EVENt, 412
SCPI.STATus.QUEStionable.RLIMit.NTRansition, 413
SCPI.STATus.QUEStionable.RLIMit.PTRansition, 414
SCPI.SYSTem.BACKlight, 415
SCPI.SYSTem.BEEPer.COMPlete.IMMediate, 416
SCPI.SYSTem.BEEPer.COMPlete.STATe, 416
SCPI.SYSTem.BEEPer.WARNing.IMMediate, 417
SCPI.SYSTem.BEEPer.WARNing.STATe, 417
SCPI.SYSTem.DATE, 418
SCPI.SYSTem.ERRor, 419
SCPI.SYSTem.KLOCk.KBD, 420
SCPI.SYSTem.KLOCk.MOUSe, 421
SCPI.SYSTem.POFF, 421
SCPI.SYSTem.PRESet, 422
SCPI.SYSTem.SECurity.LEVel, 423
SCPI.SYSTem.SERVice, 424
SCPI.SYSTem.TIME, 425
SCPI.SYSTem.UPReset, 426
SCPI.TRIGger.SEQuence.IMMediate, 427
SCPI.TRIGger.SEQuence.SINGle, 428
SCPI.TRIGger.SEQuence.SOURce, 429
SENSe(Ch).CORRection.COLLect.METHod.ERESponse, 340
UserMenu.Item(Key_id).Caption, 119
UserMenu.Item(Key_id).Enabled, 120
UserMenu.PRESet, 121
UserMenu.Press(Key_id), 122
UserMenu.Show, 122
UserMenu_OnPress(ByVal Key_id As Long), 121
VBAVersion, 123
WaitOnSRQ, 124

COM object
about COM object, 27
accessing a list of COM objects, 62
auto-complete function, 41
control methods of the E5061A/E5062A, 27
conversion rules from SCPI commands, 101
ECHO, 115
list by front panel key, 102
looking up COM objects, 20
major control difference between COM object and SCPI command, 28
object model, 100
overview of E5061A/E5062A COM object, 27

COM object reference
notational rules, 113

Complex operation library, 450
ComplexAbs(x), 451
ComplexAdd(x,y), 451
ComplexArg(x), 451
ComplexConj(x), 452
ComplexCos(x), 452
ComplexCosh(x), 452
ComplexDiv(x,y), 452
ComplexExp(x), 453
ComplexLog(x), 453
ComplexLog10(x), 453
ComplexMul(x,y), 453
ComplexNorm(x), 454
ComplexPolar(x,y), 454
ComplexSet(x,y), 454
ComplexSetArray(x), 455
ComplexSin(x), 455
ComplexSinh(x), 455
ComplexSqrt(x), 456
ComplexSub(x,y), 456

Complex type, 450
ComplexAbs(x), 451
ComplexAdd(x,y), 451
ComplexArg(x), 451
ComplexConj(x), 452
ComplexCos(x), 452
ComplexCosh(x), 452
ComplexDiv(x,y), 452
ComplexExp(x), 453
ComplexLog(x), 453
ComplexLog10(x), 453
ComplexMul(x,y), 453
ComplexNorm(x), 454
ComplexPolar(x,y), 454
ComplexSet(x,y), 454
ComplexSetArray(x), 455
ComplexSin(x), 455
ComplexSinh(x), 455
ComplexSqrt(x), 456
ComplexSub(x,y), 456

control methods, 26

control system
control system based on the macro function, 25

controlling E5061A/E5062A
control methods of the E5061A/E5062A, 27

controlling peripherals
control methods, 26
implementing a control system, 25
overview, 82

# Index


**D**

data
  reading/writing measurement data, 71
data hint, 56
DC bias
  application program for DC power supply, 84
debugging, 53
description
  COM Object reference, 113
display
  echo Window, 59
  initial screen of Visual Basic Editor, 30
  switching from Visual Basic Editor to E5061A/E5062A measurement screen, 32
documentation map, 4

**E**

echo window, 59
Echo Window button, 59
End statement
  stopping a VBA program, 52
equivalent key
  equivalent key to COM object, 114
error handling routine
  On Error GoTo sentence, 90
errors
  run-time errors, 53
  syntax errors, 53
event
  COM objects, 27
  executing a procedure with assigned softkey, 76
examples
  COM object reference, 114
export
  saving a module, 43
external instruments
  controlling external instruments from E5061A/E5062A, 81

**F**

F/W version, 275
Firmware version, 275
FirstLeftGap(Chan), 435
FirstLeftInterval(Chan), 436
FirstLRightInterval(Chan), 438
FirstRightGap(Chan), 437
For•••Next statement, 40
front panel key
  equivalent key to COM object, 114

**G**

GapMean(Chan), 439

**H**

help
  using VBA online help, 60
history
  manual printing, 2

**I**

I/O library
  controlling peripherals from E5061A/E5062A, 81
immediate window, 56
import
  loading a module, 46
index
  storing array data, 71
Initialization
  Error queue, 274
  Status register, 274
Installation and Quick Start Guide, 4

**L**

library references
  automatic library references, 63
Load & Run, 50
Load Project button, 45
loading
  loading a module, 46
  loading a project, 45
  loading a project at power-on, 46
local window
  debug toolbar, 54

**M**

macro
  introduction of the macro function, 24
Macro Break key, 51
macro function
  control system based on the macro function, 25
Macro Run key, 50
Macros dialog box, 49
manual changes, 459
manual printing history, 2
MaxEnvelopeGap(Chan), 440
MaxGap(Chan), 441
MaxLeftGap(Chan), 442
MaxPeakToPeak(Chan), 443
MaxRightGap(Chan), 444
MaxRipplePoint(Chan,Stim), 445
MaxRippleValue(Chan), 446
measurement
  detecting the end of measurement, 66
  program for measuring a band-pass filter, 88
menu bar, 30
method
  COM objects, 27
modules
  class module, 33
  deleting modules, 37
  loading a module, 46
  saving a module, 43

standard module, 33
user form, 33

**O**

object box, 38
object browser, 62
object model, 100
Operational Manual, 4
option ID, 277

**P**

peripherals
controlling peripherals from E5061A/E5062A, 81
Pole(Chan,D,LeftStim,LeftValue,RightStim,RightValue), 447
Preset User Menu button, 77
procedure box, 38
Product information, 275
programming
auto-complete function, 41
coding a VBA program, 38
Programming Manual, 4
project, 33
loading a project, 45
saving a project file, 42
project explorer, 31
property
COM objects, 27
monitoring property values, 56
property window, 31

**Q**

Quick Start Guide, 4
quick watch, 58

**R**

reading
reading/writing measurement data, 71
reference
automatic library references, 63
related objects
COM object reference, 114
Reset
Error queue, 274
Instrument setting, 277
Status register, 274
Ripple
Ripple Analysis Library, 432
ripple Analysis
setting the peak definition, 432
ripple analysis
FirstLeftGap(Chan), 435
FirstLeftInterval(Chan), 436
FirstLRightInterval(Chan), 438
FirstRightGap(Chan), 437
flow of programming, 432
GapMean(Chan), 439
MaxEnvelopeGap(Chan), 440
MaxGap(Chan), 441
MaxLeftGap(Chan), 442
MaxPeakToPeak(Chan), 443
MaxRightGap(Chan), 444
MaxRipplePoint(Chan,Stim), 445
MaxRippleValue(Chan), 446
Pole(Chan,D,LeftStim,LeftValue,RightStim,RightValue), 447
simple use example, 434
specifying analysis range, 432
running
running a VBA program, 48

**S**

sample program
apl_bsc.vba, 88
how to load, 20, 21
map_drive.vba, 96
meas_sing.vba, 69
meas_srq.vba, 66
meas_user.vba, 78
read_write.vba, 72
visa32.bas, 82
vpptype.bas, 82
sample program disk
notice, 3
Save Project button, 43
saving
saving a module, 43
saving a project file, 42
SCPI, 277
SCPI command
major control difference between COM object and SCPI command, 28
SCPI objects, 101
Select Macro button, 50
Serial number plate, 460
Service Request Enable Register
Setting, 278
softkey
executing a procedure with assigned softkey, 76
executinga VBA program in the VBA folder, 50
SRQ
suspending program until detection, 66
Standard Event Status Register
Reading out, 275
Setting valid registers, 274
standard module, 33
inserting standard module, 36
Status Byte Register
Reading out, 279
status register
detecting the end of measurement, 66
step-in
debug toolbar, 54

step-out
  debug toolbar, 54
step-over
  debug toolbar, 54
Stop button, 51
syntax
  COM object reference, 113

**T**

toolbar, 30
  debug toolbar, 54
trigger, 279
triggering
  detecting the end of measurement, 66
type of object
  COM object reference, 113

**U**

user form, 33
  inserting UserForm, 34
User Menu button, 77
user menu function, 76
UserMenu_OnPress Procedure, 76

**V**

variable
  COM object reference, 114
  monitoring variables, 56
Variant type
  storing array data, 71
VBA
  introduction of the macro function, 24
VBA Editor button, 30
VBA programmer's
  contents of this manual, 18
  how to use this manual, 20
vbreadme.txt
  CD-ROM, 83
viClose function
  VISA, 86
viOpen function
  VISA, 84
viOpenDefaultRM function
  VISA, 84
VISA
  control flow with VISA, 83
  importing definition files, 82
  notes on using VISA library with VB, 83
  online help of VISA library, 83
  programming using VISA, 83
visa.hlp
  CD-ROM, 83
Visual Basic Editor
  closing Visual Basic Editor, 32
  displaying Visual Basic Editor, 30
  initial screen, 30
viVPrintf function
  VISA, 85
viVScanf function
  VISA, 86

**W**

wait, 279
WaitOnSRQ, 66
watch window, 57
writing
  reading/writing measurement data, 71