



COPPER MOUNTAIN
TECHNOLOGIES

S2 series

(Planar 304/1, Planar 804/1
S5048, S7530, C1209, C1220)

Network Analyzer COM/DCOM Programming Manual



Software version 16.1
January, 2016

TABLE OF CONTENTS

1	Introduction.....	10
2	Scope of Manual	10
3	Registering COM Server	11
4	COM Technology Overview.....	11
5	Automation Server.....	11
6	Automation Controllers	12
7	Local and Remote Server.....	12
8	Structure of COM Objects	14
9	Accessing the Application Object.....	15
10	Object Methods.....	17
11	Object Properties	17
12	Error Handling	18
13	COM Automation Data Types.....	20
14	Measurement Data Arrays	21
15	COM Server Commands.....	22
	NAME	22
	Ready	23
	SCPI.ABORT	23
	SCPI.CALCulate(<i>Ch</i>).FSIMulator.SENDed.DEEMbed.PORT(<i>Pt</i>).USER.FILEname	24
	SCPI.CALCulate(<i>Ch</i>).FSIMulator.SENDed.DEEMbed.PORT(<i>Pt</i>).STATe	26
	SCPI.CALCulate(<i>Ch</i>).FSIMulator.SENDed.DEEMbed.STATe	27
	SCPI.CALCulate(<i>Ch</i>).FSIMulator.SENDed.PMCircuit.PORT(<i>Pt</i>).USER.FILEname	28
	SCPI.CALCulate(<i>Ch</i>).FSIMulator.SENDed.PMCircuit.PORT(<i>Pt</i>).STATe	29
	SCPI.CALCulate(<i>Ch</i>).FSIMulator.SENDed.PMCircuit.STATe	30
	SCPI.CALCulate(<i>Ch</i>).FSIMulator.SENDed.ZCONversion.PORT(<i>Pt</i>).Z0.R	31
	SCPI.CALCulate(<i>Ch</i>).FSIMulator.SENDed.ZCONversion.STATe	32
	SCPI.CALCulate(<i>Ch</i>).FSIMulator.STATe	33
	SCPI.CALCulate(<i>Ch</i>).PARameter(1).COUNT	34
	SCPI.CALCulate(<i>Ch</i>).PARameter(<i>Tt</i>).DEFine	35
	SCPI.CALCulate(<i>Ch</i>).PARameter(<i>Tt</i>).SElect	37
	SCPI.CALCulate(<i>Ch</i>).PARameter(<i>Tt</i>).SPORT	38
	SCPI.CALCulate(<i>Ch</i>).SElected.CONVersion.FUNCTion	39
	SCPI.CALCulate(<i>Ch</i>).SElected.CONVersion.STATE	40
	SCPI.CALCulate(<i>Ch</i>).SElected.CORRection.EDELay.TIME	41
	SCPI.CALCulate(<i>Ch</i>).SElected.CORRection.OFFSet.PHASe	42
	SCPI.CALCulate(<i>Ch</i>).SElected.CORRection.STATus	43
	SCPI.CALCulate(<i>Ch</i>).SElected.DATA.FDATA	44
	SCPI.CALCulate(<i>Ch</i>).SElected.DATA.FMEMory	45
	SCPI.CALCulate(<i>Ch</i>).SElected.DATA.SDATA	46
	SCPI.CALCulate(<i>Ch</i>).SElected.DATA.SMEMory	47
	SCPI.CALCulate(<i>Ch</i>).SElected.FILTter.GATE.TIME.CENTer	48
	SCPI.CALCulate(<i>Ch</i>).SElected.FILTter.GATE.TIME.SHAPe	49
	SCPI.CALCulate(<i>Ch</i>).SElected.FILTter.GATE.TIME.SPAN	50
	SCPI.CALCulate(<i>Ch</i>).SElected.FILTter.GATE.TIME.START	51
	SCPI.CALCulate(<i>Ch</i>).SElected.FILTter.GATE.TIME.STATE	52
	SCPI.CALCulate(<i>Ch</i>).SElected.FILTter.GATE.TIME.STOP	53
	SCPI.CALCulate(<i>Ch</i>).SElected.FILTter.GATE.TIME.TYPE	54
	SCPI.CALCulate(<i>Ch</i>).SElected.FORMat	55
	SCPI.CALCulate(<i>Ch</i>).SElected.FUNCTion.DATA	56
	SCPI.CALCulate(<i>Ch</i>).SElected.FUNCTion.DOMain.COUPle	57
	SCPI.CALCulate(<i>Ch</i>).SElected.FUNCTion.DOMain.START	58
	SCPI.CALCulate(<i>Ch</i>).SElected.FUNCTion.DOMain.STATE	59
	SCPI.CALCulate(<i>Ch</i>).SElected.FUNCTion.DOMain.STOP	60

SCPI.CALCulate(<i>Ch</i>).SElected.FUNCtion.EXECute	61
SCPI.CALCulate(<i>Ch</i>).SElected.FUNCtion.PEXCursion	62
SCPI.CALCulate(<i>Ch</i>).SElected.FUNCtion.POINts	63
SCPI.CALCulate(<i>Ch</i>).SElected.FUNCtion.PPOLarity	64
SCPI.CALCulate(<i>Ch</i>).SElected.FUNCtion.TARGet	65
SCPI.CALCulate(<i>Ch</i>).SElected.FUNCtion.TTTransition	66
SCPI.CALCulate(<i>Ch</i>).SElected.FUNCtion.TYPE	67
SCPI.CALCulate(<i>Ch</i>).SElected.LIMit.DATA	68
SCPI.CALCulate(<i>Ch</i>).SElected.LIMit.DISPlay.STATE	69
SCPI.CALCulate(<i>Ch</i>).SElected.LIMit.FAIL	70
SCPI.CALCulate(<i>Ch</i>).SElected.LIMit.OFFSet.AMPLitude	71
SCPI.CALCulate(<i>Ch</i>).SElected.LIMit.OFFSet.MARKer	72
SCPI.CALCulate(<i>Ch</i>).SElected.LIMit.OFFSet.STIMulus	73
SCPI.CALCulate(<i>Ch</i>).SElected.LIMit.REPort.ALL	74
SCPI.CALCulate(<i>Ch</i>).SElected.LIMit.REPort.DATA	75
SCPI.CALCulate(<i>Ch</i>).SElected.LIMit.REPort.POINts	76
SCPI.CALCulate(<i>Ch</i>).SElected.LIMit.STATE	77
SCPI.CALCulate(<i>Ch</i>).SElected.MARKer(<i>Mk</i>).ACTivate	78
SCPI.CALCulate(<i>Ch</i>).SElected.MARKer(<i>Mk</i>).BWIDth.DATA	79
SCPI.CALCulate(<i>Ch</i>).SElected.MARKer(1).BWIDth.REFERENCE	80
SCPI.CALCulate(<i>Ch</i>).SElected.MARKer(1).BWIDth.STATE	81
SCPI.CALCulate(<i>Ch</i>).SElected.MARKer(1).BWIDth.THreshold	82
SCPI.CALCulate(<i>Ch</i>).SElected.MARKer(1).BWIDth.TYPE	83
SCPI.CALCulate(<i>Ch</i>).SElected.MARKer(1).COUNT	84
SCPI.CALCulate(<i>Ch</i>).SElected.MARKer(1).COUPLE	85
SCPI.CALCulate(<i>Ch</i>).SElected.MARKer(1).DATA	86
SCPI.CALCulate(<i>Ch</i>).SElected.MARKer(1).FUNCtion.DOMain.COUPLE	87
SCPI.CALCulate(<i>Ch</i>).SElected.MARKer(1).FUNCtion.DOMain.START	88
SCPI.CALCulate(<i>Ch</i>).SElected.MARKer(1).FUNCtion.DOMain.STATE	89
SCPI.CALCulate(<i>Ch</i>).SElected.MARKer(1).FUNCtion.DOMain.STOP	90
SCPI.CALCulate(<i>Ch</i>).SElected.MARKer(<i>Mk</i>).FUNCtion.EXECute	91
SCPI.CALCulate(<i>Ch</i>).SElected.MARKer(<i>Mk</i>).FUNCtion.PEXCursion	92
SCPI.CALCulate(<i>Ch</i>).SElected.MARKer(<i>Mk</i>).FUNCtion.PPOLarity	93
SCPI.CALCulate(<i>Ch</i>).SElected.MARKer(<i>Mk</i>).FUNCtion.TARGet	94
SCPI.CALCulate(<i>Ch</i>).SElected.MARKer(<i>Mk</i>).FUNCtion.TRACKing	95
SCPI.CALCulate(<i>Ch</i>).SElected.MARKer(<i>Mk</i>).FUNCtion.TTTransition	96
SCPI.CALCulate(<i>Ch</i>).SElected.MARKer(<i>Mk</i>).FUNCtion.TYPE	97
SCPI.CALCulate(<i>Ch</i>).SElected.MARKer(1).MATH.FLATness.DATA	98
SCPI.CALCulate(<i>Ch</i>).SElected.MARKer(1).MATH.FLATness.DOMain.START	99
SCPI.CALCulate(<i>Ch</i>).SElected.MARKer(1).MATH.FLATness.DOMain.STOP	100
SCPI.CALCulate(<i>Ch</i>).SElected.MARKer(1).MATH.FLATness.STATE	101
SCPI.CALCulate(<i>Ch</i>).SElected.MARKer(1).REFerence.STATE	102
SCPI.CALCulate(<i>Ch</i>).SElected.MARKer(<i>Mk</i>).POSITION	103
SCPI.CALCulate(<i>Ch</i>).SElected.MARKer(<i>Mk</i>).STATE	104
SCPI.CALCulate(<i>Ch</i>).SElected.MARKer(<i>Mk</i>).X	105
SCPI.CALCulate(<i>Ch</i>).SElected.MARKer(<i>Mk</i>).Y	106
SCPI.CALCulate(<i>Ch</i>).SElected.MATH.FUNCtion	107
SCPI.CALCulate(<i>Ch</i>).SElected.MATH.MEMorize	108
SCPI.CALCulate(<i>Ch</i>).SElected.MStatistics.DATA	109
SCPI.CALCulate(<i>Ch</i>).SElected.MStatistics.DOMain.MARKer.START	110
SCPI.CALCulate(<i>Ch</i>).SElected.MStatistics.DOMain.MARKer.STOP	111
SCPI.CALCulate(<i>Ch</i>).SElected.MStatistics.DOMain.STATE	112
SCPI.CALCulate(<i>Ch</i>).SElected.MStatistics.STATE	113
SCPI.CALCulate(<i>Ch</i>).SElected.RLIMit.DATA	114

SCPI.CALCulate(<i>Ch</i>).SElected.RLIMit.DISPlay.LINE	115
SCPI.CALCulate(<i>Ch</i>).SElected.RLIMit.DISPlay.SElect	116
SCPI.CALCulate(<i>Ch</i>).SElected.RLIMit.DISPlay.VALue	117
SCPI.CALCulate(<i>Ch</i>).SElected.RLIMit.FAIL	118
SCPI.CALCulate(<i>Ch</i>).SElected.RLIMit.REPort.DATA	119
SCPI.CALCulate(<i>Ch</i>).SElected.RLIMit.STATe	120
SCPI.CALCulate(<i>Ch</i>).SElected.SMOothing.APERture	121
SCPI.CALCulate(<i>Ch</i>).SElected.SMOothing.STATe	122
SCPI.CALCulate(<i>Ch</i>).SElected.TRANSform.TIME.CENTer	123
SCPI.CALCulate(<i>Ch</i>).SElected.TRANSform.TIME.IMPulse.WIDTh	124
SCPI.CALCulate(<i>Ch</i>).SElected.TRANSform.TIME.KBESsel	125
SCPI.CALCulate(<i>Ch</i>).SElected.TRANSform.TIME.LPFRrequency	126
SCPI.CALCulate(<i>Ch</i>).SElected.TRANSform.TIME.REFLection.TYPE	127
SCPI.CALCulate(<i>Ch</i>).SElected.TRANSform.TIME.SPAN	128
SCPI.CALCulate(<i>Ch</i>).SElected.TRANSform.TIME.START	129
SCPI.CALCulate(<i>Ch</i>).SElected.TRANSform.TIME.STATe	130
SCPI.CALCulate(<i>Ch</i>).SElected.TRANSform.TIME.STEP.RTIme	131
SCPI.CALCulate(<i>Ch</i>).SElected.TRANSform.TIME.STIMulus	132
SCPI.CALCulate(<i>Ch</i>).SElected.TRANSform.TIME.STOP	133
SCPI.CALCulate(<i>Ch</i>).SElected.TRANSform.TIME.TYPE	134
SCPI.CALCulate(<i>Ch</i>).SElected.TRANSform.TIME.UNIT	135
SCPI.CALCulate(<i>Ch</i>).TRACe(<i>Tr</i>).DATA.FDAta	136
SCPI.CALCulate(<i>Ch</i>).TRACe(<i>Tr</i>).DATA.FMEmory	137
SCPI.CALCulate(<i>Ch</i>).TRACe(<i>Tr</i>).DATA.SDAta	138
SCPI.CALCulate(<i>Ch</i>).TRACe(<i>Tr</i>).DATA.SMEMory	139
SCPI.DISPlay.COLor.BACK	140
SCPI.DISPlay.COLor.GRATicule	141
SCPI.DISPlay.COLor.RESet	142
SCPI.DISPlay.COLor.TRACe(<i>Tr</i>).DATA	143
SCPI.DISPlay.COLor.TRACe(<i>Tr</i>).MEMory	144
SCPI.DISPlay.ENABLE	145
SCPI.DISPlay.FSIGn	146
SCPI.DISPlay.GLABel	147
SCPI.DISPlay.IMGage	148
SCPI.DISPlay.HIDE	149
SCPI.DISPlay.MAXimize	149
SCPI.DISPlay.PARTition.FONT.SIZE(<i>Param</i>)	150
SCPI.DISPlay.PARTition.VISible(<i>Param</i>)	151
SCPI.DISPlay.POSition	152
SCPI.DISPlay.SHOW	153
SCPI.DISPlay.SPLit	154
SCPI.DISPlay.REFresh.IMMediate	155
SCPI.DISPlay.WINDOW(<i>Ch</i>).ACTivate	155
SCPI.DISPlay.WINDOW(<i>Ch</i>).ANNotation.MARKer.ALIGN.TYPE	156
SCPI.DISPlay.WINDOW(<i>Ch</i>).ANNotation.MARKer.SINGle.STATE	157
SCPI.DISPlay.WINDOW(<i>Ch</i>).MAXimize	158
SCPI.DISPlay.WINDOW(<i>Ch</i>).SPLit	159
SCPI.DISPlay.WINDOW(<i>Ch</i>).TITLE.DATA	160
SCPI.DISPlay.WINDOW(<i>Ch</i>).TITLE.STATE	161
SCPI.DISPlay.WINDOW(<i>Ch</i>).TRACe(<i>Tr</i>).ANNotation.MARKer. MEMORY	162
SCPI.DISPlay.WINDOW(<i>Ch</i>).TRACe(<i>Tr</i>).ANNotation.MARKer. POSITION.X	163
SCPI.DISPlay.WINDOW(<i>Ch</i>).TRACe(<i>Tr</i>).ANNotation.MARKer. POSITION.Y	164
SCPI.DISPlay.WINDOW(<i>Ch</i>).TRACe(<i>Tr</i>).MEMORY. STATE	165
SCPI.DISPlay.WINDOW(<i>Ch</i>).TRACe(<i>Tr</i>).STATE	166

SCPI.DISPlay.WINDOW(<i>Ch</i>).TRACe(<i>Tr</i>).Y.SCALE.AUTO	167
SCPI.DISPlay.WINDOW(<i>Ch</i>).TRACe(<i>Tr</i>).Y.SCALE.PDIVision.....	168
SCPI.DISPlay.WINDOW(<i>Ch</i>).TRACe(<i>Tr</i>).Y.SCALE.RLevel.....	169
SCPI.DISPlay.WINDOW(<i>Ch</i>).TRACe(<i>Tr</i>).Y.SCALE.RPosition.....	170
SCPI.DISPlay.WINDOW(<i>Ch</i>).X.SPACing	171
SCPI.DISPlay.WINDOW(<i>Ch</i>).Y.SCALE.DIVisions	172
SCPI.HCOPy.ABORT	173
SCPI.HCOPy.DATE.STAMP.....	173
SCPI.HCOPy.IMAGe	174
SCPI.HCOPy.IMMEDIATE.....	175
SCPI.HCOPy.PAINT.....	176
SCPI.IEEE4882.CLS.....	177
SCPI.IEEE4882.IDN.....	177
SCPI.IEEE4882.OPC.....	178
SCPI.IEEE4882.RST	178
SCPI.IEEE4882.TRG	179
SCPI.IEEE4882.WAI	179
SCPI.INITiate(<i>Ch</i>).CONTinuous	180
SCPI.INITiate(<i>Ch</i>).IMMEDIATE	181
SCPI.MMEmory.CATalog(<i>Dir</i>)	182
SCPI.MMEmory.COPY(<i>Src</i> , <i>Dst</i>).....	183
SCPI.MMEmory.DElete(<i>File</i>)	183
SCPI.MMEmory.LOAD.CHANnel.STATE	184
SCPI.MMEmory.LOAD.CKIT(<i>Ck</i>).....	185
SCPI.MMEmory.LOAD.LIMIT	186
SCPI.MMEmory.LOAD.PLOSS(<i>Pt</i>).....	187
SCPI.MMEmory.LOAD.RLIMIT	188
SCPI.MMEmory.LOAD.SEGMENT	189
SCPI.MMEmory.LOAD.SNP.DATA	190
SCPI.MMEmory.LOAD.SNP.TRACe(<i>Tr</i>).MEMory	191
SCPI.MMEmory.LOAD.STATE	192
SCPI.MMEmory.MDIRectory	193
SCPI.MMEmory.STORe.CHANnel.CLEAR	194
SCPI.MMEmory.STORe.CHANnel.STATE	194
SCPI.MMEmory.STORe.CKIT(<i>Ck</i>)	195
SCPI.MMEmory.STORe.FDATA	196
SCPI.MMEmory.STORe.IMAGe	197
SCPI.MMEmory.STORe.LIMIT	198
SCPI.MMEmory.STORe.PLOSS(<i>Pt</i>).....	199
SCPI.MMEmory.STORe.RLIMIT	200
SCPI.MMEmory.STORe.SEGMENT	201
SCPI.MMEmory.STORe.SNP.DATA	202
SCPI.MMEmory.STORe.SNP.FORMAT	203
SCPI.MMEmory.STORe.SNP.TYPE.S1P	204
SCPI.MMEmory.STORe.SNP.TYPE.S2P	205
SCPI.MMEmory.STORe.STATE	206
SCPI.MMEmory.STORe.STYPE	207
SCPI.OUTPut.STATE	208
SCPI.SENSe(<i>Ch</i>).AVERage.CLEAR	209
SCPI.SENSe(<i>Ch</i>).AVERage.COUNT	209
SCPI.SENSe(<i>Ch</i>).AVERage.STATE	210
SCPI.SENSe(<i>Ch</i>).BANDwidth.RESolution	211
SCPI.SENSe(<i>Ch</i>).CORRection.CLEAR	211
SCPI.SENSe(<i>Ch</i>).CORRection.COEFFicient.DATA(<i>Str</i> , <i>Pt_r</i> , <i>Pt_s</i>)	212

SCPI.SENSE(Ch).CORRection.COEfficient.METHOD.ERESponse.....	213
SCPI.SENSE(Ch).CORRection.COEfficient.METHOD.RESPonse.OPEN	214
SCPI.SENSE(Ch).CORRection.COEfficient.METHOD.RESPonse.SHORT.....	215
SCPI.SENSE(Ch).CORRection.COEfficient.METHOD.RESPonse.THRU	216
SCPI.SENSE(Ch).CORRection.COEfficient.METHOD.SOLT1.....	217
SCPI.SENSE(Ch).CORRection.COEfficient.METHOD.SOLT2.....	218
SCPI.SENSE(Ch).CORRection.COEfficient.SAVE	219
SCPI.SENSE(Ch).CORRection.COLlect.ACQuire.ISOLation	220
SCPI.SENSE(Ch).CORRection.COLlect.ACQuire.LOAD	221
SCPI.SENSE(Ch).CORRection.COLlect.ACQuire.OPEN	222
SCPI.SENSE(Ch).CORRection.COLlect.ACQuire.SHORT.....	223
SCPI.SENSE(Ch).CORRection.COLlect.ACQuire.SUBClass	224
SCPI.SENSE(Ch).CORRection.COLlect.ACQuire.THRU	225
SCPI.SENSE(Ch).CORRection.COLlect.CKIT.LABel	226
SCPI.SENSE(Ch).CORRection.COLlect.CKIT.ORDER.LOAD(Pt).....	227
SCPI.SENSE(Ch).CORRection.COLlect.CKIT.ORDER.OPEN(Pt).....	228
SCPI.SENSE(Ch).CORRection.COLlect.CKIT.ORDER.SElect	229
SCPI.SENSE(Ch).CORRection.COLlect.CKIT.ORDER.SHORT(Pt).....	230
SCPI.SENSE(Ch).CORRection.COLlect.CKIT.ORDER.THRU(Pt_m , Pt_n).....	231
SCPI.SENSE(Ch).CORRection.COLlect.CKIT.RESet	232
SCPI.SENSE(Ch).CORRection.COLlect.CKIT.SElect	232
SCPI.SENSE(Ch).CORRection.COLlect.CKIT.STAN(Std).ARbitrary	233
SCPI.SENSE(Ch).CORRection.COLlect.CKIT.STAN(Std).CO	235
SCPI.SENSE(Ch).CORRection.COLlect.CKIT.STAN(Std).C1	236
SCPI.SENSE(Ch).CORRection.COLlect.CKIT.STAN(Std).C2	237
SCPI.SENSE(Ch).CORRection.COLlect.CKIT.STAN(Std).C3	238
SCPI.SENSE(Ch).CORRection.COLlect.CKIT.STAN(Std).DELay	239
SCPI.SENSE(Ch).CORRection.COLlect.CKIT.STAN(Std).FMAXimum	240
SCPI.SENSE(Ch).CORRection.COLlect.CKIT.STAN(Std).FMINimum	241
SCPI.SENSE(Ch).CORRection.COLlect.CKIT.STAN(Std).L0	242
SCPI.SENSE(Ch).CORRection.COLlect.CKIT.STAN(Std).L1	243
SCPI.SENSE(Ch).CORRection.COLlect.CKIT.STAN(Std).L2	244
SCPI.SENSE(Ch).CORRection.COLlect.CKIT.STAN(Std).L3	245
SCPI.SENSE(Ch).CORRection.COLlect.CKIT.STAN(Std).LABel	246
SCPI.SENSE(Ch).CORRection.COLlect.CKIT.STAN(Std).LOSS	247
SCPI.SENSE(Ch).CORRection.COLlect.CKIT.STAN(Std).TYPE	248
SCPI.SENSE(Ch).CORRection.COLlect.CKIT.STAN(Std).Z0	249
SCPI.SENSE(Ch).CORRection.COLlect.CLEAR	250
SCPI.SENSE(Ch).CORRection.COLlect.DATA.ISOLation(Pt_r , Pt_s).....	251
SCPI.SENSE(Ch).CORRection.COLlect.DATA.LOAD(Pt)	252
SCPI.SENSE(Ch).CORRection.COLlect.DATA.OPEN(Pt)	253
SCPI.SENSE(Ch).CORRection.COLlect.DATA.SHORT(Pt).....	254
SCPI.SENSE(Ch).CORRection.COLlect.DATA.THRU. MATCH(Pt_r , Pt_s).....	255
SCPI.SENSE(Ch).CORRection.COLlect.DATA.THRU. TRANsmission(Pt_r , Pt_s).....	256
SCPI.SENSE(Ch).CORRection.COLlect.ECAL.INFormation	257
SCPI.SENSE(Ch).CORRection.COLlect.ECAL.ORlenation.STATE	258
SCPI.SENSE(Ch).CORRection.COLlect.ECAL.PATH(Pt)	259
SCPI.SENSE(Ch).CORRection.COLlect.ECAL.SOLT1	260
SCPI.SENSE(Ch).CORRection.COLlect.ECAL.SOLT2	260
SCPI.SENSE(Ch).CORRection.COLlect.ECAL.UChar	261
SCPI.SENSE(Ch).CORRection.COLlect.ECAL.UTHRu.STATE	262
SCPI.SENSE(Ch).CORRection.COLlect.METHOD.ERESponse.....	263
SCPI.SENSE(Ch).CORRection.COLlect.METHOD.RESPonse.OPEN	264
SCPI.SENSE(Ch).CORRection.COLlect.METHOD.RESPonse.SHORT.....	265

SCPI.SENSe(<i>Ch</i>).CORRection.COLLect.METHOD.RESPonse.THRU	266
SCPI.SENSe(<i>Ch</i>).CORRection.COLLect.METHOD.SOLT1	267
SCPI.SENSe(<i>Ch</i>).CORRection.COLLect.METHOD.SOLT2	268
SCPI.SENSe(<i>Ch</i>).CORRection.COLLect.METHOD.TYPE	269
SCPI.SENSe(<i>Ch</i>).CORRection.COLLect.SAVE	270
SCPI.SENSe(<i>Ch</i>).CORRection.EXTension.PORT(<i>Pt</i>).FREQuency(<i>Ls</i>)	271
SCPI.SENSe(<i>Ch</i>).CORRection.EXTension.PORT(<i>Pt</i>).INCLUDE(<i>Ls</i>).STATe	272
SCPI.SENSe(<i>Ch</i>).CORRection.EXTension.PORT(<i>Pt</i>).LDC	273
SCPI.SENSe(<i>Ch</i>).CORRection.EXTension.PORT(<i>Pt</i>).LOSS(<i>Ls</i>)	274
SCPI.SENSe(<i>Ch</i>).CORRection.EXTension.PORT(<i>Pt</i>).TIME	275
SCPI.SENSe(<i>Ch</i>).CORRection.EXTension.STATE	276
SCPI.SENSe.CORRection.IMPedance.INPut.MAGNitude	277
SCPI.SENSe(<i>Ch</i>).CORRection.INFormation(<i>rPt</i> , <i>sPt</i>)	278
SCPI.SENSe(<i>Ch</i>).CORRection.OFFSet.CLEar	279
SCPI.SENSe(<i>Ch</i>).CORRection.OFFSet.COLLect.ACQuire.LOAD	280
SCPI.SENSe(<i>Ch</i>).CORRection.OFFSet.COLLect.ACQuire.OPEN	281
SCPI.SENSe(<i>Ch</i>).CORRection.OFFSet.COLLect.ACQuire.SHORT	282
SCPI.SENSe(<i>Ch</i>).CORRection.OFFSet.COLLect.ACQuire.THRU	283
SCPI.SENSe(<i>Ch</i>).CORRection.OFFSet.COLLect.ACQuire.PMETer	284
SCPI.SENSe(<i>Ch</i>).CORRection.OFFSet.COLLect.CLEar	285
SCPI.SENSe(<i>Ch</i>).CORRection.OFFSet.COLLect.METHOD.SMIX2	286
SCPI.SENSe(<i>Ch</i>).CORRection.OFFSet.COLLect.SAVE	287
SCPI.SENSe(<i>Ch</i>).CORRection.RECeiver(<i>Pt</i>).COLLect.ACQuire	288
SCPI.SENSe(<i>Ch</i>).CORRection.RECeiver(<i>Pt</i>).STATE	289
SCPI.SENSe(<i>Ch</i>).CORRection.STATE	290
SCPI.SENSe(<i>Ch</i>).CORRection.TYPE(<i>Tr</i>)	291
SCPI.SENSe(<i>Ch</i>).CORRection.TRANSform.TIME.FREQuency	292
SCPI.SENSe(<i>Ch</i>).CORRection.TRANSform.TIME.LOSS	293
SCPI.SENSe(<i>Ch</i>).CORRection.TRANSform.TIME.RVELocity	293
SCPI.SENSe(<i>Ch</i>).CORRection.TRANSform.TIME.STATE	294
SCPI.SENSe(<i>Ch</i>).DATA.CORRdata(<i>Param</i>)	295
SCPI.SENSe(<i>Ch</i>).DATA.RAWData(<i>Param</i>)	296
SCPI.SENSe(<i>Ch</i>).FREQuency.CENTer	297
SCPI.SENSe(<i>Ch</i>).FREQuency.CW	298
SCPI.SENSe(<i>Ch</i>).FREQuency.DATA	299
SCPI.SENSe(<i>Ch</i>).FREQuency.SPAN	300
SCPI.SENSe(<i>Ch</i>).FREQuency.START	301
SCPI.SENSe(<i>Ch</i>).FREQuency.STOP	302
SCPI.SENSe(<i>Ch</i>).OFFSet.ADJust.CONTinuous.PERiod	303
SCPI.SENSe(<i>Ch</i>).OFFSet.ADJust.CONTinuous.STATE	304
SCPI.SENSe(<i>Ch</i>).OFFSet.ADJust.EXECute	305
SCPI.SENSe(<i>Ch</i>).OFFSet.ADJust.PORT	305
SCPI.SENSe(<i>Ch</i>).OFFSet.ADJust.STATE	306
SCPI.SENSe(<i>Ch</i>).OFFSet.ADJust.Value	307
SCPI.SENSe(<i>Ch</i>).OFFSet.PORT(<i>Pt</i>).FREQuency.DATA	308
SCPI.SENSe(<i>Ch</i>).OFFSet.PORT(<i>Pt</i>).FREQuency.DIVisor	309
SCPI.SENSe(<i>Ch</i>).OFFSet.PORT(<i>Pt</i>).FREQuency.MULTiplier	310
SCPI.SENSe(<i>Ch</i>).OFFSet.PORT(<i>Pt</i>).FREQuency.OFFSet	311
SCPI.SENSe(<i>Ch</i>).OFFSet.PORT(<i>Pt</i>).FREQuency.START	312
SCPI.SENSe(<i>Ch</i>).OFFSet.PORT(<i>Pt</i>).FREQuency.STOP	313
SCPI.SENSe(<i>Ch</i>).OFFSet.RECeiver.FREQuency.DATA	314
SCPI.SENSe(<i>Ch</i>).OFFSet.RECeiver.FREQuency.DIVisor	315
SCPI.SENSe(<i>Ch</i>).OFFSet.RECeiver.FREQuency.MULTiplier	316
SCPI.SENSe(<i>Ch</i>).OFFSet.RECeiver.FREQuency.OFFSet	317

SCPI.SENSe(<i>Ch</i>).OFFSet.RECeiver.FREQuency.START	318
SCPI.SENSe(<i>Ch</i>).OFFSet.RECeiver.FREQuency.STOP	319
SCPI.SENSe(<i>Ch</i>).OFFSet.SOURce.FREQuency.DATA	320
SCPI.SENSe(<i>Ch</i>).OFFSet.SOURce.FREQuency.DIVisor	321
SCPI.SENSe(<i>Ch</i>).OFFSet.SOURce.FREQuency.MULTiplier	322
SCPI.SENSe(<i>Ch</i>).OFFSet.SOURce.FREQuency.OFFSet	323
SCPI.SENSe(<i>Ch</i>).OFFSet.SOURce.FREQuency.START	324
SCPI.SENSe(<i>Ch</i>).OFFSet.SOURce.FREQuency.STOP	325
SCPI.SENSe(<i>Ch</i>).OFFSet.STATE	326
SCPI.SENSe(<i>Ch</i>).OFFSet.TYPE	327
SCPI.SENSe(<i>Ch</i>).ROSCillator.SOURce	328
SCPI.SENSe(<i>Ch</i>).SEGMeNT.DATa	329
SCPI.SENSe(<i>Ch</i>).SWEep.POINT.TIME	330
SCPI.SENSe(<i>Ch</i>).SWEep.POINts	331
SCPI.SENSe(<i>Ch</i>).SWEep.TyPE	332
SCPI.SERViCe.CHANnel(1).ACTive	333
SCPI.SERViCe.CHANnel(1).COUNT	334
SCPI.SERViCe.CHANnel(<i>Ch</i>).TRACe(1).ACTIVE	334
SCPI.SERViCe.CHANnel(1).TRACe.COUNT	335
SCPI.SERViCe.CHANnel(<i>Ch</i>).TRACe(<i>Tr</i>).MARKer.ACTive	335
SCPI.SERViCe.PORT.COUNT	336
SCPI.SERViCe.SWEep.FREQency.MAXimum	336
SCPI.SERViCe.SWEep.FREQency.MINimum	337
SCPI.SERViCe.SWEep.POINts	337
SCPI.SOURce(<i>Ch</i>).POWer.CENTer	338
SCPI.SOURce(<i>Ch</i>).POWer.LEVel.IMMEDIATE.AMPLitude	339
SCPI.SOURce(<i>Ch</i>).POWer.LEVel.SLOPe.DATa	340
SCPI.SOURce(<i>Ch</i>).POWer.LEVel.SLOPe.STATE	341
SCPI.SOURce(<i>Ch</i>).POWer.PORT(<i>Pt</i>).CORRection.COLLeCT.ACQuire	342
SCPI.SOURce(<i>Ch</i>).POWer.PORT(<i>Pt</i>).CORRection.COLLeCT.TABLE. LOSS.DATa	343
SCPI.SOURce(<i>Ch</i>).POWer.PORT(<i>Pt</i>).CORRection.COLLeCT.TABLE. LOSS.STATE	344
SCPI.SOURce(<i>Ch</i>).POWer.PORT(<i>Pt</i>).CORRection.DATa	345
SCPI.SOURce(<i>Ch</i>).POWer.PORT(<i>Pt</i>).CORRection.STATE	346
SCPI.SOURce(<i>Ch</i>).POWer.PORT(1).COUPLE	347
SCPI.SOURce(<i>Ch</i>).POWer.PORT(<i>Pt</i>).LEVel.IMMEDIATE.AMPLitude	348
SCPI.SOURce(<i>Ch</i>).POWer.SPAN	349
SCPI.SOURce(<i>Ch</i>).POWer.START	350
SCPI.SOURce(<i>Ch</i>).POWer.STOP	351
SCPI.STATus.OPERation.CONDITION	352
SCPI.STATus.OPERation.ENABLE	352
SCPI.STATus.OPERation.EVENT	353
SCPI.STATus.OPERation.NTRANSITION	353
SCPI.STATus.OPERation.PTRANSITION	354
SCPI.STATus.PRESET	354
SCPI.STATus.QUESTIONable.CONDITION	355
SCPI.STATus.QUESTIONable.ENABLE	355
SCPI.STATus.QUESTIONable.EVENT	356
SCPI.STATus.QUESTIONable.LIMIT.CHANNEL(<i>Ch</i>).CONDITION	356
SCPI.STATus.QUESTIONable.LIMIT.CHANNEL(<i>Ch</i>).ENABLE	357
SCPI.STATus.QUESTIONable.LIMIT.CHANNEL(<i>Ch</i>).EVENT	357
SCPI.STATus.QUESTIONable.LIMIT.CHANNEL(<i>Ch</i>).NTRANSITION	358
SCPI.STATus.QUESTIONable.LIMIT.CHANNEL(<i>Ch</i>).PTRANSITION	359
SCPI.STATus.QUESTIONable.LIMIT.CONDITION	359
SCPI.STATus.QUESTIONable.LIMIT.ENABLE	360

SCPI.STATUs.QUESTIONable.LIMIT.EVENT	360
SCPI.STATUs.QUESTIONable.LIMIT.NTRansition	361
SCPI.STATUs.QUESTIONable.LIMIT.PTRansition	362
SCPI.STATUs.QUESTIONable.NTRansition	363
SCPI.STATUs.QUESTIONable.PTRansition	364
SCPI.STATUs.QUESTIONable.RLIMit.CHANnel(<i>Ch</i>).CONDITION	364
SCPI.STATUs.QUESTIONable.RLIMit.CHANnel(<i>Ch</i>).ENABLE	365
SCPI.STATUs.QUESTIONable.RLIMit.CHANnel(<i>Ch</i>).EVENT	365
SCPI.STATUs.QUESTIONable.RLIMit.CHANnel(<i>Ch</i>).NTRansition	366
SCPI.STATUs.QUESTIONable.RLIMit.CHANnel(<i>Ch</i>).PTRansition	367
SCPI.STATUs.QUESTIONable.RLIMit.CONDITION	367
SCPI.STATUs.QUESTIONable.RLIMit.ENABLE	368
SCPI.STATUs.QUESTIONable.RLIMit.EVENT	368
SCPI.STATUs.QUESTIONable.RLIMit.NTRansition	369
SCPI.STATUs.QUESTIONable.RLIMit.PTRansition	370
SCPI.SYSTem.BEEPer.COMplete.IMMediate	371
SCPI.SYSTem.BEEPer.COMplete.STATE	371
SCPI.SYSTem.BEEPer.WARNING.IMMediate	372
SCPI.SYSTem.BEEPer.WARNING.STATE	372
SCPI.SYSTem.COMMunicate.ECAL.TEMPerature.SENSor	373
SCPI.SYSTem.COMMunicate.ECAL.IMPedance(<i>Pt</i>)	374
SCPI.SYSTem.COMMunicate.ECAL.CHECk()	374
SCPI.SYSTem.COMMunicate.ECAL.THRU(<i>Pt1, Pt2</i>)	375
SCPI.SYSTem.CORRection.STATE	375
SCPI.SYSTem.DATE	376
SCPI.SYSTem.HIDE	376
SCPI.SYSTem.LOCal	377
SCPI.SYSTem.PRESet	377
SCPI.SYSTem.PORT.SWITchover.Delay.STATE	378
SCPI.SYSTem.REMote	379
SCPI.SYSTem.RWLock	379
SCPI.SYSTem.TEMPerature.SENSor(<i>Idx</i>)	380
SCPI.SYSTem.SHOW	380
SCPI.SYSTem.TERMinate	381
SCPI.SYSTem.TIME	381
SCPI.TRIGger.SEQuence.EXternal.Delay	382
SCPI.TRIGger.SEQuence.EXternal.POSition	383
SCPI.TRIGger.SEQuence.EXternal.SLOPe	384
SCPI.TRIGger.SEQuence.IMMediate	385
SCPI.TRIGger.SEQuence.POINT	386
SCPI.TRIGger.SEQuence.SINGLE	387
SCPI.TRIGger.SEQuence.SCOPe	388
SCPI.TRIGger.SEQuence.SOURce	389
SCPI.TRIGger.SEQuence.STATUS	390
SCPI.TRIGger.SEQuence.WAIT(STATus)	390
SCPI.TRIGger.OUTPut.FUNCTion	391
SCPI.TRIGger.OUTPut.POLarity	392
SCPI.TRIGger.OUTPut.STATE	393
16 Appendix 1. IEE488.2 Status Reporting System	394
17 Appendix 2. Error Codes	400

1 Introduction

This Manual contains information on Network Analyzer remote control and its data communication by means of user programs written with COM/DCOM technology.

COM technology is used when a user program runs together with an external measurement analyzer program on one PC. DCOM technology is used when a user program runs on a PC connected with the measurement analyzer by LAN.

Methods and techniques for writing of user programs are same for the both technologies. The only difference between the technologies is that the DCOM technology requires additional LAN setting performed by the LAN administrator.

Before reading this Manual, familiarize yourself with Analyzer Operating Manual.

2 Scope of Manual

This Manual directly applies to the PLANAR 304/1, PLANAR 804/1, S7530, S5048, C1209, C1220 models of 2-port Network Analyzer.

These models of vector network analyzers share the same executable module and the COM object name. The name of executable module is *S2VNA.exe* and the name of COM object is *S2VNA.Application*.

For example, the command for creating a COM server applies to the all models.

```
Set app = CreateObject("S2VNA.Application")
```

For the backward compatibility the old names are preserved for creating COM for each model :

```
Set app = CreateObject("Planar304.Application")
Set app = CreateObject("Planar804.Application")
Set app = CreateObject("S5048.Application")
Set app = CreateObject("S7530.Application")
```

Except where specifically noted, all commands and descriptions apply across all models covered by this document.

3 Registering COM Server

To register COM server of the analyzer run the executable module from command prompt with the `/regserver` keyword. To unregister COM server of the analyzer run the executable module from command prompt with the `/unregserver` keyword. Administrative rights is required to register/unregister COM server. Also user has ability to register COM server during the software installation procedure.

Example of the COM server registration command:

```
S2VNA.exe /regserver
```

4 COM Technology Overview

COM stands for *Component Object Model*. This programming technology was developed by Microsoft for two purposes:

- the model provides the specification for interaction of binary modules created in different programming languages;
- the model defines the interfacing between a client application and a server application running either on the same PC or on two different PCs. In the latter case, the technology has DCOM abbreviation – Distributed COM.

5 Automation Server

The network analyzer executable module contains a built-in COM server that enables other programs to access its functionality. The COM server was developed in conformity with the *COM automation* specification. COM automation is a technology allowing control over the COM server by the programs written in both traditional compiling programming languages and interpreting programming languages, such as VBScript. This enables the server applications to make their functionality accessible to many more clients.

6 Automation Controllers

Automation controllers are client programs, which use internal functionality of COM servers. Automation controller programs are developed by users for writing their own add-ons for the system.

User programs can be written in different languages:

- programming languages with built-in COM support, such as Visual Basic®, Delphi, Java;
- universal programming languages, such as C, C++;
- Microsoft Excel and Word office applications as they include built-in programming language Visual Basic for Applications®;
- program generators, such as National Analyzers LabVIEW®, MathWorks MATLAB®.

Examples written in VBA (Excel) , C++, MATLAB, Python, and other languages are available at www.coppermountaintech.com; source code of examples are also located in the Examples\COM\ folder of the application installation folder.

A Labview Driver is also included in the Labview subfolder of the Examples\COM\ folder, and can be downloaded separately from www.coppermountaintech.com. The Labview Driver contains examples of its use.

7 Local and Remote Server

The network analyzer executable module can function either as a *local* server or as a *remote* server of COM automation.

Local server runs on the same PC with the automation controller and each of the programs is executed as an individual application in a separate window. COM technology is used in this case (Figure 1).

Remote server and the automation controller run on different PCs connected by LAN. DCOM (Distributed COM) technology is used in this case (Figure 2). When using DCOM it is necessary to configure the local network by means of DCOM Windows tools.

The same automation controller is used for the both COM and DCOM technology. Some changes to the user program may be required in operators, which establish

connection with the server. Moreover, DCOM technology requires additional settings of the LAN performed by the LAN administrator.

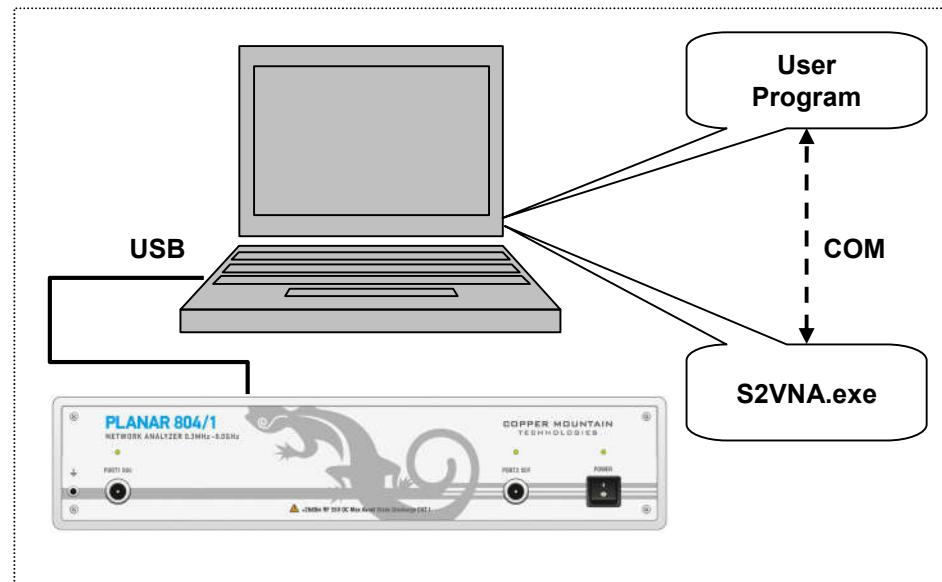


Figure 1. COM technology

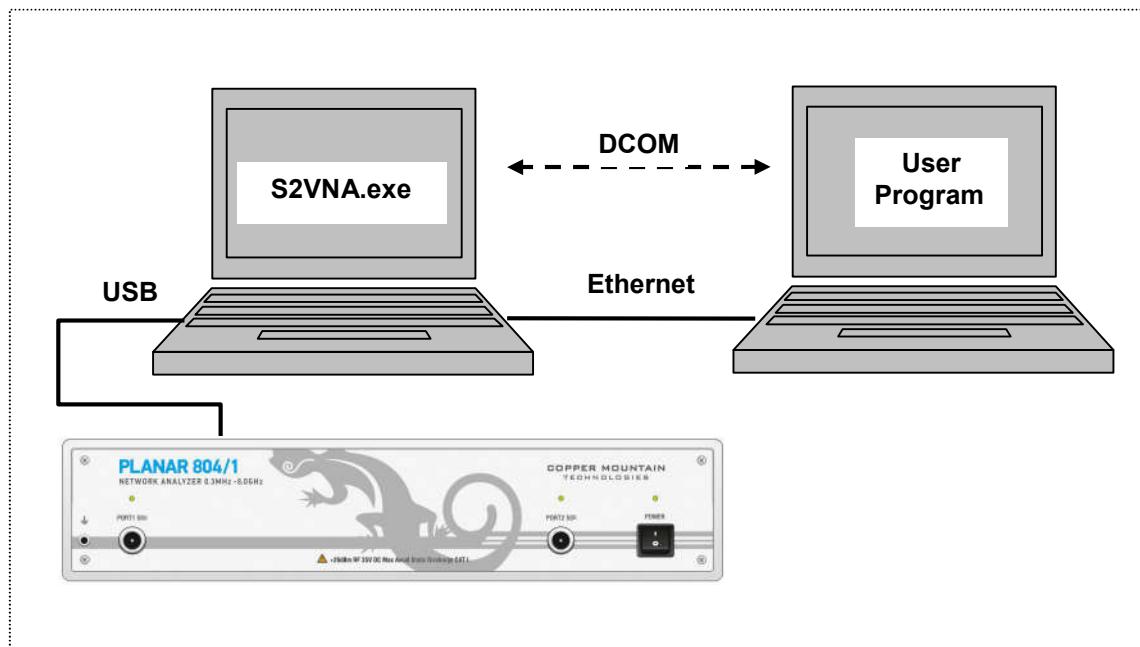


Figure 2. DCOM technology

8 Structure of COM Objects

COM server contains several *objects*, which provide different functionality of the server. The COM objects of the network analyzer executable module are organized in a hierarchical structure. Figure 3 shows the main COM objects, which comprise the first three levels of the hierarchical structure of the COM server. COM objects provide various *methods* and *properties*, which allow access to the server functions; besides, they allow access to the objects of the lower levels, which are not shown in Figure 3.

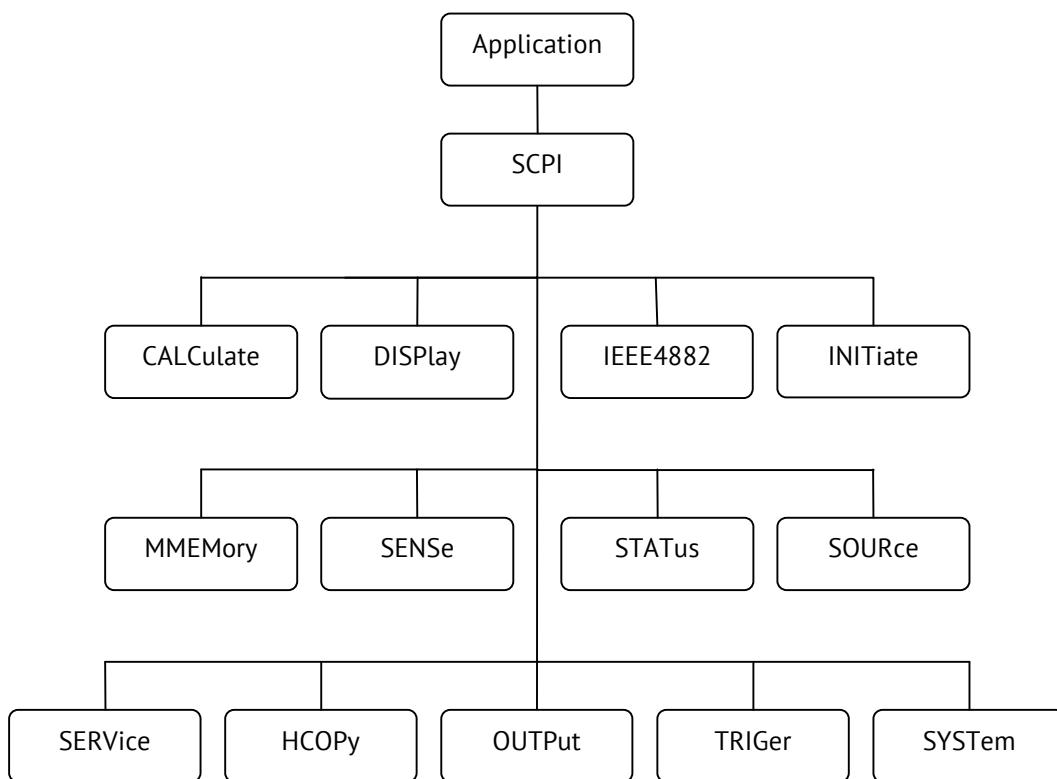


Figure 3. The structure of COM objects

The Object *Application* is in the top of the hierarchy of the COM server. Access to the lower level objects is implemented via higher level objects.

Note The hierarchy of COM objects and their names are taken from the command system used for control over the Network Analyzers via GPIB interface. GPIB communication protocol complies with the IEE488.2 standard. In accordance with this standard the SCPI programming language is used. Operators in SCPI have hierarchical chain structure, for example:

```
CALCulate:PARameter:DEFine S11
```

The same command in COM written in VB programming language is as follows:

```
app.SCPI.CALCulate.PARAMeter.DEFine = "S11"
```

9 Accessing the Application Object

To establish connection with the COM server application, create an object reference in the client program. In COM programming the object reference needs to be acquired preliminarily, to be used later to access the object functionality. To define an object in Visual Basic perform the following:

- 1) Declare a variable as an object.
- 2) Assign the object to this variable.

To declare a variable, use *Dim* operator or other declaration statement (*Public*, *Private* or *Static*). The variables used for references should of the types *Variant*, *Object*, or a type of a specific object. For example, the following three operators declare *app* variable:

```
Dim app
Dim app as Object
Dim app as S2VNA.Application
```

Use *Set* operator and *CreateObject* (*ObjectName*, *HostName*) function to assign a specific object to a variable.

<i>ObjectName</i>	Object name is always equal to “S2VNA.Application” (or “Planar804.Application”, “S5048.Application”, “S7530.Application”...)
<i>HostName</i>	Network name of the PC hosting the COM server. This parameter is not specified in case of a local server.

For example, the following operators create *Application* object and assign it to *app* variable:

```
Set app = CreateObject("S2VNA.Application")
Set app = CreateObject("S2VNA.Application", "Analyzer_Name")
Set app = CreateObject("S2VNA.Application", "192.168.1.149")
```

Note

The first form of the operator is used to create the reference to the local COM server, the second and third forms are used to create the reference to the remote DCOM server.

To allow access to the objects of a lower level of the hierarchy, these objects are specified after the reference to the higher level object and separated from it by a dot. For example:

```
Dim SystObj
Set SystObj = app.SCPI.SYSTEM
```

COM objects can have indices. For example, *CALCulate*, *INITiate*, *SENSe*, *SOURce* objects represent various aspects of the 16 measurement channels of the Analyzer. Therefore, it is necessary to write the channel index from 1 to 16 to acquire the data of these objects. For example:

```
Set SensObj1 = app.SCPI.SENSe(1)
Set SensObj2 = app.SCPI.SENSe(2)
```

Visual Basic allows omitting of such indices; in this case the indices are considered as equal to 1. For example, the following VB operators are equivalent:

```
Set SensObj = app.SCPI.SENSe(1)
Set SensObj = app.SCPI.SENSe
```

10 Object Methods

Objects have methods. Methods are actions that can be applied to objects. The object methods are specified after the object name and separated from it by a dot.

The following example shows the *PRESet* method of *SYSTem* object. This method performs setting of the Analyzer to the preset condition:

```
app.SCPI.SYSTem.PRESet
```

11 Object Properties

Along with methods, objects have properties. Properties are object characteristics that can be set or read out. The object properties are specified after the object name and separated from it by a dot.

To modify an object characteristic, write the value of the corresponding property. To define an object characteristic, read out the value of its property. The following example show the setting of the *POINts* property of *SWEp* object, i.e. the number of sweep points:

```
app.SCPI.SENSE.SWEp.POINTs = 201
```

Note

Some object properties cannot be written, and some object properties cannot be read. In such cases, the properties are indicated as “read only” or “write only”.

12 Error Handling

You can use different approaches to error handling in VB program:

- check the value of `Err.Number` variable after execution of VB operator, which contains the call to the COM server object;
- use `On Error goto` VB operator.

These approaches are represented in the examples below. The following operator causes an error in VB program as "S13" value of the `DEFine` property is incorrect.

```
app.SCPI.PARameter.DEFine = "S13"
```

In the first example, the value of the `Err.Number` variable is checked after execution of the VB operator, which contains the call to the COM server object. `On Error Resume Next` directive instructs VB not to interrupt the program execution when the error is detected but to pass control to the next operator in natural order.

```
Dim app
Public Sub HandleError1()
Set app = CreateObject("S2VNA.Application")
On Error Resume Next
app.SCPI.PARameter.DEFine = "S13"
If Err.Number <> 0 Then
    Msg = "Error # " & Str(Err.Number) & " was generated by " &_
        Err.Source & Chr(13) & Err.Description
    MsgBox Msg,, "Error"
End If
...
End Sub
```

In the second example, *On Error GoTo ErrHandler* directive instructs VB to interrupt the program execution when the error is detected and to pass control to *ErrHandler* label.

```
Dim app
Public Sub HandleError2()
Set app = CreateObject("S2VNA.Applcation")
On Error GoTo ErrHandler
app.SCPI.PARameter.DEFine = "S13"
...
Exit Sub
ErrorHandler:
Msg = "Error # " & Str(Err.Number) & " was generated by " &
      Err.Source & Chr(13) & Err.Description
MsgBox Msg,, "Error"
End Sub
```

13 COM Automation Data Types

In COM automation, there are the following data types, which can be used for client-to-server communication:

Long	32-bit signed integer, value range from -2147483648 to 2147483647
Double	64-bit double-precision floating point, value range from -1.79769313486232E308 to -4.94065645841247E-324 for negative values, and from 4.94065645841247E-324 to 1.79769313486232E308 for positive values
Boolean	16-bit integer, two values 0 – <i>false</i> , 1 – <i>true</i>
String	Variable-length string
Variant	Can be either a value of arbitrary type or an array of values of arbitrary type. In this case, the term “arbitrary type” means any allowed type of COM automation. A variable contains information about its type and array size (if it is an array). It is used for communication of data arrays between a client and a server.

14 Measurement Data Arrays

Measurement data can be either complex values or real values. This depends on the format selected by the user. For example, the data is real in logarithmic magnitude format and the data is complex in polar format.

The measurement data is transferred in a *Variant* type variable, which represents an array of *Double* type. To transfer one complex measurement, two adjacent array cells are used. To transfer one real measurement two adjacent array cells are used as well but the second cell is always equal to 0. Thus, measurement data array size is a double number of the measurement points.

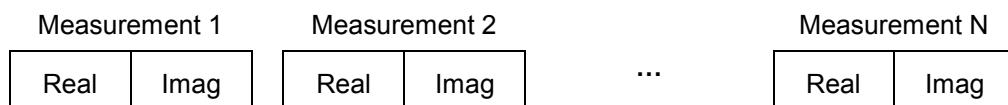


Figure 4. Array of complex measurements

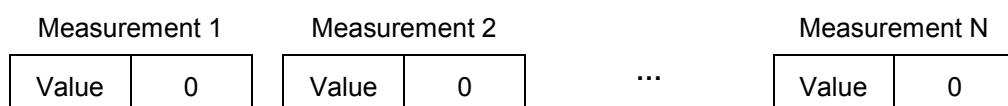


Figure 5. Array of real measurements

15 COM Server Commands

NAME

<i>Object Type</i>	Property (read only)
<i>Data Type</i>	String
<i>Description</i>	Analyzer information string. String format: manufacturer, model, serial number, number of software version / number of hardware revision.
<i>Range</i>	up to 40 characters
<i>Syntax</i>	Dim <i>ID</i> As String <i>ID</i> = <i>app.NAME</i>
<i>Reply</i>	CMT, S5048, 15097610, 15.4/02
<i>Equivalent Softkeys</i>	None

Ready

<i>Object Type</i>	Property (read only)
<i>Data Type</i>	Boolean
<i>Description</i>	Ready state of the analyzer. Reads out the <i>True</i> value after successful completion of the boot process (about 10 sec). The Analyzer must be connected to PC by a USB cable.
<i>Syntax</i>	Dim <i>State</i> as Boolean <i>State</i> = <i>app</i> .Ready
<i>Equivalent Softkeys</i>	None

SCPI.ABORT

<i>Object Type</i>	Method
<i>Description</i>	Aborts the sweep and switches the triggers of all the channels to <i>Hold</i> . The channels in <i>Continuous</i> trigger mode switch to waiting for a trigger. If the trigger source is set to <i>Internal</i> , the channel from waiting for a trigger turns to a new sweep.
<i>Syntax</i>	<i>app</i> .SCPI.ABORT
<i>Equivalent Softkeys</i>	Stimulus > Trigger > Restart

**SCPI.CALCulate(*Ch*).FSIMulator.SENDed.DEEMbed.PORT(*Pt*).
USER.FILename**

<i>Object Type</i>	Property (read/write)
<i>Data Type</i>	String
<i>Target</i>	Port <i>Pt</i> of channel <i>Ch</i> , <i>Ch</i> : channel number 1–16 (see Table 1 on page 25) <i>Pt</i> : port number 1–2 (see Table 2 on page 25)
<i>Description</i>	De-embedding function file name (*.s2p). The file contains the circuit S-parameters in Touchstone format.
<i>Range</i>	up to 256 characters
<i>Preset Value</i>	""
<i>Syntax</i>	<pre>Dim File As String File = app.SCPI.CALCulate(<i>Ch</i>).FSIMulator.SENDed.DEEMbed.PORT(<i>Pt</i>).USER.FILename app.SCPI.CALCulate(<i>Ch</i>).FSIMulator.SENDed.DEEMbed.PORT(<i>Pt</i>).USER.FILename = "network.s2p"</pre>
<i>Notes</i>	If the full path to the file is not specified, the \FixtureSim subdirectory of the main directory will be searched for the file.
<i>Equivalent Softkeys</i>	Analysis > Fixture Simulator > De-Embedding > S-parameters File

Table 1. Ch: Channel Number

<i>Data Type</i>	Long
<i>Description</i>	Channel number.
<i>Range</i>	from 1 to 16
<i>Out of Range</i>	An error occurs. Error code: 201.
<i>Notes</i>	If the channel number is not specified, it is taken as equal to 1.

Table 2. Pt: Port Number

<i>Data Type</i>	Long
<i>Description</i>	Port number.
<i>Range</i>	from 1 to 2
<i>Out of Range</i>	An error occurs. Error code: 114.
<i>Notes</i>	If the port number is not specified, it is taken as equal to 1.

SCPI.CALCulate(*Ch*).FSIMulator.SENDed.DEEMbed.PORT(*Pt*).STATe

<i>Object Type</i>	Property (read/write)
<i>Data Type</i>	Boolean
	Port <i>Pt</i> of channel <i>Ch</i> , <i>Ch</i> : channel number 1–16 (see Table 1 on page 25) <i>Pt</i> : port number 1–2 (see Table 2 on page 25)
<i>Description</i>	The ON/OFF state of the e-embedding function.
<i>Allowable Values</i>	True: De-embedding function ON False: De-embedding function OFF
<i>Preset Value</i>	False
<i>Syntax</i>	<p>Dim <i>Status</i> As Boolean</p> <pre><i>Status</i> = <i>app</i>.SCPI.CALCulate(<i>Ch</i>).FSIMulator.SENDed.DEEMbed.PORT(<i>Pt</i>).STATe <i>app</i>.SCPI.CALCulate(<i>Ch</i>).FSIMulator.SENDed.DEEMbed.PORT(<i>Pt</i>).STATe = True</pre>
<i>Equivalent Softkeys</i>	Analysis > Fixture Simulator > De-Embedding > Port n

SCPI.CALCulate(*Ch*).FSIMulator.SENDed.DEEMbed.STATE

<i>Object Type</i>	Property (read/write)
<i>Data Type</i>	Boolean
	Port <i>Pt</i> of channel <i>Ch</i> , <i>Ch</i> : channel number 1–16 (see Table 1 on page 26) <i>Pt</i> : port number 1–4 (see Table 2 on page 26)
<i>Description</i>	The ON/OFF state of the e-embedding s2p function.
<i>Allowable Values</i>	True: De-embedding function ON False: De-embedding function OFF
<i>Preset Value</i>	False
<i>Syntax</i>	Dim <i>Status</i> As Boolean <i>Status</i> = app.SCPI.CALCulate(<i>Ch</i>).FSIMulator.SENDed.DEEMbed.STATE app.SCPI.CALCulate(<i>Ch</i>).FSIMulator.SENDed.DEEMbed.STATE = True
<i>Equivalent Softkeys</i>	Analysis > Fixture Simulator > De-Embedding > De-Embedding

**SCPI.CALCulate(*Ch*).FSIMulator.SENDed.PMCircuit.PORT(*Pt*).
USER.FILename**

<i>Object Type</i>	Property (read/write)
<i>Data Type</i>	String
<i>Target</i>	Port <i>Pt</i> of channel <i>Ch</i> , <i>Ch</i> : channel number 1–16 (see Table 1 on page 25) <i>Pt</i> : port number 1–2 (see Table 2 on page 25)
<i>Description</i>	Embedding function file name (*.s2p). The file contains the circuit S-parameters in Touchstone format.
<i>Range</i>	up to 256 characters
<i>Preset Value</i>	""
<i>Syntax</i>	<pre>Dim File As String File = app.SCPI.CALCulate(<i>Ch</i>).FSIMulator.SENDed.PMCircuit.PORT(<i>Pt</i>).USER.FILename app.SCPI.CALCulate(<i>Ch</i>).FSIMulator.SENDed.PMCircuit.PORT(<i>Pt</i>).USER.FILename = "network.s2p"</pre>
<i>Notes</i>	If the full path to the file is not specified, the \FixtureSim subdirectory of the main directory will be searched for the file.
<i>Equivalent Softkeys</i>	Analysis > Fixture Simulator > Embedding > Port n

SCPI.CALCulate(*Ch*).FSIMulator.SENDed.PMCircuit.PORT(*Pt*).STATe

<i>Object Type</i>	Property (read/write)
<i>Data Type</i>	Boolean
<i>Target</i>	Port <i>Pt</i> of channel <i>Ch</i> , <i>Ch</i> : channel number 1–16 (see Table 1 on page 25) <i>Pt</i> : port number 1–2 (see Table 2 on page 25)
<i>Description</i>	The ON/OFF state of the embedding function.
<i>Allowable Values</i>	True: Embedding function ON False: Embedding function OFF
<i>Preset Value</i>	False
<i>Syntax</i>	<p>Dim <i>Status</i> As Boolean</p> <pre><i>Status</i> = <i>app</i>.SCPI.CALCulate(<i>Ch</i>).FSIMulator.SENDed.DEEMbed.PORT(<i>Pt</i>).STATe <i>app</i>.SCPI.CALCulate(<i>Ch</i>).FSIMulator.SENDed.DEEMbed.PORT(<i>Pt</i>).STATe = True</pre>
<i>Equivalent Softkeys</i>	Analysis > Fixture Simulator > Embedding > Port n

SCPI.CALCulate(*Ch*).FSIMulator.SENDed.PMCircuit.STATE

<i>Object Type</i>	Property (read/write)
<i>Data Type</i>	Boolean
<i>Target</i>	Port <i>Pt</i> of channel <i>Ch</i> , <i>Ch</i> : channel number 1–16 (see Table 1 on page 26) <i>Pt</i> : port number 1–4 (see Table 2 on page 26)
<i>Description</i>	The ON/OFF state of the embedding s2p function.
<i>Allowable Values</i>	True: Embedding function ON False: Embedding function OFF
<i>Preset Value</i>	False
<i>Syntax</i>	Dim <i>Status</i> As Boolean <i>Status</i> = app.SCPI.CALCulate(<i>Ch</i>).FSIMulator.SENDed.DEEMbed.PORT(<i>Pt</i>).STATE app.SCPI.CALCulate(<i>Ch</i>).FSIMulator.SENDed.DEEMbed.PORT(<i>Pt</i>).STATE = True
<i>Equivalent Softkeys</i>	Analysis > Fixture Simulator > Embedding > Embedding

SCPI.CALCulate(*Ch*).FSIMulator.SENDed.ZCONversion.PORT(*Pt*).Z0.R

<i>Object Type</i>	Property (read/write)
<i>Data Type</i>	Double
<i>Target</i>	Port <i>Pt</i> of channel <i>Ch</i> , <i>Ch</i> : channel number 1–16 (see Table 1 on page 25) <i>Pt</i> : port number 1–2 (see Table 2 on page 25)
<i>Description</i>	The value of the impedance for port impedance conversion function.
<i>Range</i>	from 1e–6 to 1e6
<i>Out of Range</i>	Sets the value of the limit, which is closer to the specified value.
<i>Preset Value</i>	50
<i>Unit</i>	Ω (Ohm)
<i>Syntax</i>	<p>Dim <i>Value</i> As Double</p> <pre>Value = app.SCPI.CALCulate(<i>Ch</i>).FSIMulator.SENDed.ZCONversion.PORT(<i>Pt</i>).Z0.R app.SCPI.CALCulate(<i>Ch</i>).FSIMulator.SENDed.ZCONversion.PORT(<i>Pt</i>).Z0.R = 75</pre>
<i>Equivalent Softkeys</i>	Analysis > Fixture Simulator > Port Z Conversion > Port n Z0

SCPI.CALCulate(*Ch*).FSIMulator.SENDed.ZCONversion.STATE

<i>Object Type</i>	Property (read/write)
<i>Data Type</i>	Boolean
<i>Target</i>	Channel <i>Ch</i> , <i>Ch</i> : channel number 1–16 (see Table 1 on page 25)
<i>Description</i>	The ON/OFF state of the port impedance conversion function.
<i>Allowable Values</i>	True: Port Z conversion function ON False: Port Z conversion function OFF
<i>Preset Value</i>	False
<i>Syntax</i>	<p>Dim <i>Status</i> As Boolean</p> <pre><i>Status</i> = <i>app</i>.SCPI.CALCulate(<i>Ch</i>).FSIMulator.SENDed.ZCONversion.STATE <i>app</i>.SCPI.CALCulate(<i>Ch</i>).FSIMulator.SENDed.ZCONversion.STATE = True</pre>
<i>Equivalent Softkeys</i>	Analysis > Fixture Simulator > Port Z Conversion > Port Z Conversion

SCPI.CALCulate(*Ch*).FSIMulator.STATE

<i>Object Type</i>	Property (read/write)
<i>Data Type</i>	Boolean
<i>Target</i>	Channel <i>Ch</i> , <i>Ch</i> : channel number 1–16 (see Table 1 on page 26)
<i>Description</i>	The ON/OFF state of the fixture simulator function of specified channel.
<i>Allowable Values</i>	True: fixture simulator function ON False: fixture simulator function OFF
<i>Preset Value</i>	False
<i>Syntax</i>	<p>Dim <i>Status</i> As Boolean</p> <p><i>Status</i> = app.SCPI.CALCulate(<i>Ch</i>).FSIMulator.STATE</p> <p>app.SCPI.CALCulate(<i>Ch</i>).FSIMulator.STATE = True</p>
<i>Equivalent Softkeys</i>	Analysis > Fixture Simulator > Fixture Simulator

SCPI.CALCulate(*Ch*).PARameter(1).COUNT

<i>Object Type</i>	Property (read/write)
<i>Data Type</i>	Long
<i>Target</i>	Channel <i>Ch</i> , <i>Ch</i> : channel number 1–16 (see Table 1 on page 25)
<i>Description</i>	The number of traces in the channel.
<i>Range</i>	from 1 to 16
<i>Out of Range</i>	Sets the value of the limit, which is closer to the specified value.
<i>Preset Value</i>	1
<i>Syntax</i>	<pre>Dim TraceNum As Long TraceNum = app.SCPI.CALCulate(<i>Ch</i>).PARameter.COUNT app.SCPI.CALCulate(<i>Ch</i>).PARameter.COUNT = 2</pre>
<i>Warning</i>	Object <i>PARameter</i> has an index. In Visual Basic the index is 1 by default when it is omitted. The index can not be omitted in other programming languages.
<i>Equivalent Softkeys</i>	Display > Num of Traces

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

<i>Object Type</i>	Property (read/write)
<i>Data Type</i>	String
<i>Target</i>	Trace <i>Tr</i> of channel <i>Ch</i> , <i>Ch</i> : channel number 1–16 (see Table 1 on page 25) <i>Tr</i> : trace number 1–16 (see Table 3 on page 33)
<i>Description</i>	The measurement parameter of the trace.
<i>Allowable Values</i>	"S11" : S11 parameter "S21" : S21 parameter "S12" : S12 parameter "S22" : S22 parameter "A" : Receiver A "B" : Receiver B "R1" : Receiver R1 "R2" : Receiver R2
<i>Out of Range</i>	An error occurs. Error code: 208.
<i>Preset Value</i>	Depends on the trace number. Tr 1, Tr 5, Tr 9, Tr 13: "S11" Tr 2, Tr 6, Tr 10, Tr 14: "S21" Tr 3, Tr 7, Tr 11, Tr 15: "S12" Tr 4, Tr 8, Tr 12, Tr 16: "S22"
<i>Syntax</i>	Dim Meas As String <code>Meas = app.SCPI.CALCulate(<i>Ch</i>).PARameter(<i>Tr</i>).DEFine</code> <code>app.SCPI.CALCulate(<i>Ch</i>).PARameter(<i>Tr</i>).DEFine = "S11"</code>
<i>Equivalent Softkeys</i>	Measurement > Measurement > S11 S21 S12 S22 Measurement > Measurement > Absolute > { A(1) B(1) R1(1) A(2) B(2) R2(2) }

Table 3. Tr: Trace Number

<i>Data Type</i>	Long
<i>Description</i>	Trace number
<i>Range</i>	from 1 to 16
<i>Out of Range</i>	An error occurs. Error code: 202.
<i>Notes</i>	If the trace number is not specified, it is taken as equal to 1.

SCPI.CALCulate(*Ch*).PARameter(*Tr*).SElect

<i>Object Type</i>	Method
<i>Target</i>	Trace <i>Tr</i> of channel <i>Ch</i> , <i>Ch</i> : channel number 1–16 (see Table 1 on page 25) <i>Tr</i> : trace number 1–16 (see Table 3 on page 33)
<i>Description</i>	Sets the active channel and trace.
<i>Syntax</i>	<i>app</i> .SCPI.CALCulate(<i>Ch</i>).PARameter(<i>Tr</i>).SElect
<i>Notes</i>	If the channel number is greater than the number of the channels displayed, an error occurs and the command is ignored. If the trace number is greater than the number of the traces displayed in the channel, an error occurs and the command is ignored.
<i>Equivalent Softkeys</i>	Display > Active Trace/Channel > Active Channel Display > Active Trace/Channel > Active Trace

SCPI.CALCulate(*Ch*).PARameter(*Tr*).SPORT

<i>Object Type</i>	Property (read/write)
<i>Data Type</i>	Long
<i>Target</i>	Trace <i>Tr</i> of channel <i>Ch</i> , <i>Ch</i> : channel number 1–16 (see Table 1 on page 25) <i>Tr</i> : trace number 1–16 (see Table 3 on page 33)
<i>Description</i>	The number of the source port for absolute measurements.
<i>Range</i>	from 1 to 2
<i>Out of Range</i>	An error occurs. Error code: 208.
<i>Preset Value</i>	1
<i>Syntax</i>	<p>Dim <i>StimPort</i> As Long</p> <p><i>StimPort</i> = app.SCPI.CALCulate(<i>Ch</i>).PARameter(<i>Tr</i>).SPORT</p> <p>app.SCPI.CALCulate(<i>Ch</i>).PARameter(<i>Tr</i>).SPORT = 1</p>
<i>Equivalent Softkeys</i>	Measurement > Measurement > Absolute > A(1) B(1) R1(1) A(2) B(2) R2(2)

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

<i>Object Type</i>	Property (read/write)
<i>Data Type</i>	String
<i>Target</i>	The active trace of channel <i>Ch</i> , <i>Ch</i> : channel number 1–16 (see Table 1 on page 25)
<i>Description</i>	The S-parameter conversion function type.
<i>Range</i>	<p>"ZREFlection" : Reflection equivalent impedance</p> <p>"ZTRansmit" : Transmission equivalent impedance</p> <p>"YREFlection" : Reflection equivalent admittance</p> <p>"YTRansmit" : Transmission equivalent admittance</p> <p>"INVersion" : Inverse S-parameter</p> <p>"ZTSHunt" : Shunt equivalent impedance</p> <p>"YTSHunt" : Shunt equivalent admittance</p> <p>"CONJugation" : S-parameter conjugate</p>
<i>Notes</i>	The short form of the parameter is indicated by upper case letters. There is no distinction between upper and lower case letters when the property is written. When the property is read out, the short format is indicated by upper case letters.
<i>Out of Range</i>	An error occurs. Error code: 217.
<i>Preset Value</i>	"ZREF"
<i>Syntax</i>	<pre>Dim Param As String Param = app.SCPI.CALCulate(<i>Ch</i>).SELected.CONVersion.FUNCtion app.SCPI.CALCulate(<i>Ch</i>).SELected.CONVersion.FUNCtion = "ZTR"</pre>
<i>Equivalent Softkeys</i>	Analysis > Conversion > Zr Zt Yr Yt 1/S Z Trans-Shunt Y Trans-Shunt Conjugation

SCPI.CALCulate(*Ch*).SELected.CONVersion.STATE

<i>Object Type</i>	Property (read/write)
<i>Data Type</i>	Boolean
<i>Target</i>	The active trace of channel <i>Ch</i> , <i>Ch</i> : channel number 1–16 (see Table 1 on page 25)
<i>Description</i>	The ON/OFF state of the S-parameter conversion function.
<i>Allowable Values</i>	True: S-parameter conversion function ON False: S-parameter conversion function OFF
<i>Preset Value</i>	False
<i>Syntax</i>	Dim <i>Status</i> As Boolean <i>Status</i> = <i>app.SCPI.CALCulate(Ch).SELected.CONVersion.STATE</i> <i>app.SCPI.CALCulate(Ch).SELected.CONVersion.STATE</i> = true
<i>Equivalent Softkeys</i>	Analysis > Conversion > Conversion

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

<i>Object Type</i>	Property (read/write)
<i>Data Type</i>	Double
<i>Target</i>	The active trace of channel <i>Ch</i> , <i>Ch</i> : channel number 1–16 (see Table 1 on page 25)
<i>Description</i>	The value of the electrical delay.
<i>Range</i>	from –10 to 10
<i>Out of Range</i>	Sets the value of the limit, which is closer to the specified value.
<i>Preset Value</i>	0
<i>Unit</i>	s (second)
<i>Syntax</i>	<p>Dim Value As Double</p> <p><i>Value</i> = <i>app</i>.SCPI.CALCulate(<i>Ch</i>).SELected.CORRection.EDELay.TIME</p> <p><i>app</i>.SCPI.CALCulate(<i>Ch</i>).SELected.CORRection.EDELay.TIME = 1e–9</p>
<i>Equivalent Softkeys</i>	Scale > Electrical Delay

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

<i>Object Type</i>	Property (read/write)
<i>Data Type</i>	Double
<i>Target</i>	The active trace of channel <i>Ch</i> , <i>Ch</i> : channel number 1–16 (see Table 1 on page 25)
<i>Description</i>	The value of the phase offset.
<i>Range</i>	from –360 to 360
<i>Out of Range</i>	Sets the value of the limit, which is closer to the specified value.
<i>Preset Value</i>	0
<i>Unit</i>	° (degree)
<i>Syntax</i>	<p>Dim Value As Double</p> <p><i>Value</i> = <i>app</i>.SCPI.CALCulate(<i>Ch</i>).SELected.CORRection.OFFSet.PHASe</p> <p><i>app</i>.SCPI.CALCulate(<i>Ch</i>).SELected.CORRection.OFFSet.PHASe = 360</p>
<i>Equivalent Softkeys</i>	Scale > Phase Offset

SCPI.CALCulate(*Ch*).SELected.CORRection.STATUS

<i>Object Type</i>	Property (read only)																
<i>Data Type</i>	String																
<i>Target</i>	The active trace of channel <i>Ch</i> , <i>Ch</i> : channel number 1–16 (see Table 1 on page 25)																
<i>Description</i>	The active trace correction status.																
<i>Range</i>	<p>If active trace represents a S-parameter:</p> <table> <tr> <td>"COR"</td> <td>: Correction is enabled.</td> </tr> <tr> <td>"C?"</td> <td>: Correction is enabled. Interpolation is applied.</td> </tr> <tr> <td>"C!"</td> <td>: Correction is enabled. Extrapolation is applied.</td> </tr> <tr> <td>""</td> <td>: Correction is disabled.</td> </tr> </table> <p>If active trace represents an Absolute measurement:</p> <table> <tr> <td>"RC"</td> <td>: Correction is enabled.</td> </tr> <tr> <td>"RC?"</td> <td>: Correction is enabled. Interpolation is applied.</td> </tr> <tr> <td>"RC!"</td> <td>: Correction is enabled. Extrapolation is applied.</td> </tr> <tr> <td>""</td> <td>: Correction is disabled.</td> </tr> </table>	"COR"	: Correction is enabled.	"C?"	: Correction is enabled. Interpolation is applied.	"C!"	: Correction is enabled. Extrapolation is applied.	""	: Correction is disabled.	"RC"	: Correction is enabled.	"RC?"	: Correction is enabled. Interpolation is applied.	"RC!"	: Correction is enabled. Extrapolation is applied.	""	: Correction is disabled.
"COR"	: Correction is enabled.																
"C?"	: Correction is enabled. Interpolation is applied.																
"C!"	: Correction is enabled. Extrapolation is applied.																
""	: Correction is disabled.																
"RC"	: Correction is enabled.																
"RC?"	: Correction is enabled. Interpolation is applied.																
"RC!"	: Correction is enabled. Extrapolation is applied.																
""	: Correction is disabled.																
<i>Syntax</i>	<pre>Dim Param As String Param = app.SCPI.CALCulate(<i>Ch</i>).SELected.CORRection.STATUS</pre>																
<i>Equivalent Softkeys</i>	None																

SCPI.CALCulate(*Ch*).SELected.DATA.FDATA

<i>Object Type</i>	Property (read/write)				
<i>Data Type</i>	Variant (array of double)				
<i>Target</i>	The active trace of channel <i>Ch</i> , <i>Ch</i> : channel number 1–16 (see Table 1 on page 25)				
<i>Description</i>	<p>The formatted data array. The array elements contain measurements in the current format, for example, in logarithmic magnitude format (Log Mag). Also, see section “Measurement Data Arrays” on page 19.</p> <p>The array size is $2N$, where N is the number of measurement points.</p> <p>For the n-th point, where n from 1 to N:</p> <table style="margin-left: 40px;"> <tr> <td><i>Data</i>($2n-2$)</td> <td>real number in rectangular format, real part in polar and Smith chart formats;</td> </tr> <tr> <td><i>Data</i>($2n-1$)</td> <td>0 in rectangular format, imaginary part in polar and Smith chart formats.</td> </tr> </table>	<i>Data</i> ($2n-2$)	real number in rectangular format, real part in polar and Smith chart formats;	<i>Data</i> ($2n-1$)	0 in rectangular format, imaginary part in polar and Smith chart formats.
<i>Data</i> ($2n-2$)	real number in rectangular format, real part in polar and Smith chart formats;				
<i>Data</i> ($2n-1$)	0 in rectangular format, imaginary part in polar and Smith chart formats.				
<i>Syntax</i>	<pre>Dim <i>Data</i> As Variant <i>Data</i> = <i>app</i>.SCPI.CALCulate(<i>Ch</i>).SELected.DATA.FDATA <i>app</i>.SCPI.CALCulate(<i>Ch</i>).SELected.DATA.FDATA = <i>Data</i></pre>				
<i>Equivalent Softkeys</i>	None				

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

<i>Object Type</i>	Property (read/write)
<i>Data Type</i>	Variant (array of double)
<i>Target</i>	The active trace of channel <i>Ch</i> , <i>Ch</i> : channel number 1–16 (see Table 1 on page 25)
<i>Description</i>	<p>The formatted memory array. The array elements contain saved measurements in the current format, for example, in logarithmic magnitude format (Log Mag). Also, see section “Measurement Data Arrays” on page 19.</p> <p>The array size is 2N, where N is the number of measurement points.</p> <p>For the n-th point, where n from 1 to N:</p> <ul style="list-style-type: none"> <i>Data</i>(2n-2) real number in rectangular format, real part in polar and Smith chart formats; <i>Data</i>(2n-1) 0 in rectangular format, imaginary part in polar and Smith chart formats.
<i>Syntax</i>	<pre>Dim <i>Data</i> As Variant <i>Data</i> = <i>app</i>.SCPI.CALCulate(<i>Ch</i>).SELected.DATA.FMEMory <i>app</i>.SCPI.CALCulate(<i>Ch</i>).SELected.DATA.FMEMory = <i>Data</i></pre>
<i>Equivalent Softkeys</i>	None

SCPI.CALCulate(*Ch*).SELected.DATA.SDAta

<i>Object Type</i>	Property (read/write)
<i>Data Type</i>	Variant (array of double)
<i>Target</i>	The active trace of channel <i>Ch</i> , <i>Ch</i> : channel number 1–16 (see Table 1 on page 25)
<i>Description</i>	<p>The corrected data array. The corrected measurements are complex numbers. Also, see section “Measurement Data Arrays” on page 19.</p> <p>The array size is 2N, where N is the number of measurement points.</p> <p>For the n-th point, where n from 1 to N:</p> <ul style="list-style-type: none"> <i>Data</i>(2n–2) the real part of corrected measurement; <i>Data</i>(2n–1) the imaginary part of corrected measurement.
<i>Syntax</i>	<pre>Dim <i>Data</i> As Variant <i>Data</i> = <i>app</i>.SCPI.CALCulate(<i>Ch</i>).SELected.DATA.SDAta <i>app</i>.SCPI.CALCulate(<i>Ch</i>).SELected.DATA.SDAta = <i>Data</i></pre>
<i>Equivalent Softkeys</i>	None

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

<i>Object Type</i>	Property (read/write)
<i>Data Type</i>	Variant (array of double)
<i>Target</i>	The active trace of channel <i>Ch</i> , <i>Ch</i> : channel number 1–16 (see Table 1 on page 25)
<i>Description</i>	<p>The corrected memory array. The corrected measurements are complex numbers. Also, see section “Measurement Data Arrays” on page 19.</p> <p>The array size is 2N, where N is the number of measurement points.</p> <p>For the n-th point, where n from 1 to N:</p> <ul style="list-style-type: none"> <i>Data</i>(2n–2) the real part of corrected measurement memory; <i>Data</i>(2n–1) the imaginary part of corrected measurement memory.
<i>Syntax</i>	<pre>Dim <i>Data</i> As Variant <i>Data</i> = <i>app</i>.SCPI.CALCulate(<i>Ch</i>).SELected.DATA.SMEMory <i>app</i>.SCPI.CALCulate(<i>Ch</i>).SELected.DATA.SMEMory = <i>Data</i></pre>
<i>Equivalent Softkeys</i>	None

SCPI.CALCulate(*Ch*).SELected.FILTter.GATE.TIME.CENTER

<i>Object Type</i>	Property (read/write)
<i>Data Type</i>	Double
<i>Target</i>	The active trace of channel <i>Ch</i> , <i>Ch</i> : channel number 1–16 (see Table 1 on page 25)
<i>Description</i>	The gate center value of the gating function.
<i>Range</i>	Varies depending on the frequency span and the number of points.
<i>Out of Range</i>	Sets the value of the limit, which is closer to the specified value.
<i>Preset Value</i>	0
<i>Unit</i>	s (second)
<i>Syntax</i>	<p>Dim Value As Double</p> <p><i>Value</i> = <i>app</i>.SCPI.CALCulate(<i>Ch</i>).SELected.FILTter.GATE.TIME.CENTER</p> <p><i>app</i>.SCPI.CALCulate(<i>Ch</i>).SELected.FILTter.GATE.TIME.CENTER = 1e-8</p>
<i>Equivalent Softkeys</i>	Analysis > Gating > Center

SCPI.CALCulate(*Ch*).SELected.FILTter.GATE.TIME.SHAPe

<i>Object Type</i>	Property (read/write)
<i>Data Type</i>	String
<i>Target</i>	The active trace of channel <i>Ch</i> , <i>Ch</i> : channel number 1–16 (see Table 1 on page 25)
<i>Description</i>	The gate shape of the gating function.
<i>Range</i>	"MAXimum" : Maximum shape "WIDE" : Wide shape "NORMal" : Normal shape "MINimum" : Minimum shape
<i>Notes</i>	The short form of the parameter is indicated by upper case letters. There is no distinction between upper and lower case letters when the property is written. When the property is read out, the short format is indicated by upper case letters.
<i>Out of Range</i>	An error occurs. Error code: 218.
<i>Preset Value</i>	"NORM"
<i>Syntax</i>	Dim <i>Param</i> As String <i>Param</i> = <i>app</i> .SCPI.CALCulate(<i>Ch</i>).SELected.FILTter.GATE.TIME.SHAPe <i>app</i> .SCPI.CALCulate(<i>Ch</i>).SELected.FILTter.GATE.TIME.SHAPe = "MAX"
<i>Equivalent Softkeys</i>	Analysis > Gating > Shape > Maximum Wide Normal Minimum

SCPI.CALCulate(*Ch*).SELected.FILTter.GATE.TIME.SPAN

<i>Object Type</i>	Property (read/write)
<i>Data Type</i>	Double
<i>Target</i>	The active trace of channel <i>Ch</i> , <i>Ch</i> : channel number 1–16 (see Table 1 on page 25)
<i>Description</i>	The gate span value of the gating function.
<i>Range</i>	Varies depending on the frequency span and the number of points.
<i>Out of Range</i>	Sets the value of the limit, which is closer to the specified value.
<i>Preset Value</i>	2e-8
<i>Unit</i>	s (second)
<i>Syntax</i>	<p>Dim Value As Double</p> <p><i>Value</i> = <i>app</i>.SCPI.CALCulate(<i>Ch</i>).SELected.FILTter.GATE.TIME.SPAN</p> <p><i>app</i>.SCPI.CALCulate(<i>Ch</i>).SELected.FILTter.GATE.TIME.SPAN = 1e-8</p>
<i>Equivalent Softkeys</i>	Analysis > Gating > Span

SCPI.CALCulate(*Ch*).SELected.FILTter.GATE.TIME.START

<i>Object Type</i>	Property (read/write)
<i>Data Type</i>	Double
<i>Target</i>	The active trace of channel <i>Ch</i> , <i>Ch</i> : channel number 1–16 (see Table 1 on page 25)
<i>Description</i>	The gate start value of the gating function.
<i>Range</i>	Varies depending on the frequency span and the number of points.
<i>Out of Range</i>	Sets the value of the limit, which is closer to the specified value.
<i>Preset Value</i>	-1e-8
<i>Unit</i>	s (second)
<i>Syntax</i>	<p>Dim Value As Double</p> <p><i>Value</i> = <i>app</i>.SCPI.CALCulate(<i>Ch</i>).SELected.FILTter.GATE.TIME.START</p> <p><i>app</i>.SCPI.CALCulate(<i>Ch</i>).SELected.FILTter.GATE.TIME.START = 1e-7</p>
<i>Equivalent Softkeys</i>	Analysis > Gating > Start

SCPI.CALCulate(*Ch*).SELected.FILTter.GATE.TIME.STATE

<i>Object Type</i>	Property (read/write)
<i>Data Type</i>	Boolean
<i>Target</i>	The active trace of channel <i>Ch</i> , <i>Ch</i> : channel number 1–16 (see Table 1 on page 25)
<i>Description</i>	The ON/OFF state of the gating function.
<i>Allowable Values</i>	True: Gating function ON False: Gating function OFF
<i>Preset Value</i>	False
<i>Syntax</i>	Dim <i>Status</i> As Boolean <i>Status</i> = <i>app.SCPI.CALCulate(Ch).SELected.FILTter.GATE.TIME.STATE</i> <i>app.SCPI.CALCulate(Ch).SELected.FILTter.GATE.TIME.STATE</i> = <i>Status</i>
<i>Equivalent Softkeys</i>	Analysis > Gating > Gating

SCPI.CALCulate(*Ch*).SELected.FILTter.GATE.TIME.STOP

<i>Object Type</i>	Property (read/write)
<i>Data Type</i>	Double
<i>Target</i>	The active trace of channel <i>Ch</i> , <i>Ch</i> : channel number 1–16 (see Table 1 on page 25)
<i>Description</i>	The gate stop value of the gating function.
<i>Range</i>	Varies depending on the frequency span and the number of points.
<i>Out of Range</i>	Sets the value of the limit, which is closer to the specified value.
<i>Preset Value</i>	1e–8
<i>Unit</i>	s (second)
<i>Syntax</i>	<p>Dim Value As Double</p> <p><i>Value</i> = <i>app</i>.SCPI.CALCulate(<i>Ch</i>).SELected.FILTter.GATE.TIME.STOP</p> <p><i>app</i>.SCPI.CALCulate(<i>Ch</i>).SELected.FILTter.GATE.TIME.STOP = 1e–7</p>
<i>Equivalent Softkeys</i>	Analysis > Gating > Stop

SCPI.CALCulate(*Ch*).SELected.FILTter.GATE.TIME.TYPE

<i>Object Type</i>	Property (read/write)
<i>Data Type</i>	String
<i>Target</i>	The active trace of channel <i>Ch</i> , <i>Ch</i> : channel number 1–16 (see Table 1 on page 25)
<i>Description</i>	The gate type of the gating function.
<i>Range</i>	"BPASs" : Bandpass type "NOTCh" : Notch type
<i>Notes</i>	The short form of the parameter is indicated by upper case letters. There is no distinction between upper and lower case letters when the property is written. When the property is read out, the short format is indicated by upper case letters.
<i>Out of Range</i>	An error occurs. Error code: 219.
<i>Preset Value</i>	"BPAS"
<i>Syntax</i>	Dim <i>Param</i> As String <i>Param</i> = <i>app</i> .SCPI.CALCulate(<i>Ch</i>).SELected.FILTter.GATE.TIME.TYPE <i>app</i> .SCPI.CALCulate(<i>Ch</i>).SELected.FILTter.GATE.TIME.TYPE = "bpas"
<i>Equivalent Softkeys</i>	Analysis > Gating > Type

SCPI.CALCulate(*Ch*).SElected.FORMat

<i>Object Type</i>	Property (read/write)
<i>Data Type</i>	String
<i>Target</i>	The active trace of channel <i>Ch</i> , <i>Ch</i> : channel number 1–16 (see Table 1 on page 25)
<i>Description</i>	Data format.
<i>Range</i>	"MLOGarithmic" : Logarithmic magnitude "PHASe" : Phase "GDELay" : Group delay time "SLINear" : Smith chart format (Lin) "SLOGarithmic" : Smith chart format (Log) "SCOMplex" : Smith chart format (Real/Imag) "SMITH" : Smith chart format (R + jX) "SADMittance" : Smith chart format (G + jB) "PLINear" : Polar format (Lin) "PLOGarithmic" : Polar format (Log) "POLar" : Polar format (Real/Imag) "MLINear" : Linear magnitude "SWR" : Voltage standing wave ratio "REAL" : Real part "IMAGinary" : Imaginary part "UPHase" : Expanded phase
<i>Notes</i>	The short form of the parameter is indicated by upper case letters. There is no distinction between upper and lower case letters when the property is written. When the property is read out, the short format is indicated by upper case letters.
<i>Out of Range</i>	An error occurs. Error code: 209.
<i>Preset Value</i>	"MLOG"
<i>Syntax</i>	Dim <i>Param</i> As String <i>Param</i> = <i>app</i> .SCPI.CALCulate(<i>Ch</i>).SElected.FORMat <i>app</i> .SCPI.CALCulate(<i>Ch</i>).SElected.FORMat = "PHAS"
<i>Equivalent Softkeys</i>	Format > Log Mag Phase Group Delay Lin Mag SWR Real Imag Format > Smith > Log/Phase Lin/Phase Real/Imag R+jX G+jB Format > Polar > Log/Phase Ling/Phase Real/Imag

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

<i>Object Type</i>	Property (read only)
<i>Data Type</i>	Variant (array of double)
<i>Target</i>	The active trace of channel <i>Ch</i> , <i>Ch</i> : channel number 1–16 (see Table 1 on page 25)
<i>Description</i>	<p>The data array of analysis executed by the SCPI.CALCulate(<i>Ch</i>).SELected.FUNCtion.EXECute method.</p> <p>The array size is 2N, where N is the number of points defined by the SCPI.CALCulate(<i>Ch</i>).SELected.FUNCtion.POINTs property.</p> <p>For the n-th point, where n from 1 to N:</p> <ul style="list-style-type: none"> <i>Data</i>(2n-2) the response value in the n-th measurement point; <i>Data</i>(2n-1) the stimulus value in the n-th measurement point. Always set to 0 for the analysis of mean value, standard deviation, and peak-to-peak value.
<i>Syntax</i>	<pre>Dim <i>Data</i> As Variant <i>Data</i> = <i>app</i>.SCPI.CALCulate(<i>Ch</i>).SELected.FUNCtion.DATA</pre>
<i>Equivalent Softkeys</i>	None

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

<i>Object Type</i>	Property (read/write)
<i>Data Type</i>	Boolean
<i>Target</i>	All traces of channel <i>Ch</i> , <i>Ch</i> : channel number 1–16 (see Table 1 on page 25)
<i>Description</i>	The ON/OFF coupling state of the analysis range for the SCPI.CALCulate(<i>Ch</i>).SELected.FUNCtion.EXECute method.
<i>Allowable Values</i>	True: Coupling state ON False: Coupling state OFF
<i>Preset Value</i>	True
<i>Syntax</i>	Dim <i>Status</i> As Boolean <i>Status</i> = <i>app</i> .SCPI.CALCulate(<i>Ch</i>).SELected.FUNCtion.DOMain.COUPle <i>app</i> .SCPI.CALCulate(<i>Ch</i>).SELected.FUNCtion.DOMain.COUPle = <i>Status</i>
<i>Equivalent Softkeys</i>	None

SCPI.CALCulate(*Ch*).SELected.FUNCtion.DOMain.START

<i>Object Type</i>	Property (read/write)
<i>Data Type</i>	Double
<i>Target</i>	All traces of channel <i>Ch</i> (if the coupling is set to OFF by the SCPI.CALCulate(<i>Ch</i>).SELected.FUNCtion.DOMain.COUPLE property), the active trace of channel <i>Ch</i> (if otherwise), <i>Ch</i> : channel number 1–16 (see Table 1 on page 25)
<i>Description</i>	The start value of the analysis range set by the SCPI.CALCulate(<i>Ch</i>).SELected.FUNCtion.EXECute method.
<i>Range</i>	From the stimulus current start value to the stimulus current stop value.
<i>Out of Range</i>	No limitation
<i>Preset Value</i>	0
<i>Unit</i>	Hz (Hertz) s (second) dBm (decibels above 1 milliwatt)
<i>Syntax</i>	<p>Dim <i>Value</i> As Double <i>Value</i> = <i>app</i>.SCPI.CALCulate(<i>Ch</i>).SELected.FUNCtion.DOMain.START <i>app</i>.SCPI.CALCulate(<i>Ch</i>).SELected.FUNCtion.DOMain.START = 1e9</p>
<i>Equivalent Softkeys</i>	None

SCPI.CALCulate(*Ch*).SELected.FUNCtion.DOMain.STATE

<i>Object Type</i>	Property (read/write)
<i>Data Type</i>	Boolean
<i>Target</i>	All traces of channel <i>Ch</i> (if the coupling is set to OFF by the SCPI.CALCulate(<i>Ch</i>).SELected.FUNCtion.DOMain.COUPLE property), the active trace of channel <i>Ch</i> (if otherwise), <i>Ch</i> : channel number 1–16 (see Table 1 on page 25)
<i>Description</i>	The ON/OFF state of the arbitrary range when executing the analysis by the SCPI.CALCulate(<i>Ch</i>).SELected.FUNCtion.EXECute method.
<i>Allowable Values</i>	True: Arbitrary range ON False: Arbitrary range OFF (entire sweep range)
<i>Preset Value</i>	False
<i>Syntax</i>	Dim <i>Status</i> As Boolean <i>Status</i> = <i>app</i> .SCPI.CALCulate(<i>Ch</i>).SELected.FUNCtion.DOMain.STATE <i>app</i> .SCPI.CALCulate(<i>Ch</i>).SELected.FUNCtion.DOMain.STATE = <i>true</i>
<i>Equivalent Softkeys</i>	None

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

<i>Object Type</i>	Property (read/write)
<i>Data Type</i>	Double
<i>Target</i>	All traces of channel <i>Ch</i> (if the coupling is set to OFF by the SCPI.CALCulate(<i>Ch</i>).SELected.FUNCtion.DOMain.COUPLE property), the active trace of channel <i>Ch</i> (if otherwise), <i>Ch</i> : channel number 1–16 (see Table 1 on page 25)
<i>Description</i>	The stop value of the analysis range set by the SCPI.CALCulate(<i>Ch</i>).SELected.FUNCtion.EXECute method.
<i>Range</i>	From the stimulus current start value to the stimulus current stop value.
<i>Out of Range</i>	No limitation
<i>Preset Value</i>	0
<i>Unit</i>	Hz (Hertz) s (second) dBm (decibels above 1 milliwatt)
<i>Syntax</i>	<p>Dim <i>Value</i> As Double <i>Value</i> = <i>app</i>.SCPI.CALCulate(<i>Ch</i>).SELected.FUNCtion.DOMain.STOP <i>app</i>.SCPI.CALCulate(<i>Ch</i>).SELected.FUNCtion.DOMain.STOP = 2e9</p>
<i>Equivalent Softkeys</i>	None

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

<i>Object Type</i>	Method
<i>Target</i>	The active trace of channel <i>Ch</i> , <i>Ch</i> : channel number 1–16 (see Table 1 on page 25)
<i>Description</i>	Executes the analysis specified with the SCPI.CALCulate(<i>Ch</i>).SELected.FUNCtion.TYPE property. The analysis result can then be read out with the SCPI.CALCulate(<i>Ch</i>).SELected.FUNCtion.DATA property.
<i>Syntax</i>	<i>app</i> .SCPI.CALCulate(<i>Ch</i>).SELected.FUNCtion.EXECute
<i>Equivalent Softkeys</i>	None

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

<i>Object Type</i>	Property (read/write)
<i>Data Type</i>	Double
<i>Target</i>	The active trace of channel <i>Ch</i> , <i>Ch</i> : channel number 1–16 (see Table 1 on page 25)
<i>Description</i>	The lower limit for the peak excursion value when executing the peak search by the SCPI.CALCulate(<i>Ch</i>).SELected.FUNCtion.EXECute method.
<i>Range</i>	Varies depending on the trace format.
<i>Out of Range</i>	No limitation
<i>Preset Value</i>	3
<i>Unit</i>	dB (decibel) ° (degree) s (second)
<i>Syntax</i>	<p>Dim <i>Value</i> As Double</p> <p><i>Value</i> = <i>app</i>.SCPI.CALCulate(<i>Ch</i>).SELected.FUNCtion.PEXCursion</p> <p><i>app</i>.SCPI.CALCulate(<i>Ch</i>).SELected.FUNCtion.PEXCursion = 1.5</p>
<i>Equivalent Softkeys</i>	None

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

<i>Object Type</i>	Property (read only)
<i>Data Type</i>	Long
<i>Target</i>	The active trace of channel <i>Ch</i> , <i>Ch</i> : channel number 1–16 (see Table 1 on page 25)
<i>Description</i>	The number of points (data pairs) of the analysis result by the SCPI.CALCulate(<i>Ch</i>).SELected.FUNCtion.EXECute method. Always equal to 1, when the search is executed for the maximum, minimum, mean, standard deviation, peak, and peak-to-peak values. The actual number of points is read out, when the search is executed for all peaks or all targets.
<i>Syntax</i>	Dim <i>Value</i> As Long <i>Value</i> = <i>app</i> .SCPI.CALCulate(<i>Ch</i>).SELected.FUNCtion.POINts
<i>Equivalent Softkeys</i>	None

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

<i>Object Type</i>	Property (read/write)
<i>Data Type</i>	String
<i>Target</i>	The active trace of channel <i>Ch</i> , <i>Ch</i> : channel number 1–16 (see Table 1 on page 25)
<i>Description</i>	The polarity selection when performing the peak search by the SCPI.CALCulate(<i>Ch</i>).SELected.FUNCtion.EXECute method.
<i>Range</i>	"POSitive" : Positive peaks "NEGative" : Negative peaks "BOTH" : Both positive peaks and negative peaks
<i>Notes</i>	The short form of the parameter is indicated by upper case letters. There is no distinction between upper and lower case letters when the property is written. When the property is read out, the short format is indicated by upper case letters.
<i>Out of Range</i>	The value is ignored.
<i>Preset Value</i>	"POS"
<i>Syntax</i>	Dim <i>Param</i> As String <i>Param</i> = <i>app</i> .SCPI.CALCulate(<i>Ch</i>).SELected.FUNCtion.PPOLarity <i>app</i> .SCPI.CALCulate(<i>Ch</i>).SELected.FUNCtion.PPOLarity = "NEG"
<i>Equivalent Softkeys</i>	None

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

<i>Object Type</i>	Property (read/write)
<i>Data Type</i>	Double
<i>Target</i>	The active trace of channel <i>Ch</i> , <i>Ch</i> : channel number 1–16 (see Table 1 on page 25)
<i>Description</i>	The target level when performing the search for the trace and the target level crosspoints by the SCPI.CALCulate(<i>Ch</i>).SELected.FUNCtion.EXECute method.
<i>Range</i>	Varies depending on the trace format.
<i>Out of Range</i>	No limitation
<i>Preset Value</i>	0
<i>Unit</i>	dB (decibel) ° (degree) s (second)
<i>Syntax</i>	<p>Dim <i>Value</i> As Double</p> <p><i>Value</i> = <i>app</i>.SCPI.CALCulate(<i>Ch</i>).SELected.FUNCtion.TARGet</p> <p><i>app</i>.SCPI.CALCulate(<i>Ch</i>).SELected.FUNCtion.TARGet = -10</p>
<i>Equivalent Softkeys</i>	None

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

<i>Object Type</i>	Property (read/write)
<i>Data Type</i>	String
<i>Target</i>	The active trace of channel <i>Ch</i> , <i>Ch</i> : channel number 1–16 (see Table 1 on page 25)
<i>Description</i>	The transition type selection when performing the search for the trace and the target level crosspoints by the SCPI.CALCulate(<i>Ch</i>).SELected.FUNCtion.EXECute method.
<i>Range</i>	"POSitive" : Positive peaks "NEGative" : Negative peaks "BOTH" : Both positive peaks and negative peaks
<i>Notes</i>	The short form of the parameter is indicated by upper case letters. There is no distinction between upper and lower case letters when the property is written. When the property is read out, the short format is indicated by upper case letters.
<i>Out of Range</i>	The value is ignored.
<i>Preset Value</i>	"POS"
<i>Syntax</i>	Dim <i>Param</i> As String <i>Param</i> = <i>app</i> .SCPI.CALCulate(<i>Ch</i>).SELected.FUNCtion.TTRansition <i>app</i> .SCPI.CALCulate(<i>Ch</i>).SELected.FUNCtion.TTRansition = "both"
<i>Equivalent Softkeys</i>	None

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

<i>Object Type</i>	Property (read/write)
<i>Data Type</i>	String
<i>Target</i>	The active trace of channel <i>Ch</i> , <i>Ch</i> : channel number 1–16 (see Table 1 on page 25)
<i>Description</i>	The selection of the type of analysis executed by the SCPI.CALCulate(<i>Ch</i>).SELected.FUNCtion.EXECute method.
<i>Range</i>	<p>"PTPeak" : Peak-to-peak (difference between the maximum value and the minimum value)</p> <p>"STDEV" : Standard deviation</p> <p>"MEAN" : Mean value</p> <p>"MAXimum" : Maximum value</p> <p>"MINimum" : Minimum value</p> <p>"PEAK" : Search for the peak</p> <p>"APEak" : Search for all the peaks</p> <p>"ATARget" : Search for all targets</p>
<i>Notes</i>	The short form of the parameter is indicated by upper case letters. There is no distinction between upper and lower case letters when the property is written. When the property is read out, the short format is indicated by upper case letters.
<i>Out of Range</i>	The value is ignored.
<i>Preset Value</i>	"PTP"
<i>Syntax</i>	<pre>Dim Param As String Param = app.SCPI.CALCulate(<i>Ch</i>).SELected.FUNCtion.TYPE app.SCPI.CALCulate(<i>Ch</i>).SELected.FUNCtion.TYPE = "STDEV"</pre>
<i>Equivalent Softkeys</i>	None

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

<i>Object Type</i>	Property (read/write)												
<i>Data Type</i>	Variant (array of double)												
<i>Target</i>	The active trace of channel <i>Ch</i> , <i>Ch</i> : channel number 1–16 (see Table 1 on page 25)												
<i>Description</i>	<p>The data array, which is the limit line in the limit test function. The array size is $1 + 5N$, where N is the number of measuring points.</p> <p>For the n-th point, where n from 1 to N:</p> <table> <tr> <td><i>Data(0)</i></td> <td>The number of limit line segments N is from 0 to 100. Setting 0 clears the limit line;</td> </tr> <tr> <td><i>Data(5n-4)</i></td> <td>type of the n-th limit line segment; 0: OFF 1: Upper limit 2: Lower limit 3: Single Point limit</td> </tr> <tr> <td><i>Data(5n-3)</i></td> <td>the stimulus value in the start point of the n-th segment;</td> </tr> <tr> <td><i>Data(5n-2)</i></td> <td>the stimulus value in the end point of the n-th segment;</td> </tr> <tr> <td><i>Data(5n-1)</i></td> <td>the response value in the start point of the n-th segment;</td> </tr> <tr> <td><i>Data(5n-0)</i></td> <td>the response value in the end point of the n-th segment.</td> </tr> </table>	<i>Data(0)</i>	The number of limit line segments N is from 0 to 100. Setting 0 clears the limit line;	<i>Data(5n-4)</i>	type of the n -th limit line segment; 0: OFF 1: Upper limit 2: Lower limit 3: Single Point limit	<i>Data(5n-3)</i>	the stimulus value in the start point of the n -th segment;	<i>Data(5n-2)</i>	the stimulus value in the end point of the n -th segment;	<i>Data(5n-1)</i>	the response value in the start point of the n -th segment;	<i>Data(5n-0)</i>	the response value in the end point of the n -th segment.
<i>Data(0)</i>	The number of limit line segments N is from 0 to 100. Setting 0 clears the limit line;												
<i>Data(5n-4)</i>	type of the n -th limit line segment; 0: OFF 1: Upper limit 2: Lower limit 3: Single Point limit												
<i>Data(5n-3)</i>	the stimulus value in the start point of the n -th segment;												
<i>Data(5n-2)</i>	the stimulus value in the end point of the n -th segment;												
<i>Data(5n-1)</i>	the response value in the start point of the n -th segment;												
<i>Data(5n-0)</i>	the response value in the end point of the n -th segment.												
<i>Notes</i>	If the array size is not $1 + 5N$, where N is <i>Data(0)</i> , an error occurs (error code 214). If <i>Data(5n - 4)</i> is less than 0 or more than 2, an error occurs (error code 214). When <i>Data(5n-3)</i> , <i>Data(5n-2)</i> , <i>Data(5n-1)</i> and <i>Data(5n-0)</i> elements are out of allowable range, the value is set to the limit, which is closer to the specified value.												
<i>Syntax</i>	<pre>Dim Data As Variant Data = app.SCPI.CALCulate(<i>Ch</i>).SELected.LIMit.DATA app.SCPI.CALCulate(<i>Ch</i>).SELected.LIMit.DATA = Array(1,2,800,900,-10,-10)</pre>												
<i>Equivalent Softkeys</i>	Analysis > Limit Test > Edit Limit Line												

SCPI.CALCulate(*Ch*).SELected.LIMit.DISPlay.STATE

<i>Object Type</i>	Property (read/write)
<i>Data Type</i>	Boolean
<i>Target</i>	The active trace of channel <i>Ch</i> , <i>Ch</i> : channel number 1–16 (see Table 1 on page 25)
<i>Description</i>	The ON/OFF state of the limit line display of the limit test function.
<i>Allowable Values</i>	True: Limit line display ON False: Limit line display OFF
<i>Preset Value</i>	False
<i>Syntax</i>	<p>Dim <i>Status</i> As Boolean</p> <pre><i>Status</i> = app.SCPI.CALCulate(<i>Ch</i>).SELected.LIMit.DISPlay.STATE app.SCPI.CALCulate(<i>Ch</i>).SELected.LIMit.DISPlay.STATE = true</pre>
<i>Equivalent Softkeys</i>	Analysis > Limit Test > Limit Line

SCPI.CALCulate(*Ch*).SELected.LIMit.FAIL

<i>Object Type</i>	Property (read only)
<i>Data Type</i>	Boolean
<i>Target</i>	The active trace of channel <i>Ch</i> , <i>Ch</i> : channel number 1–16 (see Table 1 on page 25)
<i>Description</i>	The limit test result.
<i>Allowable Values</i>	True: Fail False: Pass
<i>Syntax</i>	<pre>Dim Status As Boolean Status = app.SCPI.CALCulate(Ch).SELected.LIMit.FAIL</pre>
<i>Equivalent Softkeys</i>	None

SCPI.CALCulate(*Ch*).SELected.LIMit.OFFSet.AMPLitude

<i>Object Type</i>	Property (read/write)
<i>Data Type</i>	Double
<i>Target</i>	The active trace of channel <i>Ch</i> , <i>Ch</i> : channel number 1–16 (see Table 1 on page 25)
<i>Description</i>	The value of the limit line offset along Y-axis.
<i>Range</i>	Varies depending on the trace format.
<i>Out of Range</i>	No limitation
<i>Preset Value</i>	0
<i>Unit</i>	dB (decibel) ° (degree) s (second)
<i>Syntax</i>	<p>Dim Value As Double</p> <p><i>Value</i> = <i>app</i>.SCPI.CALCulate(<i>Ch</i>).SELected.LIMit.OFFSet.AMPLitude</p> <p><i>app</i>.SCPI.CALCulate(<i>Ch</i>).SELected.LIMit.OFFSet.AMPLitude = -10</p>
<i>Equivalent Softkeys</i>	Analysis > Limit Test > Limit Line Offsets > Response Offset

SCPI.CALCulate(*Ch*).SELected.LIMit.OFFSet.MARKer

<i>Object Type</i>	Method
<i>Target</i>	The active trace of channel <i>Ch</i> , <i>Ch</i> : channel number 1–16 (see Table 1 on page 25)
<i>Description</i>	Sets the value of the limit line offset along Y-axis to the active marker value.
<i>Syntax</i>	<code>app.SCPI.CALCulate(<i>Ch</i>).SELected.LIMit.OFFSet.MARKer</code>
<i>Equivalent Softkeys</i>	Analysis > Limit Test > Limit Line Offsets > Marker > Response Ofs

SCPI.CALCulate(*Ch*).SElected.LIMit.OFFSet.STIMulus

<i>Object Type</i>	Property (read/write)
<i>Data Type</i>	Double
<i>Target</i>	The active trace of channel <i>Ch</i> , <i>Ch</i> : channel number 1–16 (see Table 1 on page 25)
<i>Description</i>	The value of the limit line offset along X-axis.
<i>Range</i>	From the stimulus current start value to the stimulus current stop value.
<i>Out of Range</i>	No limitation
<i>Preset Value</i>	0
<i>Unit</i>	Hz (Hertz) s (second) dBm (decibels above 1 milliwatt)
<i>Syntax</i>	<p>Dim <i>Value</i> As Double <i>Value</i> = <i>app</i>.SCPI.CALCulate(<i>Ch</i>).SElected.LIMit.OFFSet.STIMulus <i>app</i>.SCPI.CALCulate(<i>Ch</i>).SElected.LIMit.OFFSet.STIMulus = 1e6</p>
<i>Equivalent Softkeys</i>	Analysis > Limit Test > Limit Lines Offsets > Stimulus Offset

SCPI.CALCulate(*Ch*).SELected.LIMit.REPort.ALL

<i>Object Type</i>	Property (read only)														
<i>Data Type</i>	Variant (array of double)														
<i>Target</i>	The active trace of channel <i>Ch</i> , <i>Ch</i> : channel number 1–16 (see Table 1 on page 25)														
<i>Description</i>	<p>The data array, which is the limit test results. The array size is 4N, where N is the number of measurement points.</p> <p>For the n-th point, where n from 1 to N:</p> <table style="margin-left: 40px;"> <tr><td><i>Data(4n-3)</i></td><td>the stimulus value in the n-th point</td></tr> <tr><td><i>Data(4n-2)</i></td><td>the limit test result in the n-th point</td></tr> <tr><td>-1:</td><td>No limit</td></tr> <tr><td>0:</td><td>Fail</td></tr> <tr><td>1:</td><td>Pass</td></tr> <tr><td><i>Data(4n-1)</i></td><td>the upper limit value in the n-th point (0 – if there is no limit)</td></tr> <tr><td><i>Data(4n-0)</i></td><td>the lower limit value in the n-th point (0 – if there is no limit)</td></tr> </table>	<i>Data(4n-3)</i>	the stimulus value in the n-th point	<i>Data(4n-2)</i>	the limit test result in the n-th point	-1:	No limit	0:	Fail	1:	Pass	<i>Data(4n-1)</i>	the upper limit value in the n-th point (0 – if there is no limit)	<i>Data(4n-0)</i>	the lower limit value in the n-th point (0 – if there is no limit)
<i>Data(4n-3)</i>	the stimulus value in the n-th point														
<i>Data(4n-2)</i>	the limit test result in the n-th point														
-1:	No limit														
0:	Fail														
1:	Pass														
<i>Data(4n-1)</i>	the upper limit value in the n-th point (0 – if there is no limit)														
<i>Data(4n-0)</i>	the lower limit value in the n-th point (0 – if there is no limit)														
<i>Syntax</i>	<pre>Dim <i>Data</i> As Variant <i>Data</i> = <i>app</i>.SCPI.CALCulate(<i>Ch</i>).SELected.LIMit.REPort.ALL</pre>														
<i>Equivalent Softkeys</i>	None														

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

<i>Object Type</i>	Property (read only)
<i>Data Type</i>	Variant (array of double)
<i>Target</i>	The active trace of channel <i>Ch</i> , <i>Ch</i> : channel number 1–16 (see Table 1 on page 25)
<i>Description</i>	The data array, which is the stimulus values at all the measurement points that failed the limit test. The array size is defined by the SCPI.CALCulate(<i>Ch</i>).SELected.LIMit.REPort.POINTs property.
<i>Syntax</i>	<p>Dim <i>Data</i> As Variant</p> <p><i>Data</i> = <i>app</i>.SCPI.CALCulate(<i>Ch</i>).SELected.LIMit.REPort.DATA</p>
<i>Equivalent Softkeys</i>	None

SCPI.CALCulate(*Ch*).SELected.LIMit.REPort.POINts

<i>Object Type</i>	Property (read only)
<i>Data Type</i>	Long
<i>Target</i>	The active trace of channel <i>Ch</i> , <i>Ch</i> : channel number 1–16 (see Table 1 on page 25)
<i>Description</i>	The number of the measurement points that failed the limit test. The array of stimulus values of the points can be read out by the SCPI.CALCulate(<i>Ch</i>).SELected.LIMit.REPort.DATA property.
<i>Syntax</i>	Dim <i>Cnt</i> As Long <i>Cnt</i> = <i>app</i> .SCPI.CALCulate(<i>Ch</i>).SELected.LIMit.REPort.POINts
<i>Equivalent Softkeys</i>	None

SCPI.CALCulate(*Ch*).SELected.LIMit.STATE

<i>Object Type</i>	Property (read/write)
<i>Data Type</i>	Boolean
<i>Target</i>	The active trace of channel <i>Ch</i> , <i>Ch</i> : channel number 1–16 (see Table 1 on page 25)
<i>Description</i>	The ON/OFF state of the limit test function.
<i>Allowable Values</i>	True: Limit test function ON False: Limit test function OFF
<i>Preset Value</i>	False
<i>Syntax</i>	<p>Dim <i>Status</i> As Boolean</p> <p><i>Status</i> = <i>app</i>.SCPI.CALCulate(<i>Ch</i>).SELected.LIMit.STATE</p> <p><i>app</i>.SCPI.CALCulate(<i>Ch</i>).SELected.LIMit.STATE = true</p>
<i>Equivalent Softkeys</i>	Analysis > Limit Test > Limit Test

SCPI.CALCulate(*Ch*).SELected.MARKer(*Mk*).ACTivate

<i>Object Type</i>	Method
<i>Target</i>	Marker <i>Mk</i> of the active trace of channel <i>Ch</i> , <i>Ch</i> : channel number 1–16 (see Table 1 on page 25) <i>Mk</i> : marker number 1–15, or reference marker number 16 (see Table 4 on page 74)
<i>Description</i>	Sets the active marker. If a marker is OFF this function will turn it ON. Turning ON a marker with the number from 1 to 15 will turn ON all the markers of smaller numbers. Turning ON the reference marker with number 16 does not turn ON the markers with the numbers from 1 to 15, but switches these markers to the relative measurement mode.
<i>Syntax</i>	<code>app.SCPI.CALCulate(<i>Ch</i>).SELected.MARKer(<i>Mk</i>).ACTivate</code>
<i>Equivalent Softkeys</i>	Markers > Select > Marker n Markers > Reference Marker

Table 4. *Mk*: Marker Number

<i>Data Type</i>	Long
<i>Description</i>	Marker number. Numbers from 1 to 15 are for regular markers, number 16 is for the reference marker.
<i>Range</i>	from 1 to 16
<i>Out of Range</i>	An error occurs. Error code: 203.
<i>Notes</i>	If the marker number is not specified, it is taken as equal to 1.

SCPI.CALCulate(*Ch*).SELected.MARKer(*Mk*).BWIDth.DATA

<i>Object Type</i>	Property (read only)
<i>Data Type</i>	Variant (array of double)
<i>Target</i>	Marker <i>Mk</i> of the active trace of channel <i>Ch</i> , <i>Ch</i> : channel number 1–16 (see Table 1 on page 25) <i>Mk</i> : marker number 1–15, or reference marker number 16 (see Table 4 on page 74)
<i>Description</i>	The bandwidth search result. The bandwidth search can be performed relatively to the marker <i>Mk</i> , or relatively to the absolute maximum value of the trace (in this case the marker number is ignored), what is set by the SCPI.CALCulate(<i>Ch</i>).SELected.MARKer(<i>Mk</i>).BWIDth.REFerence property. The array contains 4 elements: <i>Data(0)</i> Bandwidth; <i>Data(1)</i> Center frequency; <i>Data(2)</i> Q value; <i>Data(3)</i> Loss.
<i>Notes</i>	If the bandwidth search is impossible, all the read out values are 0. If the search is performed relatively to a maker, which is OFF, an error occurs (error code 204).
<i>Syntax</i>	Dim <i>Data</i> As Variant <i>Data</i> = <i>app</i> .SCPI.CALCulate(<i>Ch</i>).SELected.MARKer(<i>Mk</i>).BWIDth.DATA
<i>Equivalent Softkeys</i>	None

SCPI.CALCulate(*Ch*).SELected.MARKer(1).BWIDth.REFERENCE

<i>Object Type</i>	Property (read/write)
<i>Data Type</i>	String
<i>Target</i>	The active trace of channel <i>Ch</i> , <i>Ch</i> : channel number 1–16 (see Table 1 on page 25)
<i>Description</i>	The selection of the reference point for the bandwidth search function: reference marker or absolute maximum value of the trace.
<i>Range</i>	"MARKer" : Bandwidth search relative to the reference marker "MAXimum" : Bandwidth search relative to the absolute maximum of the trace
<i>Notes</i>	The short form of the parameter is indicated by upper case letters. There is no distinction between upper and lower case letters when the property is written. When the property is read out, the short format is indicated by upper case letters.
<i>Out of Range</i>	The value is ignored.
<i>Preset Value</i>	"MAX"
<i>Syntax</i>	<pre>Dim Param As String Param = app.SCPI.CALCulate(<i>Ch</i>).SELected.MARKer.BWIDth.REFERENCE app.SCPI.CALCulate(<i>Ch</i>).SELected.MARKer.BWIDth.REFERENCE = "marker"</pre>
<i>Warning</i>	Object <i>MARKer</i> has an index. In Visual Basic the index is 1 by default when it is omitted. The index can not be omitted in other programming languages.
<i>Equivalent Softkeys</i>	Markers > Marker Math > Bandwidth Search > Search Ref To

SCPI.CALCulate(*Ch*).SELected.MARKer(1).BWIDth.STATE

<i>Object Type</i>	Property (read/write)
<i>Data Type</i>	Boolean
<i>Target</i>	The active trace of channel <i>Ch</i> , <i>Ch</i> : channel number 1–16 (see Table 1 on page 25)
<i>Description</i>	The ON/OFF state of the bandwidth search function.
<i>Allowable Values</i>	True: Bandwidth search function ON False: Bandwidth search function OFF
<i>Preset Value</i>	False
<i>Syntax</i>	<p>Dim <i>Status</i> As Boolean</p> <p><i>Status</i> = <i>app</i>.SCPI.CALCulate(<i>Ch</i>).SELected.MARKer.BWIDth.STATE</p> <p><i>app</i>.SCPI.CALCulate(<i>Ch</i>).SELected.MARKer.BWIDth.STATE = true</p>
<i>Warning</i>	Object <i>MARKer</i> has an index. In Visual Basic the index is 1 by default when it is omitted. The index can not be omitted in other programming languages.
<i>Equivalent Softkeys</i>	Markers > Marker Math > Bandwidth Search > Bandwidth Search

SCPI.CALCulate(*Ch*).SELected.MARKer(1).BWIDth.THReshold

<i>Object Type</i>	Property (read/write)
<i>Data Type</i>	Double
<i>Target</i>	The active trace of channel <i>Ch</i> , <i>Ch</i> : channel number 1–16 (see Table 1 on page 25)
<i>Description</i>	The bandwidth definition value.
<i>Range</i>	Varies depending on the trace format.
<i>Out of Range</i>	No limitation
<i>Preset Value</i>	–3
<i>Unit</i>	dB (decibel) ° (degree) s (second)
<i>Syntax</i>	<p>Dim <i>Value</i> As Double</p> <p><i>Value</i> = <i>app</i>.SCPI.CALCulate(<i>Ch</i>).SELected.MARKer(<i>Mk</i>).BWIDth.THReshold</p> <p><i>app</i>.SCPI.CALCulate(<i>Ch</i>).SELected.MARKer(<i>Mk</i>).BWIDth.THReshold = –6.0</p>
<i>Warning</i>	Object <i>MARKer</i> has an index. In Visual Basic the index is 1 by default when it is omitted. The index can not be omitted in other programming languages.
<i>Equivalent Softkeys</i>	Markers > Marker Math > Bandwidth Search > Bandwidth Value

SCPI.CALCulate(*Ch*).SELected.MARKer(1).BWIDth.TYPE

<i>Object Type</i>	Property (read/write)
<i>Data Type</i>	String
<i>Target</i>	The active trace of channel <i>Ch</i> , <i>Ch</i> : channel number 1–16 (see Table 1 on page 25)
<i>Description</i>	The type of the bandwidth search function.
<i>Range</i>	"BPASs" : Bandpass "NOTCh" : Notch
<i>Notes</i>	The short form of the parameter is indicated by upper case letters. There is no distinction between upper and lower case letters when the property is written. When the property is read out, the short format is indicated by upper case letters.
<i>Out of Range</i>	The value is ignored.
<i>Preset Value</i>	"BPAS"
<i>Syntax</i>	<pre>Dim Param As String Param = app.SCPI.CALCulate(<i>Ch</i>).SELected.MARKer.BWIDth.TYPE app.SCPI.CALCulate(<i>Ch</i>).SELected.MARKer.BWIDth.TYPE = "notc"</pre>
<i>Warning</i>	Object <i>MARKer</i> has an index. In Visual Basic the index is 1 by default when it is omitted. The index can not be omitted in other programming languages.
<i>Equivalent Softkeys</i>	Markers > Marker Math > Bandwidth Search > Type

SCPI.CALCulate(*Ch*).SELected.MARKer(1).COUNT

<i>Object Type</i>	Property (read/write)
<i>Data Type</i>	Long
<i>Target</i>	The active trace of channel <i>Ch</i> , <i>Ch</i> : channel number 1–16 (see Table 1 on page 25)
<i>Description</i>	The number of the turned ON markers. Writing value from 1 to 15 will turn ON regular markers. Writing value 16 will turn ON 15 regular markers and the reference marker. Turning ON the reference marker switches all markers to the relative measurement mode.
<i>Range</i>	from 0 to 16
<i>Out of Range</i>	Sets the value of the limit, which is closer to the specified value.
<i>Preset Value</i>	0
<i>Syntax</i>	<p>Dim MarkerCnt As Long</p> <p><i>MarkerCnt</i> = <i>app</i>.SCPI.CALCulate(<i>Ch</i>).SELected.MARKer.COUNT</p> <p><i>app</i>.SCPI.CALCulate(<i>Ch</i>).SELected.MARKer.COUNT = 5</p>
<i>Warning</i>	Object MARKer has an index. In Visual Basic the index is 1 by default when it is omitted. The index can not be omitted in other programming languages.
<i>Equivalent Softkeys</i>	None

SCPI.CALCulate(*Ch*).SELected.MARKer(1).COUPle

<i>Object Type</i>	Property (read/write)
<i>Data Type</i>	Boolean
<i>Target</i>	All traces of channel <i>Ch</i> , <i>Ch</i> : channel number 1–16 (see Table 1 on page 25)
<i>Description</i>	The ON/OFF state of the marker coupling function.
<i>Allowable Values</i>	True: Marker coupling ON False: Marker coupling OFF
<i>Preset Value</i>	True
<i>Syntax</i>	Dim <i>Status</i> As Boolean <i>Status</i> = <i>app</i> .SCPI.CALCulate(<i>Ch</i>).SELected.MARKer.COUPle <i>app</i> .SCPI.CALCulate(<i>Ch</i>).SELected.MARKer.COUPle = false
<i>Warning</i>	Object <i>MARKer</i> has an index. In Visual Basic the index is 1 by default when it is omitted. The index can not be omitted in other programming languages.
<i>Equivalent Softkeys</i>	Marker > Properties > Marker Couple

SCPI.CALCulate(*Ch*).SELected.MARKer(1).DATA

<i>Object Type</i>	Property (read only)
<i>Data Type</i>	Variant (array of double)
<i>Target</i>	The active trace of channel <i>Ch</i> , <i>Ch</i> : channel number 1–16 (see Table 1 on page 25)
<i>Description</i>	<p>The data array of all turned ON markers. The array size is $3N + 1$, where N is the number of turned ON markers including the reference marker. If reference marker is turned ON the last three elements of array contain the reference marker data and the rest elements of array contain the relative values.</p> <p>For the n-th marker, where n from 1 to N:</p> <ul style="list-style-type: none"> <i>Data(0)</i> the number of turned ON markers including the reference marker; <i>Data(3n-2)</i> the stimulus value of the n-th marker; <i>Data(3n-1)</i> the real data in rectangular format, real part in polar and Smith chart formats of the n-th marker; <i>Data(3n-0)</i> 0 in rectangular format, imaginary part in polar and Smith chart formats of the n-th marker;
<i>Syntax</i>	<pre>Dim Data As Variant Data = app.SCPI.CALCulate(<i>Ch</i>).SELected.MARKer.DATA</pre>
<i>Warning</i>	Object <i>MARKEr</i> has an index. In Visual Basic the index is 1 by default when it is omitted. The index can not be omitted in other programming languages.
<i>Equivalent Softkeys</i>	None

SCPI.CALCulate(*Ch*).SELected.MARKer(1).FUNCTION.DOMain.COUPLE

<i>Object Type</i>	Property (read/write)
<i>Data Type</i>	Boolean
<i>Target</i>	All traces of channel <i>Ch</i> , <i>Ch</i> : channel number 1–16 (see Table 1 on page 25)
<i>Description</i>	The ON/OFF state of the marker search range coupling for different traces.
<i>Allowable Values</i>	True: Marker search range coupling ON False: Marker search range coupling OFF
<i>Preset Value</i>	True
<i>Syntax</i>	Dim <i>Status</i> As Boolean <i>Status</i> = <i>app</i> .SCPI.CALCulate(<i>Ch</i>).SELected.MARKer.FUNCTION.DOMain.COUPLE <i>app</i> .SCPI.CALCulate(<i>Ch</i>).SELected.MARKer.FUNCTION.DOMain.COUPLE = false
<i>Warning</i>	Object <i>MARKer</i> has an index. In Visual Basic the index is 1 by default when it is omitted. The index can not be omitted in other programming languages.
<i>Equivalent Softkeys</i>	Markers > Marker Search > Couple

SCPI.CALCulate(*Ch*).SELected.MARKer(1).FUNCtion.DOMain.START

<i>Object Type</i>	Property (read/write)
<i>Data Type</i>	Double
<i>Target</i>	All traces of channel <i>Ch</i> (if the marker search range coupling is set to OFF by the SCPI.CALCulate(<i>Ch</i>).SELected.FUNCtion.DOMain.COUPLE property), the active trace of channel <i>Ch</i> (if otherwise), <i>Ch</i> : channel number 1–16 (see Table 1 on page 25)
<i>Description</i>	The start value of the marker search range.
<i>Range</i>	From the stimulus current start value to the stimulus current stop value.
<i>Out of Range</i>	No limitation
<i>Preset Value</i>	Minimum frequency.
<i>Unit</i>	Hz (Hertz) s (second) dBm (decibels above 1 milliwatt)
<i>Syntax</i>	<p>Dim <i>Value</i> As Double <i>Value</i> = <i>app</i>.SCPI.CALCulate(<i>Ch</i>).SELected.MARKer.FUNCtion.DOMain.START <i>app</i>.SCPI.CALCulate(<i>Ch</i>).SELected.MARKer.FUNCtion.DOMain.START = 1e6</p>
<i>Warning</i>	Object MARKer has an index. In Visual Basic the index is 1 by default when it is omitted. The index can not be omitted in other programming languages.
<i>Equivalent Softkeys</i>	Markers > Marker Search > Search Start

SCPI.CALCulate(*Ch*).SElected.MARKer(1).FUNCtion.DOMain.STATE

<i>Object Type</i>	Property (read/write)
<i>Data Type</i>	Boolean
<i>Target</i>	All traces of channel <i>Ch</i> (if the marker search range coupling is set to OFF by the SCPI.CALCulate(<i>Ch</i>).SElected.MARKer.FUNCtion.DOMain.COUPle property), the active trace of channel <i>Ch</i> (if otherwise), <i>Ch</i> : channel number 1–16 (see Table 1 on page 25)
<i>Description</i>	The ON/OFF state of the arbitrary range when executing the marker search.
<i>Allowable Values</i>	True: Marker search range ON False: Marker search range OFF (entire sweep range)
<i>Preset Value</i>	False
<i>Syntax</i>	<p>Dim <i>Status</i> As Boolean</p> <p><i>Status</i> = <i>app</i>.SCPI.CALCulate(<i>Ch</i>).SElected.MARKer.FUNCtion.DOMain.STATE</p> <p><i>app</i>.SCPI.CALCulate(<i>Ch</i>).SElected.MARKer.FUNCtion.DOMain.STATE = true</p>
<i>Warning</i>	Object MARKer has an index. In Visual Basic the index is 1 by default when it is omitted. The index can not be omitted in other programming languages.
<i>Equivalent Softkeys</i>	Markers > Marker Search > Search Range

SCPI.CALCulate(*Ch*).SELected.MARKer(1).FUNCtion.DOMain.STOP

<i>Object Type</i>	Property (read/write)
<i>Data Type</i>	Double
<i>Target</i>	All traces of channel <i>Ch</i> (if the marker search range coupling is set to OFF by the SCPI.CALCulate(<i>Ch</i>).SELected.MARKer.FUNCtion.DOMain.COUPle property), the active trace of channel <i>Ch</i> (if otherwise), <i>Ch</i> : channel number 1–16 (see Table 1 on page 25)
<i>Description</i>	The stop value of the marker search range.
<i>Range</i>	From the stimulus current start value to the stimulus current stop value.
<i>Out of Range</i>	No limitation
<i>Preset Value</i>	Maximum frequency
<i>Unit</i>	Hz (Hertz) s (second) dBm (decibels above 1 milliwatt)
<i>Syntax</i>	<p>Dim <i>Value</i> As Double <i>Value</i> = <i>app</i>.SCPI.CALCulate(<i>Ch</i>).SELected.MARKer.FUNCtion.DOMain.STOP <i>app</i>.SCPI.CALCulate(<i>Ch</i>).SELected.MARKer.FUNCtion.DOMain.STOP = 1e6</p>
<i>Warning</i>	Object MARKer has an index. In Visual Basic the index is 1 by default when it is omitted. The index can not be omitted in other programming languages.
<i>Equivalent Softkeys</i>	Markers > Marker Search > Search Stop

SCPI.CALCulate(*Ch*).SELected.MARKer(*Mk*).FUNCtion.EXECute

<i>Object Type</i>	Method
<i>Target</i>	Marker <i>Mk</i> of the active trace of channel <i>Ch</i> , <i>Ch</i> : channel number 1–16 (see Table 1 on page 25) <i>Mk</i> : marker number 1–15, or reference marker number 16 (see Table 4 on page 74)
<i>Description</i>	Executes the marker search according to the specified criterion. The type of the marker search is set by the SCPI.CALCulate(<i>Ch</i>).SELected.MARKer(<i>Mk</i>).FUNCtion.TYPE property.
<i>Syntax</i>	<i>app</i> .SCPI.CALCulate(<i>Ch</i>).SELected.MARKer(<i>Mk</i>).FUNCtion.EXECute
<i>Equivalent Softkeys</i>	Markers > Marker Search > Maximum Minimum Markers > Marker Search > Peak > Search Peak Search Max Peak Search Peak Left Search Peak Right Markers > Marker Search > Target > Search Target Search Target Left Search Target Right

SCPI.CALCulate(*Ch*).SELected.MARKer(*Mk*).FUNCtion.PEXcursion

<i>Object Type</i>	Property (read/write)
<i>Data Type</i>	Double
<i>Target</i>	Marker <i>Mk</i> of the active trace of channel <i>Ch</i> , <i>Ch</i> : channel number 1–16 (see Table 1 on page 25) <i>Mk</i> : marker number 1–15, or reference marker number 16 (see Table 4 on page 74)
<i>Description</i>	The peak excursion value, when the marker search for peak is performed by the SCPI.CALCulate(<i>Ch</i>).SELected.MARKer(<i>Mk</i>).FUNCtion.EXECute method.
<i>Range</i>	Varies depending on the trace format.
<i>Out of Range</i>	No limitation
<i>Preset Value</i>	1
<i>Unit</i>	dB (decibel) ° (degree) s (second)
<i>Syntax</i>	<p>Dim <i>Value</i> As Double <i>Value</i> = <i>app</i>.SCPI.CALCulate(<i>Ch</i>).SELected.MARKer(<i>Mk</i>).FUNCtion.PEXcursion <i>app</i>.SCPI.CALCulate(<i>Ch</i>).SELected.MARKer(<i>Mk</i>).FUNCtion.PEXcursion = 3.0</p>
<i>Equivalent Softkeys</i>	Markers > Marker Search > Peak > Peak Excursion

SCPI.CALCulate(*Ch*).SELected.MARKer(*Mk*).FUNCtion.PPOLarity

<i>Object Type</i>	Property (read/write)
<i>Data Type</i>	String
<i>Target</i>	Marker <i>Mk</i> of the active trace of channel <i>Ch</i> , <i>Ch</i> : channel number 1–16 (see Table 1 on page 25) <i>Mk</i> : marker number 1–15, or reference marker number 16 (see Table 4 on page 74)
<i>Description</i>	The peak polarity selection, when the marker search for peak is performed by the SCPI.CALCulate(<i>Ch</i>).SELected.MARKer(<i>Mk</i>).FUNCtion.EXECute method.
<i>Range</i>	"POSitive" : Positive polarity "NEGative" : Negative polarity "BOTH" : Both positive polarity and negative polarity
<i>Notes</i>	The short form of the parameter is indicated by upper case letters. There is no distinction between upper and lower case letters when the property is written. When the property is read out, the short format is indicated by upper case letters.
<i>Out of Range</i>	The value is ignored.
<i>Preset Value</i>	"POS"
<i>Syntax</i>	<pre>Dim Param As String Param = app.SCPI.CALCulate(<i>Ch</i>).SELected.MARKer(<i>Mk</i>).FUNCtion.PPOLarity app.SCPI.CALCulate(<i>Ch</i>).SELected.MARKer(<i>Mk</i>).FUNCtion.PPOLarity = "neg"</pre>
<i>Equivalent Softkeys</i>	Markers > Marker Search > Peak > Peak Polarity > Positive Negative Both

SCPI.CALCulate(*Ch*).SELected.MARKer(*Mk*).FUNCtion.TARGet

<i>Object Type</i>	Property (read/write)
<i>Data Type</i>	Double
<i>Target</i>	Marker <i>Mk</i> of the active trace of channel <i>Ch</i> , <i>Ch</i> : channel number 1–16 (see Table 1 on page 25) <i>Mk</i> : marker number 1–15, or reference marker number 16 (see Table 4 on page 74)
<i>Description</i>	The target value, when the marker search for target is performed by the SCPI.CALCulate(<i>Ch</i>).SELected.MARKer(<i>Mk</i>).FUNCtion.EXECute method.
<i>Range</i>	Varies depending on the trace format.
<i>Out of Range</i>	No limitation
<i>Preset Value</i>	0
<i>Unit</i>	dB (decibel) ° (degree) s (second)
<i>Syntax</i>	<p>Dim <i>Value</i> As Double <i>Value</i> = <i>app</i>.SCPI.CALCulate(<i>Ch</i>).SELected.MARKer(<i>Mk</i>).FUNCtion.TARGet <i>app</i>.SCPI.CALCulate(<i>Ch</i>).SELected.MARKer(<i>Mk</i>).FUNCtion.TARGet = -10</p>
<i>Equivalent Softkeys</i>	Markers > Marker Search > Target > Target Value

SCPI.CALCulate(*Ch*).SELected.MARKer(*Mk*).FUNCtion.TRACKing

<i>Object Type</i>	Property (read/write)
<i>Data Type</i>	Boolean
<i>Target</i>	Marker <i>Mk</i> of the active trace of channel <i>Ch</i> , <i>Ch</i> : channel number 1–16 (see Table 1 on page 25) <i>Mk</i> : marker number 1–15, or reference marker number 16 (see Table 4 on page 74)
<i>Description</i>	The ON/OFF state of the marker search tracking function.
<i>Allowable Values</i>	True: Marker search tracking ON False: Marker search tracking OFF
<i>Preset Value</i>	False
<i>Syntax</i>	Dim <i>Status</i> As Boolean <i>Status</i> = <i>app</i> .SCPI.CALCulate(<i>Ch</i>).SELected.MARKer(<i>Mk</i>).FUNCtion.TRACKing <i>app</i> .SCPI.CALCulate(<i>Ch</i>).SELected.MARKer(<i>Mk</i>).FUNCtion.TRACKing = true
<i>Equivalent Softkeys</i>	Markers > Marker Search > Tracking

SCPI.CALCulate(*Ch*).SELected.MARKer(*Mk*).FUNCtion.TTRansition

<i>Object Type</i>	Property (read/write)
<i>Data Type</i>	String
<i>Target</i>	Marker <i>Mk</i> of the active trace of channel <i>Ch</i> , <i>Ch</i> : channel number 1–16 (see Table 1 on page 25) <i>Mk</i> : marker number 1–15, or reference marker number 16 (see Table 4 on page 74)
<i>Description</i>	The selection of the type of the target transition, when the marker search for transition is performed by the SCPI.CALCulate(<i>Ch</i>).SELected.MARKer(<i>Mk</i>).FUNCtion.EXECute method.
<i>Range</i>	"POSitive" : Positive target transition "NEGative" : Negative target transition "BOTH" : Both positive target transition and negative target transition
<i>Notes</i>	The short form of the parameter is indicated by upper case letters. There is no distinction between upper and lower case letters when the property is written. When the property is read out, the short format is indicated by upper case letters.
<i>Out of Range</i>	The value is ignored.
<i>Preset Value</i>	"POS"
<i>Syntax</i>	<p>Dim <i>Param</i> As String</p> <p><i>Param</i> = <i>app</i>.SCPI.CALCulate(<i>Ch</i>).SELected.MARKer(<i>Mk</i>).FUNCtion.TTRansition</p> <p><i>app</i>.SCPI.CALCulate(<i>Ch</i>).SELected.MARKer(<i>Mk</i>).FUNCtion.TTRansition = "neg"</p>
<i>Equivalent Softkeys</i>	Marker > Marker Search > Target > Target Transition

SCPI.CALCulate(*Ch*).SELected.MARKer(*Mk*).FUNCtion.TYPE

<i>Object Type</i>	Property (read/write)
<i>Data Type</i>	String
<i>Target</i>	Marker <i>Mk</i> of the active trace of channel <i>Ch</i> , <i>Ch</i> : channel number 1–16 (see Table 1 on page 25) <i>Mk</i> : marker number 1–15, or reference marker number 16 (see Table 4 on page 74)
<i>Description</i>	The selection of the type of the marker search, which is performed by the SCPI.CALCulate(<i>Ch</i>).SELected.MARKer(<i>Mk</i>).FUNCtion.EXECute method.
<i>Range</i>	"MAXimum" : Maximum value search "MINimum" : Minimum value search "PEAK" : Peak search "LPEak" : Peak search to the left from the marker "RPEak" : Peak search to the right from the marker "TARGet" : Target search "LTARget" : Target search to the left from the marker "RTARget" : Target search to the right from the marker
<i>Notes</i>	The short form of the parameter is indicated by upper case letters. There is no distinction between upper and lower case letters when the property is written. When the property is read out, the short format is indicated by upper case letters.
<i>Out of Range</i>	The value is ignored.
<i>Preset Value</i>	"MAX"
<i>Syntax</i>	Dim <i>Param</i> As String <i>Param</i> = app.SCPI.CALCulate(<i>Ch</i>).SELected.MARKer(<i>Mk</i>).FUNCtion.TYPE app.SCPI.CALCulate(<i>Ch</i>).SELected.MARKer(<i>Mk</i>).FUNCtion.TYPE = "MIN"
<i>Equivalent Softkeys</i>	Markers > Marker Search > Maximum Minimum Markers > Marker Search > Peak > Search Peak Search Max Peak Search Peak Left Search Peak Right Markers > Marker Search > Target > Search Target Search Target Left Search Target Right

SCPI.CALCulate(*Ch*).SELected.MARKer(1).MATH.FLATness.DATA

<i>Object Type</i>	Property (read only)								
<i>Data Type</i>	Variant (array of double)								
<i>Target</i>	The active trace of channel <i>Ch</i> , <i>Ch</i> : channel number 1–16 (see Table 1 on page 25)								
<i>Description</i>	<p>The FLATNESS function data array. The FLATNESS function is applied within the range determined by two markers (see SCPI.CALCulate(<i>Ch</i>).SELected.MARKer.MATH.FLATness.DOMain.START and SCPI.CALCulate(<i>Ch</i>).SELected.MARKer.MATH.FLATness.DOMain.STOP properties).</p> <p>The array includes 4 elements:</p> <table style="margin-left: 40px;"> <tr> <td><i>Data(0)</i></td> <td>Span;</td> </tr> <tr> <td><i>Data(1)</i></td> <td>Gain;</td> </tr> <tr> <td><i>Data(2)</i></td> <td>Slope;</td> </tr> <tr> <td><i>Data(3)</i></td> <td>Flatness.</td> </tr> </table>	<i>Data(0)</i>	Span;	<i>Data(1)</i>	Gain;	<i>Data(2)</i>	Slope;	<i>Data(3)</i>	Flatness.
<i>Data(0)</i>	Span;								
<i>Data(1)</i>	Gain;								
<i>Data(2)</i>	Slope;								
<i>Data(3)</i>	Flatness.								
<i>Syntax</i>	<p>Dim <i>Data</i> As Variant</p> <p><i>Data</i> = app.SCPI.CALCulate(<i>Ch</i>).SELected.MARKer.MATH.FLATness.DATA</p>								
<i>Warning</i>	Object MARKer has an index. In Visual Basic the index is 1 by default when it is omitted. The index can not be omitted in other programming languages.								
<i>Equivalent Softkeys</i>	None								

SCPI.CALCulate(*Ch*).SELected.MARKer(1).MATH.FLATness.DOMain.START

<i>Object Type</i>	Property (read/write)
<i>Data Type</i>	Long
<i>Target</i>	The active trace of channel <i>Ch</i> , <i>Ch</i> : channel number 1–16 (see Table 1 on page 25)
<i>Description</i>	The number of the marker, which specifies the start frequency of the FLATNESS function range.
<i>Range</i>	from 1 to 16
<i>Out of Range</i>	Sets the value of the limit, which is closer to the specified value.
<i>Preset Value</i>	1
<i>Syntax</i>	<pre>Dim MkrNum As Long MkrNum = app.SCPI.CALCulate(<i>Ch</i>).SELected.MARKer.MATH.FLATness.DOMain.START app.SCPI.CALCulate(<i>Ch</i>).SELected.MARKer.MATH.FLATness.DOMain.START = 1</pre>
<i>Warning</i>	Object <i>MARKer</i> has an index. In Visual Basic the index is 1 by default when it is omitted. The index can not be omitted in other programming languages.
<i>Equivalent Softkeys</i>	Markers > Marker Math > Flatness > Flatness Start

SCPI.CALCulate(*Ch*).SELected.MARKer(1).MATH.FLATness.DOMain.STOP

<i>Object Type</i>	Property (read/write)
<i>Data Type</i>	Long
<i>Target</i>	The active trace of channel <i>Ch</i> , <i>Ch</i> : channel number 1–16 (see Table 1 on page 25)
<i>Description</i>	The number of the marker, which specifies the stop frequency of the FLATNESS function range.
<i>Range</i>	from 1 to 16
<i>Out of Range</i>	Sets the value of the limit, which is closer to the specified value.
<i>Preset Value</i>	2
<i>Syntax</i>	<pre>Dim MkrNum As Long MkrNum = app.SCPI.CALCulate(<i>Ch</i>).SELected.MARKer.MATH.FLATness.DOMain.STOP app.SCPI.CALCulate(<i>Ch</i>).SELected.MARKer.MATH.FLATness.DOMain.STOP = 1</pre>
<i>Warning</i>	Object <i>MARKer</i> has an index. In Visual Basic the index is 1 by default when it is omitted. The index can not be omitted in other programming languages.
<i>Equivalent Softkeys</i>	Markers > Marker Math > Flatness > Flatness Stop

SCPI.CALCulate(*Ch*).SELected.MARKer(1).MATH.FLATness.STATE

<i>Object Type</i>	Property (read/write)
<i>Data Type</i>	Boolean
<i>Target</i>	The active trace of channel <i>Ch</i> , <i>Ch</i> : channel number 1–16 (see Table 1 on page 25)
<i>Description</i>	The ON/OFF state of the FLATNESS function.
<i>Allowable Values</i>	True: FLATNESS function ON False: FLATNESS function OFF
<i>Preset Value</i>	False
<i>Syntax</i>	<pre>Dim Status As Boolean Status = app.SCPI.CALCulate(<i>Ch</i>).SELected.MARKer.MATH.FLATness.STATE app.SCPI.CALCulate(<i>Ch</i>).SELected.MARKer.MATH.FLATness.STATE = true</pre>
<i>Warning</i>	Object <i>MARKer</i> has an index. In Visual Basic the index is 1 by default when it is omitted. The index can not be omitted in other programming languages.
<i>Equivalent Softkeys</i>	Markers > Marker Math > Flatness > Flatness

SCPI.CALCulate(*Ch*).SELected.MARKer(1).REFerence.STATE

<i>Object Type</i>	Property (read/write)
<i>Data Type</i>	Boolean
<i>Target</i>	The active trace of channel <i>Ch</i> , <i>Ch</i> : channel number 1–16 (see Table 1 on page 25)
<i>Description</i>	The ON/OFF state of the reference marker. When the reference marker is turned ON, all the values of the other markers turn to relative values.
<i>Allowable Values</i>	True: Reference marker ON False: Reference marker OFF
<i>Preset Value</i>	False
<i>Syntax</i>	Dim <i>Status</i> As Boolean <i>Status</i> = <i>app</i> .SCPI.CALCulate(<i>Ch</i>).SELected.MARKer.REFerence.STATE <i>app</i> .SCPI.CALCulate(<i>Ch</i>).SELected.MARKer.REFerence.STATE = true
<i>Warning</i>	Object MARKer has an index. In Visual Basic the index is 1 by default when it is omitted. The index can not be omitted in other programming languages.
<i>Equivalent Softkeys</i>	Markers > Reference Marker

SCPI.CALCulate(*Ch*).SELected.MARKer(*Mk*).POSIon**SCPI.CALCulate(*Ch*).SELected.MARKer(*Mk*)._SET**

<i>Object Type</i>	Property (write only)
<i>Data Type</i>	String
<i>Target</i>	Marker <i>Mk</i> of the active trace of channel <i>Ch</i> , <i>Ch</i> : channel number 1–16 (see Table 1 on page 25) <i>Mk</i> : marker number 1–15, or reference marker number 16 (see Table 4 on page 74)
<i>Description</i>	The setting of the value of the specified item to the value of the position of the marker.
<i>Range</i>	"STARt" : Sweep start value set to the stimulus value of the marker position. "STOP" : Sweep stop value set to the stimulus value of the marker position. "CENTER" : Sweep center value set to the stimulus value of the marker position. "RLEVel" : Reference value set to the response value of the marker position. "DELAY" : Delay value set to the response value of the marker position.
<i>Notes</i>	The short form of the parameter is indicated by upper case letters. There is no distinction between upper and lower case letters when the property is written. When the property is read out, the short format is indicated by upper case letters.
<i>Out of Range</i>	The value is ignored.
<i>Syntax</i>	<i>app</i> .SCPI.CALCulate(<i>Ch</i>).SELected.MARKer(<i>Mk</i>).POSIon = "STOP" <i>app</i> .SCPI.CALCulate(<i>Ch</i>).SELected.MARKer(<i>Mk</i>)._SET = "STOP"
<i>Equivalent Softkeys</i>	Markers > Marker Functions > Marker->Start Marker->Stop Marker->Center Marker->Ref Value Marker->Delay

SCPI.CALCulate(*Ch*).SELected.MARKer(*Mk*).STATe

<i>Object Type</i>	Property (read/write)
<i>Data Type</i>	Boolean
<i>Target</i>	Marker <i>Mk</i> of the active trace of channel <i>Ch</i> , <i>Ch</i> : channel number 1–16 (see Table 1 on page 25) <i>Mk</i> : marker number 1–15, or reference marker number 16 (see Table 4 on page 74)
<i>Description</i>	The ON/OFF state of a marker. Turning ON a marker with the number from 1 to 15 will turn ON all the markers of smaller numbers. Turning OFF a marker with the number from 1 to 15 will turn OFF all the markers of greater numbers (except for the reference marker). Turning ON/OFF the reference marker with number 16 does not turn ON/OFF the markers with the numbers from 1 to 15, but switches these markers to the relative measurement mode.
<i>Allowable Values</i>	True: Marker ON False: Marker OFF
<i>Preset Value</i>	False
<i>Syntax</i>	Dim <i>Status</i> As Boolean <i>Status</i> = <i>app</i> .SCPI.CALCulate(<i>Ch</i>).SELected.MARKer(<i>Mk</i>).STATe <i>app</i> .SCPI.CALCulate(<i>Ch</i>).SELected.MARKer(<i>Mk</i>).STATe = true
<i>Equivalent Softkeys</i>	Markers > Add Marker Remove Marker Markers > Reference Marker

SCPI.CALCulate(*Ch*).SELected.MARKer(*Mk*).X

<i>Object Type</i>	Property (read/write)
<i>Data Type</i>	Double
<i>Target</i>	Marker <i>Mk</i> of the active trace of channel <i>Ch</i> , <i>Ch</i> : channel number 1–16 (see Table 1 on page 25) <i>Mk</i> : marker number 1–15, or reference marker number 16 (see Table 4 on page 74)
<i>Description</i>	The stimulus value of the marker.
<i>Range</i>	From the stimulus current start value to the stimulus current stop value.
<i>Out of Value</i>	Sets the value of the limit, which is closer to the specified value.
<i>Preset Value</i>	Stimulus center value
<i>Unit</i>	Hz (Hertz) s (second) dBm (decibels above 1 milliwatt)
<i>Syntax</i>	<p>Dim <i>Value</i> As Double <i>Value</i> = <i>app</i>.SCPI.CALCulate(<i>Ch</i>).SELected.MARKer(<i>Mk</i>).X <i>app</i>.SCPI.CALCulate(<i>Ch</i>).SELected.MARKer(<i>Mk</i>).X = 1e9</p>
<i>Equivalent Softkeys</i>	Markers > Edit Stimulus

SCPI.CALCulate(*Ch*).SELected.MARKer(*Mk*).Y

<i>Object Type</i>	Property (read only)				
<i>Data Type</i>	Variant (array of double)				
<i>Target</i>	Marker <i>Mk</i> of the active trace of channel <i>Ch</i> , <i>Ch</i> : channel number 1–16 (see Table 1 on page 25) <i>Mk</i> : marker number 1–15, or reference marker number 16 (see Table 4 on page 74)				
<i>Description</i>	The response value of the marker. If the reference marker is turned ON, the values of the markers from 1 to 15 are read out as relative values to the reference marker. The array includes 2 elements: <table style="margin-left: 40px;"> <tr> <td><i>Data(0)</i></td> <td>real number in rectangular format, real part in polar and Smith chart formats;</td> </tr> <tr> <td><i>Data(1)</i></td> <td>0 in rectangular format, imaginary part in polar and Smith chart formats.</td> </tr> </table>	<i>Data(0)</i>	real number in rectangular format, real part in polar and Smith chart formats;	<i>Data(1)</i>	0 in rectangular format, imaginary part in polar and Smith chart formats.
<i>Data(0)</i>	real number in rectangular format, real part in polar and Smith chart formats;				
<i>Data(1)</i>	0 in rectangular format, imaginary part in polar and Smith chart formats.				
<i>Syntax</i>	Dim <i>Data</i> As Variant <i>Data</i> = <i>app</i> .SCPI.CALCulate(<i>Ch</i>).SELected.MARKer(<i>Mk</i>).Y				
<i>Equivalent Softkeys</i>	None				

SCPI.CALCulate(*Ch*).SELected.MATH.FUNction

<i>Object Type</i>	Property (read/write)
<i>Data Type</i>	String
<i>Target</i>	The active trace of channel <i>Ch</i> , <i>Ch</i> : channel number 1–16 (see Table 1 on page 25)
<i>Description</i>	The selection of the math operation between the measurement data and the memory trace data. The math result replaces the data trace. If the data trace is not saved, the command is ignored.
<i>Range</i>	"DIVide" : Division <i>Data / Mem.</i> "MULTiply" : Multiplication <i>Data x Mem.</i> "ADD" : Addition <i>Data + Mem.</i> "SUBTract" : Subtraction <i>Data – Mem.</i> "NORMal" : No math
<i>Notes</i>	The short form of the parameter is indicated by upper case letters. There is no distinction between upper and lower case letters when the property is written. When the property is read out, the short format is indicated by upper case letters.
<i>Out of Range</i>	An error occurs. Error code 210.
<i>Preset Value</i>	"NORM"
<i>Syntax</i>	Dim <i>Param</i> As String <i>Param</i> = <i>app</i> .SCPI.CALCulate(<i>Ch</i>).SELected.MATH.FUNction <i>app</i> .SCPI.CALCulate(<i>Ch</i>).SELected.MATH.FUNction= "DIV"
<i>Equivalent Softkeys</i>	Display > Data Math > Data/Mem Data*Mem Data+Mem Data-Mem OFF

SCPI.CALCulate(*Ch*).SELected.MATH.MEMorize

<i>Object Type</i>	Method
<i>Target</i>	The active trace of channel <i>Ch</i> , <i>Ch</i> : channel number 1–16 (see Table 1 on page 25)
<i>Description</i>	Saves the measurement data to the memory trace. Automatically turns on the display of the memory trace.
<i>Syntax</i>	<i>app</i> .SCPI.CALCulate(<i>Ch</i>).SELected.MATH.MEMorize
<i>Equivalent Softkeys</i>	Display > Data->Memory

SCPI.CALCulate(*Ch*).SELected.MSTatistics.DATA

<i>Object Type</i>	Property (read only)
<i>Data Type</i>	Variant (array of double)
<i>Target</i>	The active trace of channel <i>Ch</i> , <i>Ch</i> : channel number 1–16 (see Table 1 on page 25)
<i>Description</i>	<p>The math statistics data array. The statistics function is applied either over the whole range (for all the trace), or within the range specified by the SCPI.CALCulate(<i>Ch</i>).SELected.MSTatistics.DOMain.STATE property (the range limits are determined by two markers).</p> <p>The array includes 3 elements:</p> <ul style="list-style-type: none"> <i>Data(0)</i> Mean value; <i>Data(1)</i> Standard deviation; <i>Data(2)</i> Peak-to-peak (difference between the maximum value and the minimum value).
<i>Syntax</i>	<pre>Dim <i>Data</i> As Variant <i>Data</i> = <i>app</i>.SCPI.CALCulate(<i>Ch</i>).SELected.MSTatistics.DATA</pre>
<i>Equivalent Softkeys</i>	None

SCPI.CALCulate(*Ch*).SELected.MSTatistics.DOMain.MARKer.START

<i>Object Type</i>	Property (read/write)
<i>Data Type</i>	Long
<i>Target</i>	The active trace of channel <i>Ch</i> , <i>Ch</i> : channel number 1–16 (see Table 1 on page 25)
<i>Description</i>	The number of the marker, which specifies the start frequency of the math statistics range.
<i>Range</i>	from 1 to 16
<i>Out of Range</i>	Sets the value of the limit, which is closer to the specified value.
<i>Preset Value</i>	1
<i>Syntax</i>	<p>Dim <i>MkrNum</i> As Long</p> <pre><i>MkrNum</i> = app.SCPI.CALCulate(<i>Ch</i>).SELected. MSTatistics.DOMain.MARKer.START app.SCPI.CALCulate(<i>Ch</i>).SELected. MSTatistics.DOMain.MARKer.START = 3</pre>
<i>Equivalent Softkeys</i>	Markers > Marker Math > Statistics > Statistics Start

SCPI.CALCulate(*Ch*).SELected.MSTatistics.DOMain.MARKer.STOP

<i>Object Type</i>	Property (read/write)
<i>Data Type</i>	Long
<i>Target</i>	The active trace of channel <i>Ch</i> , <i>Ch</i> : channel number 1–16 (see Table 1 on page 25)
<i>Description</i>	The number of the marker, which specifies the stop frequency of the math statistics range.
<i>Range</i>	from 1 to 16
<i>Out of Range</i>	Sets the value of the limit, which is closer to the specified value.
<i>Preset Value</i>	2
<i>Syntax</i>	<pre>Dim MarkerNum As Long MarkerNum = app.SCPI.CALCulate(<i>Ch</i>).SELected. MSTatistics.DOMain.MARKer.STOP app.SCPI.CALCulate(<i>Ch</i>).SELected. MSTatistics.DOMain.MARKer.STOP = 4</pre>
<i>Equivalent Softkeys</i>	Markers > Marker Math > Statistics > Statistics Stop

SCPI.CALCulate(*Ch*).SELected.MStatistics.DOMain.STATE

<i>Object Type</i>	Property (read/write)
<i>Data Type</i>	Boolean
<i>Target</i>	The active trace of channel <i>Ch</i> , <i>Ch</i> : channel number 1–16 (see Table 1 on page 25)
<i>Description</i>	The ON/OFF state of the math statistics range.
<i>Allowable Values</i>	True: Statistics range ON False: Statistics range OFF
<i>Preset Value</i>	False
<i>Syntax</i>	<p>Dim <i>Status</i> As Boolean</p> <p><i>Status</i> = <i>app</i>.SCPI.CALCulate(<i>Ch</i>).SELected. MStatistics.DOMain.STATE</p> <p><i>app</i>.SCPI.CALCulate(<i>Ch</i>).SELected. MStatistics.DOMain.STATE = true</p>
<i>Equivalent Softkeys</i>	Markers > Marker Math > Statistics > Statistics Range

SCPI.CALCulate(*Ch*).SELected.MStatistics.STATE

<i>Object Type</i>	Property (read/write)
<i>Data Type</i>	Boolean
<i>Target</i>	The active trace of channel <i>Ch</i> , <i>Ch</i> : channel number 1–16 (see Table 1 on page 25)
<i>Description</i>	The ON/OFF state of the math statistics display.
<i>Allowable Values</i>	True: Statistics display ON False: Statistics display OFF
<i>Preset Value</i>	False
<i>Syntax</i>	<p>Dim <i>Status</i> As Boolean</p> <p><i>Status</i> = <i>app</i>.SCPI.CALCulate(<i>Ch</i>).SELected.MStatistics.STATE</p> <p><i>app</i>.SCPI.CALCulate(<i>Ch</i>).SELected.MStatistics.STATE = true</p>
<i>Equivalent Softkeys</i>	Markers > Marker Math > Statistics > Statistics

SCPI.CALCulate(*Ch*).SELected.RLIMit.DATA

<i>Object Type</i>	Property (read/write)
<i>Data Type</i>	Variant (array of double)
<i>Target</i>	The active trace of channel <i>Ch</i> , <i>Ch</i> : channel number 1–16 (see Table 1 on page 25)
<i>Description</i>	<p>The data array, which is the limit line for the ripple limit function. The array size is $1 + 4N$, where N is the number of limit line segments.</p> <p>For the n-th point, where n from 1 to N:</p> <ul style="list-style-type: none"> <i>Data(0)</i> the number of limit line segments N is the integer from 0 to 12. Setting 0 clears the limit line; <i>Data(4n-3)</i> type of the n-th limit line segment; 0: Off 1: On <i>Data(4n-2)</i> the stimulus value in the beginning point of the n-th segment; <i>Data(4n-1)</i> the stimulus value in the end point of the n-th segment; <i>Data(4n-0)</i> the ripple limit value of the n-th segment.
<i>Notes</i>	If the array size is not $1 + 4N$, where N is <i>Data(0)</i> , an error occurs (error code 214). If <i>Data(4n - 3)</i> is less than 0 or more than 1, an error occurs (error code 214). When <i>Data(4n-2)</i> , <i>Data(4n-1)</i> , and <i>Data(4n-0)</i> elements are out of allowable range, the value is set to the limit, which is closer to the specified value.
<i>Syntax</i>	<pre>Dim Data As Variant Data = app.SCPI.CALCulate(<i>Ch</i>).SELected.RLIMit.DATA app.SCPI.CALCulate(<i>Ch</i>).SELected.RLIMit.DATA = Array(1,1,800,900,10)</pre>
<i>Equivalent Softkeys</i>	Analysis > Ripple Limit > Edit Ripple Limit

SCPI.CALCulate(*Ch*).SELected.RLIMit.DISPlay.LINE

<i>Object Type</i>	Property (read/write)
<i>Data Type</i>	Boolean
<i>Target</i>	The active trace of channel <i>Ch</i> , <i>Ch</i> : channel number 1–16 (see Table 1 on page 25)
<i>Description</i>	The ON/OFF state of the ripple limit line display.
<i>Allowable Values</i>	True: Ripple limit line ON False: Ripple limit line OFF
<i>Preset Value</i>	False
<i>Syntax</i>	<p>Dim <i>Status</i> As Boolean</p> <p><i>Status</i> = <i>app</i>.SCPI.CALCulate(<i>Ch</i>).SELected.RLIMit.DISPlay.LINE</p> <p><i>app</i>.SCPI.CALCulate(<i>Ch</i>).SELected.RLIMit.DISPlay.LINE = true</p>
<i>Equivalent Softkeys</i>	Analysis > Ripple Limit > Ripple Limit

SCPI.CALCulate(*Ch*).SELected.RLIMit.DISPlay.SElect

<i>Object Type</i>	Property (read/write)
<i>Data Type</i>	Long
<i>Target</i>	The active trace of channel <i>Ch</i> , <i>Ch</i> : channel number 1–16 (see Table 1 on page 25)
<i>Description</i>	The number of the ripple limit test band selected for the ripple value display.
<i>Range</i>	from 1 to 12
<i>Out of Range</i>	Sets the value of the limit, which is closer to the specified value.
<i>Preset Value</i>	1
<i>Syntax</i>	<p>Dim <i>Value</i> As Long</p> <p><i>Value</i> = <i>app</i>.SCPI.CALCulate(<i>Ch</i>).SELected.RLIMit.DISPlay.SElect</p> <p><i>app</i>.SCPI.CALCulate(<i>Ch</i>).SELected.RLIMit.DISPlay.SElect = 2</p>
<i>Equivalent Softkeys</i>	Analysis > Ripple Limit > Ripple Value Band

SCPI.CALCulate(*Ch*).SELected.RLIMit.DISPlay.VALue

<i>Object Type</i>	Property (read/write)
<i>Data Type</i>	String
<i>Target</i>	The active trace of channel <i>Ch</i> , <i>Ch</i> : channel number 1–16 (see Table 1 on page 25)
<i>Description</i>	The selection of the display type of the ripple value in the specified band.
<i>Range</i>	"OFF" : Ripple value display OFF "ABSolute" : Absolute value "MARgin" : Margin (difference between the ripple limit and the absolute value)
<i>Notes</i>	The short form of the parameter is indicated by upper case letters. There is no distinction between upper and lower case letters when the property is written. When the property is read out, the short format is indicated by upper case letters.
<i>Out of Range</i>	The value is ignored.
<i>Preset Value</i>	"OFF"
<i>Syntax</i>	<pre>Dim Param As String Param = app.SCPI.CALCulate(<i>Ch</i>).SELected.RLIMit.DISPlay.VALue app.SCPI.CALCulate(<i>Ch</i>).SELected.RLIMit.DISPlay.VALue = "ABS"</pre>
<i>Equivalent Softkeys</i>	Analysis > Ripple Limit > Ripple Value

SCPI.CALCulate(*Ch*).SELected.RLIMit.FAIL

<i>Object Type</i>	Property (read only)
<i>Data Type</i>	Boolean
<i>Target</i>	The active trace of channel <i>Ch</i> , <i>Ch</i> : channel number 1–16 (see Table 1 on page 25)
<i>Description</i>	Ripple limit test result.
<i>Allowable Values</i>	True: Fail False: Pass
<i>Syntax</i>	Dim <i>Status</i> As Boolean <i>Status</i> = <i>app</i> .SCPI.CALCulate(<i>Ch</i>).SELected.RLIMit.FAIL
<i>Equivalent Softkeys</i>	None

SCPI.CALCulate(*Ch*).SELected.RLIMit.REPort.DATA

<i>Object Type</i>	Property (read only)								
<i>Data Type</i>	Variant (array of double)								
<i>Target</i>	The active trace of channel <i>Ch</i> , <i>Ch</i> : channel number 1–16 (see Table 1 on page 25)								
<i>Description</i>	<p>The data array, which is the ripple limit test results. The array size is 1+3N, where N is the number of ripple limit bands.</p> <p>For the n-th point, where n from 1 to N:</p> <table style="margin-left: 40px;"> <tr> <td><i>Data(0)</i></td> <td>N total number of the bands;</td> </tr> <tr> <td><i>Data(3n-2)</i></td> <td>n number of the band;</td> </tr> <tr> <td><i>Data(3n-1)</i></td> <td>Ripple value in the n-th band;</td> </tr> <tr> <td><i>Data(3n-0)</i></td> <td>Ripple limit test result in the n-th band: 0: Pass 1: Fail</td> </tr> </table>	<i>Data(0)</i>	N total number of the bands;	<i>Data(3n-2)</i>	n number of the band;	<i>Data(3n-1)</i>	Ripple value in the n-th band;	<i>Data(3n-0)</i>	Ripple limit test result in the n-th band: 0: Pass 1: Fail
<i>Data(0)</i>	N total number of the bands;								
<i>Data(3n-2)</i>	n number of the band;								
<i>Data(3n-1)</i>	Ripple value in the n-th band;								
<i>Data(3n-0)</i>	Ripple limit test result in the n-th band: 0: Pass 1: Fail								
<i>Syntax</i>	<pre>Dim <i>Data</i> As Variant <i>Data</i> = <i>app</i>.SCPI.CALCulate(<i>Ch</i>).SELected.RLIMit.REPort.DATA</pre>								
<i>Equivalent Softkeys</i>	None								

SCPI.CALCulate(*Ch*).SELected.RLIMit.STATE

<i>Object Type</i>	Property (read/write)
<i>Data Type</i>	Boolean
<i>Target</i>	The active trace of channel <i>Ch</i> , <i>Ch</i> : channel number 1–16 (see Table 1 on page 25)
<i>Description</i>	The ON/OFF state of the ripple limit test.
<i>Allowable Values</i>	True: Ripple limit test ON False: Ripple limit test OFF
<i>Preset Value</i>	False
<i>Syntax</i>	<p>Dim <i>Status</i> As Boolean</p> <p><i>Status</i> = <i>app</i>.SCPI.CALCulate(<i>Ch</i>).SELected.RLIMit.STATE</p> <p><i>app</i>.SCPI.CALCulate(<i>Ch</i>).SELected.RLIMit.STATE = true</p>
<i>Equivalent Softkeys</i>	Analysis > Ripple Limit > Ripple Test

SCPI.CALCulate(*Ch*).SELected.SMOothing.APERture

<i>Object Type</i>	Property (read/write)
<i>Data Type</i>	Double
<i>Target</i>	The active trace of channel <i>Ch</i> , <i>Ch</i> : channel number 1–16 (see Table 1 on page 25)
<i>Description</i>	The smoothing aperture for the smoothing function.
<i>Range</i>	from 0.01 to 20
<i>Out of Range</i>	Sets the value of the limit, which is closer to the specified value.
<i>Preset Value</i>	1
<i>Unit</i>	%
<i>Syntax</i>	<p>Dim <i>Value</i> As Double</p> <p><i>Value</i> = <i>app</i>.SCPI.CALCulate(<i>Ch</i>).SELected.SMOothing.APERture</p> <p><i>app</i>.SCPI.CALCulate(<i>Ch</i>).SELected.SMOothing.APERture = 1.5</p>
<i>Equivalent Softkeys</i>	Average > Smo Aperture

SCPI.CALCulate(*Ch*).SELected.SMOothing.STATE

<i>Object Type</i>	Property (read/write)
<i>Data Type</i>	Boolean
<i>Target</i>	The active trace of channel <i>Ch</i> , <i>Ch</i> : channel number 1–16 (see Table 1 on page 25)
<i>Description</i>	The ON/OFF state of the trace smoothing function.
<i>Allowable Values</i>	True: Trace smoothing ON False: Trace smoothing OFF
<i>Preset Value</i>	False
<i>Syntax</i>	<p>Dim <i>Status</i> As Boolean</p> <pre><i>Status</i> = app.SCPI.CALCulate(<i>Ch</i>).SELected.SMOothing.STATE app.SCPI.CALCulate(<i>Ch</i>).SELected.SMOothing.STATE = true</pre>
<i>Equivalent Softkeys</i>	Average > Smoothing

SCPI.CALCulate(*Ch*).SELected.TRANSform.TIME.CENTer

<i>Object Type</i>	Property (read/write)
<i>Data Type</i>	Double
<i>Target</i>	The active trace of channel <i>Ch</i> , <i>Ch</i> : channel number 1–16 (see Table 1 on page 25)
<i>Description</i>	The time (distance) domain center value, when the time domain transformation function is turned ON.
<i>Range</i>	Varies depending on the specified frequency range and the number of points.
<i>Out of Range</i>	Sets the value of the limit, which is closer to the specified value.
<i>Preset Value</i>	0
<i>Unit</i>	s (seconds), m (meters) or ft (feet)
<i>Syntax</i>	<p>Dim <i>Value</i> As Double</p> <p><i>Value</i> = <i>app</i>.SCPI.CALCulate(<i>Ch</i>).SELected.TRANSform.TIME.CENTer</p> <p><i>app</i>.SCPI.CALCulate(<i>Ch</i>).SELected.TRANSform.TIME.CENTer = 1e–8</p>
<i>Related Commands</i>	SCPI.CALCulate(<i>Ch</i>).SELected.TRANSform.TIME.UNIT
<i>Equivalent Softkeys</i>	Analysis > Time Domain > Center

SCPI.CALCulate(*Ch*).SELected.TRANSform.TIME.IMPulse.WIDTh

<i>Object Type</i>	Property (read/write)
<i>Data Type</i>	Double
<i>Target</i>	The active trace of channel <i>Ch</i> , <i>Ch</i> : channel number 1–16 (see Table 1 on page 25)
<i>Description</i>	The impulse width (time domain transformation resolution), coupled with the Kaiser–Bessel window shape β parameter. The impulse width setting changes the β parameter, and setting of β parameter changes the impulse width.
<i>Range</i>	Varies depending on the specified frequency range and the number of points.
<i>Out of Range</i>	Sets the value of the limit, which is closer to the specified value.
<i>Preset Value</i>	None
<i>Unit</i>	s (second)
<i>Syntax</i>	<p>Dim <i>Value</i> As Double</p> <pre>Value = app.SCPI.CALCulate(<i>Ch</i>).SELected.TRANSform.TIME.IMPulse.WIDTh app.SCPI.CALCulate(<i>Ch</i>).SELected.TRANSform.TIME.IMPulse.WIDTh = 1e-8</pre>
<i>Equivalent Softkeys</i>	Analysis > Time Domain > Window > Impulse Width (when the transformation type is set to Bandpass or Lowpass Impulse)

SCPI.CALCulate(*Ch*).SELected.TRANSform.TIME.KBESsel

<i>Object Type</i>	Property (read/write)
<i>Data Type</i>	Double
<i>Target</i>	The active trace of channel <i>Ch</i> , <i>Ch</i> : channel number 1–16 (see Table 1 on page 25)
<i>Description</i>	The β parameter, which controls the Kaiser–Bessel window shape, when performing time domain transformation.
<i>Range</i>	from 0 to 13
<i>Out of Range</i>	Sets the value of the limit, which is closer to the specified value.
<i>Preset Value</i>	6
<i>Unit</i>	s (second)
<i>Syntax</i>	<p>Dim <i>Value</i> As Double</p> <pre>Value = app.SCPI.CALCulate(<i>Ch</i>).SELected.TRANSform.TIME.KBESsel app.SCPI.CALCulate(<i>Ch</i>).SELected.TRANSform.TIME.KBESsel = 13</pre>
<i>Equivalent Softkeys</i>	Analysis > Time Domain > Window > Kaiser Beta

SCPI.CALCulate(*Ch*).SELected.TRANSform.TIME.LPFREquency

<i>Object Type</i>	Method
<i>Target</i>	The active trace of channel <i>Ch</i> , <i>Ch</i> : channel number 1–16 (see Table 1 on page 25)
<i>Description</i>	Changes the frequency range to match with the lowpass type of the time domain transformation function.
<i>Syntax</i>	<i>app</i> .SCPI.CALCulate(<i>Ch</i>).SELected.TRANSform.TIME.LPFREquency
<i>Equivalent Softkeys</i>	Analysis > Time Domain > Set Frequency Low Pass

SCPI.CALCulate(*Ch*).SELected.TRANSform.TIME.REFLection.TYPE

<i>Object Type</i>	Property (read/write)
<i>Data Type</i>	String
<i>Target</i>	The active trace of channel <i>Ch</i> , <i>Ch</i> : channel number 1–16 (see Table 1 on page 25)
<i>Description</i>	The selection of the reflection distance either one way or round trip for the time domain transformation function.
<i>Range</i>	"RTRip" Round Trip "OWAY" One Way
<i>Notes</i>	The short form of the parameter is indicated by upper case letters. There is no distinction between upper and lower case letters when the property is written. When the property is read out, the short format is indicated by upper case letters.
<i>Out of Range</i>	The value is ignored.
<i>Preset Value</i>	"RTRip"
<i>Syntax</i>	<p>Dim <i>Param</i> As String</p> <pre>Param = app.SCPI.CALCulate(<i>Ch</i>).SELected.TRANSform.TIME.REFLection.TYPE app.SCPI.CALCulate(<i>Ch</i>).SELected.TRANSform.TIME.REFLection.TYPE = "RTR"</pre>
<i>Equivalent Softkeys</i>	Analysis > Time Domain >Reflection Type > Round Trip One Way

SCPI.CALCulate(*Ch*).SELected.TRANSform.TIME.SPAN

<i>Object Type</i>	Property (read/write)
<i>Data Type</i>	Double
<i>Target</i>	The active trace of channel <i>Ch</i> , <i>Ch</i> : channel number 1–16 (see Table 1 on page 25)
<i>Description</i>	The time (distance) domain span value, when the time domain transformation function is turned ON.
<i>Range</i>	Varies depending on the specified frequency range and the number of points.
<i>Out of Range</i>	Sets the value of the limit, which is closer to the specified value.
<i>Unit</i>	s (seconds), m (meters) or ft (feet)
<i>Syntax</i>	<p>Dim Value As Double</p> <p><i>Value</i> = <i>app</i>.SCPI.CALCulate(<i>Ch</i>).SELected.TRANSform.TIME.SPAN</p> <p><i>app</i>. SCPI.CALCulate(<i>Ch</i>).SELected.TRANSform.TIME.SPAN = 1e-8</p>
<i>Related Commands</i>	SCPI.CALCulate(<i>Ch</i>).SELected.TRANSform.TIME.UNIT
<i>Equivalent Softkeys</i>	Analysis > Time Domain > Span

SCPI.CALCulate(*Ch*).SELected.TRANSform.TIME.START

<i>Object Type</i>	Property (read/write)
<i>Data Type</i>	Double
<i>Target</i>	The active trace of channel <i>Ch</i> , <i>Ch</i> : channel number 1–16 (see Table 1 on page 25)
<i>Description</i>	The time (distance) domain start value, when the time domain transformation function is turned ON.
<i>Range</i>	Varies depending on the specified frequency range and the number of points.
<i>Out of Range</i>	Sets the value of the limit, which is closer to the specified value.
<i>Unit</i>	s (seconds), m (meters) or ft (feet)
<i>Syntax</i>	<p>Dim Value As Double</p> <p><i>Value</i> = <i>app</i>.SCPI.CALCulate(<i>Ch</i>).SELected.TRANSform.TIME.START</p> <p><i>app</i>.SCPI.CALCulate(<i>Ch</i>).SELected.TRANSform.TIME.START = 1e-8</p>
<i>Related Commands</i>	SCPI.CALCulate(<i>Ch</i>).SELected.TRANSform.TIME.UNIT
<i>Equivalent Softkeys</i>	Analysis > Time Domain > Start

SCPI.CALCulate(*Ch*).SELected.TRANSform.TIME.STATE

<i>Object Type</i>	Property (read/write)
<i>Data Type</i>	Boolean
<i>Target</i>	The active trace of channel <i>Ch</i> , <i>Ch</i> : channel number 1–16 (see Table 1 on page 25)
<i>Description</i>	The ON/OFF state of the time domain transformation function.
<i>Allowable Values</i>	True: Time domain transformation ON False: Time domain transformation OFF
<i>Preset Value</i>	False
<i>Syntax</i>	<p>Dim <i>Status</i> As Boolean</p> <p><i>Status</i> = <i>app</i>.SCPI.CALCulate(<i>Ch</i>).SELected.TRANSform.TIME.STATE</p> <p><i>app</i>.SCPI.CALCulate(<i>Ch</i>).SELected.TRANSform.TIME.STATE = true</p>
<i>Equivalent Softkeys</i>	Analysis > Time Domain > Time Domain

SCPI.CALCulate(*Ch*).SELected.TRANSform.TIME.STEP.RTIme

<i>Object Type</i>	Property (read/write)
<i>Data Type</i>	Double
<i>Target</i>	The active trace of channel <i>Ch</i> , <i>Ch</i> : channel number 1–16 (see Table 1 on page 25)
<i>Description</i>	The rise time of the step signal (time domain transformation resolution), coupled with the Kaiser-Bessel window shape β parameter. The impulse width setting changes the β parameter, and setting of β parameter changes the impulse width.
<i>Range</i>	Varies depending on the specified frequency range and the number of points.
<i>Out of Range</i>	Sets the value of the limit, which is closer to the specified value.
<i>Preset Value</i>	None
<i>Unit</i>	s (second)
<i>Syntax</i>	<p>Dim <i>Value</i> As Double</p> <pre>Value = app.SCPI.CALCulate(<i>Ch</i>).SELected.TRANSform.TIME.IMPulse.WIDTh app.SCPI.CALCulate(<i>Ch</i>).SELected.TRANSform.TIME.IMPulse.WIDTh = 1e-8</pre>
<i>Equivalent Softkeys</i>	Analysis > Time Domain > Window > Impulse Width (when the transformation type is set to Lowpass Step)

SCPI.CALCulate(*Ch*).SELected.TRANSform.TIME.STIMulus

<i>Object Type</i>	Property (read/write)
<i>Data Type</i>	String
<i>Target</i>	The active trace of channel <i>Ch</i> , <i>Ch</i> : channel number 1–16 (see Table 1 on page 25)
<i>Description</i>	The selection of the stimulus type for the time domain transformation function: impulse or step.
<i>Range</i>	"IMPulse" Impulse "STEP" Step
<i>Notes</i>	The short form of the parameter is indicated by upper case letters. There is no distinction between upper and lower case letters when the property is written. When the property is read out, the short format is indicated by upper case letters.
<i>Out of Range</i>	The value is ignored.
<i>Preset Value</i>	"IMP"
<i>Syntax</i>	Dim <i>Param</i> As String <i>Param</i> = <i>app</i> .SCPI.CALCulate(<i>Ch</i>).SELected.TRANSform.TIME.STIMulus <i>app</i> .SCPI.CALCulate(<i>Ch</i>).SELected.TRANSform.TIME.STIMulus = "STEP"
<i>Equivalent Softkeys</i>	Analysis > Time Domain > Type > Bandpass Lowpass Step Lowpass Impulse

SCPI.CALCulate(*Ch*).SELected.TRANSform.TIME.STOP

<i>Object Type</i>	Property (read/write)
<i>Data Type</i>	Double
<i>Target</i>	The active trace of channel <i>Ch</i> , <i>Ch</i> : channel number 1–16 (see Table 1 on page 25)
<i>Description</i>	The time (distance) domain stop value, when the time domain transformation function is turned ON.
<i>Range</i>	Varies depending on the specified frequency range and the number of points.
<i>Out of Range</i>	Sets the value of the limit, which is closer to the specified value.
<i>Unit</i>	s (seconds), m (meters) or ft (feet)
<i>Syntax</i>	<p>Dim <i>Value</i> As Double <i>Value</i> = <i>app</i>.SCPI.CALCulate(<i>Ch</i>).SELected.TRANSform.TIME.STOP <i>app</i>.SCPI.CALCulate(<i>Ch</i>).SELected.TRANSform.TIME.STOP = 2e-8</p>
<i>Related Commands</i>	SCPI.CALCulate(<i>Ch</i>).SELected.TRANSform.TIME.UNIT
<i>Equivalent Softkeys</i>	Analysis > Time Domain > Stop

SCPI.CALCulate(*Ch*).SELected.TRANSform.TIME.TYPE

<i>Object Type</i>	Property (read/write)
<i>Data Type</i>	String
<i>Target</i>	The active trace of channel <i>Ch</i> , <i>Ch</i> : channel number 1–16 (see Table 1 on page 25)
<i>Description</i>	The selection of the transformation type for the time domain transformation function: bandpass response or direct current circuit.
<i>Range</i>	"BPAsS" Bandpass "LPAsS" Lowpass
<i>Notes</i>	The short form of the parameter is indicated by upper case letters. There is no distinction between upper and lower case letters when the property is written. When the property is read out, the short format is indicated by upper case letters.
<i>Out of Range</i>	The value is ignored.
<i>Preset Value</i>	"BPAsS"
<i>Syntax</i>	<pre>Dim Param As String Param = app.SCPI.CALCulate(<i>Ch</i>).SELected.TRANSform.TIME.TYPE app.SCPI.CALCulate(<i>Ch</i>).SELected.TRANSform.TIME.TYPE = "STEP"</pre>
<i>Equivalent Softkeys</i>	Analysis > Time Domain > Type > Bandpass Lowpass Step Lowpass Impulse

SCPI.CALCulate(*Ch*).SELected.TRANSform.TIME.UNIT

<i>Object Type</i>	Property (read/write)
<i>Data Type</i>	String
<i>Target</i>	The active trace of channel <i>Ch</i> , <i>Ch</i> : channel number 1–16 (see Table 1 on page 25)
<i>Description</i>	The selection of the transformation unit for the time domain transformation function: seconds, meters, feet.
<i>Range</i>	"SEConds" Seconds "METers" Meters "FEET" Feet
<i>Notes</i>	The short form of the parameter is indicated by upper case letters. There is no distinction between upper and lower case letters when the property is written. When the property is read out, the short format is indicated by upper case letters.
<i>Out of Range</i>	The value is ignored.
<i>Preset Value</i>	"SEConds"
<i>Syntax</i>	Dim <i>Param</i> As String <i>Param</i> = <i>app</i> .SCPI.CALCulate(<i>Ch</i>).SELected.TRANSform.TIME.UNIT <i>app</i> .SCPI.CALCulate(<i>Ch</i>).SELected.TRANSform.TIME.UNIT = "MET"
<i>Equivalent Softkeys</i>	Analysis > Time Domain > Unit > Seconds Meters Feet

SCPI.CALCulate(*Ch*).TRACe(*Tr*).DATA.FDATA

<i>Object Type</i>	Property (read/write)
<i>Data Type</i>	Variant (array of double)
<i>Target</i>	The specified trace <i>Tr</i> of channel <i>Ch</i> , <i>Tr</i> : trace number 1–16 (see Table 3 on page 33) <i>Ch</i> : channel number 1–16 (see Table 1 on page 25)
<i>Description</i>	The formatted data array. The array elements contain measurements in the current format, for example, in logarithmic magnitude format (Log Mag). Also, see section “Measurement Data Arrays” on page 19. The array size is 2N, where N is the number of measurement points. For the n-th point, where n from 1 to N: <i>Data</i> (2n-2) real number in rectangular format, real part in polar and Smith chart formats; <i>Data</i> (2n-1) 0 in rectangular format, imaginary part in polar and Smith chart formats.
<i>Syntax</i>	Dim <i>Data</i> As Variant <i>Data</i> = <i>app</i> .SCPI.CALCulate(<i>Ch</i>).Trace(<i>Tr</i>).DATA.FDATA <i>app</i> .SCPI.CALCulate(<i>Ch</i>).Trace(<i>Tr</i>).DATA.FDATA = <i>Data</i>
<i>Equivalent Softkeys</i>	None

SCPI.CALCulate(*Ch*).TRACe(*Tr*).DATA.FMEMory

<i>Object Type</i>	Property (read/write)
<i>Data Type</i>	Variant (array of double)
<i>Target</i>	The specified trace <i>Tr</i> of channel <i>Ch</i> , <i>Tr</i> : trace number 1–16 (see Table 3 on page 33) <i>Ch</i> : channel number 1–16 (see Table 1 on page 25)
<i>Description</i>	<p>The formatted memory array. The array elements contain saved measurements in the current format, for example, in logarithmic magnitude format (Log Mag). Also, see section “Measurement Data Arrays” on page 19.</p> <p>The array size is 2N, where N is the number of measurement points.</p> <p>For the n-th point, where n from 1 to N:</p> <ul style="list-style-type: none"> <i>Data</i>(2n-2) real number in rectangular format, real part in polar and Smith chart formats; <i>Data</i>(2n-1) 0 in rectangular format, imaginary part in polar and Smith chart formats.
<i>Syntax</i>	<pre>Dim <i>Data</i> As Variant <i>Data</i> = <i>app</i>.SCPI.CALCulate(<i>Ch</i>).Trace(<i>Tr</i>).DATA.FMEMory <i>app</i>.SCPI.CALCulate(<i>Ch</i>).Trace(<i>Tr</i>).DATA.FMEMory = <i>Data</i></pre>
<i>Equivalent Softkeys</i>	None

SCPI.CALCulate(*Ch*). TRACe(*Tr*).DATA.SDATA

<i>Object Type</i>	Property (read/write)
<i>Data Type</i>	Variant (array of double)
<i>Target</i>	The specified trace <i>Tr</i> of channel <i>Ch</i> , <i>Tr</i> : trace number 1–16 (see Table 3 on page 33) <i>Ch</i> : channel number 1–16 (see Table 1 on page 25)
<i>Description</i>	The corrected data array. The corrected measurements are complex numbers. Also, see section “Measurement Data Arrays” on page 19. The array size is 2N, where N is the number of measurement points. For the n-th point, where n from 1 to N: <i>Data</i> (2n-2) the real part of corrected measurement; <i>Data</i> (2n-1) the imaginary part of corrected measurement.
<i>Syntax</i>	Dim <i>Data</i> As Variant <i>Data</i> = <i>app</i> .SCPI.CALCulate(<i>Ch</i>). Trace(<i>Tr</i>).DATA.SDATA <i>app</i> .SCPI.CALCulate(<i>Ch</i>). Trace(<i>Tr</i>).DATA.SDATA = <i>Data</i>
<i>Equivalent Softkeys</i>	None

SCPI.CALCulate(*Ch*). TRACe(*Tr*).DATA.SMEMory

<i>Object Type</i>	Property (read/write)
<i>Data Type</i>	Variant (array of double)
<i>Target</i>	The specified trace <i>Tr</i> of channel <i>Ch</i> , <i>Tr</i> : trace number 1–16 (see Table 1 on page 33) <i>Ch</i> : channel number 1–16 (see Table 1 on page 25)
<i>Description</i>	The corrected memory array. The corrected measurements are complex numbers. Also, see section “Measurement Data Arrays” on page 19. The array size is 2N, where N is the number of measurement points. For the n-th point, where n from 1 to N: <i>Data</i> (2n-2) the real part of corrected measurement memory; <i>Data</i> (2n-1) the imaginary part of corrected measurement memory.
<i>Syntax</i>	Dim <i>Data</i> As Variant <i>Data</i> = <i>app</i> .SCPI.CALCulate(<i>Ch</i>). Trace(<i>Tr</i>).DATA.SMEMory <i>app</i> .SCPI.CALCulate(<i>Ch</i>). Trace(<i>Tr</i>).DATA.SMEMory = <i>Data</i>
<i>Equivalent Softkeys</i>	None

SCPI.DISPlay.COLor.BACK

<i>Object Type</i>	Property (read/write)						
<i>Data Type</i>	Variant (array of long)						
<i>Target</i>	Analyzer						
<i>Description</i>	<p>The background color for trace display.</p> <p>The array contains 3 elements:</p> <table style="margin-left: 40px;"> <tr> <td><i>Data(0)</i></td> <td>Red value R;</td> </tr> <tr> <td><i>Data(1)</i></td> <td>Green value G;</td> </tr> <tr> <td><i>Data(2)</i></td> <td>Blue value B.</td> </tr> </table>	<i>Data(0)</i>	Red value R;	<i>Data(1)</i>	Green value G;	<i>Data(2)</i>	Blue value B.
<i>Data(0)</i>	Red value R;						
<i>Data(1)</i>	Green value G;						
<i>Data(2)</i>	Blue value B.						
<i>Range</i>	For all the array elements from 0 to 255.						
<i>Out of Range</i>	Sets the value of the limit, which is closer to the specified value.						
<i>Preset Value</i>	0, 0, 0						
<i>Syntax</i>	<p>Dim <i>Data</i> As Variant</p> <p><i>Data</i> = <i>app.SCPI.DISPlay.COLor.BACK</i></p> <p><i>app.SCPI.DISPlay.COLor.BACK</i> = Array(0, 0, 0)</p>						
<i>Equivalent Softkeys</i>	Display > Properties > Color > Background > Red Green Blue						

SCPI.DISPlay.COLor.GRATicule

<i>Object Type</i>	Property (read/write)						
<i>Data Type</i>	Variant (array of long)						
<i>Target</i>	Analyzer						
<i>Description</i>	<p>The grid and the graticule label color for trace display.</p> <p>The array contains 3 elements:</p> <table style="margin-left: 40px;"> <tr> <td><i>Data(0)</i></td> <td>Red value R;</td> </tr> <tr> <td><i>Data(1)</i></td> <td>Green value G;</td> </tr> <tr> <td><i>Data(2)</i></td> <td>Blue value B.</td> </tr> </table>	<i>Data(0)</i>	Red value R;	<i>Data(1)</i>	Green value G;	<i>Data(2)</i>	Blue value B.
<i>Data(0)</i>	Red value R;						
<i>Data(1)</i>	Green value G;						
<i>Data(2)</i>	Blue value B.						
<i>Range</i>	For all array elements from 0 to 255.						
<i>Out of Range</i>	Sets the value of the limit, which is closer to the specified value.						
<i>Preset Value</i>	<table style="margin-left: 40px;"> <tr> <td><i>Data(0)</i></td> <td>160;</td> </tr> <tr> <td><i>Data(1)</i></td> <td>160;</td> </tr> <tr> <td><i>Data(2)</i></td> <td>164.</td> </tr> </table>	<i>Data(0)</i>	160;	<i>Data(1)</i>	160;	<i>Data(2)</i>	164.
<i>Data(0)</i>	160;						
<i>Data(1)</i>	160;						
<i>Data(2)</i>	164.						
<i>Syntax</i>	<p>Dim <i>Data</i> As Variant</p> <p><i>Data</i> = <i>app.SCPI.DISPlay.COLor.GRATicule</i></p> <p><i>app.SCPI.DISPlay.COLor.GRATicule</i> = Array(128, 128, 128)</p>						
<i>Equivalent Softkeys</i>	Display > Properties > Color > Grid > Red Green Blue						

SCPI.DISPlay.COLor.RESet

<i>Object Type</i>	Method
<i>Target</i>	Analyzer
<i>Description</i>	Restores the display settings to the default values.
<i>Syntax</i>	<code>app.SCPI.DISPlay.COLor.RESet</code>
<i>Equivalent Softkeys</i>	Display > Properties > Set Defaults

SCPI.DISPlay.COLOr.TRACe(*Tr*).DATA

<i>Object Type</i>	Property (read/write)
<i>Data Type</i>	Variant (array of long)
<i>Target</i>	Trace number <i>Tr</i> in all channels, <i>Tr</i> : trace number 1–16 (see Table 3 on page 33)
<i>Description</i>	The data trace color. The array contains 3 elements: <i>Data(0)</i> Red value R; <i>Data(1)</i> Green value G; <i>Data(2)</i> Blue value B.
<i>Range</i>	For all array elements from 0 to 255.
<i>Out of Range</i>	Sets the value of the limit, which is closer to the specified value.
<i>Preset Value</i>	Varies depending on the trace number.
<i>Syntax</i>	Dim <i>Data</i> As Variant <i>Data</i> = <i>app.SCPI.DISPlay.COLOr.TRACe(Tr).DATA</i> <i>app.SCPI.DISPlay.COLOr.TRACe(Tr).DATA</i> = Array(255, 255, 0)
<i>Equivalent Softkeys</i>	Display > Properties > Color > Data Trace > Red Green Blue

SCPI.DISPlay.COLor.TRACe(*Tr*).MEMory

<i>Object Type</i>	Property (read/write)
<i>Data Type</i>	Variant (array of long)
<i>Target</i>	Trace number <i>Tr</i> in all channels <i>Tr</i> : trace number 1–16 (see Table 3 on page 33)
<i>Description</i>	The memory trace color. The array contains 3 elements: <i>Data(0)</i> Red value R; <i>Data(1)</i> Green value G; <i>Data(2)</i> Blue value B.
<i>Range</i>	For all array elements from 0 to 255.
<i>Out of Range</i>	Sets the value of the limit, which is closer to the specified value.
<i>Preset Value</i>	Varies depending on the trace number.
<i>Syntax</i>	Dim <i>Data</i> As Variant <i>Data</i> = <i>app.SCPI.DISPlay.COLor.TRACe(Tr).MEMory</i> <i>app.SCPI.DISPlay.COLor.TRACe(Tr).MEMory</i> = Array(255, 255, 0)
<i>Equivalent Softkeys</i>	Display > Properties > Color > Data Trace > Red Green Blue

SCPI.DISPlay.ENABLE

<i>Object Type</i>	Property (read/write)
<i>Data Type</i>	Boolean
<i>Target</i>	Analyzer
<i>Description</i>	The ON/OFF state of the display update function.
<i>Allowable Values</i>	True: Display update ON False: Display update OFF
<i>Preset Value</i>	True
<i>Syntax</i>	<p>Dim <i>Status</i> As Boolean</p> <p><i>Status</i> = <i>app.SCPI.DISPlay.ENABLE</i></p> <p><i>app.SCPI.DISPlay.ENABLE</i> = true</p>
<i>Equivalent Softkeys</i>	Display > Update

SCPI.DISPlay.FSIGn

<i>Object Type</i>	Property (read/write)
<i>Data Type</i>	Boolean
<i>Target</i>	Analyzer
<i>Description</i>	The ON/OFF state of the <i>Fail</i> sign display, when performing limit test or ripple limit test.
<i>Allowable Values</i>	True: <i>Fail</i> sign display ON False: <i>Fail</i> sign display OFF
<i>Preset Value</i>	False
<i>Syntax</i>	Dim <i>Status</i> As Boolean <i>Status</i> = <i>app.SCPI.DISPlay.FSIGn</i> <i>app.SCPI.DISPlay.FSIGn</i> = true
<i>Equivalent Softkeys</i>	Analysis > Limit Test > Fail Sign Analysis > Ripple Limit > Fail Sign

SCPI.DISPlay.GLabel

<i>Object Type</i>	Property (read/write)
<i>Data Type</i>	String
<i>Target</i>	Analyzer GUI
<i>Description</i>	The Graticule Label state.
<i>Range</i>	"OFF" : Graticule label is OFF "ACTive" : Only active trace has graticule label "ALL" : All traces have graticule label
<i>Notes</i>	The short form of the parameter is indicated by upper case letters. There is no distinction between upper and lower case letters when the property is written. When the property is read out, the short format is indicated by upper case letters.
<i>Out of Range</i>	The value is ignored.
<i>Preset Value</i>	"ACTive"
<i>Syntax</i>	<pre>Dim Param As String Param = app.SCPI.DISPlay.GLabel app.SCPI.DISPlay.GLabel = "OFF"</pre>
<i>Equivalent Softkeys</i>	Display > Properties > Graticule Label

SCPI.DISPlay.IMAGe

<i>Object Type</i>	Property (read/write)
<i>Data Type</i>	String
<i>Target</i>	Analyzer
<i>Description</i>	The inverted color display of the data traces.
<i>Range</i>	"NORMal" : Normal display "INVert" : Inverted color display
<i>Notes</i>	The short form of the parameter is indicated by upper case letters. There is no distinction between upper and lower case letters when the property is written. When the property is read out, the short format is indicated by upper case letters.
<i>Out of Range</i>	The value is ignored.
<i>Preset Value</i>	"NORM"
<i>Syntax</i>	Dim Param As String <i>Param</i> = app.SCPI.DISPlay.IMAGe app.SCPI.DISPlay.IMAGe = "INV"
<i>Equivalent Softkeys</i>	Display > Properties > Invert Color

SCPI.DISPlay.HIDE

<i>Object Type</i>	Method
<i>Target</i>	Analyzer GUI
<i>Description</i>	Hides the analyzer GUI. Blanks the main window and outputs "Remote Control".
<i>Syntax</i>	<code>app.SCPI.DISPlay.HIDE</code>
<i>Related Commands</i>	SCPI.DISPlay.SHOW
<i>Equivalent Softkeys</i>	None

SCPI.DISPlay.MAXimize

<i>Object Type</i>	Property (read/write)
<i>Data Type</i>	Boolean
<i>Target</i>	Active channel window
<i>Description</i>	The ON/OFF state of the window maximization of the active channel.
<i>Allowable Values</i>	True: Maximization ON False Maximization OFF
<i>Preset Value</i>	False
<i>Syntax</i>	Dim Status As Boolean <code>Status = app.SCPI.DISPlay.MAXimize</code> <code>app.SCPI.DISPlay. MAXimize = true</code>
<i>Equivalent Softkeys</i>	Display > Active Trace/Channel > Maximize channel

SCPI.DISPlay.PARTition.FONT.SIZE(*Param*)

<i>Object Type</i>	Property (read/write)
<i>Data Type</i>	Long
<i>Target</i>	Analyzer GUI
<i>Description</i>	The font size of the category specified by Parameter.
<i>Parameter</i>	"CHANnel" : Channel window "BUTTOn" : Soft buttons "MENU" : Menu bar "CSTatus" : Channel status "ASTatus" : Analyzer status
<i>Notes</i>	The short form of the parameter is indicated by upper case letters. There is no distinction between upper and lower case letters when the property is written. When the property is read out, the short format is indicated by upper case letters.
<i>Range</i>	From 10 to 22.
<i>Preset Value</i>	"11"
<i>Syntax</i>	$Size = app.SCPI.DISPlay.PARTition.FONT.SIZE("CHAN")$ $app.SCPI.DISPlay.PARTition.FONT.SIZE("CHAN") = 20$
<i>Equivalent Softkeys</i>	Display > Properties > Font > Size

SCPI.DISPlay.PARTition.VISible(*Param*)

<i>Object Type</i>	Property (read/write)
<i>Data Type</i>	Boolean
<i>Target</i>	Analyzer GUI
<i>Description</i>	Shows or hides the display partition specified by Parameter.
<i>Parameter</i>	"BUTTon" : Soft buttons "MENU" : Menu bar "CSTatus" : Channel status "ASTatus" : Analyzer status "TITLE" : Main window title "FLABel" : Frequency label "MTABle" : Marker table
<i>Notes</i>	The short form of the parameter is indicated by upper case letters. There is no distinction between upper and lower case letters when the property is written. When the property is read out, the short format is indicated by upper case letters.
<i>Syntax</i>	<pre>State = app.SCPI.DISPlay.PARTition.VISible("MENU") app.SCPI.DISPlay.PARTition.VISible("MENU") = true</pre>
<i>Equivalent Softkeys</i>	Display > Properties > Menu Bar Display > Display Properties > Frequency Label Markers > Properties > Marker Table or None

SCPI.DISPlay.POSition

<i>Object Type</i>	Property (read/write)
<i>Data Type</i>	Variant (array of long)
<i>Target</i>	Analyzer GUI
<i>Description</i>	<p>The main window position on the screen and dimension.</p> <p>The array contains 4 elements:</p> <ul style="list-style-type: none"> Data(0) Specifies the coordinate of the left side of the window; Data(1) Specifies the coordinate of the top of the window; Data(2) Specifies the width of the window; Data(3) Specifies the height of the window.
<i>Range</i>	from 0 to the screen resolution;
<i>Out of Range</i>	Sets the value of the limit, which is closer to the specified value.
<i>Preset Value</i>	<p>Data(0) – (screen width – 800) / 2, Data(1) – (screen height – 600) / 2, Data(2) – 800, Data(3) – 600,</p> <p>Preset: Display > Properties > Set Defaults</p>
<i>Syntax</i>	<p>Dim Pos As Variant <i>Pos = app.SCPI.DISPlay.POSition</i> <i>app.SCPI.DISPlay.POSition = Array(0, 0, 800, 600)</i></p>
<i>Equivalent Softkeys</i>	None

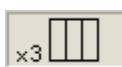
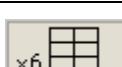
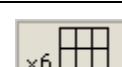
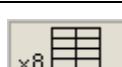
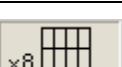
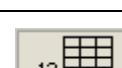
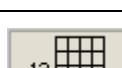
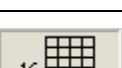
SCPI.DISPlay.SHOW

<i>Object Type</i>	Method
<i>Target</i>	Analyzer GUI
<i>Description</i>	Shows the analyzer GUI hidden by the SCPI.DISPlay.HIDE command.
<i>Syntax</i>	<i>app</i> .SCPI.DISPlay.SHOW
<i>Related Commands</i>	SCPI.DISPlay.HIDE
<i>Equivalent Softkeys</i>	None

SCPI.DISPlay.SPLit

<i>Object Type</i>	Property (read/write)
<i>Data Type</i>	Long
<i>Target</i>	Analyzer
<i>Description</i>	The number and layout of the channel windows on the screen. The channel window layout is in Table 5 below.
<i>Range</i>	from 1 to 16
<i>Out of Range</i>	Sets the value of the limit, which is closer to the specified value.
<i>Preset Value</i>	1
<i>Syntax</i>	<p>Dim Value As Long</p> <p>Value = app.SCPI.DISPlay.SPLit</p> <p>app.SCPI.DISPlay.SPLit = 2</p>
<i>Equivalent Softkeys</i>	Display > Allocate channels

Table 5. Channel Window Layout on the Screen

1: 	2: 	3: 	4: 
5: 	6: 	7: 	8: 
9: 	10: 	11: 	12: 
13: 	14: 	15: 	16: 

SCPI.DISPlay.REFResh.IMMEDIATE**SCPI.DISPlay._UPDate.IMMEDIATE**

<i>Object Type</i>	Method
<i>Target</i>	Analyzer
<i>Description</i>	Updates the display once, when the display update is set to OFF (SCPI.DISPlay.ENABLe property is set to False).
<i>Syntax</i>	<code>app.SCPI.DISPlay.REFResh.IMMEDIATE</code> <code>app.SCPI.DISPlay._UPDate.IMMEDIATE</code>
<i>Equivalent Softkeys</i>	None

SCPI.DISPlay.WINDOW(*Ch*).ACTivate

<i>Object Type</i>	Method
<i>Target</i>	Channel <i>Ch</i> , <i>Ch</i> : channel number 1–16 (see Table 1 on page 25)
<i>Description</i>	Sets the active channel.
<i>Notes</i>	The channel window must be displayed. At attempt to set to the active channel the channel, which is not displayed, an error occurs.
<i>Syntax</i>	<code>app.SCPI.DISPlay.WINDOW(<i>Ch</i>).ACTivate</code>
<i>Equivalent Softkeys</i>	Display > Active Trace / Channel > Active Channel

SCPI.DISPlay.WINDOW(*Ch*).ANNotation.MARKer.ALIGn.TYPE

<i>Object Type</i>	Property (read/write)
<i>Data Type</i>	String
<i>Target</i>	Channel <i>Ch</i> , <i>Ch</i> : channel number 1–16 (see Table 1 on page 25)
<i>Description</i>	The alignment mode of the marker display position of each trace, when the only active trace display feature is turned OFF (SCPI.DISPlay.WINDOW(<i>Ch</i>).ANNotation.MARKer.SINGLe.STATE property is set to False).
<i>Range</i>	"VERTical" : Vertical alignment "HORIZONTAL" : Horizontal alignment "NONE" : No alignment
<i>Notes</i>	The short form of the parameter is indicated by upper case letters. There is no distinction between upper and lower case letters when the property is written. When the property is read out, the short format is indicated by upper case letters.
<i>Out of Range</i>	The value is ignored.
<i>Preset Value</i>	"NONE"
<i>Syntax</i>	<pre>Dim Param As String Param = app.SCPI.DISPlay.WINDOW(<i>Ch</i>).ANNotation.MARKer.ALIGn.TYPE app.SCPI.DISPlay.WINDOW(<i>Ch</i>).ANNotation.MARKer.ALIGn.TYPE = "VERT"</pre>
<i>Equivalent Softkeys</i>	Markers > Properties > Align > Vertical Horizontal OFF

SCPI.DISPlay.WINDoW(*Ch*).ANNotation.MARKer.SINGle.STATE

<i>Object Type</i>	Property (read/write)
<i>Data Type</i>	Boolean
<i>Target</i>	Channel <i>Ch</i> , <i>Ch</i> : channel number 1–16 (see Table 1 on page 25)
<i>Description</i>	The ON/OFF state of the marker display for the active trace only.
<i>Allowable Values</i>	True: Only active trace markers display ON False: Only active trace markers display OFF
<i>Preset Value</i>	True
<i>Syntax</i>	Dim <i>Status</i> As Boolean <i>Status</i> = <i>app</i> .SCPI SCPI.DISPlay.WINDoW(<i>Ch</i>).ANNotation.MARKer.SINGle.STATE <i>app</i> .SCPI SCPI.DISPlay.WINDoW(<i>Ch</i>).ANNotation.MARKer.SINGle.STATE = true
<i>Equivalent Softkeys</i>	Markers > Properties > Active Only

SCPI.DISPlay.WINDOW(*Ch*).MAXimize

<i>Object Type</i>	Property (read/write)
<i>Data Type</i>	Boolean
<i>Target</i>	The active trace of channel <i>Ch</i> , <i>Ch</i> : channel number 1–16 (see Table 1 on page 25)
<i>Description</i>	The ON/OFF state of the active trace maximization of the specified channel.
<i>Allowable Values</i>	True: Maximization ON False Maximization OFF
<i>Preset Value</i>	False
<i>Syntax</i>	Dim <i>Status</i> As Boolean <i>Status</i> = <i>app.SCPI.DISPlay.WINDOW(Ch).MAXimize</i> <i>app.SCPI.DISPlay.WINDOW(Ch).MAXimize</i> = true
<i>Equivalent Softkeys</i>	Display > Active Trace/Channel > Maximize Trace

SCPI.DISPlay.WINDOW(*Ch*).SPLit

<i>Object Type</i>	Property (read/write)
<i>Data Type</i>	Long
<i>Target</i>	Channel <i>Ch</i> , <i>Ch</i> : channel number 1–16 (see Table 1 on page 25)
<i>Description</i>	The code of the graph layout of channels. The channel window layout is in Table 5 on page 150.
<i>Range</i>	from 1 to 16
<i>Out of Range</i>	Sets the value of the limit, which is closer to the specified value.
<i>Preset Value</i>	1
<i>Notes</i>	This property does not define the number of traces in the channel window, the number of traces is defined by the SCPI.CALCulate(<i>Ch</i>).PARameter.COUNT property.
<i>Syntax</i>	<p>Dim <i>Value</i> As Long</p> <p><i>Value</i> = <i>app</i>.SCPI.DISPlay.WINDOW(<i>Ch</i>).SPLit</p> <p><i>app</i>.SCPI.DISPlay.WINDOW(<i>Ch</i>).SPLit = 2</p>
<i>Equivalent Softkeys</i>	Display > Allocate Traces

SCPI.DISPlay.WINDoW(*Ch*).TITLE.DATA

<i>Object Type</i>	Property (read/write)
<i>Data Type</i>	String
<i>Target</i>	Channel <i>Ch</i> , <i>Ch</i> : channel number 1–16 (see Table 1 on page 25)
<i>Description</i>	The channel title label.
<i>Range</i>	up to 254 characters
<i>Preset Value</i>	""
<i>Syntax</i>	<pre>Dim <i>Text</i> As String <i>Text</i> = app.SCPI.DISPlay.WINDoW(<i>Ch</i>).TITLE.DATA app.SCPI.DISPlay.WINDoW(<i>Ch</i>).TITLE.DATA = "Example1"</pre>
<i>Equivalent Softkeys</i>	Display > Edit Title Label

SCPI.DISPlay.WINDoW(*Ch*).TITLE.STATE

<i>Object Type</i>	Property (read/write)
<i>Data Type</i>	Boolean
<i>Target</i>	Channel <i>Ch</i> , <i>Ch</i> : channel number 1–16 (see Table 1 on page 25)
<i>Description</i>	The ON/OFF state of the title label display.
<i>Allowable Values</i>	True: Title label display ON False: Title label display OFF
<i>Preset Value</i>	False
<i>Syntax</i>	Dim <i>Status</i> As Boolean <i>Status</i> = <i>app</i> .SCPI.DISPlay.WINDoW(<i>Ch</i>).TITLE.STATE <i>app</i> .SCPI.DISPlay.WINDoW(<i>Ch</i>).TITLE.STATE = true
<i>Equivalent Softkeys</i>	Display > Title Label

SCPI.DISPlay.WINDoW(*Ch*).TRACe(*Tr*).ANNotation.MARKer. MEMory

<i>Object Type</i>	Property (read/write)
<i>Data Type</i>	Boolean
<i>Target</i>	Trace <i>Tr</i> of channel <i>Ch</i> , <i>Ch</i> : channel number 1–16 (see Table 1 on page 25) <i>Tr</i> : trace number 1–16 (see Table 3 on page 33)
<i>Description</i>	The ON/OFF state of the memory value display on the marker.
<i>Allowable Values</i>	True: Memory value display ON False: Memory value display OFF
<i>Syntax</i>	Dim <i>Status</i> As Boolean <i>Status</i> = <i>app.SCPI.DISPlay.WINDoW(Ch).TRACe(Tr).ANNotation.MARKer.MEMory</i> <i>app.SCPI.DISPlay.WINDoW(Ch).TRACe(Tr).ANNotation.MARKer.MEMory</i> = true
<i>Equivalent Softkeys</i>	Markers > Properties > Memory Value

SCPI.DISPlay.WINDOW(*Ch*).TRACe(*Tr*).ANNotation.MARKer. POSition.X

<i>Object Type</i>	Property (read/write)
<i>Data Type</i>	Double
<i>Target</i>	Trace <i>Tr</i> of channel <i>Ch</i> , <i>Ch</i> : channel number 1–16 (see Table 1 on page 25) <i>Tr</i> : trace number 1–16 (see Table 3 on page 33)
<i>Description</i>	The display position of the marker value on the X-axis by a percentage of the display width.
<i>Range</i>	from 0 to 100
<i>Out of Range</i>	Sets the value of the limit, which is closer to the specified value.
<i>Preset Value</i>	0
<i>Unit</i>	%
<i>Syntax</i>	Dim <i>Value</i> As Double <i>Value</i> = <i>app</i> .SCPI.DISPlay.WINDOW(<i>Ch</i>).TRACe(<i>Tr</i>).ANNotation.MARKer.POSition.X <i>app</i> .SCPI.DISPlay.WINDOW(<i>Ch</i>).TRACe(<i>Tr</i>).ANNotation.MARKer.POSition.X = 50
<i>Equivalent Softkeys</i>	Markers > Properties > Data X Position

SCPI.DISPlay.WINDOW(*Ch*).TRACe(*Tr*).ANNotation.MARKer. POSition.Y

<i>Object Type</i>	Property (read/write)
<i>Data Type</i>	Double
<i>Target</i>	Trace <i>Tr</i> of channel <i>Ch</i> , <i>Ch</i> : channel number 1–16 (see Table 1 on page 25) <i>Tr</i> : trace number 1–16 (see Table 3 on page 33)
<i>Description</i>	The display position of the marker value on the Y-axis by a percentage of the display height.
<i>Range</i>	from 0 to 100
<i>Out of Range</i>	Sets the value of the limit, which is closer to the specified value.
<i>Preset Value</i>	0
<i>Unit</i>	%
<i>Syntax</i>	Dim <i>Value</i> As Double <i>Value</i> = <i>app</i> .SCPI.DISPlay.WINDOW(<i>Ch</i>).TRACe(<i>Tr</i>).ANNotation.MARKer.POSition.Y <i>app</i> .SCPI.DISPlay.WINDOW(<i>Ch</i>).TRACe(<i>Tr</i>).ANNotation.MARKer.POSition.Y = 50
<i>Equivalent Softkeys</i>	Markers > Properties > Data Y Position

SCPI.DISPlay.WINDoW(*Ch*).TRACe(*Tr*).MEMory. STATe

<i>Object Type</i>	Property (read/write)
<i>Data Type</i>	Boolean
<i>Target</i>	Trace <i>Tr</i> of channel <i>Ch</i> , <i>Ch</i> : channel number 1–16 (see Table 1 on page 25) <i>Tr</i> : trace number 1–16 (see Table 3 on page 33)
<i>Description</i>	The ON/OFF state of the memory trace display.
<i>Allowable Values</i>	True: Memory trace display ON False: Memory trace display OFF
<i>Preset Value</i>	False
<i>Notes</i>	If the memory is empty, an error occurs and the object is ignored.
<i>Syntax</i>	<p>Dim <i>Status</i> As Boolean</p> <p><i>Status</i> = app.SCPI.DISPlay.WINDoW(<i>Ch</i>).TRACe(<i>Tr</i>).MEMory.STATe</p> <p>app.SCPI.DISPlay.WINDoW(<i>Ch</i>).TRACe(<i>Tr</i>).MEMory.STATe = true</p>
<i>Equivalent Softkeys</i>	Display > Display > Memory Data & Memory (ON) Display > Display > Data OFF (OFF)

SCPI.DISPlay.WINDOW(*Ch*).TRACe(*Tr*).STATe

<i>Object Type</i>	Property (read/write)
<i>Data Type</i>	Boolean
<i>Target</i>	Trace <i>Tr</i> of channel <i>Ch</i> , <i>Ch</i> : channel number 1–16 (see Table 1 on page 25) <i>Tr</i> : trace number 1–16 (see Table 3 on page 33)
<i>Description</i>	The ON/OFF state of the data trace display.
<i>Allowable Values</i>	True: Data trace display ON False: Data trace display OFF
<i>Preset Value</i>	True
<i>Syntax</i>	Dim <i>Status</i> As Boolean <i>Status</i> = <i>app</i> .SCPI.DISPlay.WINDOW(<i>Ch</i>).TRACe(<i>Tr</i>).STATe <i>app</i> .SCPI.DISPlay.WINDOW(<i>Ch</i>).TRACe(<i>Tr</i>).STATe = false
<i>Equivalent Softkeys</i>	Display > Display > Data Data & Memory (ON) Display > Display > Memory OFF (OFF)

SCPI.DISPlay.WINDow(*Ch*).TRACe(*Tr*).Y.SCALE.AUTO

<i>Object Type</i>	Method
<i>Target</i>	Trace <i>Tr</i> of channel <i>Ch</i> , <i>Ch</i> : channel number 1–16 (see Table 1 on page 25) <i>Tr</i> : trace number 1–16 (see Table 3 on page 33)
<i>Description</i>	Executes the auto scale function for the trace.
<i>Syntax</i>	<i>app</i> .SCPI.DISPlay.WINDow(<i>Ch</i>).TRACe(<i>Tr</i>).Y.SCALE.AUTO
<i>Equivalent Softkeys</i>	Scale > Auto Scale

SCPI.DISPlay.WINDOW(*Ch*).TRACe(*Tr*).Y.SCALE.PDIVision

<i>Object Type</i>	Property (read/write)
<i>Data Type</i>	Double
<i>Target</i>	Trace <i>Tr</i> of channel <i>Ch</i> , <i>Ch</i> : channel number 1–16 (see Table 1 on page 25) <i>Tr</i> : trace number 1–16 (see Table 3 on page 33)
<i>Description</i>	The trace scale. Sets the scale per division, when the data format is the rectangular format. Sets the full scale value, when the data format is the Smith chart format or the polar format.
<i>Range</i>	from 10E-18 to 1E18
<i>Out of Range</i>	Sets the value of the limit, which is closer to the specified value.
<i>Preset Value</i>	Varies depending on the format. Logarithmic Magnitude: 10 dB/Div Phase: 40 °/Div Expand Phase: 100 °/Div Group Delay: 10e-9 s/Div Smith Chart, Polar, SWR: 1 /Div Linear Magnitude: 0.1 /Div Real part, Imaginary part: 0.2 /Div
<i>Unit</i>	dB/Div (decibel per division), °/Div (degree per division), s/Div (second per division)
<i>Syntax</i>	<p>Dim <i>Value</i> As Double</p> <pre>Value = app.SCPI.DISPlay.WINDOW(<i>Ch</i>).TRACe(<i>Tr</i>).Y.SCALE.PDIVision app.SCPI.DISPlay.WINDOW(<i>Ch</i>).TRACe(<i>Tr</i>).Y.SCALE.PDIVision = 20</pre>
<i>Equivalent Softkeys</i>	Scale > Scale

SCPI.DISPlay.WINDOW(*Ch*).TRACe(*Tr*).Y.SCALE.RLEVel

<i>Object Type</i>	Property (read/write)
<i>Data Type</i>	Double
<i>Target</i>	Trace <i>Tr</i> of channel <i>Ch</i> , <i>Ch</i> : channel number 1–16 (see Table 1 on page 25) <i>Tr</i> : trace number 1–16 (see Table 3 on page 33)
<i>Description</i>	The value of the reference line (response value on the reference line). For the rectangular format only.
<i>Range</i>	from –1E–18 to 1E18
<i>Out of Range</i>	Sets the value of the limit, which is closer to the specified value.
<i>Preset Value</i>	0 (except for SWR: 1)
<i>Unit</i>	dB (decibel) ° (degree) s (second)
<i>Syntax</i>	<p>Dim <i>Value</i> As Double</p> <p><i>Value</i> = <i>app</i>.SCPI.DISPlay.WINDOW(<i>Ch</i>).TRACe(<i>Tr</i>).Y.SCALE.RLEVel</p> <p><i>app</i>.SCPI.DISPlay.WINDOW(<i>Ch</i>).TRACe(<i>Tr</i>).Y.SCALE.RLEVel = 10</p>
<i>Equivalent Softkeys</i>	Scale > Ref Value

SCPI.DISPlay.WINDOW(*Ch*).TRACe(*Tr*).Y.SCALE.RPOsition

<i>Object Type</i>	Property (read/write)
<i>Data Type</i>	Long
<i>Target</i>	Trace <i>Tr</i> of channel <i>Ch</i> , <i>Ch</i> : channel number 1–16 (see Table 1 on page 25) <i>Tr</i> : trace number 1–16 (see Table 3 on page 33)
<i>Description</i>	The position of the reference line. For the rectangular format only.
<i>Range</i>	From 0 to the number of the scale divisions (set by the SCPI.DISPlay.WINDOW(<i>Ch</i>).Y.SCALE.DIVisions property, 10 by default).
<i>Out of Range</i>	Sets the value of the limit, which is closer to the specified value.
<i>Preset Value</i>	5 (except for SWR: 0)
<i>Syntax</i>	Dim <i>Value</i> As Long <i>Value</i> = <i>app</i> .SCPI.DISPlay.WINDOW(<i>Ch</i>).TRACe(<i>Tr</i>).Y.SCALE.RPOsition <i>app</i> .SCPI.DISPlay.WINDOW(<i>Ch</i>).TRACe(<i>Tr</i>).Y.SCALE.RPOsition = 10
<i>Equivalent Softkeys</i>	Scale > Ref Position

SCPI.DISPlay.WINDOW(*Ch*).X.SPACing

<i>Object Type</i>	Property (read/write)
<i>Data Type</i>	String
<i>Target</i>	Channel <i>Ch</i> , <i>Ch</i> : channel number 1–16 (see Table 1 on page 25).
<i>Description</i>	The selection of the display method of the graph horizontal axis for the segment sweep.
<i>Range</i>	"LINear" : Frequency base (linear frequency axis) "OBASe" : Order base (linear axis of the point numbers)
<i>Notes</i>	The short form of the parameter is indicated by upper case letters. There is no distinction between upper and lower case letters when the property is written. When the property is read out, the short format is indicated by upper case letters.
<i>Out of Range</i>	The value is ignored.
<i>Preset Value</i>	"LIN"
<i>Syntax</i>	Dim <i>Param</i> As String <i>Param</i> = <i>app.SCPI.DISPlay.WINDOW(Ch).X.SPACing</i> <i>app.SCPI.DISPlay.WINDOW(Ch).X.SPACing</i> = "OBAS"
<i>Equivalent Softkeys</i>	Stimulus > Segment Table > Segment Display

SCPI.DISPlay.WINDOW(*Ch*).Y.SCALE.DIVisions

<i>Object Type</i>	Property (read/write)
<i>Data Type</i>	Long
<i>Target</i>	Channel <i>Ch</i> , <i>Ch</i> : channel number 1–16 (see Table 1 on page 25).
<i>Description</i>	The number of the vertical scale divisions. For the rectangular format only.
<i>Range</i>	from 4 to 30
<i>Out of Range</i>	Sets the value of the limit, which is closer to the specified value.
<i>Preset Value</i>	10
<i>Resolution</i>	2
<i>Syntax</i>	<p>Dim <i>Value</i> As Long</p> <p><i>Value</i> = <i>app.SCPI.DISPlay.WINDOW(Ch).Y.SCALE.DIVisions</i></p> <p><i>app.SCPI.DISPlay.WINDOW(Ch).Y.SCALE.DIVisions</i> = 12</p>
<i>Equivalent Softkeys</i>	Scale > Divisions

SCPI.HCOPy.ABORt

<i>Object Type</i>	Method
<i>Target</i>	Analyzer
<i>Description</i>	Aborts the printout.
<i>Syntax</i>	<code>app.SCPI.HCOPy.ABORt</code>
<i>Equivalent Softkeys</i>	None

SCPI.HCOPy.DATE.STAMp

<i>Object Type</i>	Property (read/write)
<i>Data Type</i>	Boolean
<i>Target</i>	Analyzer
<i>Description</i>	The ON/OFF state of the current date and time printout in the upper right corner.
<i>Allowable Values</i>	True: Date & time printout ON False: Date & time printout OFF
<i>Preset Value</i>	True
<i>Syntax</i>	<pre>Dim Status As Boolean Status = app.SCPI.HCOPy.DATE.STAMp app.SCPI.HCOPy.DATE.STAMp = False</pre>
<i>Equivalent Softkeys</i>	System > Print > Print Date & Time

SCPI.HCOPy.IMAGe

<i>Object Type</i>	Property (read/write)
<i>Data Type</i>	String
<i>Target</i>	Analyzer
<i>Description</i>	The inverted color image printout.
<i>Range</i>	"NORMal" : Normal printout "INVert" : Inverted color printout
<i>Notes</i>	The short form of the parameter is indicated by upper case letters. There is no distinction between upper and lower case letters when the property is written. When the property is read out, the short format is indicated by upper case letters.
<i>Out of Range</i>	The value is ignored.
<i>Preset Value</i>	"NORM"
<i>Syntax</i>	Dim Param As String <i>Param</i> = app.SCPI.HCOPy.IMAGe app.SCPI.HCOPy.IMAGe = "INV"
<i>Equivalent Softkeys</i>	System > Print > Invert Image

SCPI.HCOPy.IMMEDIATE

<i>Object Type</i>	Method
<i>Target</i>	Analyzer
<i>Description</i>	Prints out the image displayed on the screen without previewing.
<i>Syntax</i>	<i>app.SCPI.HCOPy.IMMEDIATE</i>
<i>Equivalent Softkeys</i>	System > Print > Print Embedded

SCPI.HCOPy.PAINt

<i>Object Type</i>	Property (read/write)
<i>Data Type</i>	String
<i>Target</i>	Analyzer
<i>Description</i>	The color chart for the image printout.
<i>Range</i>	"COLOR" : Color printout "GRAY" : Grayscale printout "BW" : Black&white printout
<i>Notes</i>	The short form of the parameter is indicated by upper case letters. There is no distinction between upper and lower case letters when the property is written. When the property is read out, the short format is indicated by upper case letters.
<i>Out of Range</i>	The value is ignored.
<i>Preset Value</i>	"BW"
<i>Syntax</i>	Dim Param As String <code>Param = app.SCPI.HCOPy.PAINt</code> <code>app.SCPI.HCOPy.PAINt = "COL"</code>
<i>Equivalent Softkeys</i>	System > Print > Print Color

SCPI.IEEE4882.CLS

<i>Object Type</i>	Method
<i>Target</i>	Status Reporting System
<i>Description</i>	<p>Clears the following:</p> <ul style="list-style-type: none"> • 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
<i>Syntax</i>	<code>app.SCPI.IEEE4882.CLS</code>
<i>Equivalent Softkeys</i>	None

SCPI.IEEE4882.IDN

<i>Object Type</i>	Property (read only)
<i>Data Type</i>	String
<i>Target</i>	Analyzer
<i>Description</i>	The analyzer information string. The string format: "{manufacturer}, {model}, {serial number}, {software version/firmware version}".
<i>Range</i>	up to 40 characters
<i>Syntax</i>	<p>Dim <i>ID</i> As String <i>ID</i> = app.NAME</p>
<i>Equivalent Softkeys</i>	None

SCPI.IEEE4882.OPC

<i>Object Type</i>	Property (read/write)
<i>Data Type</i>	Long
<i>Target</i>	Status Reporting System
<i>Description</i>	<p>Read 1 when all pending operations are complete. Note: since COM server executes commands sequentially and any operation is complete before COM server returns control the OPC read command doesn't wait anything.</p> <p>Write form of the command sets 1 the OPC bit (bit 0) of the Standard Event Status Register when all of pending operations complete. In the COM programming this is dummy operation.</p>
<i>Syntax</i>	$Value = app.SCPI.IEEE4882.OPC$ $app.SCPI.IEEE4882.OPC = \text{Dummy}$
<i>Equivalent Softkeys</i>	None

SCPI.IEEE4882.RST

<i>Object Type</i>	Method
<i>Target</i>	Analyzer
<i>Description</i>	Restores the default settings of the analyzer. There is difference from presetting the analyzer with the SCPI.SYSTem.PRESet method – in this case the trigger mode is set to <i>Hold</i> .
<i>Syntax</i>	$app.SCPI.IEEE4882.RST$
<i>Equivalent Softkeys</i>	None

SCPI.IEEE4882.TRG

<i>Object Type</i>	Method
<i>Target</i>	Analyzer
<i>Description</i>	If the trigger source is set to GPIB/LAN (SCPI.TRIGger.SEQuence.SOURce property is set to "BUS"), triggers a sweep. If the trigger source is not set to the bus (SCPI.TRIGger.SEQuence.SOURce property is not set to "BUS") or the analyzer is not waiting for a trigger, the method is ignored.
<i>Syntax</i>	<i>app</i> .SCPI.IEEE4882.TRG
<i>Related Commands</i>	SCPI.TRIGger.SEQuence.SOURce SCPI.INITiate(<i>Ch</i>).CONTinuous SCPI.INITiate(<i>Ch</i>).IMMEDIATE
<i>Equivalent Softkeys</i>	None

SCPI.IEEE4882.WAI

<i>Object Type</i>	Method
<i>Target</i>	Analyzer
<i>Description</i>	Waits for the execution of all commands sent before this command. Note: since COM server executes commands sequentially and any operation is complete before COM server returns control the WAI command doesn't wait anything.
<i>Syntax</i>	<i>app</i> .SCPI.IEEE4882.WAI
<i>Equivalent Softkeys</i>	None

SCPI.INITiate(*Ch*).CONTinuous

<i>Object Type</i>	Property (read/write)
<i>Data Type</i>	Boolean
<i>Target</i>	Channel <i>Ch</i> , <i>Ch</i> : channel number 1–16 (see Table 1 on page 25)
<i>Description</i>	The ON/OFF state of the continuous trigger initiation mode. If the continuous trigger initiation mode is set to OFF, the channel turns to the hold state.
<i>Allowable Values</i>	True: Continuous trigger initiation mode ON False: Continuous trigger initiation mode OFF
<i>Preset Value</i>	True
<i>Syntax</i>	Dim <i>Status</i> As Boolean <i>Status</i> = <i>app.SCPI.INITiate(Ch).CONTinuous</i> <i>app.SCPI.INITiate(Ch).CONTinuous</i> = False
<i>Notes</i>	The sweep start in continuous trigger initiation mode depends on the trigger source. If the trigger is set to internal, the sweeps will go immediately one after another. If the trigger is set otherwise, the sweep will start when the trigger signal is received.
<i>Equivalent Softkeys</i>	Stimulus > Trigger > Continuous Stimulus > Trigger > Hold

SCPI.INITiate(*Ch*).IMMEDIATE

<i>Object Type</i>	Method
<i>Target</i>	Channel <i>Ch</i> , <i>Ch</i> : channel number 1–16 (see Table 1 on page 25)
<i>Description</i>	<p>Sets the channel to the single trigger mode. Before this method is called, the channel must be in hold state, otherwise an error occurs (error code 213) and the method is ignored.</p> <p>On completion of the sweep, the channel goes back into the hold state.</p> <p>The method returns control before the end of the sweep.</p>
<i>Syntax</i>	<code>app.SCPI.INITiate(<i>Ch</i>).IMMEDIATE</code>
<i>Notes</i>	The sweep start in the single trigger mode depends on the trigger source. If the trigger is set to internal, the sweep will start immediately after the method is called. If the trigger is set otherwise, the sweep will start when the trigger signal is received.
<i>Equivalent Softkeys</i>	Stimulus > Trigger > Single

SCPI.MMEMORY.CATalog(*Dir*)

<i>Object Type</i>	Property (read only)
<i>Data Type</i>	String
<i>Description</i>	<p>This command reads out the following information on the hard drive:</p> <ul style="list-style-type: none"> • Space in use • Available space • Name and size of all files (including directories) in the specified directory <p>Format:</p> <p>("<i>{A}</i>,""<i>{B}</i>,""<i>Name 1</i>,""<i>Size 1</i>,""<i>Name 2</i>,""<i>Size 2</i>,"... ,"<i>Name N</i>,""<i>Size N</i>")</p> <p>Where N is the number of all files in the specified directory and n is an integer between 1 and N.</p> <p><i>{A}</i>: Space in use of the hard drive (byte).</p> <p><i>{B}</i>: Available space of the hard drive (byte).</p> <p><i>{Name n}</i>: Name of the n-th file (directory).</p> <p><i>{Size n}</i>: Size (byte) of the n-th file (directory). Always 0 for directories.</p>
<i>Parameter</i>	<i>Dir</i> – Directory name whose information you want to read out
<i>Syntax</i>	<p><i>Dim Cat as String</i></p> <p><i>Cat = app.SCPI.MMEMORY.CATalog("".")</i></p>
<i>Equivalent Softkeys</i>	None

SCPI.MMEMORY.COPY(*Src, Dst*)

<i>Object Type</i>	Method
<i>Target</i>	Analyzer
<i>Description</i>	Copies a file.
<i>Syntax</i>	<i>app.SCPI.MMEMORY.COPY(<i>Src, Dst</i>)</i>
<i>Parameter</i>	<i>Src</i> – Source file name. String data type. <i>Dst</i> – Destination file name. String data type.
<i>Equivalent Softkeys</i>	None

SCPI.MMEMORY.DELETE(*File*)

<i>Object Type</i>	Method
<i>Target</i>	Analyzer
<i>Description</i>	Deletes a file.
<i>Syntax</i>	<i>app.SCPI.MMEMORY.DELETE(<i>File</i>)</i>
<i>Parameter</i>	<i>File</i> – File name. String data type.
<i>Equivalent Softkeys</i>	None

SCPI.MMEMORY.LOAD.CHANNEL.STATE

<i>Object Type</i>	Property (write only)
<i>Data Type</i>	String
<i>Target</i>	Active channel
<i>Description</i>	Recalls the analyzer state for the active channel, saved in one of the four memory registers by the SCPI.MMEMORY.STORE.CHANNEL.STATE property.
<i>Range</i>	"A" : Recall from register A "B" : Recall from register B "C" : Recall from register C "D" : Recall from register D
<i>Out of Range</i>	The value is ignored.
<i>Syntax</i>	<code>app.SCPI.MMEMORY.LOAD.CHANNEL.STATE = "A"</code>
<i>Equivalent Softkeys</i>	Save/Recall > Recall Channel > State A B C D

SCPI.MMEMORY.LOAD.CKIT(Ck)

<i>Object Type</i>	Property (write only)
<i>Data Type</i>	String
<i>Target</i>	Calibration kit Ck, Ck: calibration kit number 1–11 (see Table 6 on page 181)
<i>Description</i>	Recalls the definition file for the calibration kit. The file must be saved by the SCPI.MMEMORY.STORe.CKIT(Ck) property.
<i>Range</i>	up to 254 characters
<i>Syntax</i>	<code>app.SCPI.MMEMORY.LOAD.CKIT(Ck) = File</code>
<i>Notes</i>	If the full path to the file is not specified, the \CalKit subdirectory of the main directory will be searched for the file. The calibration kit definition file has *.ckd extension by default.
<i>Equivalent Softkeys</i>	None

Table 6. Ck: Calibration Kit Number

<i>Data Type</i>	Long
<i>Description</i>	Calibration kit number.
<i>Range</i>	from 1 to 11
<i>Out of Range</i>	An error occurs. Error code: 114.
<i>Notes</i>	If the calibration kit number is not specified, it is taken as equal to 1.

SCPI.MMEMORY.LOAD.LIMit

<i>Object Type</i>	Property (write only)
<i>Data Type</i>	String
<i>Target</i>	Active trace of the active channel.
<i>Description</i>	Recalls the specified limit table file. The file must be saved by the SCPI.MMEMORY.STORE.LIMit property.
<i>Range</i>	up to 254 characters
<i>Syntax</i>	<i>app.SCPI.MMEMORY.LOAD.LIMit = File</i>
<i>Notes</i>	If the full path to the file is not specified, the \Limit subdirectory of the main directory will be searched for the file. The limit table files have *.lim extension by default.
<i>Equivalent Softkeys</i>	Analysis > Limit Test > Edit Limit Line > Restore Limit Table

SCPI.MMEMORY.LOAD.PLOSS(*Pt*)

<i>Object Type</i>	Property (write only)
<i>Data Type</i>	String
<i>Target</i>	Port <i>Pt</i> of the active channel, <i>Pt</i> : port number 1–2 (see Table 2 on page 25).
<i>Description</i>	Recalls the specified loss compensation table file. The file must be saved by the SCPI.MMEMORY.STORE.PLOSS(<i>Pt</i>) property.
<i>Range</i>	up to 254 characters
<i>Syntax</i>	<code>app.SCPI.MMEMORY.LOAD.PLOSS(<i>Pt</i>) = <i>File</i></code>
<i>Notes</i>	If the full path to the file is not specified, the \CalKit subdirectory of the main directory will be searched for the file. The loss compensation file has *.lct extension by default.
<i>Equivalent Softkeys</i>	Calibration > Power Calibration > Loss Compen > Import Loss Table

SCPI.MMEMORY.LOAD.RLIMit

<i>Object Type</i>	Property (write only)
<i>Data Type</i>	String
<i>Target</i>	Active trace of the active channel.
<i>Description</i>	Recalls the ripple limit table file. The file must be saved by the SCPI.MMEMORY.STORE.RLIMit property.
<i>Range</i>	up to 254 characters
<i>Syntax</i>	<i>app.SCPI.MMEMORY.LOAD.RLIMit = File</i>
<i>Notes</i>	If the full path to the file is not specified, the \Limit subdirectory of the main directory will be searched for the file. The ripple limit files have *.rlm extension by default.
<i>Equivalent Softkeys</i>	Analysis > Ripple Limit > Edit Ripple Limit > Restore Ripple Limit Table

SCPI.MMEMORY.LOAD SEGMENT

<i>Object Type</i>	Property (write only)
<i>Data Type</i>	String
<i>Target</i>	Active channel
<i>Description</i>	Recalls the segment table file. The file must be saved by the SCPI.MMEMORY.STORE SEGMENT property.
<i>Range</i>	up to 254 characters
<i>Syntax</i>	<i>app.SCPI.MMEMORY.LOAD SEGMENT = File</i>
<i>Notes</i>	If the full path to the file is not specified, the \Segment subdirectory of the main directory will be searched for the file. The segment files have *.seg extension by default.
<i>Equivalent Softkeys</i>	Stimulus > Segment Table > Recall...

SCPI.MMEMORY.LOAD.SNP.DATA

<i>Object Type</i>	Property (write only)
<i>Data Type</i>	String
<i>Target</i>	Active channel
<i>Description</i>	Loads the Touchstone file with the specified name to the measured S-parameters of the active channel. The Touchstone file types *.s1p (1-port) and *.s2p (2-port) are supported. *.s1p file loads the S11 parameter only. *.s2p file loads all S11, S21, S12 and S22 parameters. On completion of the command, the channel goes to the hold state.
<i>Range</i>	up to 254 characters
<i>Syntax</i>	app.SCPI.MMEMORY.LOAD.SNP.DATA = File
<i>Equivalent Softkeys</i>	Save/Recall > Load Data From Touchstone File > To S-parameters...

SCPI.MMEMORY.LOAD.SNP.TRACe(*Tr*).MMEMORY

<i>Object Type</i>	Property (write only)
<i>Data Type</i>	String
<i>Target</i>	The specified memory trace <i>Tr</i> of active channel, <i>Tr</i> : trace number 1–16 (see Table 3 on page 33) Active channel set by command SCPI.DISPlay.WINDOW(Ch).ACTivate
<i>Description</i>	Loads the Touchstone file with the specified name to the memory trace. The Touchstone file types *.s1p (1-port) and *.s2p (2-port) are supported. The current measured S-parameter of data trace selects the appropriate S-parameter from Touchstone file. After successful load the display of memory trace is automatically switched on.
<i>Range</i>	up to 254 characters
<i>Syntax</i>	<i>app</i> .SCPI.MMEMORY.LOAD.SNP.TRACe(<i>Tr</i>).MMEMORY = <i>File</i>
<i>Equivalent Softkeys</i>	Save/Recall > Load Data From Touchstone File > To Active Trace...

SCPI.MMEMORY.LOAD.STATE

<i>Object Type</i>	Property (write only)
<i>Data Type</i>	String
<i>Target</i>	Analyzer
<i>Description</i>	Recalls the specified analyzer state file. The file must be saved by the SCPI.MMEMORY.STORE.STATE property.
<i>Range</i>	up to 254 characters
<i>Syntax</i>	<i>app.SCPI.MMEMORY.LOAD.STATE = File</i>
<i>Notes</i>	If the full path to the file is not specified, the \State subdirectory of the main directory will be searched for the file. The analyzer state files have *.sta extension by default.
<i>Equivalent Softkeys</i>	Save/Recall > Recall State > State...

SCPI.MMEMORY.MDIRECTORY

<i>Object Type</i>	Property (write only)
<i>Data Type</i>	String
<i>Target</i>	Analyzer
<i>Description</i>	Creates a new directory (folder). Contains the full path to the folder being created.
<i>Range</i>	up to 254 characters
<i>Syntax</i>	<i>app.SCPI.MMEMORY.MDIRECTORY = Path</i>
<i>Equivalent Softkeys</i>	None

SCPI.MMEMORY.STORE.CHANNEL.CLEAR

<i>Object Type</i>	Method
<i>Target</i>	Analyzer
<i>Description</i>	Clears the memory of the channel state saved by the SCPI.MMEMORY.STORE.CHANNEL.STATE property.
<i>Syntax</i>	<code>app.SCPI.MMEMORY.STORE.CHANNEL.CLEAR</code>
<i>Equivalent Softkeys</i>	Save/Recall > Save Channel > Clear States

SCPI.MMEMORY.STORE.CHANNEL.STATE

<i>Object Type</i>	Property (write only)
<i>Data Type</i>	String
<i>Target</i>	Active channel
<i>Description</i>	Saves the analyzer state of the items set for the active channel into one of the four memory registers.
<i>Range</i>	"A" : Save to register A "B" : Save to register B "C" : Save to register C "D" : Save to register D
<i>Out of Range</i>	The value is ignored.
<i>Syntax</i>	<code>app.SCPI.MMEMORY.STORE.CHANNEL.STATE = "A"</code>
<i>Equivalent Softkeys</i>	Save/Recall > Save Channel > State A B C D

SCPI.MMEMORY.STORE.CKIT(Ck)

<i>Object Type</i>	Property (write only)
<i>Data Type</i>	String
<i>Target</i>	Calibration kit Ck, Ck: calibration kit number 1–11 (see Table 6 on page 181)
<i>Description</i>	Saves the definition file for the calibration kit parameters.
<i>Range</i>	up to 254 characters
<i>Syntax</i>	<i>app.SCPI.MMEMORY.STORE.CKIT(Ck) = File</i>
<i>Notes</i>	If the full path to the file is not specified, the file will be saved to the \CalKit subdirectory of the main directory. The calibration kit definition file has *.ckd extension by default.
<i>Equivalent Softkeys</i>	None

SCPI.MMEmory.STORe.FDATa

<i>Object Type</i>	Property (write only)
<i>Data Type</i>	String
<i>Target</i>	Active trace of the active channel
<i>Description</i>	Saves the CSV formatted data into a file.
<i>Range</i>	up to 254 characters
<i>Syntax</i>	<i>app.SCPI.MMEmory.STORe.FDATa = File</i>
<i>Notes</i>	If the full path to the file is not specified, the file will be saved to the \CSV subdirectory of the main directory. The files have *.csv extension by default.
<i>Equivalent Softkeys</i>	Save/Recall > Save Trace Data

SCPI.MMEMORY.STORE.IMAGE

<i>Object Type</i>	Property (write only)
<i>Data Type</i>	String
<i>Target</i>	Analyzer
<i>Description</i>	Saves the display image in BMP or PNG format into a file.
<i>Range</i>	up to 254 characters
<i>Syntax</i>	<i>app.SCPI.MMEMORY.STORE.IMAGE = File</i>
<i>Notes</i>	If the full path to the file is not specified, the file will be saved to the \Image subdirectory of the main directory. If the file has *.png extension, the file has PNG format, in all the other cases the file has BMP format.
<i>Equivalent Softkeys</i>	System > Print > Print Windows > Save as...

SCPI.MMEMORY.STORE.LIMIT

<i>Object Type</i>	Property (write only)
<i>Data Type</i>	String
<i>Target</i>	Active trace of the active channel
<i>Description</i>	Saves the limit table into a file with the specified name.
<i>Range</i>	up to 254 characters
<i>Syntax</i>	<i>app.SCPI.MMEMORY.STORE.LIMIT = File</i>
<i>Notes</i>	If the full path to the file is not specified, the file will be saved to the \Limit subdirectory of the main directory. The files have *.lim extension by default.
<i>Equivalent Softkeys</i>	Analysis > Limit Test > Edit Limit Line > Save Limit Table

SCPI.MMEMORY.STORE.PLOSS(*Pt*)

<i>Object Type</i>	Property (write only)
<i>Data Type</i>	String
<i>Target</i>	Port <i>Pt</i> of the active channel, <i>Pt</i> : port number 1–2 (see Table 2 on page 25)
<i>Description</i>	Saves the loss compensation table into a file with the specified name.
<i>Range</i>	up to 254 characters
<i>Syntax</i>	<i>app.SCPI.MMEMORY.STORE.PLOSS(Pt) = File</i>
<i>Notes</i>	If the full path to the file is not specified, the file will be saved to the \CalKit subdirectory of the main directory. The loss compensation files have *.lct extension by default.
<i>Equivalent Softkeys</i>	Calibration > Power Calibration > Loss Compen > Export Loss Table

SCPI.MMEMORY.STORE.RLIMIT

<i>Object Type</i>	Property (write only)
<i>Data Type</i>	String
<i>Target</i>	Active trace of the active channel
<i>Description</i>	Saves the ripple limit table into a file with the specified name.
<i>Range</i>	up to 254 characters
<i>Syntax</i>	<i>app.SCPI.MMEMORY.STORE.RLIMIT = File</i>
<i>Notes</i>	If the full path to the file is not specified, the file will be saved to the \Limit subdirectory of the main directory. The ripple limit files have *.rlm extension by default.
<i>Equivalent Softkeys</i>	Analysis > Ripple Limit > Edit Ripple Limit > Save Ripple Limit Table

SCPI.MMEMORY.STORESEGMENT

<i>Object Type</i>	Property (write only)
<i>Data Type</i>	String
<i>Target</i>	Active channel
<i>Description</i>	Saves the segment table in a file with the specified name.
<i>Range</i>	up to 254 characters
<i>Syntax</i>	<i>app.SCPI.MMEMORY.STORESEGMENT = File</i>
<i>Notes</i>	If the full path to the file is not specified, the file will be saved to the \Segment subdirectory of the main directory. The segment files have *.seg extension by default.
<i>Equivalent Softkeys</i>	Stimulus > Segment Table > Save...

SCPI.MMEMORY.STORE.SNP.DATA

<i>Object Type</i>	Property (write only)
<i>Data Type</i>	String
<i>Target</i>	Active channel
<i>Description</i>	Saves the measured S-parameters of the active channel into a Touchstone file with the specified name. The file type (1-port or 2-port) is set by the SCPI.MMEMORY.STORE.SNP.TYPE.S1P and SCPI.MMEMORY.STORE.SNP.TYPE.S2P properties. 1-port type file saves one reflection parameter: S11 or S22. 2-port type file saves all the four parameters: S11, S21, S12, S22.
<i>Range</i>	up to 254 characters
<i>Syntax</i>	<code>app.SCPI.MMEMORY.STORE.SNP.DATA = File</code>
<i>Notes</i>	If the full path to the file is not specified, the file will be saved to the \FixtureSim subdirectory of the main directory. The 1-port measurement files have *.slp extension; the 2-port measurement files have *.s2p extension.
<i>Equivalent Softkeys</i>	Save/Recall > Save Data to Touchstone File > Save File...

SCPI.MMEMORY.STORE.SNP.FORMAT

<i>Object Type</i>	Property (read/write)
<i>Data Type</i>	String
<i>Target</i>	Active channel
<i>Description</i>	The data format for the S-parameters saving by the SCPI.MMEMORY.STORE.SNP.DATA property.
<i>Range</i>	" MA" : Logarithmic Magnitude / Angle format " DB" : Linear Magnitude / Angle format " RI" : Real part /Imaginary part format
<i>Out of Range</i>	The value is ignored.
<i>Preset Value</i>	"RI"
<i>Syntax</i>	Dim Param As String <code>Param = app.SCPI.MMEMORY.STORE.SNP.FORMAT</code> <code>app.SCPI.MMEMORY.STORE.SNP.FORMAT = "DB"</code>
<i>Equivalent Softkeys</i>	Save/Recall > Save Data to Touchstone File > Format

SCPI.MMEMory.STORe.SNP.TYPE.S1P

<i>Object Type</i>	Property (read/write)
<i>Data Type</i>	Long
<i>Target</i>	Active channel
<i>Description</i>	Selects 1-port measurements and sets port numbers (selects S11 or S22 parameter) when saving the measured S-parameters by the SCPI.MMEMory.STORe.SNP.DATA property.
<i>Range</i>	from 1 to 2
<i>Out of Range</i>	An error occurs. Error code 222.
<i>Preset Value</i>	1
<i>Syntax</i>	<p>Dim Value As Long</p> <p><i>Value</i> = <i>app.SCPI.MMEMory.STORe.SNP.TYPE.S1P</i></p> <p><i>app.SCPI.MMEMory.STORe.SNP.TYPE.S1P</i> = 2</p>
<i>Equivalent Softkeys</i>	<p>Save/Recall > Save Data to Touchstone File > Type > 1-Port (s1p)</p> <p>Save/Recall > Save Data to Touchstone File > Select Port</p>

SCPI.MMEMORY.STORE.SNP.TYPE.S2P

<i>Object Type</i>	Property (read/write)				
<i>Data Type</i>	Variant (array of long)				
<i>Target</i>	Active channel				
<i>Description</i>	<p>Selects 2-port measurements and sets port numbers when saving the measured S-parameters by the SCPI.MMEMORY.STORE.SNP.DATA property.</p> <p>The array contains 2 elements:</p> <table style="margin-left: 40px;"> <tr> <td><i>Data(0)</i></td> <td>First port number (reads out 1, ignored when written);</td> </tr> <tr> <td><i>Data(1)</i></td> <td>Second port number (reads out 2, ignored when written).</td> </tr> </table>	<i>Data(0)</i>	First port number (reads out 1, ignored when written);	<i>Data(1)</i>	Second port number (reads out 2, ignored when written).
<i>Data(0)</i>	First port number (reads out 1, ignored when written);				
<i>Data(1)</i>	Second port number (reads out 2, ignored when written).				
<i>Syntax</i>	<pre>Dim Data As Variant Data = app.SCPI.MMEMORY.STORE.SNP.TYPE.S2P app.SCPI.MMEMORY.STORE.SNP.TYPE.S2P = Array(1, 2)</pre>				
<i>Equivalent Softkeys</i>	Save/Recall > Save Data to Touchstone File > Type > 2-Port (s2p)				

SCPI.MMEMORY.STORE.STATE

<i>Object Type</i>	Property (write only)
<i>Data Type</i>	String
<i>Target</i>	Analyzer
<i>Description</i>	Saves the analyzer state into a file with the specified name.
<i>Range</i>	up to 254 characters
<i>Syntax</i>	<i>app.SCPI.MMEMORY.STORE.STATE = File</i>
<i>Notes</i>	If the full path to the file is not specified, the file will be saved to the \State subdirectory of the main directory. The state files have *.sta extension by default.
<i>Equivalent Softkeys</i>	Save/Recall > Save State > State...

SCPI.MMEMORY.STORE.STYPE

<i>Object Type</i>	Property (read/write)
<i>Data Type</i>	String
<i>Target</i>	Analyzer
<i>Description</i>	Selects the type of the analyzer or channel state saving by the SCPI.MMEMORY.STORE.STATE or SCPI.MMEMORY.STORE.CHANNEL.STATE property.
<i>Range</i>	"STATe" : Measurement conditions "CSTate" : Measurement conditions and calibration tables "DSTate" : Measurement conditions and data traces "CDSTate" : Measurement conditions, calibration tables and data traces
<i>Notes</i>	The short form of the parameter is indicated by upper case letters. There is no distinction between upper and lower case letters when the property is written. When the property is read out, the short format is indicated by upper case letters.
<i>Out of Range</i>	An error occurs. Error code 205.
<i>Preset Value</i>	"CST"
<i>Syntax</i>	Dim Param As String <i>Param</i> = app.SCPI.MMEMORY.STORE.STYPE app.SCPI.MMEMORY.STORE.STYPE = "STATE"
<i>Equivalent Softkeys</i>	Save/Recall > Save Type

SCPI.OUTPut.STATE

<i>Object Type</i>	Property (read/write)
<i>Data Type</i>	Boolean
<i>Target</i>	Analyzer
<i>Description</i>	Sets the ON/OFF state of the stimulus signal output. Measurements cannot be performed when the stimulus signal output is set to OFF.
<i>Allowable Values</i>	True: Stimulus signal output ON False: Stimulus signal output OFF
<i>Preset Value</i>	True
<i>Syntax</i>	<p>Dim <i>Status</i> As Boolean</p> <p><i>Status</i> = <i>app.SCPI.OUTPut.STATE</i></p> <p><i>app.SCPI.OUTPut.STATE</i> = False</p>
<i>Equivalent Softkeys</i>	Stimulus > Power > RF Out

SCPI.SENSE(*Ch*).AVERage.CLEar

<i>Object Type</i>	Method
<i>Target</i>	Channel <i>Ch</i> , <i>Ch</i> : channel number 1–16 (see Table 1 on page 25)
<i>Description</i>	Resets the averaging data count to 0. Restarts the averaging process.
<i>Syntax</i>	<i>app.SCPI.SENSE(Ch).AVERage.CLEar</i>
<i>Equivalent Softkeys</i>	None

SCPI.SENSE(*Ch*).AVERage.COUnT

<i>Object Type</i>	Property (read/write)
<i>Data Type</i>	Long
<i>Target</i>	Channel <i>Ch</i> , <i>Ch</i> : channel number 1–16 (see Table 1 on page 25)
<i>Description</i>	The averaging factor, when the averaging function is set to ON by the SCPI.SENSE(<i>Ch</i>).AVERage.STATE property.
<i>Range</i>	from 1 to 999
<i>Out of Range</i>	Sets the value of the limit, which is closer to the specified value.
<i>Preset Value</i>	10
<i>Syntax</i>	Dim Value As Long <i>Value</i> = <i>app.SCPI.SENSE(Ch).AVERage.COUnT</i> <i>app.SCPI.SENSE(Ch).AVERage.COUnT</i> = 2
<i>Equivalent Softkeys</i>	Average > Avg Factor

SCPI.SENSe(*Ch*).AVERage.STATE

<i>Object Type</i>	Property (read/write)
<i>Data Type</i>	Boolean
<i>Target</i>	Channel <i>Ch</i> , <i>Ch</i> : channel number 1–16 (see Table 1 on page 25)
<i>Description</i>	The ON/OFF state of the averaging function.
<i>Allowable Values</i>	True: Averaging ON False: Averaging OFF
<i>Preset Value</i>	False
<i>Syntax</i>	<p>Dim <i>Status</i> As Boolean</p> <p><i>Status</i> = <i>app</i>.SCPI.SENSe(<i>Ch</i>).AVERage.STATE</p> <p><i>app</i>.SCPI.SENSe(<i>Ch</i>).AVERage.STATE = False</p>
<i>Equivalent Softkeys</i>	Average > Averaging

SCPI.SENSE(*Ch*).BANDwidth.RESolution

<i>Object Type</i>	Property (read/write)
<i>Data Type</i>	Double
<i>Target</i>	Channel <i>Ch</i> , <i>Ch</i> : channel number 1–16 (see Table 1 on page 25)
<i>Description</i>	The IF bandwidth value.
<i>Range</i>	from 1 to 30000
<i>Resolution</i>	In steps of 1, 1.5, 2, 3, 5, 7.
<i>Out of Range</i>	Sets the value of the limit, which is closer to the specified value.
<i>Preset Value</i>	10000
<i>Unit</i>	Hz (Hertz)
<i>Syntax</i>	<p>Dim <i>Value</i> As Double <i>Value</i> = <i>app</i>.SCPI.SENSE(<i>Ch</i>).BANDwidth.RESolution <i>app</i>.SCPI.SENSE(<i>Ch</i>).BANDwidth.RESolution = 100</p>
<i>Equivalent Softkeys</i>	Average > IF Bandwidth

SCPI.SENSE(*Ch*).CORRection.CLEar

<i>Object Type</i>	Method
<i>Target</i>	Channel <i>Ch</i> , <i>Ch</i> : channel number 1–16 (see Table 1 on page 25)
<i>Description</i>	Clears the calibration coefficient table.
<i>Syntax</i>	<i>app</i> .SCPI.SENSE(<i>Ch</i>).CORRection.CLEar
<i>Equivalent Softkeys</i>	None

SCPI.SENSE(Ch).CORRection.COEfficient.DATA(Str, Pt_r, Pt_s)

<i>Object Type</i>	Property (read/write)
<i>Data Type</i>	Variant (array of double)
<i>Target</i>	Channel <i>Ch</i> , <i>Ch</i> : channel number 1–16 (see Table 1 on page 25)
<i>Description</i>	<p>The calibration coefficient data array set by the type of the corrected error <i>Str</i>, the number of the receiver port <i>Pt_r</i> and the number of the source port <i>Pt_s</i>,</p> <p><i>Str</i> : error type (see below) <i>Pt_r</i> : the number of the receiver port 1–2 (see Table 2 on page 25) <i>Pt_s</i> : the number of the source port 1–2 (see Table 2 on page 25)</p> <p>The array size is 2N, where N is the number of measurement points.</p> <p>For the n-th point, where n from 1 to N:</p> <p style="padding-left: 40px;"><i>Data</i>(2n–2) real part of the calibration coefficients <i>Data</i>(2n–1) imaginary part of the calibration coefficients</p>
<i>Parameter</i>	<p>String <i>Str</i> – corrected error type:</p> <p>"ES": Source match "ER": Reflection tracking "ED": Directivity "EL": Load match "ET": Transmission tracking "EX": Isolation</p> <p>When ES, ER, or ED is used, the numbers of the ports <i>Pt_r</i> and <i>Pt_s</i> must be the same. When EL, ET, or EX is used, the numbers of the ports <i>Pt_r</i> and <i>Pt_s</i> must be different.</p>
<i>Syntax</i>	<pre>Dim <i>Data</i> As Variant <i>Data</i> = app.SCPI.SENSE(<i>Ch</i>).CORRection.COEfficient.DATA (<i>Str</i>, <i>Pt_r</i>, <i>Pt_s</i>) app.SCPI.SENSE(<i>Ch</i>).CORRection.COEfficient.DATA (<i>Str</i>, <i>Pt_r</i>, <i>Pt_s</i>) = <i>Data</i></pre>
<i>Notes</i>	The written calibration coefficients become effective only after the SCPI.SENSE(<i>Ch</i>).CORRection.COEfficient.SAVE method is invoked.
<i>Equivalent Softkeys</i>	None

SCPI.SENSe(Ch).CORRection.COEfficient.METHod.ERESponse

<i>Object Type</i>	Property (write only)
<i>Data Type</i>	Variant (array of long)
<i>Target</i>	Channel <i>Ch</i> , <i>Ch</i> : channel number 1–16 (see Table 1 on page 25)
<i>Description</i>	Selects the port numbers and sets the <i>1-path 2-port calibration</i> type, when the written calibration coefficients are made effective by the SCPI.SENSe(Ch).CORRection.COEfficient.SAVE method. The array contains 2 elements: <i>Data(0)</i> the number of the receiver port; <i>Data(1)</i> the number of the source port.
<i>Range</i>	Port number is 1 or 2. Array elements can not contain the same port numbers.
<i>Out of Range</i>	If an incorrect port number is specified, an error occurs (error code: 222). If the same port numbers are specified, an error occurs (error code: 220).
<i>Syntax</i>	<pre>Dim Ports As Variant Ports = Array(2, 1) app.SCPI.SENSe(Ch).CORRection.COEfficient.METHOD.ERESponse = Ports</pre>
<i>Equivalent Softkeys</i>	None

SCPI.SENSE(*Ch*).CORRection.COEFficient.METHod.RESPonse.OPEN

<i>Object Type</i>	Property (write only)
<i>Data Type</i>	Long
<i>Target</i>	Channel <i>Ch</i> , <i>Ch</i> : channel number 1–16 (see Table 1 on page 25)
<i>Description</i>	Selects the port and sets the <i>response calibration (Open)</i> type, when the written calibration coefficients are made effective by the SCPI.SENSE(<i>Ch</i>).CORRection.COEFficient.SAVE method.
<i>Range</i>	from 1 to 2
<i>Out of Range</i>	An error occurs. Error code: 222.
<i>Syntax</i>	<pre>Dim Port As Long Port = 1 app.SCPI.SENSE(Ch).CORRection.COEFficient.METHod.RESPonse.OPEN = Port</pre>
<i>Equivalent Softkeys</i>	None

SCPI.SENSE(*Ch*).CORRection.COEFficient.METHOD.RESPonse.SHORt

<i>Object Type</i>	Property (write only)
<i>Data Type</i>	Long
<i>Target</i>	Channel <i>Ch</i> , <i>Ch</i> : channel number 1–16 (see Table 1 on page 25)
<i>Description</i>	Selects the port and sets the <i>response calibration (Short)</i> type, when the written calibration coefficients are made effective by the SCPI.SENSE(<i>Ch</i>).CORRection.COEFficient.SAVE method.
<i>Range</i>	from 1 to 2
<i>Out of Range</i>	An error occurs. Error code: 222.
<i>Syntax</i>	<pre>Dim Port As Long Port = 1 app.SCPI.SENSE(Ch).CORRection.COEFficient.METHOD.RESPonse.SHORt = Port</pre>
<i>Equivalent Softkeys</i>	None

SCPI.SENSE(Ch).CORRection.COEFficient.METHod.RESPonse.THRU

<i>Object Type</i>	Property (write only)
<i>Data Type</i>	Variant (array of long)
<i>Target</i>	Channel <i>Ch</i> , <i>Ch</i> : channel number 1–16 (see Table 1 on page 25)
<i>Description</i>	Selects the ports and sets the <i>response calibration (Thru)</i> type, when the written calibration coefficients are made effective by the SCPI.SENSE(Ch).CORRection.COEFficient.SAVE method. The array contains 2 elements: <i>Data(0)</i> the number of the receiver port; <i>Data(1)</i> the number of the source port.
<i>Range</i>	Port number is 1 or 2. The array elements can not contain the same port numbers.
<i>Out of Range</i>	If an incorrect port number is specified, an error occurs (error code: 222). If the same port numbers are specified, an error occurs (error code: 220).
<i>Syntax</i>	<pre>Dim Ports As Variant Ports = Array(2, 1) app.SCPI.SENSE(Ch).CORRection.COEFficient.METHod.RESPonse.THRU = Ports</pre>
<i>Equivalent Softkeys</i>	None

SCPI.SENSE(*Ch*).CORRection.COEFficient.METHOD.SOLT1

<i>Object Type</i>	Property (write only)
<i>Data Type</i>	Long
<i>Target</i>	Channel <i>Ch</i> , <i>Ch</i> : channel number 1–16 (see Table 1 on page 25)
<i>Description</i>	Selects the port and sets the <i>full 1-port calibration</i> type, when the written calibration coefficients are made effective by the SCPI.SENSE(<i>Ch</i>).CORRection.COEFficient.SAVE method.
<i>Range</i>	from 1 to 2
<i>Out of Range</i>	An error occurs. Error code: 222.
<i>Syntax</i>	<pre>Dim Port As Long Port = 1 app.SCPI.SENSE(Ch).CORRection.COEFficient.METHOD.SOLT1= Port</pre>
<i>Equivalent Softkeys</i>	None

SCPI.SENSe(*Ch*).CORRection.COEFficient.METHOD.SOLT2

<i>Object Type</i>	Property (write only)
<i>Data Type</i>	Variant (array of long)
<i>Target</i>	Channel <i>Ch</i> , <i>Ch</i> : channel number 1–16 (see Table 1 on page 25)
<i>Description</i>	Selects the ports and sets the <i>full 2-port calibration</i> type, when the written calibration coefficients are made effective by the SCPI.SENSe(<i>Ch</i>).CORRection.COEFficient.SAVE method. The array contains 2 elements: <i>Data(0)</i> the number of the receiver port; <i>Data(1)</i> the number of the source port.
<i>Range</i>	Port number is 1 or 2. The array elements can not contain the same port numbers.
<i>Out of Range</i>	If an incorrect port number is specified, an error occurs (error code: 222). If the same port numbers are specified, an error occurs (error code: 220).
<i>Syntax</i>	<i>Dim Ports As Variant</i> <i>Ports</i> = Array(2, 1) <i>app.SCPI.SENSe(Ch).CORRection.COEFficient.METHOD.SOLT2 = Ports</i>
<i>Equivalent Softkeys</i>	None

SCPI.SENSe(*Ch*).CORRection.COEFficient.SAVE

<i>Object Type</i>	Method
<i>Target</i>	Channel <i>Ch</i> , <i>Ch</i> : channel number 1–16 (see Table 1 on page 25)
<i>Description</i>	<p>Enables the written calibration coefficients depending on the selected calibration type.</p> <p>On completion of the method the written calibration coefficients are cleared, the error correction automatically turns ON.</p> <p>At the attempt to execute this method before all the needed calibration coefficients are written, an error occurs and the method is ignored.</p>
<i>Syntax</i>	<i>app</i> .SCPI.SENSe(<i>Ch</i>).CORRection.COEFficient.SAVE
<i>Related Commands</i>	<p>Calibration type selection:</p> <p>SCPI.SENSe(<i>Ch</i>).CORRection.COEFficient.METHOD.ERESponse SCPI.SENSe(<i>Ch</i>).CORRection.COEFficient.METHOD.RESPonse.OPEN SCPI.SENSe(<i>Ch</i>).CORRection.COEFficient.METHOD.RESPonse.SHORT SCPI.SENSe(<i>Ch</i>).CORRection.COEFficient.METHOD.RESPonse.THRU SCPI.SENSe(<i>Ch</i>).CORRection.COEFficient.METHOD.SOLT1 SCPI.SENSe(<i>Ch</i>).CORRection.COEFficient.METHOD.SOLT2</p> <p>Calibration coefficient writing:</p> <p>SCPI.SENSe(<i>Ch</i>).CORRection.COEFficient.DATA (<i>Str</i>, <i>Pt_r</i>, <i>Pt_s</i>)</p>
<i>Equivalent Softkeys</i>	None

SCPI.SENSE(*Ch*).CORRection.COLLect.ACQuire.ISOLation

<i>Object Type</i>	Property (write only)
<i>Data Type</i>	Variant (array of long)
<i>Target</i>	Channel <i>Ch</i> , <i>Ch</i> : channel number 1–16 (see Table 1 on page 25)
<i>Description</i>	Measures the isolation calibration data between the source port and the receiver port. The array contains 2 elements: <i>Data(0)</i> the number of the receiver port; <i>Data(1)</i> the number of the source port.
<i>Range</i>	Port number is 1 or 2. The array elements can not contain the same port numbers.
<i>Out of Range</i>	If an incorrect port number is specified, an error occurs (error code: 222). If the same port numbers are specified, an error occurs (error code: 220).
<i>Syntax</i>	<code>app.SCPI.SENSE(<i>Ch</i>).CORRection.COLLect.ACQuire.ISOLation = Array(1, 2)</code>
<i>Notes</i>	The property writing starts the measurement for the channel independently of the trigger initiation and trigger source settings. The function of the property writing waits for the completion of the measurement.
<i>Equivalent Softkeys</i>	Calibration > Calibrate > Response (Thru) > Isolation (Optional) Calibration > Calibrate > One Path 2-Port Cal > Isolation (Optional) Calibration > Calibrate > Full 2-Port Cal > Isolation (Optional)

SCPI.SENSE(*Ch*).CORRection.COLLect.ACQuire.LOAD

<i>Object Type</i>	Property (write only)
<i>Data Type</i>	Long
<i>Target</i>	Channel <i>Ch</i> , <i>Ch</i> : channel number 1–16 (see Table 1 on page 25)
<i>Description</i>	Measures the calibration data of the <i>load</i> standard for the specified port.
<i>Range</i>	Port number is 1 or 2.
<i>Out of Range</i>	An error occurs (error code: 222).
<i>Syntax</i>	<i>app</i> .SCPI.SENSE(<i>Ch</i>).CORRection.COLLect.ACQuire.LOAD = 1
<i>Notes</i>	The property writing starts the measurement for the channel independently of the trigger initiation and trigger source settings. The function of the property writing waits for the completion of the measurement.
<i>Equivalent Softkeys</i>	Calibration > Calibrate > Response (Open) > Load (Optional) Calibration > Calibrate > Response (Short) > Load (Optional) Calibration > Calibrate > Full 1-Port Cal > Load Calibration > Calibrate > One Path 2-Port Cal > Load Calibration > Calibrate > Full 2-Port Cal > Port n Load

SCPI.SENSE(*Ch*).CORRection.COLLect.ACQuire.OPEN

<i>Object Type</i>	Property (write only)
<i>Data Type</i>	Long
<i>Target</i>	Channel <i>Ch</i> , <i>Ch</i> : channel number 1–16 (see Table 1 on page 25)
<i>Description</i>	Measures the calibration data of the <i>open</i> standard for the specified port.
<i>Range</i>	Port number is 1 or 2.
<i>Out of Range</i>	An error occurs (error code: 222).
<i>Syntax</i>	<i>app.SCPI.SENSE(Ch).CORRection.COLLect.ACQuire.OPEN= 1</i>
<i>Notes</i>	The property writing starts the measurement for the channel independently of the trigger initiation and trigger source settings. The function of the property writing waits for the completion of the measurement.
<i>Equivalent Softkeys</i>	Calibration > Calibrate > Response (Open) > Open Calibration > Calibrate > Full 1-Port Cal > Open Calibration > Calibrate > One Path 2-Port Cal > Open Calibration > Calibrate > Full 2-Port Cal > Port n Open

SCPI.SENSE(*Ch*).CORRection.COLLect.ACQuire. SHORt

<i>Object Type</i>	Property (write only)
<i>Data Type</i>	Long
<i>Target</i>	Channel <i>Ch</i> , <i>Ch</i> : channel number 1–16 (see Table 1 on page 25)
<i>Description</i>	Measures the calibration data of the <i>short</i> standard for the specified port.
<i>Range</i>	Port number is 1 or 2.
<i>Out of Range</i>	An error occurs (error code: 222).
<i>Syntax</i>	<i>app</i> .SCPI.SENSE(<i>Ch</i>).CORRection.COLLect.ACQuire.SHORt = 1
<i>Notes</i>	The property writing starts the measurement for the channel independently of the trigger initiation and trigger source settings. The function of the property writing waits for the completion of the measurement.
<i>Equivalent Softkeys</i>	Calibration > Calibrate > Response (Short) > Short Calibration > Calibrate > Full 1-Port Cal > Short Calibration > Calibrate > One Path 2-Port Cal > Short Calibration > Calibrate > Full 2-Port Cal > Port n Short

SCPI.SENSe(*Ch*).CORRection.COLLect.ACQuire.SUBClass

<i>Object Type</i>	Property (read/write)
<i>Data Type</i>	Long
<i>Target</i>	Channel <i>Ch</i> , <i>Ch</i> : channel number 1–16 (see Table 1 on page 26)
<i>Description</i>	The subclass number which is used for the calibration of the selected channel (Ch). For example, if two different subclasses are set in advance, say Thru 1 & Thru 2, which are visible at the calibration softkey, this command can select either Thru1 or Thru2. When performing Thru cal, either Thru 1 or Thru 2 set with this command is used for the calibration.
<i>Range</i>	1 to 8.
<i>Syntax</i>	<code>app.SCPI.SENSe(<i>Ch</i>).CORRection.COLLect.ACQuire.SUBClass = 2</code> <code>Subclass = app.SCPI.SENSe(<i>Ch</i>).CORRection.COLLect.ACQuire.SUBClass</code>
<i>Equivalent Softkeys</i>	None

SCPI.SENSE(*Ch*).CORRection.COLlect.ACQuire.THRU

<i>Object Type</i>	Property (write only)
<i>Data Type</i>	Variant (array of long)
<i>Target</i>	Channel <i>Ch</i> , <i>Ch</i> : channel number 1–16 (see Table 1 on page 25)
<i>Description</i>	Measures the calibration data of the <i>thru</i> standard between the source port and the receiver port. The array contains 2 elements: <i>Data(0)</i> the number of the receiver port; <i>Data(1)</i> the number of the source port.
<i>Range</i>	Port number is 1 or 2. The array elements can not contain the same port numbers.
<i>Out of Range</i>	If an incorrect port number is specified, an error occurs (error code: 222). If the same port numbers are specified, an error occurs (error code: 220).
<i>Syntax</i>	<code>app.SCPI.SENSE(<i>Ch</i>).CORRection.COLlect.ACQuire.THRU= Array(1, 2)</code>
<i>Notes</i>	The property writing starts the measurement for the channel independently of the trigger initiation and trigger source settings. The function of the property writing waits for the completion of the measurement.
<i>Equivalent Softkeys</i>	Calibration > Calibrate > Response (Thru) > Thru Calibration > Calibrate > One Path 2-Port Cal > Thru Calibration > Calibrate > Full 2-Port Cal > Port 1-2 Thru

SCPI.SENSE(*Ch*).CORRection.COLLect.CKIT.LABel

<i>Object Type</i>	Property (read/write)
<i>Data Type</i>	String
<i>Target</i>	Calibration kit, selected for channel <i>Ch</i> , <i>Ch</i> : channel number 1–16 (see Table 1 on page 25)
<i>Description</i>	The calibration kit label.
<i>Range</i>	up to 254 characters
<i>Preset Value</i>	Varies depending on the number of the calibration kit.
<i>Syntax</i>	<pre>Dim Lab As String Lab = app.SCPI.SENSE(<i>Ch</i>).CORRection.COLLect.CKIT.LABel app.SCPI.SENSE(<i>Ch</i>).CORRection.COLLect.CKIT.LABel = "User1"</pre>
<i>Equivalent Softkeys</i>	Calibration > Edit Cal Kit > Label

SCPI.SENSe(*Ch*).CORRection.COLLect.CKIT.ORDer.LOAD(*Pt*)

<i>Object Type</i>	Property (read/write)
<i>Data Type</i>	Long
<i>Target</i>	Calibration kit, selected for channel <i>Ch</i> , <i>Ch</i> : channel number 1–16 (see Table 1 on page 25)
<i>Description</i>	The number of the calibration standard of the load type, used for the measurement of the specified port <i>Pt</i> , <i>Pt</i> : port number 1–2 (see Table 1 on page 25)
<i>Range</i>	From 1 to the number of standards in the calibration kit.
<i>Out of Range</i>	If the specified standard number is greater than the number of standards in the kit, an error occurs (error code: 222). If the specified standard number is not the load standard number, an error occurs (error code: 220).
<i>Syntax</i>	<p>Dim <i>Num</i> As Long</p> <p><i>Num</i> = <i>app</i>.SCPI.SENSe(<i>Ch</i>).CORRection.COLLect.CKIT.ORDer.LOAD(<i>Pt</i>)</p> <p><i>app</i>.SCPI.SENSe(<i>Ch</i>).CORRection.COLLect.CKIT.ORDer.LOAD(<i>Pt</i>) = 1</p>
<i>Equivalent Softkeys</i>	Calibration > Cal Kit > Specify CLSs > Load Port x (Row)

SCPI.SENSe(*Ch*).CORRection.COLLect.CKIT.ORDer.OPEN(*Pt*)

<i>Object Type</i>	Property (read/write)
<i>Data Type</i>	Long
<i>Target</i>	Calibration kit, selected for channel <i>Ch</i> , <i>Ch</i> : channel number 1–16 (see Table 1 on page 25)
<i>Description</i>	The number of the calibration standard of the open type, used for the measurement of the specified port <i>Pt</i> , <i>Pt</i> : port number 1–2 (see Table 1 on page 25)
<i>Range</i>	From 1 to the number of standards in the calibration kit.
<i>Out of Range</i>	If the specified standard number is greater than the number of standards in the kit, an error occurs (error code: 222). If the specified standard number is not the open standard number, an error occurs (error code: 220).
<i>Syntax</i>	<p>Dim <i>Num</i> As Long</p> <p><i>Num</i> = <i>app</i>.SCPI.SENSe(<i>Ch</i>).CORRection.COLLect.CKIT.ORDer.OPEN(<i>Pt</i>)</p> <p><i>app</i>.SCPI.SENSe(<i>Ch</i>).CORRection.COLLect.CKIT.ORDer.OPEN(<i>Pt</i>) = 1</p>
<i>Equivalent Softkeys</i>	Calibration > Cal Kit > Specify CLSs > Open Port x (Row)

SCPI.SENSe(*Ch*).CORRection.COLLect.CKIT.ORDer.SElect

<i>Object Type</i>	Property (read/write)
<i>Data Type</i>	Long
<i>Target</i>	Calibration kit, selected for channel <i>Ch</i> , <i>Ch</i> : channel number 1–16 (see Table 1 on page 26)
<i>Description</i>	The subclass used to specify classes of calibration standards by the commands SCPI.SENSe(<i>Ch</i>).CORRection.COLLect.CKIT.ORDer.XXXX.
<i>Range</i>	1 to 8.
<i>Syntax</i>	<p>Dim <i>Num</i> As Long</p> <p><i>Num</i> = app.SCPI.SENSe(<i>Ch</i>).CORRection.COLLect.CKIT.ORDer.SElect</p> <p>app.SCPI.SENSe(<i>Ch</i>).CORRection.COLLect.CKIT.ORDer.SElect = 1</p>
<i>Equivalent Softkeys</i>	Calibration > Cal Kit > Specify CLSs > Subclass n (Column)

SCPI.SENSe(*Ch*).CORRection.COLLect.CKIT.ORDer.SHORt(*Pt*)

<i>Object Type</i>	Property (read/write)
<i>Data Type</i>	Long
<i>Target</i>	Calibration kit, selected for channel <i>Ch</i> , <i>Ch</i> : channel number 1–16 (see Table 1 on page 25)
<i>Description</i>	The number of the calibration standard of the short type, used for the measurement of the specified port <i>Pt</i> , <i>Pt</i> : port number 1–2 (see Table 1 on page 25)
<i>Range</i>	From 1 to the number of standards in the calibration kit.
<i>Out of Range</i>	If the specified standard number is greater than the number of standards in the kit, an error occurs (error code: 222). If the specified standard number is not the short standard number, an error occurs (error code: 220).
<i>Syntax</i>	<p>Dim <i>Num</i> As Long</p> <p><i>Num</i> = <i>app.SCPI.SENSe(Ch).CORRection.COLLect.CKIT.ORDer.SHORt(Pt)</i></p> <p><i>app.SCPI.SENSe(Ch).CORRection.COLLect.CKIT.ORDer.SHORt(Pt)</i> = 1</p>
<i>Equivalent Softkeys</i>	Calibration > Cal Kit > Specify CLSs > Short Port x (Row)

SCPI.SENSE(*Ch*).CORRection.COLLect.CKIT.ORDer.THRU(*Pt_m*, *Pt_n*)

<i>Object Type</i>	Property (read/write)
<i>Data Type</i>	Long
<i>Target</i>	Calibration kit, selected for channel <i>Ch</i> , <i>Ch</i> : channel number 1–16 (see Table 1 on page 25)
<i>Description</i>	The number of the calibration standard of the thru type, used for the measurement between the <i>Pt_m</i> and <i>Pt_n</i> ports, <i>Pt_m</i> : port number 1–2 (see Table 2 on page 25) <i>Pt_n</i> : port number 1–2 (see Table 2 on page 25)
<i>Range</i>	From 1 to the number of standards in the calibration kit.
<i>Out of Range</i>	If the specified standard number is greater than the number of standards in the kit, an error occurs (error code: 222). If the specified standard number is not the thru standard number, an error occurs (error code: 220).
<i>Syntax</i>	<pre>Dim Num As Long Num = app.SCPI.SENSE(<i>Ch</i>).CORRection.COLLect.CKIT.ORDer.THRU(1, 2) app.SCPI.SENSE(<i>Ch</i>).CORRection.COLLect.CKIT.ORDer.THRU(1, 2) = 1</pre>
<i>Equivalent Softkeys</i>	Calibration > Cal Kit > Specify CLSs > Thru Port x-y (Row)

SCPI.SENSE(*Ch*).CORRection.COLLect.CKIT.RESet

<i>Object Type</i>	Method
<i>Target</i>	Calibration kit, selected for channel <i>Ch</i> , <i>Ch</i> : channel number 1–16 (see Table 1 on page 25)
<i>Description</i>	Resets the calibration kit to the factory settings.
<i>Syntax</i>	<code>app.SCPI.SENSE(<i>Ch</i>).CORRection.COLLect.CKIT.RESet</code>
<i>Equivalent Softkeys</i>	Calibration > Restore Cal Kit

SCPI.SENSE(*Ch*).CORRection.COLLect.CKIT.SElect

<i>Object Type</i>	Property (read/write)
<i>Data Type</i>	Long
<i>Target</i>	Channel <i>Ch</i> , <i>Ch</i> : channel number 1–16 (see Table 1 on page 25)
<i>Description</i>	The selected calibration kit for the channel.
<i>Range</i>	from 1 to 50
<i>Out of Range</i>	An error occurs. Error code: 222.
<i>Preset Value</i>	1
<i>Syntax</i>	Dim <i>Value</i> As Long <code>Value = app.SCPI.SENSE(<i>Ch</i>).CORRection.COLLect.CKIT.SElect</code> <code>app.SCPI.SENSE(<i>Ch</i>).CORRection.COLLect.CKIT.SElect = 3</code>
<i>Equivalent Softkeys</i>	Calibration > Cal Kit > Select n

SCPI.SENSE(*Ch*).CORRection.COLLect.CKIT.STAN(*Std*).ARBitrary

<i>Object Type</i>	Property (read/write)
<i>Data Type</i>	Double
<i>Target</i>	Standard <i>Std</i> of the calibration kit specified for channel <i>Ch</i> , <i>Ch</i> : channel number 1–16 (see Table 1 on page 25) <i>Std</i> : standard number (see Table 7 on page 228)
<i>Description</i>	The value of the arbitrary impedance for the load standard.
<i>Range</i>	from –1E18 to 1E18
<i>Out of Range</i>	Sets the value of the limit, which is closer to the specified value.
<i>Preset Value</i>	50 or 75, depending on the selected calibration kit.
<i>Unit</i>	Ω (Ohm)
<i>Syntax</i>	<p>Dim <i>Value</i> As Double</p> <pre>Value = app.SCPI.SENSE(Ch).CORRection.COLLect.CKIT.STAN(Std).ARBitrary app.SCPI.SENSE(Ch).CORRection.COLLect.CKIT.STAN(Std).ARBitrary = 100</pre>
<i>Equivalent Softkeys</i>	Calibration > Cal Kit > Define STDs > Load > Terminal Impedance

Table 7. Std: Calibration Standard Number

<i>Data Type</i>	Long
<i>Description</i>	The number of the standard.
<i>Range</i>	Varies depending on the number of the standards in the calibration kit.
<i>Out of Range</i>	If the specified standard number is greater than the number of standards in the kit, an error occurs (error code: 222).
<i>Notes</i>	If the standard number is not specified, it is taken as equal to 1.

SCPI.SENSe(*Ch*).CORRection.COLLect.CKIT.STAN(*Std*).C0

<i>Object Type</i>	Property (read/write)
<i>Data Type</i>	Double
<i>Target</i>	Standard <i>Std</i> of the calibration kit specified for channel <i>Ch</i> , <i>Ch</i> : channel number 1–16 (see Table 1 on page 25) <i>Std</i> : standard number (see Table 7 on page 228)
<i>Description</i>	The C0 value of the open calibration standard.
<i>Range</i>	from –1E18 to 1E18
<i>Out of Range</i>	Sets the value of the limit, which is closer to the specified value.
<i>Preset Value</i>	Varies depending on the selected calibration kit and the standard.
<i>Unit</i>	1E–15 F (Farad)
<i>Syntax</i>	<p>Dim <i>Value</i> As Double <i>Value</i> = <i>app</i>.SCPI.SENSe(<i>Ch</i>).CORRection.COLLect.CKIT.STAN(<i>Std</i>).C0 <i>app</i>.SCPI.SENSe(<i>Ch</i>).CORRection.COLLect.CKIT.STAN(<i>Std</i>).C0 = 100</p>
<i>Equivalent Softkeys</i>	Calibration > Cal Kit > Define STDs > C1 10⁻²⁷ F/Hz

SCPI.SENSe(*Ch*).CORRection.COLLect.CKIT.STAN(*Std*).C1

<i>Object Type</i>	Property (read/write)
<i>Data Type</i>	Double
<i>Target</i>	Standard <i>Std</i> of the calibration kit specified for channel <i>Ch</i> , <i>Ch</i> : channel number 1–16 (see Table 1 on page 25) <i>Std</i> : standard number (see Table 7 on page 228)
<i>Description</i>	The C1 value of the open calibration standard.
<i>Range</i>	from –1E18 to 1E18
<i>Out of Range</i>	Sets the value of the limit, which is closer to the specified value.
<i>Preset Value</i>	Varies depending on the selected calibration kit and the standard.
<i>Unit</i>	1E–27 F/Hz (Farad/Hertz)
<i>Syntax</i>	<p>Dim <i>Value</i> As Double</p> <pre>Value = app.SCPI.SENSe(<i>Ch</i>).CORRection.COLLect.CKIT.STAN(<i>Std</i>).C1 app.SCPI.SENSe(<i>Ch</i>).CORRection.COLLect.CKIT.STAN(<i>Std</i>).C1 = 100</pre>
<i>Equivalent Softkeys</i>	Calibration > Cal Kit > Define STDs > C1 10⁻²⁷ F/Hz

SCPI.SENSe(*Ch*).CORRection.COLLect.CKIT.STAN(*Std*).C2

<i>Object Type</i>	Property (read/write)
<i>Data Type</i>	Double
<i>Target</i>	Standard <i>Std</i> of the calibration kit specified for channel <i>Ch</i> , <i>Ch</i> : channel number 1–16 (see Table 1 on page 25) <i>Std</i> : standard number (see Table 7 on page 228)
<i>Description</i>	The C2 value of the open calibration standard.
<i>Range</i>	from –1E18 to 1E18
<i>Out of Range</i>	Sets the value of the limit, which is closer to the specified value.
<i>Preset Value</i>	Varies depending on the selected calibration kit and the standard.
<i>Unit</i>	1E–36 F/Hz ² (Farad/Hertz ²)
<i>Syntax</i>	<p>Dim <i>Value</i> As Double <i>Value</i> = app.SCPI.SENSe(<i>Ch</i>).CORRection.COLLect.CKIT.STAN(<i>Std</i>).C2 app.SCPI.SENSe(<i>Ch</i>).CORRection.COLLect.CKIT.STAN(<i>Std</i>).C2 = 100</p>
<i>Equivalent Softkeys</i>	Calibration > Cal Kit > Define STDs > C2 10⁻³⁶ F/Hz²

SCPI.SENSe(*Ch*).CORRection.COLLect.CKIT.STAN(*Std*).C3

<i>Object Type</i>	Property (read/write)
<i>Data Type</i>	Double
<i>Target</i>	Standard <i>Std</i> of the calibration kit specified for channel <i>Ch</i> , <i>Ch</i> : channel number 1–16 (see Table 1 on page 25) <i>Std</i> : standard number (see Table 7 on page 228)
<i>Description</i>	The C3 value of the open calibration standard.
<i>Range</i>	from –1E18 to 1E18
<i>Out of Range</i>	Sets the value of the limit, which is closer to the specified value.
<i>Preset Value</i>	Varies depending on the selected calibration kit and the standard.
<i>Unit</i>	$1\text{E}-45 \text{ F/Hz}^3$ (Farad/Hertz ³)
<i>Syntax</i>	<p>Dim <i>Value</i> As Double <i>Value</i> = <i>app</i>.SCPI.SENSe(<i>Ch</i>).CORRection.COLLect.CKIT.STAN(<i>Std</i>).C3 <i>app</i>.SCPI.SENSe(<i>Ch</i>).CORRection.COLLect.CKIT.STAN(<i>Std</i>).C3 = 100</p>
<i>Equivalent Softkeys</i>	Calibration > Cal Kit > Define STDs > C3 10^{-45} F/Hz^3

SCPI.SENSe(*Ch*).CORRection.COLLect.CKIT.STAN(*Std*).DELay

<i>Object Type</i>	Property (read/write)
<i>Data Type</i>	Double
<i>Target</i>	Standard <i>Std</i> of the calibration kit specified for channel <i>Ch</i> , <i>Ch</i> : channel number 1–16 (see Table 1 on page 25) <i>Std</i> : standard number (see Table 7 on page 228)
<i>Description</i>	The offset delay value of the calibration standard.
<i>Range</i>	from –1E18 to 1E18
<i>Out of Range</i>	Sets the value of the limit, which is closer to the specified value.
<i>Preset Value</i>	Varies depending on the selected calibration kit and the standard.
<i>Unit</i>	s (second)
<i>Syntax</i>	Dim <i>Value</i> As Double <i>Value</i> = app.SCPI.SENSe(<i>Ch</i>).CORRection.COLLect.CKIT.STAN(<i>Std</i>).DELay <i>app.SCPI.SENSe(<i>Ch</i>).CORRection.COLLect.CKIT.STAN(<i>Std</i>).DELay</i> = 93E–12
<i>Equivalent Softkeys</i>	Calibration > Cal Kit > Define STDs > Offset Delay

SCPI.SENSe(*Ch*).CORRection.COLLect.CKIT.STAN(*Std*).FMAXimum

<i>Object Type</i>	Property (read/write)
<i>Data Type</i>	Double
<i>Target</i>	Standard <i>Std</i> of the calibration kit specified for channel <i>Ch</i> , <i>Ch</i> : channel number 1–16 (see Table 1 on page 25) <i>Std</i> : standard number (see Table 7 on page 228)
<i>Description</i>	The maximum frequency value of the calibration standard.
<i>Range</i>	from 0 to 1E14
<i>Out of Range</i>	Sets the value of the limit, which is closer to the specified value.
<i>Preset Value</i>	999GHz
<i>Unit</i>	Hz (Hertz)
<i>Syntax</i>	<p>Dim <i>Value</i> As Double <i>Value</i> = app.SCPI.SENSe(<i>Ch</i>).CORRection.COLLect.CKIT.STAN(<i>Std</i>).FMAXimum app.SCPI.SENSe(<i>Ch</i>).CORRection.COLLect.CKIT.STAN(<i>Std</i>).FMAXimum = 3E9</p>
<i>Equivalent Softkeys</i>	Calibration > Cal Kit > Define STDs > Frequency max

SCPI.SENSe(*Ch*).CORRection.COLLect.CKIT.STAN(*Std*).FMINimum

<i>Object Type</i>	Property (read/write)
<i>Data Type</i>	Double
<i>Target</i>	Standard <i>Std</i> of the calibration kit specified for channel <i>Ch</i> , <i>Ch</i> : channel number 1–16 (see Table 1 on page 25) <i>Std</i> : standard number (see Table 7 on page 228)
<i>Description</i>	The minimum frequency value of the calibration standard.
<i>Range</i>	from 0 to 1E14
<i>Out of Range</i>	Sets the value of the limit, which is closer to the specified value.
<i>Preset Value</i>	0 Hz
<i>Unit</i>	Hz (Hertz)
<i>Syntax</i>	<p>Dim <i>Value</i> As Double</p> <pre>Value = app.SCPI.SENSe(<i>Ch</i>).CORRection.COLLect.CKIT.STAN(<i>Std</i>).FMINimum app.SCPI.SENSe(<i>Ch</i>).CORRection.COLLect.CKIT.STAN(<i>Std</i>).FMINimum = 3E9</pre>
<i>Equivalent Softkeys</i>	Calibration > Cal Kit > Define STDs > Frequency min

SCPI.SENSE(*Ch*).CORRection.COLLect.CKIT.STAN(*Std*).L0

<i>Object Type</i>	Property (read/write)
<i>Data Type</i>	Double
<i>Target</i>	Standard <i>Std</i> of the calibration kit specified for channel <i>Ch</i> , <i>Ch</i> : channel number 1–16 (see Table 1 on page 25) <i>Std</i> : standard number (see Table 7 on page 228)
<i>Description</i>	The L0 value of the short calibration standard.
<i>Range</i>	from –1E18 to 1E18
<i>Out of Range</i>	Sets the value of the limit, which is closer to the specified value.
<i>Preset Value</i>	Varies depending on the selected calibration kit and the standard.
<i>Unit</i>	1E–12 H (Henry)
<i>Syntax</i>	<p>Dim <i>Value</i> As Double <i>Value</i> = <i>app</i>.SCPI.SENSE(<i>Ch</i>).CORRection.COLLect.CKIT.STAN(<i>Std</i>).L0 <i>app</i>.SCPI.SENSE(<i>Ch</i>).CORRection.COLLect.CKIT.STAN(<i>Std</i>).L0 = 100</p>
<i>Equivalent Softkeys</i>	Calibration > Edit Cal Kit > Short n > L0 10⁻¹² H

SCPI.SENSe(*Ch*).CORRection.COLLect.CKIT.STAN(*Std*).L1

<i>Object Type</i>	Property (read/write)
<i>Data Type</i>	Double
<i>Target</i>	Standard <i>Std</i> of the calibration kit specified for channel <i>Ch</i> , <i>Ch</i> : channel number 1–16 (see Table 1 on page 25) <i>Std</i> : standard number (see Table 7 on page 228)
<i>Description</i>	The L1 value of the short calibration standard.
<i>Range</i>	from –1E18 to 1E18
<i>Out of Range</i>	Sets the value of the limit, which is closer to the specified value.
<i>Preset Value</i>	Varies depending on the selected calibration kit and the standard.
<i>Unit</i>	1E–24 H/Hz (Henry/Hertz)
<i>Syntax</i>	<p>Dim <i>Value</i> As Double <i>Value</i> = app.SCPI.SENSe(<i>Ch</i>).CORRection.COLLect.CKIT.STAN(<i>Std</i>).L1 <i>app.SCPI.SENSe(<i>Ch</i>).CORRection.COLLect.CKIT.STAN(<i>Std</i>).L1</i> = 100</p>
<i>Equivalent Softkeys</i>	Calibration > Edit Cal Kit > Short n > L1 10⁻²⁴ H/Hz

SCPI.SENSe(*Ch*).CORRection.COLLect.CKIT.STAN(*Std*).L2

<i>Object Type</i>	Property (read/write)
<i>Data Type</i>	Double
<i>Target</i>	Standard <i>Std</i> of the calibration kit specified for channel <i>Ch</i> , <i>Ch</i> : channel number 1–16 (see Table 1 on page 25) <i>Std</i> : standard number (see Table 7 on page 228)
<i>Description</i>	The L2 value of the short calibration standard.
<i>Range</i>	from –1E18 to 1E18
<i>Out of Range</i>	Sets the value of the limit, which is closer to the specified value.
<i>Preset Value</i>	Varies depending on the selected calibration kit and the standard.
<i>Unit</i>	1E–33 H/Hz ² (Henry/Hertz ²)
<i>Syntax</i>	<p>Dim <i>Value</i> As Double <i>Value</i> = <i>app</i>.SCPI.SENSe(<i>Ch</i>).CORRection.COLLect.CKIT.STAN(<i>Std</i>).L2 <i>app</i>.SCPI.SENSe(<i>Ch</i>).CORRection.COLLect.CKIT.STAN(<i>Std</i>).L2 = 100</p>
<i>Equivalent Softkeys</i>	Calibration > Edit Cal Kit > Short n > L2 10^{–33} H/Hz²

SCPI.SENSe(*Ch*).CORRection.COLLect.CKIT.STAN(*Std*).L3

<i>Object Type</i>	Property (read/write)
<i>Data Type</i>	Double
<i>Target</i>	Standard <i>Std</i> of the calibration kit specified for channel <i>Ch</i> , <i>Ch</i> : channel number 1–16 (see Table 1 on page 25) <i>Std</i> : standard number (see Table 7 on page 228)
<i>Description</i>	The L3 value of the short calibration standard.
<i>Range</i>	from –1E18 to 1E18
<i>Out of Range</i>	Sets the value of the limit, which is closer to the specified value.
<i>Preset Value</i>	Varies depending on the selected calibration kit and the standard.
<i>Unit</i>	1E–42 H/Hz ³ (Henry/Hertz ³)
<i>Syntax</i>	<p>Dim <i>Value</i> As Double <i>Value</i> = app.SCPI.SENSe(<i>Ch</i>).CORRection.COLLect.CKIT.STAN(<i>Std</i>).L3 <i>app.SCPI.SENSe(<i>Ch</i>).CORRection.COLLect.CKIT.STAN(<i>Std</i>).L3</i> = 100</p>
<i>Equivalent Softkeys</i>	Calibration > Edit Cal Kit > Short n > L3 10^{–42} H/Hz³

SCPI.SENSe(*Ch*).CORRection.COLLect.CKIT.STAN(*Std*).LABel

<i>Object Type</i>	Property (read/write)
<i>Data Type</i>	String
<i>Target</i>	Standard <i>Std</i> of the calibration kit specified for channel <i>Ch</i> , <i>Ch</i> : channel number 1–16 (see Table 1 on page 25) <i>Std</i> : standard number (see Table 7 on page 228)
<i>Description</i>	The label of the calibration standard.
<i>Range</i>	up to 254 characters
<i>Preset Value</i>	Varies depending on the selected calibration kit and the standard.
<i>Syntax</i>	<pre>Dim <i>Lab</i> As String <i>Lab</i> = <i>app</i>.SCPI.SENSe(<i>Ch</i>).CORRection.COLLect.CKIT.STAN(<i>Std</i>).LABel <i>app</i>.SCPI.SENSe(<i>Ch</i>).CORRection.COLLect.CKIT.STAN(<i>Std</i>).LABel = "Open"</pre>
<i>Equivalent Softkeys</i>	Calibration > Cal Kit > Define STDs > Standard Label

SCPI.SENSE(*Ch*).CORRection.COLLect.CKIT.STAN(*Std*).LOSS

<i>Object Type</i>	Property (read/write)
<i>Data Type</i>	Double
<i>Target</i>	Standard <i>Std</i> of the calibration kit specified for channel <i>Ch</i> , <i>Ch</i> : channel number 1–16 (see Table 1 on page 25) <i>Std</i> : standard number (see Table 7 on page 228)
<i>Description</i>	The offset loss value of the calibration standard.
<i>Range</i>	from –1E18 to 1E18
<i>Out of Range</i>	Sets the value of the limit, which is closer to the specified value.
<i>Preset Value</i>	Varies depending on the selected calibration kit and the standard.
<i>Unit</i>	Ω/s (Ohm/second)
<i>Syntax</i>	<p>Dim <i>Value</i> As Double <i>Value</i> = app.SCPI.SENSE(<i>Ch</i>).CORRection.COLLect.CKIT.STAN(<i>Std</i>).LOSS app.SCPI.SENSE(<i>Ch</i>).CORRection.COLLect.CKIT.STAN(<i>Std</i>).LOSS = 700E6</p>
<i>Equivalent Softkeys</i>	Calibration > Cal Kit > Define STDs > Offset Loss

SCPI.SENSe(*Ch*).CORRection.COLLect.CKIT.STAN(*Std*).TYPE

<i>Object Type</i>	Property (read/write)
<i>Data Type</i>	String
<i>Target</i>	Standard <i>Std</i> of the calibration kit specified for channel <i>Ch</i> , <i>Ch</i> : channel number 1–16 (see Table 1 on page 25) <i>Std</i> : standard number (see Table 7 on page 228)
<i>Description</i>	The type of the calibration standard.
<i>Range</i>	"OPEN" : Open "SHORt" : Short "LOAD" : Load "THRU" : Thru "UTHR" : Unknown Thru "SLID" : Sliding Load "DATA" : Data based "NONE" : Not defined
<i>Notes</i>	The short form of the parameter is indicated by upper case letters. There is no distinction between upper and lower case letters when the property is written. When the property is read out, the short format is indicated by upper case letters.
<i>Out of Range</i>	An error occurs. Error code: 216.
<i>Preset Value</i>	Varies depending on the selected calibration kit and the standard.
<i>Syntax</i>	Dim <i>Param</i> As String <i>Param</i> = <i>app</i> .SCPI.SENSe(<i>Ch</i>).CORRection.COLLect.CKIT.STAN(<i>Std</i>).TYPE <i>app</i> .SCPI.SENSe(<i>Ch</i>).CORRection.COLLect.CKIT.STAN(<i>Std</i>).TYPE = "OPEN"
<i>Equivalent Softkeys</i>	Calibration > Cal Kit > Define STDs > Standard Type

SCPI.SENSe(*Ch*).CORRection.COLLect.CKIT.STAN(*Std*).Z0

<i>Object Type</i>	Property (read/write)
<i>Data Type</i>	Double
<i>Target</i>	Standard <i>Std</i> of the calibration kit specified for channel <i>Ch</i> , <i>Ch</i> : channel number 1–16 (see Table 1 on page 25) <i>Std</i> : standard number (see Table 7 on page 228)
<i>Description</i>	The offset Z0 value of the calibration standard.
<i>Range</i>	from –1E18 to 1E18
<i>Out of Range</i>	Sets the value of the limit, which is closer to the specified value.
<i>Preset Value</i>	50 or 75, depending on the selected calibration kit.
<i>Unit</i>	Ω (Ohm)
<i>Syntax</i>	<p>Dim <i>Value</i> As Double <i>Value</i> = <i>app</i>.SCPI.SENSe(<i>Ch</i>).CORRection.COLLect.CKIT.STAN(<i>Std</i>).Z0 <i>app</i>.SCPI.SENSe(<i>Ch</i>).CORRection.COLLect.CKIT.STAN(<i>Std</i>).Z0 = 50</p>
<i>Equivalent Softkeys</i>	Calibration > Cal Kit > Define STDs > Offset Z0

SCPI.SENSe(*Ch*).CORRection.COLLect.CLEar

<i>Object Type</i>	Method
<i>Target</i>	Channel <i>Ch</i> , <i>Ch</i> : channel number 1–16 (see Table 1 on page 25)
<i>Description</i>	Clears the measurement values of the calibration standards.
<i>Syntax</i>	<code>app.SCPI.SENSe(<i>Ch</i>).CORRection.COLLect.CLEar</code>
<i>Equivalent Softkeys</i>	Calibration > Calibrate > Response (Open) Response (Short) Response (Thru) One Path 2-Port Cal Full 1-Port Cal Full 2-Port Cal > Cancel > OK

SCPI.SENSE(*Ch*).CORRection.COLLect.DATA.ISOLation(*Pt_r*, *Pt_s*)

<i>Object Type</i>	Property (read/write)
<i>Data Type</i>	Variant (array of double)
<i>Target</i>	Channel <i>Ch</i> , <i>Ch</i> : channel number 1–16 (see Table 1 on page 25)
<i>Description</i>	<p>The array of the isolation calibration measurements performed between the receiver port <i>Pt_r</i> and the source port <i>Pt_s</i>,</p> <p><i>Pt_r</i>: the number of the receiver port 1–2 (see Table 2 on page 25)</p> <p><i>Pt_s</i>: the number of the source port 1–2 (see Table 2 on page 25)</p> <p>The array elements are complex numbers. Also see section “Measurement Data Arrays” on page 19.</p> <p>The array size is 2N, where N is the number of measurement points.</p> <p>For the n-th point, where n from 1 to N:</p> <p style="padding-left: 20px;"><i>Data</i>(2n–2) real part of the measurement;</p> <p style="padding-left: 20px;"><i>Data</i>(2n–1) imaginary part of the measurement.</p>
<i>Syntax</i>	<p>Dim <i>Data</i> As Variant</p> <p><i>Data</i> = <i>app</i>.SCPI.SENSE(<i>Ch</i>).CORRection.COLLect.DATA.ISOLation(<i>Pt_r</i>, <i>Pt_s</i>)</p> <p><i>app</i>.SCPI.SENSE(<i>Ch</i>).CORRection.COLLect.DATA.ISOLation(<i>Pt_r</i>, <i>Pt_s</i>) = <i>Data</i></p>
<i>Related Commands</i>	SCPI.SENSE(<i>Ch</i>).CORRection.COLLect.ACQuire.ISOLation
<i>Equivalent Softkeys</i>	None

SCPI.SENSE(*Ch*).CORRection.COLLect.DATA.LOAD(*Pt*)

<i>Object Type</i>	Property (read/write)
<i>Data Type</i>	Variant (array of double)
<i>Target</i>	Channel <i>Ch</i> , <i>Ch</i> : channel number 1–16 (see Table 1 on page 25)
<i>Description</i>	<p>The array of the <i>load</i> calibration standard measurements for the port <i>Pt</i>, <i>Pt</i>: port number 1–2 (see Table 2 on page 25)</p> <p>The array elements are complex numbers. Also see section “Measurement Data Arrays” on page 19.</p> <p>The array size is 2N, where N is the number of measurement points. For the n-th point, where n from 1 to N:</p> <p style="padding-left: 40px;"><i>Data</i>(2n–2) real part of the measurement;</p> <p style="padding-left: 40px;"><i>Data</i>(2n–1) imaginary part of the measurement.</p>
<i>Syntax</i>	<pre>Dim <i>Data</i> As Variant <i>Data</i> = <i>app</i>.SCPI.SENSE(<i>Ch</i>).CORRection.COLLect.DATA.LOAD(<i>Pt</i>) <i>app</i>.SCPI.SENSE(<i>Ch</i>).CORRection.COLLect.DATA.LOAD(<i>Pt</i>) = <i>Data</i></pre>
<i>Related Commands</i>	SCPI.SENSE(<i>Ch</i>).CORRection.COLLect.ACQuire.LOAD
<i>Equivalent Softkeys</i>	None

SCPI.SENSE(*Ch*).CORRection.COLLect.DATA.OPEN(*Pt*)

<i>Object Type</i>	Property (read/write)
<i>Data Type</i>	Variant (array of double)
<i>Target</i>	Channel <i>Ch</i> , <i>Ch</i> : channel number 1–16 (see Table 1 on page 25)
<i>Description</i>	<p>The array of the <i>open</i> calibration standard measurements for the port <i>Pt</i>, <i>Pt</i>: port number 1–2 (see Table 2 on page 25)</p> <p>The array elements are complex numbers. Also see section “Measurement Data Arrays” on page 19.</p> <p>The array size is 2N, where N is the number of measurement points. For the n-th point, where n from 1 to N:</p> <p style="padding-left: 40px;"><i>Data</i>(2n–2) real part of the measurement;</p> <p style="padding-left: 40px;"><i>Data</i>(2n–1) imaginary part of the measurement.</p>
<i>Syntax</i>	<p>Dim <i>Data</i> As Variant</p> <p><i>Data</i> = <i>app</i>.SCPI.SENSE(<i>Ch</i>).CORRection.COLLect.DATA.OPEN(<i>Pt</i>)</p> <p><i>app</i>.SCPI.SENSE(<i>Ch</i>).CORRection.COLLect.DATA.OPEN(<i>Pt</i>) = <i>Data</i></p>
<i>Related Commands</i>	SCPI.SENSE(<i>Ch</i>).CORRection.COLLect.ACQuire.OPEN
<i>Equivalent Softkeys</i>	None

SCPI.SENSE(*Ch*).CORRection.COLLect.DATA.SHORt(*Pt*)

<i>Object Type</i>	Property (read/write)
<i>Data Type</i>	Variant (array of double)
<i>Target</i>	Channel <i>Ch</i> , <i>Ch</i> : channel number 1–16 (see Table 1 on page 25)
<i>Description</i>	<p>The array of the <i>short</i> calibration standard measurements for the port <i>Pt</i>, <i>Pt</i>: port number 1–2 (see Table 2 on page 25)</p> <p>The array elements are complex numbers. Also see section “Measurement Data Arrays” on page 19.</p> <p>The array size is 2N, where N is the number of measurement points. For the n-th point, where n from 1 to N:</p> <ul style="list-style-type: none"> <i>Data</i>(2n–2) real part of the measurement; <i>Data</i>(2n–1) imaginary part of the measurement.
<i>Syntax</i>	<pre>Dim <i>Data</i> As Variant <i>Data</i> = <i>app</i>.SCPI.SENSE(<i>Ch</i>).CORRection.COLLect.DATA.SHORt(<i>Pt</i>) <i>app</i>.SCPI.SENSE(<i>Ch</i>).CORRection.COLLect.DATA.SHORt(<i>Pt</i>) = <i>Data</i></pre>
<i>Related Commands</i>	SCPI.SENSE(<i>Ch</i>).CORRection.COLLect.ACQuire.SHORT
<i>Equivalent Softkeys</i>	None

SCPI.SENSE(*Ch*).CORRection.COLlect.DATA.THRU. MATCH(*Pt_r,Pt_s*)

<i>Object Type</i>	Property (read/write)
<i>Data Type</i>	Variant (array of double)
<i>Target</i>	Channel <i>Ch</i> , <i>Ch</i> : channel number 1–16 (see Table 1 on page 25)
<i>Description</i>	<p>The array of the reflection measurements of the <i>thru</i> standard connected between the receiver port <i>Pt_r</i> and the source port <i>Pt_s</i>,</p> <p><i>Pt_r</i>: the number of the receiver port 1–2 (see Table 2 on page 25)</p> <p><i>Pt_s</i>: the number of the source port 1–2 (see Table 2 on page 25)</p> <p>The array elements are complex numbers. Also see section “Measurement Data Arrays” on page 19.</p> <p>The array size is 2<i>N</i>, where <i>N</i> is the number of measurement points.</p> <p>For the <i>n</i>-th point, where <i>n</i> from 1 to <i>N</i>:</p> <p style="padding-left: 40px;"><i>Data</i>(2<i>n</i>–2) real part of the measurement;</p> <p style="padding-left: 40px;"><i>Data</i>(2<i>n</i>–1) imaginary part of the measurement.</p>
<i>Syntax</i>	<pre>Dim <i>Data</i> As Variant <i>Data</i> = app.SCPI.SENSE(<i>Ch</i>).CORRection.COLlect.DATA.THRU.MATCH(<i>Pt_r, Pt_s</i>) app.SCPI.SENSE(<i>Ch</i>).CORRection.COLlect.DATA.THRU.MATCH(<i>Pt_r, Pt_s</i>) = <i>Data</i></pre>
<i>Related Commands</i>	SCPI.SENSE(<i>Ch</i>).CORRection.COLlect.ACQuire.THRU
<i>Equivalent Softkeys</i>	None

SCPI.SENSe(*Ch*).CORRection.COLLect.DATA.THRU.TRANSmission(*Pt_r*, *Pt_s*)

<i>Object Type</i>	Property (read/write)
<i>Data Type</i>	Variant (array of double)
<i>Target</i>	Channel <i>Ch</i> , <i>Ch</i> : channel number 1–16 (see Table 1 on page 25)
<i>Description</i>	<p>The array of the transmission measurements performed between the receiver port <i>Pt_r</i> and the source port <i>Pt_s</i>, using the <i>thru</i> standard,</p> <p><i>Pt_r</i>: the number of the receiver port 1–2 (see Table 2 on page 25)</p> <p><i>Pt_s</i>: the number of the source port 1–2 (see Table 2 on page 25)</p> <p>The array elements are complex numbers. Also see section “Measurement Data Arrays” on page 19.</p> <p>The array size is 2N, where N is the number of measurement points.</p> <p>For the n-th point, where n from 1 to N:</p> <p style="padding-left: 40px;"><i>Data</i>(2n–2) real part of the measurement;</p> <p style="padding-left: 40px;"><i>Data</i>(2n–1) imaginary part of the measurement.</p>
<i>Syntax</i>	<pre>Dim <i>Data</i> As Variant <i>Data</i> = <i>app</i>.SCPI.SENSe(<i>Ch</i>).CORRection.COLLect.DATA.THRU.TRANSmission(<i>Pt_r</i>, <i>Pt_s</i>) <i>app</i>.SCPI.SENSe(<i>Ch</i>).CORRection.COLLect.DATA.THRU.TRANSmission(<i>Pt_r</i>, <i>Pt_s</i>) = <i>Data</i></pre>
<i>Related Commands</i>	SCPI.SENSe(<i>Ch</i>).CORRection.COLLect.ACQuire.THRU
<i>Equivalent Softkeys</i>	None

SCPI.SENSe(*Ch*).CORRection.COLLect.ECAL.INFormation

<i>Object Type</i>	Property (read only)
<i>Data Type</i>	String
<i>Target</i>	AutoCal module (<i>Ch</i> – arbitrary number 1 to 16)
<i>Description</i>	<p>Gets information about the AutoCal Module connected to the Network Analyzer in a string with comma separated fields.</p> <p>Autocal Module Information:</p> <ul style="list-style-type: none"> Model Name, Serial Number, Current Temperature of AutoCal Module, <p>Selected Characterization Information:</p> <ul style="list-style-type: none"> Characterization Name, Characterization Date and Time, Min Frequency, Max Frequency, Number of Points, Characterization Temperature, PortA Connector, PortB Connector, PortA Adapter, PortB Adapter, Analyzer, Location, Operator.
<i>Syntax</i>	<pre>Dim <i>ID</i> As String <i>ID</i> = <i>app</i>.SCPI.SENSe(<i>Ch</i>).CORRection.COLLect.ECAL.INFormation</pre>
<i>Equivalent Softkeys</i>	Calibration > AutoCal > Characterization Info...

SCPI.SENSe(*Ch*).CORRection.COLLect.ECAL.ORIentation.STATE

<i>Object Type</i>	Property (read/write)
<i>Data Type</i>	Boolean
<i>Target</i>	AutoCal (<i>Ch</i> – arbitrary number 1 to 16)
<i>Description</i>	The ON/OFF state of the Auto-Orientation function used when executing AutoCal.
<i>Allowable Values</i>	True: Auto-Orientation function ON False: Auto-Orientation function OFF
<i>Preset Value</i>	False
<i>Syntax</i>	<p>Dim <i>Status</i> As Boolean</p> <p><i>Status</i> = <code>app.SCPI.SENSe(<i>Ch</i>).CORRection.COLLect.ECAL.ORIentation.STATE</code></p> <p><code>app.SCPI.SENSe(<i>Ch</i>).CORRection.COLLect.ECAL.ORIentation.STATE =</code> False</p>
<i>Equivalent Softkeys</i>	Calibration > AutoCal > Orientation > Auto-Orientation

SCPI.SENSE(*Ch*).CORRection.COLLect.ECAL.PATH(*Pt*)

<i>Object Type</i>	Property (read/write)
<i>Data Type</i>	Long
<i>Target</i>	AutoCal (<i>Ch</i> – arbitrary number 1 to 16)
<i>Description</i>	Sets or reads out the AutoCal module port number which is connected to a selected Network Analyzer port (<i>Pt</i>).
<i>Allowable Values</i>	1: Port A of AutoCal Module 2: Port B of AutoCal Module 3: Port C of AutoCal Module (4 port AutoCal module only) 4: Port D of AutoCal Module (4 port AutoCal module only)
<i>Syntax</i>	Dim Value As Long Value = app.SCPI.SENSE(<i>Ch</i>).CORRection.COLLect.ECAL.PATH(<i>Pt</i>) app.SCPI.SENSE(<i>Ch</i>).CORRection.COLLect.ECAL.PATH(<i>Pt</i>) = 2
<i>Equivalent Softkeys</i>	Calibration > AutoCal > Orientation > Port 1 Port 2

SCPI.SENSE(*Ch*).CORRection.COLLect.ECAL.SOLT1

<i>Object Type</i>	Property (write only)
<i>Data Type</i>	Long
<i>Target</i>	Channel <i>Ch</i> , <i>Ch</i> : channel number 1–16 (see Table 1 on page 25)
<i>Description</i>	Executes 1-port calibration of the specified port of selected channel (<i>Ch</i>) using the AutoCal module.
<i>Syntax</i>	Dim <i>Port</i> As Long <code>app.SCPI.SENSE(<i>Ch</i>).CORRection.COLLect.ECAL.SOLT1 = <i>Port</i></code>
<i>Equivalent Softkeys</i>	Calibration > AutoCal > 1-Port AutoCal > Port 1 Port 2

SCPI.SENSE(*Ch*).CORRection.COLLect.ECAL.SOLT2

<i>Object Type</i>	Property (write only)
<i>Data Type</i>	Variant (array of long)
<i>Target</i>	Channel <i>Ch</i> , <i>Ch</i> : channel number 1–16 (see Table 1 on page 25)
<i>Description</i>	Executes full 2-port calibration between the specified 2 ports of selected channel (<i>Ch</i>) using the AutoCal module.
<i>Syntax</i>	<code>app.SCPI.SENSE(<i>Ch</i>).CORRection.COLLect.ECAL.SOLT2 = Array(2, 1)</code>
<i>Equivalent Softkeys</i>	Calibration > AutoCal > 2-Port AutoCal

SCPI.SENSE(*Ch*).CORRection.COLLect.ECAL.UChar

<i>Object Type</i>	Property (read/write)
<i>Data Type</i>	String
<i>Target</i>	AutoCal (<i>Ch</i> – arbitrary number 1 to 16)
<i>Description</i>	Sets or reads out the Characteristic used when executing AutoCal (factory or user characterization)..
<i>Range</i>	"CHAR0" : Factory characterization "CHAR1" : User characterization 1 "CHAR2" : User characterization 2 "CHAR3" : User characterization 3
<i>Preset Value</i>	CHAR0
<i>Syntax</i>	<pre>Dim Param As String Param = app.SCPI.SENSE(<i>Ch</i>).CORRection.COLLect.ECAL.UChar app.SCPI.SENSE(<i>Ch</i>).CORRection.COLLect.ECAL.UChar = "CHAR0"</pre>
<i>Equivalent Softkeys</i>	Calibration > AutoCal > Characterization

SCPI.SENSe(*Ch*).CORRection.COLLect.ECAL.UTHRu.STATE

<i>Object Type</i>	Property (read/write)
<i>Data Type</i>	Boolean
<i>Target</i>	AutoCal (<i>Ch</i> – arbitrary number 1 to 16)
<i>Description</i>	The ON/OFF state of the Unknown Thru feature used when executing AutoCal.
<i>Allowable Values</i>	True: Unknown Thru feature ON False: Unknown Thru feature OFF
<i>Preset Value</i>	False
<i>Syntax</i>	Dim <i>Status</i> As Boolean <i>Status</i> = <i>app</i> .SCPI.SENSe(<i>Ch</i>).CORRection.COLLect.ECAL.UTHRu.STATE <i>app</i> .SCPI.SENSe(<i>Ch</i>).CORRection.COLLect.ECAL.UTHRu.STATE = False
<i>Note</i>	PLANAR 304/1 does not support
<i>Equivalent Softkeys</i>	Calibration > AutoCal > Unkn Thru

SCPI.SENSe(*Ch*).CORRection.COLLect.METHOD.ERESponse

<i>Object Type</i>	Property (write only)
<i>Data Type</i>	Variant (array of long)
<i>Target</i>	Channel <i>Ch</i> , <i>Ch</i> : channel number 1–16 (see Table 1 on page 25)
<i>Description</i>	Selects the ports and sets the <i>one path 2-port calibration</i> type for the calculation of the calibration coefficients on completion of the calibration executed by the SCPI.SENSe(<i>Ch</i>).CORRection.COLLect.SAVE method. The array contains 2 elements: <i>Data(0)</i> the number of the receiver port; <i>Data(1)</i> the number of the source port.
<i>Range</i>	Port number is 1 or 2. The array elements can not contain the same port numbers.
<i>Out of Range</i>	If an incorrect port number is specified, an error occurs (error code: 222). If the same port numbers are specified, an error occurs (error code: 220).
<i>Syntax</i>	<code>app.SCPI.SENSe(<i>Ch</i>).CORRection.COLLect.METHOD.ERESponse = Array(2, 1)</code>
<i>Equivalent Softkeys</i>	Calibration > Calibrate > One Path 2-Port Cal > Select Port

SCPI.SENSe(*Ch*).CORRection.COLLect.METHOD.RESPonse.OPEN

<i>Object Type</i>	Property (write only)
<i>Data Type</i>	Long
<i>Target</i>	Channel <i>Ch</i> , <i>Ch</i> : channel number 1–16 (see Table 1 on page 25)
<i>Description</i>	Selects the port and sets the <i>response calibration (Open)</i> type for the calculation of the calibration coefficients on completion of the calibration executed by the SCPI.SENSe(<i>Ch</i>).CORRection.COLLect.SAVE method.
<i>Range</i>	from 1 to 2
<i>Out of Range</i>	An error occurs. Error code: 222.
<i>Syntax</i>	<i>app</i> .SCPI.SENSe(<i>Ch</i>).CORRection.COLLect.METHOD.RESPonse.OPEN = 1
<i>Equivalent Softkeys</i>	Calibration > Calibrate > Response (Open) > Select Port

SCPI.SENSe(*Ch*).CORRection.COLLect.METHOD.RESPonse.SHORT

<i>Object Type</i>	Property (write only)
<i>Data Type</i>	Long
<i>Target</i>	Channel <i>Ch</i> , <i>Ch</i> : channel number 1–16 (see Table 1 on page 25)
<i>Description</i>	Selects the port and sets the <i>response calibration (Short)</i> type for the calculation of the calibration coefficients on completion of the calibration executed by the SCPI.SENSe(<i>Ch</i>).CORRection.COLLect.SAVE method.
<i>Range</i>	from 1 to 2
<i>Out of Range</i>	An error occurs. Error code: 222.
<i>Syntax</i>	<i>app</i> .SCPI.SENSe(<i>Ch</i>).CORRection.COLLect.METHOD.RESPonse.SHORT = 1
<i>Equivalent Softkeys</i>	Calibration > Calibrate > Response (Short) > Select Port

SCPI.SENSe(*Ch*).CORRection.COLLect.METHOD.RESPonse.THRU

<i>Object Type</i>	Property (write only)
<i>Data Type</i>	Variant (array of long)
<i>Target</i>	Channel <i>Ch</i> , <i>Ch</i> : channel number 1–16 (see Table 1 on page 25)
<i>Description</i>	Selects the ports and sets the <i>response calibration (Thru)</i> type for the calculation of the calibration coefficients on completion of the calibration executed by the SCPI.SENSe(<i>Ch</i>).CORRection.COLLect.SAVE method. The array contains 2 elements: <i>Data(0)</i> the number of the receiver port; <i>Data(1)</i> the number of the source port.
<i>Range</i>	Port number is 1 or 2. Array elements can not contain the same port numbers.
<i>Out of Range</i>	If an incorrect port number is specified, an error occurs (error code: 222). If the same port numbers are specified, an error occurs (error code: 220).
<i>Syntax</i>	<code>app.SCPI.SENSe(<i>Ch</i>).CORRection.COLLect.METHOD.RESPonse.THRU = Array(2, 1)</code>
<i>Equivalent Softkeys</i>	Calibration > Calibrate > Response (Thru) > Select Port

SCPI.SENSE(*Ch*).CORRection.COLLect.METHOD.SOLT1

<i>Object Type</i>	Property (write only)
<i>Data Type</i>	Long
<i>Target</i>	Channel <i>Ch</i> , <i>Ch</i> : channel number 1–16 (see Table 1 on page 25)
<i>Description</i>	Selects the port and sets the <i>full 1-port calibration</i> type for the calculation of the calibration coefficients on completion of the calibration executed by the SCPI.SENSE(<i>Ch</i>).CORRection.COLLect.SAVE method.
<i>Range</i>	from 1 to 2
<i>Out of Range</i>	An error occurs. Error code: 222.
<i>Syntax</i>	<i>app</i> .SCPI.SENSE(<i>Ch</i>).CORRection.COLLect.METHOD.SOLT1 = 1
<i>Equivalent Softkeys</i>	Calibration > Calibrate > Full 1-Port Cal > Select Port

SCPI.SENSE(*Ch*).CORRection.COLLect.METHOD.SOLT2

<i>Object Type</i>	Property (write only)
<i>Data Type</i>	Variant (array of long)
<i>Target</i>	Channel <i>Ch</i> , <i>Ch</i> : channel number 1–16 (see Table 1 on page 25)
<i>Description</i>	Selects the port and sets the <i>full 2-port calibration</i> type for the calculation of the calibration coefficients on completion of the calibration executed by the SCPI.SENSE(<i>Ch</i>).CORRection.COLLect.SAVE method. The array contains 2 elements: <i>Data(0)</i> the number of the receiver port; <i>Data(1)</i> the number of the source port.
<i>Range</i>	Port number is 1 or 2. Array elements can not contain the same port numbers.
<i>Out of Range</i>	If an incorrect port number is specified, an error occurs (error code: 222). If the same port numbers are specified, an error occurs (error code: 220).
<i>Syntax</i>	<code>app.SCPI.SENSE(<i>Ch</i>).CORRection.COLLect.METHOD.SOLT2 = Array(2, 1)</code>
<i>Equivalent Softkeys</i>	Calibration > Calibrate > Full 2-Port Cal

SCPI.SENSE(*Ch*).CORRection.COLlect.METHOD.TYPE

<i>Object Type</i>	Property (read only)
<i>Data Type</i>	String
<i>Target</i>	Channel <i>Ch</i> , <i>Ch</i> : channel number 1–16 (see Table 1 on page 25)
<i>Description</i>	The calibration type selected for calculating of the calibration coefficients on completion of the calibration executed by the SCPI.SENSE(<i>Ch</i>).CORRection.COLlect.SAVE method.
<i>Range</i>	"RESPO" : Response (Open) "RESPS" : Response (Short) "RESPT" : Response (Thru) "SOLT1" : Full 1–port calibration "SOLT2" : Full 2–port calibration "1PATH" : One path 2–port calibration "NONE" : Not defined
<i>Syntax</i>	<pre>Dim Param As String Param = app.SCPI.SENSE(<i>Ch</i>).CORRection.COLlect.METHOD.TYPE</pre>
<i>Equivalent Softkeys</i>	None

SCPI.SENSE(*Ch*).CORRection.COLLect.SAVE

<i>Object Type</i>	Method
<i>Target</i>	Channel <i>Ch</i> , <i>Ch</i> : channel number 1–16 (see Table 1 on page 25)
<i>Description</i>	<p>Calculates the calibration coefficients from the calibration standards measurements depending on the selected calibration type.</p> <p>On completion of the method, all the calibration standards measurements are cleared and the error correction automatically turns ON.</p> <p>At the attempt to execute this method before all the needed standards are measured, an error occurs and the method is ignored.</p> <p>Before executing this command it is necessary to select calibration type by one of commands SCPI.SENSE(<i>Ch</i>).CORRection.COLLect.METHOD.XXXX.</p>
<i>Syntax</i>	<i>app</i> .SCPI.SENSE(<i>Ch</i>).CORRection.COLLect.SAVE
<i>Related Commands</i>	<p>Calibration type selection:</p> <p>SCPI.SENSE(<i>Ch</i>).CORRection.COLLect.METHOD.RESPonse.OPEN SCPI.SENSE(<i>Ch</i>).CORRection.COLLect.METHOD.RESPonse.SHORT SCPI.SENSE(<i>Ch</i>).CORRection.COLLect.METHOD.RESPonse.THRU SCPI.SENSE(<i>Ch</i>).CORRection.COLLect.METHOD.ERESponse SCPI.SENSE(<i>Ch</i>).CORRection.COLLect.METHOD.SOLT1 SCPI.SENSE(<i>Ch</i>).CORRection.COLLect.METHOD.SOLT2</p> <p>Calibration standards measurement:</p> <p>SCPI.SENSE(<i>Ch</i>).CORRection.COLLect.ACQuire.ISOLation SCPI.SENSE(<i>Ch</i>).CORRection.COLLect.ACQuire.LOAD SCPI.SENSE(<i>Ch</i>).CORRection.COLLect.ACQuire.OPEN SCPI.SENSE(<i>Ch</i>).CORRection.COLLect.ACQuire.SHORT SCPI.SENSE(<i>Ch</i>).CORRection.COLLect.ACQuire.THRU</p>
<i>Equivalent Softkeys</i>	Calibration > Calibrate > Response (Open) Response (Short) Response (Thru) One Path 2-Port Cal Full 1-Port Cal Full 2-Port Cal > Apply

SCPI.SENSE(Ch).CORRection.EXTension.PORT(Pt).FREQuency(Ls)

<i>Object Type</i>	Property (read/write)
<i>Data Type</i>	Double
<i>Target</i>	Port <i>Pt</i> of channel <i>Ch</i> , <i>Ch</i> : channel number 1–16 (see Table 1 on page 25) <i>Pt</i> : port number 1–2 (see Table 2 on page 25)
<i>Description</i>	The value of the frequency at the point number <i>Ls</i> for calculation of the loss for the port extension function, <i>Ls</i> : point number 1–2 (see Table 8 on page 265)
<i>Range</i>	from minimum to maximum analyzer frequency
<i>Out of Range</i>	Sets the value of the limit, which is closer to the specified value.
<i>Preset Value</i>	1E9
<i>Unit</i>	Hz (Hertz)
<i>Syntax</i>	<pre>Dim Value As Double Value = app.SCPI.SENSE(Ch).CORRection.EXTension.PORT(Pt).FREQuency(Ls) app.SCPI.SENSE(Ch).CORRection.EXTension.PORT(Pt).FREQuency(Ls) = 100E6</pre>
<i>Equivalent Softkeys</i>	Calibration > Port Extensions > Loss > Freq1 Freq2

Table 8. Ls: Point Number

<i>Data Type</i>	Long
<i>Description</i>	The point number for the loss setting in the port extension function.
<i>Range</i>	1 or 2
<i>Out of Range</i>	An error occurs. Error code: 222.
<i>Notes</i>	If the point number is not specified, it is taken as equal to 1.

SCPI.SENSe(*Ch*).CORRection.EXTension.PORT(*Pt*).INCLude(*Ls*).STATe

<i>Object Type</i>	Property (read/write)
<i>Data Type</i>	Boolean
<i>Target</i>	Port <i>Pt</i> of channel <i>Ch</i> , <i>Ch</i> : channel number 1–16 (see Table 1 on page 25) <i>Pt</i> : port number 1–2 (see Table 2 on page 25)
<i>Description</i>	The ON/OFF state of the loss compensation at the point number <i>Ls</i> to calculate the loss for the port extension function, <i>Ls</i> : point number 1–2 (see Table 8 on page 265)
<i>Allowable Values</i>	True: Loss compensation ON False: Loss compensation OFF
<i>Preset Value</i>	False
<i>Syntax</i>	Dim <i>Status</i> As Boolean <i>Status</i> = <i>app</i> .SCPI.SENSe(<i>Ch</i>).CORRection.EXTension.PORT(<i>Pt</i>).INCLude(<i>Ls</i>).STATe <i>app</i> .SCPI.SENSe(<i>Ch</i>).CORRection.EXTension.PORT(<i>Pt</i>).INCLude(<i>Ls</i>).STATe = True
<i>Equivalent Softkeys</i>	Calibration > Port Extensions > Loss > Loss1 Loss2

SCPI.SENSe(*Ch*).CORRection.EXTension.PORT(*Pt*).LDC

<i>Object Type</i>	Property (read/write)
<i>Data Type</i>	Double
<i>Target</i>	Port <i>Pt</i> of channel <i>Ch</i> , <i>Ch</i> : channel number 1–16 (see Table 1 on page 25) <i>Pt</i> : port number 1–2 (see Table 2 on page 25)
<i>Description</i>	The loss value at DC for the loss calculation of the port extension function.
<i>Range</i>	from –200 to 200
<i>Out of Range</i>	Sets the value of the limit, which is closer to the specified value.
<i>Preset Value</i>	0
<i>Unit</i>	dB (decibel)
<i>Syntax</i>	<p>Dim Value As Double</p> <p><i>Value</i> = <i>app.SCPI.SENSe(Ch).CORRection.EXTension.PORT(Pt).LDC</i></p> <p><i>app.SCPI.SENSe(Ch).CORRection.EXTension.PORT(Pt).LDC</i> = 10</p>
<i>Equivalent Softkeys</i>	Calibration > Port Extensions > Loss > Loss at DC

SCPI.SENSe(*Ch*).CORRection.EXTension.PORT(*Pt*).LOSS(*Ls*)

<i>Object Type</i>	Property (read/write)
<i>Data Type</i>	Double
<i>Target</i>	Port <i>Pt</i> of channel <i>Ch</i> , <i>Ch</i> : channel number 1–16 (see Table 1 on page 25) <i>Pt</i> : port number 1–2 (see Table 2 on page 25)
<i>Description</i>	The loss value at the point number <i>Ls</i> to calculate the loss for the port extension function, <i>Ls</i> : point number 1–2 (see Table 8 on page 265)
<i>Range</i>	from –200 to 200
<i>Out of Range</i>	Sets the value of the limit, which is closer to the specified value.
<i>Preset Value</i>	0
<i>Unit</i>	dB (decibel)
<i>Syntax</i>	Dim <i>Value</i> As Double <i>Value</i> = <i>app</i> .SCPI.SENSe(<i>Ch</i>).CORRection.EXTension.PORT(<i>Pt</i>).LOSS(<i>Ls</i>) <i>app</i> .SCPI.SENSe(<i>Ch</i>).CORRection.EXTension.PORT(<i>Pt</i>).LOSS(<i>Ls</i>) = 10
<i>Equivalent Softkeys</i>	Calibration > Port Extensions > Loss > Loss1 Loss2

SCPI.SENSe(*Ch*).CORRection.EXTension.PORT(*Pt*).TIME

<i>Object Type</i>	Property (read/write)
<i>Data Type</i>	Double
<i>Target</i>	Port <i>Pt</i> of channel <i>Ch</i> , <i>Ch</i> : channel number 1–16 (see Table 1 on page 25) <i>Pt</i> : port number 1–2 (see Table 2 on page 25)
<i>Description</i>	The electrical delay value for the port extension function.
<i>Range</i>	from –10 to 10
<i>Out of Range</i>	Sets the value of the limit, which is closer to the specified value.
<i>Preset Value</i>	0
<i>Unit</i>	s (second)
<i>Syntax</i>	<p>Dim <i>Value</i> As Double</p> <p><i>Value</i> = <i>app</i>.SCPI.SENSe(<i>Ch</i>).CORRection.EXTension.PORT(<i>Pt</i>).TIME</p> <p><i>app</i>.SCPI.SENSe(<i>Ch</i>).CORRection.EXTension.PORT(<i>Pt</i>).TIME = 10</p>
<i>Equivalent Softkeys</i>	Calibration > Port Extensions > Extension Port1 Extension Port2

SCPI.SENSe(*Ch*).CORRection.EXTension.STATE

<i>Object Type</i>	Property (read/write)
<i>Data Type</i>	Boolean
<i>Target</i>	Channel <i>Ch</i> , <i>Ch</i> : channel number 1–16 (see Table 1 on page 25)
<i>Description</i>	The ON/OFF state of the port extension function.
<i>Allowable Values</i>	True: Port extension function ON False: Port extension function OFF
<i>Preset Value</i>	False
<i>Syntax</i>	<p>Dim <i>Status</i> As Boolean</p> <p><i>Status</i> = <i>app</i>.SCPI.SENSe(<i>Ch</i>).CORRection.EXTension.STATE</p> <p><i>app</i>.SCPI.SENSe(<i>Ch</i>).CORRection.EXTension.STATE = True</p>
<i>Equivalent Softkeys</i>	Calibration > Port Extensions > Extension

SCPI.SENSe.CORRection.IMPedance.INPut.MAGNitude

<i>Object Type</i>	Property (read/write)
<i>Data Type</i>	Double
<i>Target</i>	Analyzer
<i>Description</i>	The system impedance Z0.
<i>Range</i>	from 0.001 to 1000
<i>Out of Range</i>	Sets the value of the limit, which is closer to the specified value.
<i>Preset Value</i>	50
<i>Unit</i>	Ω (Ohm)
<i>Syntax</i>	<p>Dim Value As Double</p> <p><i>Value</i> = <i>app</i>.SCPI.SENSe.CORRection.IMPedance.INPut.MAGNitude</p> <p><i>app</i>.SCPI.SENSe.CORRection.IMPedance.INPut.MAGNitude = 75</p>
<i>Warning</i>	Object <i>SENSe</i> has an index. In Visual Basic the index is 1 by default when it is omitted. The index can not be omitted in other programming languages.
<i>Equivalent Softkeys</i>	Calibration > System Z0

SCPI.SENSE(*Ch*).CORRection.INFormation(*rPt*, *sPt*)

<i>Object Type</i>	Property (read only)
<i>Data Type</i>	String
<i>Target</i>	Channel <i>Ch</i> , <i>Ch</i> : channel number 1–16 (see Table 1 on page 25)
<i>Description</i>	The calibration information string.
<i>Syntax</i>	<i>Dim CallInfo As String</i> <i>CallInfo = app.SCPI.SENSE(Ch).CORRection.INFormation(rPt, sPt)</i>
<i>Equivalent Softkeys</i>	none

SCPI.SENSe(*Ch*).CORRection.OFFSet.CLEar

<i>Object Type</i>	Method
<i>Target</i>	Channel <i>Ch</i> , <i>Ch</i> : channel number 1–16 (see Table 1 on page 25)
<i>Description</i>	Clears the scalar mixer calibration coefficient table.
<i>Syntax</i>	<i>app</i> .SCPI.SENSe(<i>Ch</i>).CORRection.OFFSet.CLEar
<i>Equivalent Softkeys</i>	None

SCPI.SENSE(*Ch*).CORRection.OFFSet.COLLect.ACQuire.LOAD

<i>Object Type</i>	Property (write only)
<i>Data Type</i>	Variant (array of long)
<i>Target</i>	Channel <i>Ch</i> , <i>Ch</i> : channel number 1–16 (see Table 1 on page 25)
<i>Description</i>	Measures the calibration data of the load standard of the specified port when the frequency offset feature is on (scalar mixer cal). The array contains 2 elements: Port(0) Measurement port number; Port(1) Frequency port number.
<i>Range</i>	Port number is 1 or 2. The array elements can contain the same port numbers.
<i>Out of Range</i>	If an incorrect port number is specified, an error occurs (error code: 222).
<i>Syntax</i>	<i>app</i> .SCPI.SENSE(<i>Ch</i>).CORRection.OFFSet.COLLect.ACQuire.LOAD = Array(1, 2)
<i>Notes</i>	The property writing starts the measurement for the channel independently of the trigger initiation and trigger source settings. The function of the property writing waits for the completion of the measurement.
<i>Equivalent Softkeys</i>	Calibration > Mixer/Converter Calibration > Scalar Mixer Calibration > Reflection Port n > Port n @Freq m Load

SCPI.SENSE(*Ch*).CORRection.OFFSet.COLLect.ACQuire.OPEN

<i>Object Type</i>	Property (write only)				
<i>Data Type</i>	Variant (array of long)				
<i>Target</i>	Channel <i>Ch</i> , <i>Ch</i> : channel number 1–16 (see Table 1 on page 25)				
<i>Description</i>	Measures the calibration data of the open standard of the specified port when the frequency offset feature is on (scalar mixer cal). The array contains 2 elements: <table style="margin-left: 40px;"> <tr> <td><i>Port(0)</i></td> <td>Measurement port number;</td> </tr> <tr> <td><i>Port(1)</i></td> <td>Frequency port number.</td> </tr> </table>	<i>Port(0)</i>	Measurement port number;	<i>Port(1)</i>	Frequency port number.
<i>Port(0)</i>	Measurement port number;				
<i>Port(1)</i>	Frequency port number.				
<i>Range</i>	Port number is 1 or 2. The array elements can contain the same port numbers.				
<i>Out of Range</i>	If an incorrect port number is specified, an error occurs (error code: 222).				
<i>Syntax</i>	<i>app</i> .SCPI.SENSE(<i>Ch</i>).CORRection.OFFSet.COLLect.ACQuire.OPEN = Array(1, 2)				
<i>Notes</i>	The property writing starts the measurement for the channel independently of the trigger initiation and trigger source settings. The function of the property writing waits for the completion of the measurement.				
<i>Equivalent Softkeys</i>	Calibration > Mixer/Converter Calibration > Scalar Mixer Calibration > Reflection Port n > Port n @Freq m Open				

SCPI.SENSE(*Ch*).CORRection.OFFSet.COLLect.ACQuire.SHORt

<i>Object Type</i>	Property (write only)				
<i>Data Type</i>	Variant (array of long)				
<i>Target</i>	Channel <i>Ch</i> , <i>Ch</i> : channel number 1–16 (see Table 1 on page 25)				
<i>Description</i>	Measures the calibration data of the short standard of the specified port when the frequency offset feature is on (scalar mixer cal). The array contains 2 elements: <table style="margin-left: 40px;"> <tr> <td><i>Port(0)</i></td> <td>Measurement port number;</td> </tr> <tr> <td><i>Port(1)</i></td> <td>Frequency port number.</td> </tr> </table>	<i>Port(0)</i>	Measurement port number;	<i>Port(1)</i>	Frequency port number.
<i>Port(0)</i>	Measurement port number;				
<i>Port(1)</i>	Frequency port number.				
<i>Range</i>	Port number is 1 or 2. The array elements can contain the same port numbers.				
<i>Out of Range</i>	If an incorrect port number is specified, an error occurs (error code: 222).				
<i>Syntax</i>	<i>app</i> .SCPI.SENSE(<i>Ch</i>).CORRection.OFFSet.COLLect.ACQuire.SHORt = Array(1, 2)				
<i>Notes</i>	The property writing starts the measurement for the channel independently of the trigger initiation and trigger source settings. The function of the property writing waits for the completion of the measurement.				
<i>Equivalent Softkeys</i>	Calibration > Mixer/Converter Calibration > Scalar Mixer Calibration > Reflection Port n > Port n @Freq m Short				

SCPI.SENSE(*Ch*).CORRection.OFFSet.COLLect.ACQuire.THRU

<i>Object Type</i>	Property (write only)				
<i>Data Type</i>	Variant (array of long)				
<i>Target</i>	Channel <i>Ch</i> , <i>Ch</i> : channel number 1–16 (see Table 1 on page 25)				
<i>Description</i>	Measures the calibration data of the thru standard of the specified port when the frequency offset feature is on (scalar mixer cal). The array contains 2 elements: <table style="margin-left: 40px;"> <tr> <td><i>Port(0)</i></td> <td>Response port number;</td> </tr> <tr> <td><i>Port(1)</i></td> <td>Stimulus port number.</td> </tr> </table>	<i>Port(0)</i>	Response port number;	<i>Port(1)</i>	Stimulus port number.
<i>Port(0)</i>	Response port number;				
<i>Port(1)</i>	Stimulus port number.				
<i>Range</i>	Port number is 1 or 2. The array elements can not contain the same port numbers.				
<i>Out of Range</i>	If an incorrect port number is specified, an error occurs (error code: 222). If the same port numbers are specified, an error occurs (error code: 220).				
<i>Syntax</i>	<code>app.SCPI.SENSE(<i>Ch</i>).CORRection.OFFSet.COLLect.ACQuire.THRU = Array(1, 2)</code>				
<i>Notes</i>	The property writing starts the measurement for the channel independently of the trigger initiation and trigger source settings. The function of the property writing waits for the completion of the measurement.				
<i>Equivalent Softkeys</i>	Calibration > Mixer/Converter Calibration > Scalar Mixer Calibration > Reflection Port n > Port n @Freq m Thru				

SCPI.SENSe(*Ch*).CORRection.OFFSet.COLLect.ACQuire.PMETer

<i>Object Type</i>	Property (write only)
<i>Data Type</i>	Variant (array of long)
<i>Target</i>	Channel <i>Ch</i> , <i>Ch</i> : channel number 1–16 (see Table 1 on page 25)
<i>Description</i>	Measures the scalar-mixer calibration data using the power meter when the frequency offset feature is ON. The array contains 3 elements: <i>Port(0)</i> Measurement port number; <i>Port(1)</i> Frequency port number. <i>Port(2)</i> Reserved.
<i>Range</i>	Port number is 1 or 2. The array elements can contain the same port numbers.
<i>Out of Range</i>	If an incorrect port number is specified, an error occurs (error code: 222).
<i>Syntax</i>	<i>app.SCPI.SENSe(<i>Ch</i>).CORRection.OFFSet.COLLect.ACQuire.PMETer</i> = Array(1, 2, 0)
<i>Notes</i>	The property writing starts the measurement for the channel independently of the trigger initiation and trigger source settings. The function of the property writing waits for the completion of the measurement.
<i>Equivalent Softkeys</i>	Calibration > Mixer/Converter Calibration > Scalar Mixer Calibration > Power > Port n @Freq m

SCPI.SENSe(*Ch*).CORRection.OFFSet.COLLect.CLEar

<i>Object Type</i>	Method
<i>Target</i>	Channel <i>Ch</i> , <i>Ch</i> : channel number 1–16 (see Table 1 on page 25)
<i>Description</i>	Clears the calibration measurement data when the frequency offset feature is ON (scalar mixer calibration).
<i>Syntax</i>	<i>app</i> .SCPI.SENSe(<i>Ch</i>).CORRection.OFFSet.COLLect.CLEar
<i>Equivalent Softkeys</i>	Calibration > Mixer/Converter Calibration > Scalar Mixer Calibration > Cancel

SCPI.SENSE(*Ch*).CORRection.OFFSet.COLLect.METHOD.SMIX2

<i>Object Type</i>	Property (write only)
<i>Data Type</i>	Variant (array of long)
<i>Target</i>	Channel <i>Ch</i> , <i>Ch</i> : channel number 1–16 (see Table 1 on page 25)
<i>Description</i>	Selects the ports and sets the <i>scalar mixer calibration</i> type when the frequency offset feature is on for the calculation of the calibration coefficients on completion of the calibration executed by the SCPI.SENSE(<i>Ch</i>).CORRection.OFFSet.COLLect.SAVE method. The array contains 2 elements: <i>Port(0)</i> Port number 1; <i>Port(1)</i> Port number 2.
<i>Range</i>	Port number is 1 or 2. The array elements can not contain the same port numbers.
<i>Out of Range</i>	If an incorrect port number is specified, an error occurs (error code: 222). If the same port numbers are specified, an error occurs (error code: 220).
<i>Syntax</i>	<code>app.SCPI.SENSE(<i>Ch</i>).CORRection.OFFSet.COLLect.METHOD.SMIX2 = Array(2, 1)</code>
<i>Equivalent Softkeys</i>	Calibration > Mixer/Converter Calibration > Scalar Mixer Calibration

SCPI.SENSE(*Ch*).CORRection.OFFSet.COLLect.SAVE

<i>Object Type</i>	Method
<i>Target</i>	Channel <i>Ch</i> , <i>Ch</i> : channel number 1–16 (see Table 1 on page 25)
<i>Description</i>	<p>Calculates the calibration coefficient for the selected calibration type (scalar mixer calibration only) from the calibration data measured with the frequency offset feature is ON.</p> <p>If this command is executed before all necessary calibration data for calculating the calibration coefficient is measured, an error occurs when executed.</p>
<i>Syntax</i>	<code>app.SCPI.SENSE(<i>Ch</i>).CORRection.OFFSet.COLLect.SAVE</code>
<i>Related Commands</i>	SCPI.SENSE(<i>Ch</i>).CORRection.OFFSet.COLLect.METHOD.SMIX2 SCPI.SENSE(<i>Ch</i>).CORRection.OFFSet.COLLect.ACQuire.LOAD SCPI.SENSE(<i>Ch</i>).CORRection.OFFSet.COLLect.ACQuire.OPEN SCPI.SENSE(<i>Ch</i>).CORRection.OFFSet.COLLect.ACQuire.SHORT SCPI.SENSE(<i>Ch</i>).CORRection.OFFSet.COLLect.ACQuire.THRU SCPI.SENSE(<i>Ch</i>).CORRection.OFFSet.COLLect.ACQuire.PMETer
<i>Equivalent Softkeys</i>	Calibration > Mixer/Converter Calibration > Scalar Mixer Calibration > Apply

SCPI.SENSe(*Ch*).CORRection.RECeiver(*Pt*).COLLect.ACQuire

<i>Object Type</i>	Property (write only)
<i>Data Type</i>	Long
<i>Target</i>	Port <i>Pt</i> of channel <i>Ch</i> , <i>Ch</i> : channel number 1–16 (see Table 1 on page 25) <i>Pt</i> : port number 1–2 (see Table 2 on page 25)
<i>Description</i>	The property writing executes the receiver calibration for the specified port <i>Pt</i> , using the specified source port <i>Src</i> to the right of the equal sign.
<i>Range</i>	The number of the source port from 1 to 2.
<i>Out of Range</i>	Error occurs. Error code: 222.
<i>Syntax</i>	<i>app.SCPI.SENSe(Ch).CORRection.RECeiver(Pt).COLLect.ACQuire = Src</i>
<i>Notes</i>	The property writing triggers the channel sweep independently of its current mode (continuous, single, hold) and trigger source setting. The function of the property writing waits for the completion of the calibration.
<i>Equivalent Softkeys</i>	Calibration > Receiver Calibration > Take Cal Sweep

SCPI.SENSE(*Ch*).CORRection.RECeiver(*Pt*).STATe

<i>Object Type</i>	Property (read/write)
<i>Data Type</i>	Boolean
<i>Target</i>	Port <i>Pt</i> of channel <i>Ch</i> , <i>Ch</i> : channel number 1–16 (see Table 1 on page 25) <i>Pt</i> : port number 1–2 (see Table 2 on page 25)
<i>Description</i>	The ON/OFF state of the receiver correction for the port <i>Pt</i> .
<i>Allowable Values</i>	True: Receiver correction ON False: Receiver correction OFF
<i>Preset Value</i>	False
<i>Syntax</i>	Dim <i>Status</i> As Boolean <i>Status</i> = <i>app</i> .SCPI.SENSE(<i>Ch</i>).CORRection.RECeiver(<i>Pt</i>).STATe <i>app</i> .SCPI.SENSE(<i>Ch</i>).CORRection.RECeiver(<i>Pt</i>).STATe = True
<i>Equivalent Softkeys</i>	Calibration > Receiver Calibration > Correction

SCPI.SENSe(*Ch*).CORRection.STATE

<i>Object Type</i>	Property (read/write)
<i>Data Type</i>	Boolean
<i>Target</i>	Channel <i>Ch</i> , <i>Ch</i> : channel number 1–16 (see Table 1 on page 25)
<i>Description</i>	The ON/OFF state of the error correction.
<i>Allowable Values</i>	True: Error correction ON False: Error correction OFF
<i>Preset Value</i>	False
<i>Syntax</i>	<p>Dim <i>Status</i> As Boolean</p> <p><i>Status</i> = <i>app</i>.SCPI.SENSe(<i>Ch</i>).CORRection.STATE</p> <p><i>app</i>.SCPI.SENSe(<i>Ch</i>).CORRection.STATE = True</p>
<i>Equivalent Softkeys</i>	Calibration > Correction

SCPI.SENSe(*Ch*).CORRection.TYPE(*Tr*)

<i>Object Type</i>	Property (read only)
<i>Data Type</i>	Variant (Variant array)
<i>Target</i>	Trace <i>Tr</i> of channel <i>Ch</i> , <i>Ch</i> : channel number 1–16 (see Table 1 on page 25) <i>Tr</i> : trace number 1–16 (see Table 3 on page 33)
<i>Description</i>	The information about the applied calibration type and the port numbers for the specified trace. The array contains 3 elements: <i>Data(0)</i> calibration type (see below); <i>Data(1)</i> the number of the receiver port to be calibrated; <i>Data(2)</i> the number of the source port to be calibrated.
<i>Range</i>	Calibration type in the element <i>Data(0)</i> : "RESPO" : Response (Open) "RESPS" : Response (Short) "RESPT" : Response (Thru) "SOLT1" : Full 1–port calibration "SOLT2" : Full 2–port calibration "1PATH" : One path 2–port calibration "NONE" : Not defined
<i>Syntax</i>	Dim <i>CallInfo</i> As Variant <i>CallInfo</i> = <i>app</i> .SCPI.SENSe(<i>Ch</i>).CORRection.TYPE(<i>Tr</i>)
<i>Equivalent Softkeys</i>	None

SCPI.SENSe(*Ch*).CORRection.TRANSform.TIME.FREQuency

<i>Object Type</i>	Property (read/write)
<i>Data Type</i>	Double
<i>Target</i>	Channel <i>Ch</i> , <i>Ch</i> : channel number 1–16 (see Table 1 on page 25)
<i>Description</i>	The frequency at which the cable loss specified for the cable correction function, when the time domain transformation function is turned ON.
<i>Preset Value</i>	1 GHz.
<i>Unit</i>	Hz (Hertz)
<i>Syntax</i>	Dim <i>Value</i> As Double <i>Value</i> = <i>app</i> .SCPI.SENSe(<i>Ch</i>).CORRection.TRANSform.TIME.FREQuency <i>app</i> .SCPI.SENSe(<i>Ch</i>).CORRection.TRANSform.TIME.FREQuency = 1E9
<i>Equivalent Softkeys</i>	Analysis > Time Domain > Cable Correction > Frequency

SCPI.SENSE(*Ch*). CORRection.TRANSform.TIME.LOSS

<i>Object Type</i>	Property (read/write)
<i>Data Type</i>	Double
<i>Target</i>	Channel <i>Ch</i> , <i>Ch</i> : channel number 1–16 (see Table 1 on page 25)
<i>Description</i>	The cable loss value for the cable correction function, when the time domain transformation function is turned ON.
<i>Preset Value</i>	0
<i>Unit</i>	dB/m (decibell / meter)
<i>Syntax</i>	Dim <i>Value</i> As Double <i>Value</i> = <i>app.SCPI.SENSE(Ch).CORRection.TRANSform.TIME.LOSS</i> <i>app.SCPI.SENSE(Ch).CORRection.TRANSform.TIME.LOSS</i> = 1.4
<i>Equivalent Softkeys</i>	Analysis > Time Domain > Cable Correction > Cable Loss

SCPI.SENSE(*Ch*). CORRection.TRANSform.TIME.RVELocity

<i>Object Type</i>	Property (read/write)
<i>Data Type</i>	Double
<i>Target</i>	Channel <i>Ch</i> , <i>Ch</i> : channel number 1–16 (see Table 1 on page 25)
<i>Description</i>	The cable relative wave speed velocity for the cable correction function, when the time domain transformation function is turned ON.
<i>Preset Value</i>	1
<i>Syntax</i>	Dim <i>Value</i> As Double <i>Value</i> = <i>app.SCPI.SENSE(Ch).CORRection.TRANSform.TIME.RVELocity</i> <i>app.SCPI.SENSE(Ch).CORRection.TRANSform.TIME.RVELocity</i> = 0.66
<i>Equivalent Softkeys</i>	Analysis > Time Domain > Cable Correction > Velocity Factor

SCPI.SENSe(*Ch*).CORRection.TRANSform.TIME.STATE

<i>Object Type</i>	Property (read/write)
<i>Data Type</i>	Boolean
<i>Target</i>	Channel <i>Ch</i> , <i>Ch</i> : channel number 1–16 (see Table 1 on page 25)
<i>Description</i>	The ON/OFF state of the cable correction when the time domain transformation function is turned ON.
<i>Allowable Values</i>	True: cable correction ON False: cable correction OFF
<i>Preset Value</i>	False
<i>Syntax</i>	Dim <i>Status</i> As Boolean <i>Status</i> = <i>app</i> .SCPI.SENSe(<i>Ch</i>).CORRection.TRANSform.TIME.STATE <i>app</i> .SCPI.SENSe(<i>Ch</i>).CORRection.TRANSform.TIME.STATE = True
<i>Equivalent Softkeys</i>	Analysis > Time Domain > Cable Correction > Cable Correction

SCPI.SENSE(*Ch*).DATA.CORRdata(*Param*)

<i>Object Type</i>	Property (read only)
<i>Data Type</i>	Variant (array of double)
<i>Target</i>	Channel <i>Ch</i> , <i>Ch</i> : channel number 1–16 (see Table 1 on page 25)
<i>Parameter</i>	<i>Param</i> of String type selects S-parameter: "S11", "S12", "S21", "S22"
<i>Description</i>	<p>The corrected S-parameters array. The corrected measurements are complex numbers. Also, see section “Measurement Data Arrays” on page 20.</p> <p>The array size is 2N, where N is the number of measurement points.</p> <p>For the n-th point, where n from 1 to N:</p> <ul style="list-style-type: none"> <i>Data</i>(2n-2) the real part of corrected measurement; <i>Data</i>(2n-1) the imaginary part of corrected measurement. <p>The index of the array starts from 0.</p>
<i>Syntax</i>	<pre>Dim <i>Data</i> As Variant <i>Data</i> = app.SCPI.SENSE(<i>ch</i>).DATA.CORRdata("S11")</pre>
<i>Equivalent Softkeys</i>	None

SCPI.SENSE(*Ch*).DATA.RAWData(*Param*)

<i>Object Type</i>	Property (read only)
<i>Data Type</i>	Variant (array of double)
<i>Target</i>	Channel <i>Ch</i> , <i>Ch</i> : channel number 1–16 (see Table 1 on page 25)
<i>Parameter</i>	<i>Param</i> of String type selects S-parameter: "S11", "S12", "S21", "S22"
<i>Description</i>	<p>The raw S-parameters array (without correction). The raw measurements are complex numbers. Also, see section “Measurement Data Arrays” on page 20.</p> <p>The array size is 2N, where N is the number of measurement points.</p> <p>For the n-th point, where n from 1 to N:</p> <ul style="list-style-type: none"> <i>Data</i>(2n-2) the real part of corrected measurement; <i>Data</i>(2n-1) the imaginary part of corrected measurement. <p>The index of the array starts from 0.</p>
<i>Syntax</i>	<pre>Dim <i>Data</i> As Variant <i>Data</i> = app.SCPI.SENSE(<i>ch</i>).DATA.RAWData("S11")</pre>
<i>Equivalent Softkeys</i>	None

SCPI.SENSe(*Ch*).FREQuency.CENTer

<i>Object Type</i>	Property (read/write)
<i>Data Type</i>	Double
<i>Target</i>	Channel <i>Ch</i> , <i>Ch</i> : channel number 1–16 (see Table 1 on page 25)
<i>Description</i>	The stimulus center value of the sweep range for linear or logarithmic sweep types.
<i>Range</i>	From the minimum value to the maximum value of the analyzer frequency range.
<i>Out of Range</i>	Sets the value of the limit, which is closer to the specified value.
<i>Preset Value</i>	The center value of the analyzer frequency range.
<i>Unit</i>	Hz (Hertz)
<i>Syntax</i>	<p>Dim <i>Value</i> As Double</p> <p><i>Value</i> = <i>app</i>.SCPI.SENSe(<i>Ch</i>).FREQuency.CENTer</p> <p><i>app</i>.SCPI.SENSe(<i>Ch</i>).FREQuency.CENTer = 1E9</p>
<i>Equivalent Softkeys</i>	Stimulus > Center

SCPI.SENSE(*Ch*).FREQuency.CW

<i>Object Type</i>	Property (read/write)
<i>Data Type</i>	Double
<i>Target</i>	Channel <i>Ch</i> , <i>Ch</i> : channel number 1–16 (see Table 1 on page 25)
<i>Description</i>	The value of the fixed frequency for the power sweep.
<i>Range</i>	From the minimum value to the maximum value of the analyzer frequency range.
<i>Out of Range</i>	Sets the value of the limit, which is closer to the specified value.
<i>Preset Value</i>	The minimum value of the analyzer frequency range.
<i>Unit</i>	Hz (Hertz)
<i>Syntax</i>	Dim <i>Value</i> As Double <i>Value</i> = <i>app.SCPI.SENSE(Ch).FREQuency.CW</i> <i>app.SCPI.SENSE(Ch).FREQuency.CW</i> = 1E9
<i>Equivalent Softkeys</i>	Stimulus > Power > CW Freq

SCPI.SENSe(*Ch*).FREQuency.DATA

<i>Object Type</i>	Property (read only)
<i>Data Type</i>	Variant (array of double)
<i>Target</i>	Channel <i>Ch</i> , <i>Ch</i> : channel number 1–16 (see Table 1 on page 25)
<i>Description</i>	<p>The array of the measurement points frequency for linear, logarithmic or segment sweep type.</p> <p>The array size is N, where N is the number of measurement points.</p> <p>For the n-th point, where n from 1 to N:</p> $\text{Data}(n-1) \quad \text{the frequency value at the n-th measurement point.}$
<i>Syntax</i>	<p>Dim <i>Data</i> As Variant</p> <p><i>Data</i> = <i>app</i>. SCPI.SENSe(<i>Ch</i>).FREQuency.DATA</p>
<i>Equivalent Softkeys</i>	None

SCPI.SENSE(*Ch*).FREQuency.SPAN

<i>Object Type</i>	Property (read/write)
<i>Data Type</i>	Double
<i>Target</i>	Channel <i>Ch</i> , <i>Ch</i> : channel number 1–16 (see Table 1 on page 25)
<i>Description</i>	The stimulus span value of the sweep range for linear or logarithmic sweep types.
<i>Range</i>	From 0 to the maximum span of the analyzer frequency range.
<i>Out of Range</i>	Sets the value of the limit, which is closer to the specified value.
<i>Preset Value</i>	The maximum span of the analyzer frequency range.
<i>Unit</i>	Hz (Hertz)
<i>Syntax</i>	Dim <i>Value</i> As Double <i>Value</i> = <i>app</i> .SCPI.SENSE(<i>Ch</i>).FREQuency.SPAN <i>app</i> .SCPI.SENSE(<i>Ch</i>).FREQuency.SPAN = 2E9
<i>Equivalent Softkeys</i>	Stimulus > Span

SCPI.SENSe(*Ch*).FREQuency.START

<i>Object Type</i>	Property (read/write)
<i>Data Type</i>	Double
<i>Target</i>	Channel <i>Ch</i> , <i>Ch</i> : channel number 1–16 (see Table 1 on page 25)
<i>Description</i>	The stimulus start value of the sweep range for linear or logarithmic sweep types.
<i>Range</i>	From the minimum value to the maximum value of the analyzer frequency range.
<i>Out of Range</i>	Sets the value of the limit, which is closer to the specified value.
<i>Preset Value</i>	The minimum value of the analyzer frequency range.
<i>Unit</i>	Hz (Hertz)
<i>Syntax</i>	<p>Dim <i>Value</i> As Double <i>Value</i> = <i>app</i>.SCPI.SENSe(<i>Ch</i>).FREQuency.START <i>app</i>.SCPI.SENSe(<i>Ch</i>).FREQuency.START = 1E6</p>
<i>Equivalent Softkeys</i>	Stimulus > Start

SCPI.SENSe(*Ch*).FREQuency.STOP

<i>Object Type</i>	Property (read/write)
<i>Data Type</i>	Double
<i>Target</i>	Channel <i>Ch</i> , <i>Ch</i> : channel number 1–16 (see Table 1 on page 25)
<i>Description</i>	The stimulus stop value of the sweep range for linear or logarithmic sweep types.
<i>Range</i>	From the minimum value to the maximum value of the analyzer frequency range.
<i>Out of Range</i>	Sets the value of the limit, which is closer to the specified value.
<i>Preset Value</i>	The maximum value of the analyzer frequency range
<i>Unit</i>	Hz (Hertz)
<i>Syntax</i>	<p>Dim <i>Value</i> As Double <i>Value</i> = <i>app</i>.SCPI.SENSe(<i>Ch</i>).FREQuency.STOP <i>app</i>.SCPI.SENSe(<i>Ch</i>).FREQuency.STOP = 1E6</p>
<i>Equivalent Softkeys</i>	Stimulus > Stop

SCPI.SENSe(*Ch*).OFFSet.ADJust.CONTinuous.PERiod

<i>Object Type</i>	Property (read/write)
<i>Data Type</i>	Long
<i>Target</i>	Channel <i>Ch</i> , <i>Ch</i> : channel number 1–16 (see Table 1 on page 25)
<i>Description</i>	Selects the period for continuous offset adjust feature: 1 – 3 sec; 2 – 10 sec; 3 – 30 sec.
<i>Range</i>	from 1 to 3
<i>Out of Range</i>	Sets the value of the limit, which is closer to the specified value.
<i>Preset Value</i>	2
<i>Unit</i>	n/a
<i>Syntax</i>	<p>Dim <i>Value</i> As Long</p> <p><i>Value</i> = <i>app</i>.SCPI.SENSe(<i>Ch</i>).OFFSet.ADJust.PERiod</p> <p><i>app</i>.SCPI.SENSe(<i>Ch</i>).OFFSet.ADJust.PERiod = 1</p>
<i>Equivalent Softkeys</i>	Stimulus > Frequency Offset > Offset Adjust > Adjust Period

SCPI.SENSe(*Ch*).OFFSet.ADJust.CONTinuous.STATE

<i>Object Type</i>	Property (read/write)
<i>Data Type</i>	Boolean
<i>Target</i>	Channel <i>Ch</i> , <i>Ch</i> : channel number 1–16 (see Table 1 on page 25)
<i>Description</i>	The ON/OFF state of the continuous frequency offset adjust feature.
<i>Allowable Values</i>	True: continuous offset adjust feature is ON False: continuous offset adjust feature is OFF
<i>Preset Value</i>	False
<i>Syntax</i>	<p>Dim <i>Status</i> As Boolean</p> <p><i>Status</i> = <i>app</i>.SCPI.SENSe(<i>Ch</i>).OFFSet.ADJust.CONTinuous.STATE</p> <p><i>app</i>.SCPI.SENSe(<i>Ch</i>).OFFSet.ADJust.CONTinuous.STATE = True</p>
<i>Equivalent Softkeys</i>	Stimulus > Frequency Offset > Offset Adjust > Continuous Adjust

SCPI.SENSe(*Ch*).OFFSet.ADJust.EXECute

<i>Object Type</i>	Method
<i>Target</i>	Channel <i>Ch</i> , <i>Ch</i> : channel number 1–16 (see Table 1 on page 25)
<i>Description</i>	Executes the offset adjust when the frequency offset feature is ON.
<i>Syntax</i>	<i>app.SCPI.SENSe(Ch).OFFset.ADJust.EXECute</i>
<i>Equivalent Softkeys</i>	Stimulus > Frequency Offset > Offset Adjust > Adjust Immediate

SCPI.SENSe(*Ch*).OFFSet.ADJust.PORT

<i>Object Type</i>	Property (read/write)
<i>Data Type</i>	Long
<i>Target</i>	Channel <i>Ch</i> , <i>Ch</i> : channel number 1–16 (see Table 1 on page 25)
<i>Description</i>	Selects the port number for frequency offset adjust feature.
<i>Range</i>	from 1 to 2
<i>Out of Range</i>	Sets the value of the limit, which is closer to the specified value.
<i>Preset Value</i>	1
<i>Unit</i>	n/a
<i>Syntax</i>	Dim Value As Long <i>Value</i> = <i>app.SCPI.SENSe(Ch).OFFset.ADJust.PORT</i> <i>app.SCPI.SENSe(Ch).OFFset.ADJust.PORT</i> = 2
<i>Equivalent Softkeys</i>	Stimulus > Frequency Offset > Offset Adjust > Select Port

SCPI.SENSe(*Ch*).OFFSet.ADJust.STATE

<i>Object Type</i>	Property (read/write)
<i>Data Type</i>	Boolean
<i>Target</i>	Channel <i>Ch</i> , <i>Ch</i> : channel number 1–16 (see Table 1 on page 25)
<i>Description</i>	The ON/OFF state of the frequency offset adjust feature.
<i>Allowable Values</i>	True: offset adjust feature is ON False: offset adjust feature is OFF
<i>Preset Value</i>	False
<i>Syntax</i>	<p>Dim <i>Status</i> As Boolean</p> <p><i>Status</i> = <i>app.SCPI.SENSe(Ch).OFFSet.ADJust.STATE</i></p> <p><i>app.SCPI.SENSe(Ch).OFFSet.ADJust.STATE</i> = True</p>
<i>Equivalent Softkeys</i>	Stimulus > Frequency Offset > Offset Adjust > Offset Adjust

SCPI.SENSe(*Ch*).OFFSet.ADJust.Value

<i>Object Type</i>	Property (read/write)
<i>Data Type</i>	Double
<i>Target</i>	Channel <i>Ch</i> , <i>Ch</i> : channel number 1–16 (see Table 1 on page 25)
<i>Description</i>	The frequency offset adjust value of the offset adjust feature.
<i>Range</i>	from –5E5 to 5E5
<i>Out of Range</i>	Sets the value of the limit, which is closer to the specified value.
<i>Preset Value</i>	0
<i>Unit</i>	Hz
<i>Syntax</i>	Dim <i>Value</i> As Double <i>Value</i> = <i>app</i> .SCPI.SENSe(<i>Ch</i>).OFFSet.ADJust.VALUE <i>app</i> .SCPI.SENSe(<i>Ch</i>).OFFSet.ADJust.VALUE = 1000
<i>Equivalent Softkeys</i>	Stimulus > Frequency Offset > Offset Adjust > Adjust Value

SCPI.SENSe(*Ch*).OFFSet.PORT(*Pt*).FREQuency.DATA

<i>Object Type</i>	Property (read only)
<i>Data Type</i>	Variant (array of double)
<i>Target</i>	Port <i>Pt</i> of channel <i>Ch</i> , <i>Ch</i> : channel number 1–16 (see Table 1 on page 25) <i>Pt</i> : port number 1–2 (see Table 2 on page 25)
<i>Description</i>	Reads the frequency data for the selected port <i>Pt</i> when the frequency offset feature is ON and offset type is "PORT", for the selected channel <i>Ch</i> . The array size is N, where N is the number of measurement points. For the n-th point, where n from 1 to N: <i>Data</i> (<i>n</i> –1) the frequency value at the n-th measurement point.
<i>Syntax</i>	Dim <i>Data</i> As Variant <i>Data</i> = <i>app</i> .SCPI.SENSe(<i>Ch</i>).OFFSet.PORT(<i>Pt</i>).FREQuency.DATA
<i>Related Commands</i>	SCPI.SENSe(<i>Ch</i>).OFFSet.STATE SCPI.SENSe(<i>Ch</i>).OFFset.TYPE
<i>Equivalent Softkeys</i>	None

SCPI.SENSe(*Ch*).OFFSet.PORT(*Pt*).FREQuency.DIVisor

<i>Object Type</i>	Property (read/write)
<i>Data Type</i>	Double
<i>Target</i>	Port <i>Pt</i> of channel <i>Ch</i> , <i>Ch</i> : channel number 1–16 (see Table 1 on page 25) <i>Pt</i> : port number 1–2 (see Table 2 on page 25)
<i>Description</i>	The basic frequency range divisor when the frequency offset feature is ON and offset type is "PORT".
<i>Range</i>	from 1 to 1000
<i>Out of Range</i>	Sets the value of the limit, which is closer to the specified value.
<i>Preset Value</i>	1
<i>Unit</i>	n/a
<i>Syntax</i>	Dim <i>Value</i> As Double <i>Value</i> = <i>app</i> .SCPI.SENSe(<i>Ch</i>).OFFset.PORT(<i>Pt</i>).FREQuency.DIVisor <i>app</i> .SCPI.SENSe(<i>Ch</i>).OFFset.PORT(<i>Pt</i>).FREQuency.DIVisor = 2
<i>Related Commands</i>	SCPI.SENSe(<i>Ch</i>).OFFset.STATE SCPI.SENSe(<i>Ch</i>).OFFset.TYPE
<i>Equivalent Softkeys</i>	Stimulus > Frequency Offset > Port n > Divider

SCPI.SENSe(*Ch*).OFFSet.PORT(*Pt*).FREQuency.MULTIplier

<i>Object Type</i>	Property (read/write)
<i>Data Type</i>	Double
<i>Target</i>	Port <i>Pt</i> of channel <i>Ch</i> , <i>Ch</i> : channel number 1–16 (see Table 1 on page 25) <i>Pt</i> : port number 1–2 (see Table 2 on page 25)
<i>Description</i>	The basic range frequency multiplier when the frequency offset feature is ON and offset type is "PORT".
<i>Range</i>	from –1000 to 1000
<i>Out of Range</i>	Sets the value of the limit, which is closer to the specified value.
<i>Preset Value</i>	1
<i>Unit</i>	n/a
<i>Syntax</i>	Dim <i>Value</i> As Double <i>Value</i> = <i>app</i> .SCPI.SENSe(<i>Ch</i>).OFFset.PORT(<i>Pt</i>).FREQuency.MULTIplier <i>app</i> .SCPI.SENSe(<i>Ch</i>).OFFset.PORT(<i>Pt</i>).FREQuency.MULTIplier = 2
<i>Related Commands</i>	SCPI.SENSe(<i>Ch</i>).OFFset.STATE SCPI.SENSe(<i>Ch</i>).OFFset.TYPE
<i>Equivalent Softkeys</i>	Stimulus > Frequency Offset > Port n > Multiplier

SCPI.SENSe(*Ch*).OFFSet.PORT(*Pt*).FREQuency.OFFSet

<i>Object Type</i>	Property (read/write)
<i>Data Type</i>	Double
<i>Target</i>	Port <i>Pt</i> of channel <i>Ch</i> , <i>Ch</i> : channel number 1–16 (see Table 1 on page 25) <i>Pt</i> : port number 1–2 (see Table 2 on page 25)
<i>Description</i>	The basic frequency range offset when the frequency offset feature is ON and offset type is "PORT".
<i>Range</i>	from –1e–12 to 1e12
<i>Out of Range</i>	Sets the value of the limit, which is closer to the specified value.
<i>Preset Value</i>	0
<i>Unit</i>	Hz
<i>Syntax</i>	Dim <i>Value</i> As Double <i>Value</i> = <i>app</i> .SCPI.SENSe(<i>Ch</i>).OFFset.PORT(<i>Pt</i>).FREQuency.OFFSet <i>app</i> .SCPI.SENSe(<i>Ch</i>).OFFset.PORT(<i>Pt</i>).FREQuency.OFFSet = 1e9
<i>Related Commands</i>	SCPI.SENSe(<i>Ch</i>).OFFset.STATE SCPI.SENSe(<i>Ch</i>).OFFset.TYPE
<i>Equivalent Softkeys</i>	Stimulus > Frequency Offset > Port n > Offset

SCPI.SENSE(*Ch*).OFFSet.PORT(*Pt*).FREQuency.START

<i>Object Type</i>	Property (read/write)
<i>Data Type</i>	Double
<i>Target</i>	Port <i>Pt</i> of channel <i>Ch</i> , <i>Ch</i> : channel number 1–16 (see Table 1 on page 25) <i>Pt</i> : port number 1–2 (see Table 2 on page 25)
<i>Description</i>	The frequency sweep start when the frequency offset feature is ON and offset type is "PORT".
<i>Unit</i>	Hz
<i>Syntax</i>	Dim <i>Value</i> As Double <i>Value</i> = <i>app</i> .SCPI.SENSE(<i>Ch</i>).OFFset.PORT(<i>Pt</i>).FREQuency.START
<i>Related Commands</i>	SCPI.SENSE(<i>Ch</i>).OFFset.STATE SCPI.SENSE(<i>Ch</i>).OFFset.TYPE
<i>Equivalent Softkeys</i>	Stimulus > Frequency Offset > Port n > Start

SCPI.SENSE(*Ch*).OFFSET.PORT(*Pt*).FREQuency.STOP

<i>Object Type</i>	Property (read/write)
<i>Data Type</i>	Double
<i>Target</i>	Port <i>Pt</i> of channel <i>Ch</i> , <i>Ch</i> : channel number 1–16 (see Table 1 on page 25) <i>Pt</i> : port number 1–2 (see Table 2 on page 25)
<i>Description</i>	The frequency sweep stop when the frequency offset feature is ON and offset type is "PORT".
<i>Unit</i>	Hz
<i>Syntax</i>	Dim <i>Value</i> As Double <i>Value</i> = <i>app</i> .SCPI.SENSE(<i>Ch</i>).OFFSET.PORT(<i>Pt</i>).FREQuency.STOP
<i>Related Commands</i>	SCPI.SENSE(<i>Ch</i>).OFFSET.STATE SCPI.SENSE(<i>Ch</i>).OFFSET.TYPE
<i>Equivalent Softkeys</i>	Stimulus > Frequency Offset > Port n > Stop

SCPI.SENSe(*Ch*).OFFSet.RECeiver.FREQuency.DATA

<i>Object Type</i>	Property (read only)
<i>Data Type</i>	Variant (array of double)
<i>Target</i>	Channel <i>Ch</i> , <i>Ch</i> : channel number 1–16 (see Table 1 on page 25)
<i>Description</i>	<p>Reads the receiver frequency data when the frequency offset feature is ON and offset type is "SRCRcv", for the selected channel <i>Ch</i>.</p> <p>The array size is N, where N is the number of measurement points.</p> <p>For the n-th point, where n from 1 to N:</p> $\text{Data}(n-1) \quad \text{the frequency value at the } n\text{-th measurement point.}$
<i>Syntax</i>	<pre>Dim Data As Variant Data = app.SCPI.SENSe(<i>Ch</i>).OFFSet.RECeiver.FREQuency.DATA</pre>
<i>Related Commands</i>	SCPI.SENSe(<i>Ch</i>).OFFSet.STATE SCPI.SENSe(<i>Ch</i>).OFFset.TYPE
<i>Equivalent Softkeys</i>	None

SCPI.SENSe(*Ch*).OFFSet.RECeiver.FREQuency.DIVisor

<i>Object Type</i>	Property (read/write)
<i>Data Type</i>	Double
<i>Target</i>	Channel <i>Ch</i> , <i>Ch</i> : channel number 1–16 (see Table 1 on page 25)
<i>Description</i>	The basic frequency range divisor when the frequency offset feature is ON and offset type is "SRCRcv".
<i>Range</i>	from 1 to 1000
<i>Out of Range</i>	Sets the value of the limit, which is closer to the specified value.
<i>Preset Value</i>	1
<i>Unit</i>	n/a
<i>Syntax</i>	Dim <i>Value</i> As Double <i>Value</i> = <i>app</i> .SCPI.SENSe(<i>Ch</i>).OFFset.RECeiver.FREQuency.DIVisor <i>app</i> .SCPI.SENSe(<i>Ch</i>).OFFset.RECeiver.FREQuency.DIVisor = 2
<i>Related Commands</i>	SCPI.SENSe(<i>Ch</i>).OFFset.STATe SCPI.SENSe(<i>Ch</i>).OFFset.TYPe
<i>Equivalent Softkeys</i>	Stimulus > Frequency Offset > Receivers > Divider

SCPI.SENSe(*Ch*).OFFSet.RECeiver.FREQuency.MULTiplier

<i>Object Type</i>	Property (read/write)
<i>Data Type</i>	Double
<i>Target</i>	Channel <i>Ch</i> , <i>Ch</i> : channel number 1–16 (see Table 1 on page 25)
<i>Description</i>	The basic range frequency multiplier when the frequency offset feature is ON and offset type is "SRCRcv".
<i>Range</i>	from –1000 to 1000
<i>Out of Range</i>	Sets the value of the limit, which is closer to the specified value.
<i>Preset Value</i>	1
<i>Unit</i>	n/a
<i>Syntax</i>	<p>Dim <i>Value</i> As Double</p> <p><i>Value</i> = <i>app</i>.SCPI.SENSe(<i>Ch</i>).OFFset.RECeiver.FREQuency.MULTiplier</p> <p><i>app</i>.SCPI.SENSe(<i>Ch</i>).OFFset.RECeiver.FREQuency.MULTiplier = 2</p>
<i>Related Commands</i>	SCPI.SENSe(<i>Ch</i>).OFFset.STATe SCPI.SENSe(<i>Ch</i>).OFFset.TYPe
<i>Equivalent Softkeys</i>	Stimulus > Frequency Offset > Receivers > Multiplier

SCPI.SENSe(*Ch*).OFFSet.RECeiver.FREQuency.OFFSet

<i>Object Type</i>	Property (read/write)
<i>Data Type</i>	Double
<i>Target</i>	Channel <i>Ch</i> , <i>Ch</i> : channel number 1–16 (see Table 1 on page 25)
<i>Description</i>	The basic frequency range offset when the frequency offset feature is ON and offset type is "SRCRcv".
<i>Range</i>	from -1e-12 to 1e12
<i>Out of Range</i>	Sets the value of the limit, which is closer to the specified value.
<i>Preset Value</i>	0
<i>Unit</i>	Hz
<i>Syntax</i>	<p>Dim <i>Value</i> As Double</p> <p><i>Value</i> = app.SCPI.SENSe(<i>Ch</i>).OFFset.RECeiver.FREQuency.OFFSet</p> <p>app.SCPI.SENSe(<i>Ch</i>).OFFset.RECeiver.FREQuency.OFFSet = 1e9</p>
<i>Related Commands</i>	SCPI.SENSe(<i>Ch</i>).OFFset.STATe SCPI.SENSe(<i>Ch</i>).OFFset.TYPe
<i>Equivalent Softkeys</i>	Stimulus > Frequency Offset > Receivers > Offset

SCPI.SENSe(*Ch*).OFFSet.RECeiver.FREQuency.START

<i>Object Type</i>	Property (read/write)
<i>Data Type</i>	Double
<i>Target</i>	Channel <i>Ch</i> , <i>Ch</i> : channel number 1–16 (see Table 1 on page 25)
<i>Description</i>	The frequency sweep start when the frequency offset feature is ON and offset type is "SRCRcv".
<i>Unit</i>	Hz
<i>Syntax</i>	<p>Dim <i>Value</i> As Double <i>Value</i> = app.SCPI.SENSe(<i>Ch</i>).OFFset.RECeiver.FREQuency.START <i>app.SCPI.SENSe(Ch).OFFset.RECeiver.FREQuency.START</i> = 1e9</p>
<i>Related Commands</i>	SCPI.SENSe(<i>Ch</i>).OFFset.STATE SCPI.SENSe(<i>Ch</i>).OFFset.TYPE
<i>Equivalent Softkeys</i>	Stimulus > Frequency Offset > Receivers > Start

SCPI.SENSe(*Ch*).OFFSet.RECeiver.FREQuency.STOP

<i>Object Type</i>	Property (read/write)
<i>Data Type</i>	Double
<i>Target</i>	Channel <i>Ch</i> , <i>Ch</i> : channel number 1–16 (see Table 1 on page 25)
<i>Description</i>	The frequency sweep stop when the frequency offset feature is ON and offset type is "SRCRcv".
<i>Unit</i>	Hz
<i>Syntax</i>	<p>Dim <i>Value</i> As Double <i>Value</i> = <i>app</i>.SCPI.SENSe(<i>Ch</i>).OFFset.RECeiver.FREQuency.STOP <i>app</i>.SCPI.SENSe(<i>Ch</i>).OFFset.RECeiver.FREQuency.STOP = 1e9</p>
<i>Related Commands</i>	SCPI.SENSe(<i>Ch</i>).OFFset.STATE SCPI.SENSe(<i>Ch</i>).OFFset.TYPE
<i>Equivalent Softkeys</i>	Stimulus > Frequency Offset > Receivers > Stop

SCPI.SENSe(*Ch*).OFFSet.SOURce.FREQuency.DATA

<i>Object Type</i>	Property (read only)
<i>Data Type</i>	Variant (array of double)
<i>Target</i>	Channel <i>Ch</i> , <i>Ch</i> : channel number 1–16 (see Table 1 on page 25)
<i>Description</i>	Reads the source frequency data when the frequency offset feature is ON and offset type is "SRCRcv", for the selected channel <i>Ch</i> . The array size is N, where N is the number of measurement points. For the n-th point, where n from 1 to N: $Data(n-1)$ the frequency value at the n-th measurement point.
<i>Syntax</i>	Dim <i>Data</i> As Variant <i>Data</i> = <i>app</i> .SCPI.SENSe(<i>Ch</i>).OFFSet.SOURce.FREQuency.DATA
<i>Related Commands</i>	SCPI.SENSe(<i>Ch</i>).OFFset.STATE SCPI.SENSe(<i>Ch</i>).OFFset.TYPE
<i>Equivalent Softkeys</i>	None

SCPI.SENSe(*Ch*).OFFSet.SOURce.FREQuency.DIVisor

<i>Object Type</i>	Property (read/write)
<i>Data Type</i>	Double
<i>Target</i>	Channel <i>Ch</i> , <i>Ch</i> : channel number 1–16 (see Table 1 on page 25)
<i>Description</i>	The basic frequency range divisor when the frequency offset feature is ON and offset type is "SRCRcv".
<i>Range</i>	from 1 to 1000
<i>Out of Range</i>	Sets the value of the limit, which is closer to the specified value.
<i>Preset Value</i>	1
<i>Unit</i>	n/a
<i>Syntax</i>	<p>Dim <i>Value</i> As Double</p> <p><i>Value</i> = <i>app</i>.SCPI.SENSe(<i>Ch</i>).OFFset.SOURce.FREQuency.DIVisor</p> <p><i>app</i>.SCPI.SENSe(<i>Ch</i>).OFFset.SOURce.FREQuency.DIVisor = 2</p>
<i>Related Commands</i>	SCPI.SENSe(<i>Ch</i>).OFFset.STATe SCPI.SENSe(<i>Ch</i>).OFFset.TYPe
<i>Equivalent Softkeys</i>	Stimulus > Frequency Offset > Source > Divider

SCPI.SENSe(*Ch*).OFFSet.SOURce.FREQuency.MULTiplier

<i>Object Type</i>	Property (read/write)
<i>Data Type</i>	Double
<i>Target</i>	Channel <i>Ch</i> , <i>Ch</i> : channel number 1–16 (see Table 1 on page 25)
<i>Description</i>	The basic range frequency multiplier when the frequency offset feature is ON and offset type is "SRCRcv".
<i>Range</i>	from –1000 to 1000
<i>Out of Range</i>	Sets the value of the limit, which is closer to the specified value.
<i>Preset Value</i>	1
<i>Unit</i>	n/a
<i>Syntax</i>	<p>Dim <i>Value</i> As Double</p> <p><i>Value</i> = <i>app</i>.SCPI.SENSe(<i>Ch</i>).OFFset.SOURce.FREQuency.MULTiplier</p> <p><i>app</i>.SCPI.SENSe(<i>Ch</i>).OFFset.SOURce.FREQuency.MULTiplier = 2</p>
<i>Related Commands</i>	SCPI.SENSe(<i>Ch</i>).OFFset.STATe SCPI.SENSe(<i>Ch</i>).OFFset.TYPe
<i>Equivalent Softkeys</i>	Stimulus > Frequency Offset > Source > Multiplier

SCPI.SENSe(*Ch*).OFFSet.SOURce.FREQuency.OFFSet

<i>Object Type</i>	Property (read/write)
<i>Data Type</i>	Double
<i>Target</i>	Channel <i>Ch</i> , <i>Ch</i> : channel number 1–16 (see Table 1 on page 25)
<i>Description</i>	The basic frequency range offset when the frequency offset feature is ON and offset type is "SRCRcv".
<i>Range</i>	from -1e-12 to 1e12
<i>Out of Range</i>	Sets the value of the limit, which is closer to the specified value.
<i>Preset Value</i>	0
<i>Unit</i>	Hz
<i>Syntax</i>	<p>Dim <i>Value</i> As Double</p> <p><i>Value</i> = app.SCPI.SENSe(<i>Ch</i>).OFFset.SOURce.FREQuency.OFFSet</p> <p>app.SCPI.SENSe(<i>Ch</i>).OFFset.SOURce.FREQuency.OFFSet = 1e9</p>
<i>Related Commands</i>	SCPI.SENSe(<i>Ch</i>).OFFset.STATe SCPI.SENSe(<i>Ch</i>).OFFset.TYPE
<i>Equivalent Softkeys</i>	Stimulus > Frequency Offset > Source > Offset

SCPI.SENSe(*Ch*).OFFSet.SOURce.FREQuency.START

<i>Object Type</i>	Property (read/write)
<i>Data Type</i>	Double
<i>Target</i>	Channel <i>Ch</i> , <i>Ch</i> : channel number 1–16 (see Table 1 on page 25)
<i>Description</i>	The frequency sweep start when the frequency offset feature is ON and offset type is "SRCRcv".
<i>Unit</i>	Hz
<i>Syntax</i>	<p>Dim <i>Value</i> As Double <i>Value</i> = app.SCPI.SENSe(<i>Ch</i>).OFFset.SOURce.FREQuency.START <i>app.SCPI.SENSe(Ch).OFFset.SOURce.FREQuency.START</i> = 1e9</p>
<i>Related Commands</i>	SCPI.SENSe(<i>Ch</i>).OFFset.STATE SCPI.SENSe(<i>Ch</i>).OFFset.TYPE
<i>Equivalent Softkeys</i>	Stimulus > Frequency Offset > Source > Start

SCPI.SENSe(*Ch*).OFFSet.SOURce.FREQuency.STOP

<i>Object Type</i>	Property (read/write)
<i>Data Type</i>	Double
<i>Target</i>	Channel <i>Ch</i> , <i>Ch</i> : channel number 1–16 (see Table 1 on page 25)
<i>Description</i>	The frequency sweep stop when the frequency offset feature is ON and offset type is " SRCRcv".
<i>Unit</i>	Hz
<i>Syntax</i>	<p>Dim Value As Double</p> <p><i>Value</i> = <i>app</i>.SCPI.SENSe(<i>Ch</i>).OFFSet.SOURce.FREQuency.STOP</p> <p><i>app</i>.SCPI.SENSe(<i>Ch</i>).OFFSet.SOURce.FREQuency.STOP = 1e9</p>
<i>Related Commands</i>	SCPI.SENSe(<i>Ch</i>).OFFSet.STATE SCPI.SENSe(<i>Ch</i>).OFFSet.TYPE
<i>Equivalent Softkeys</i>	Stimulus > Frequency Offset > Source > Stop

SCPI.SENSe(*Ch*).OFFSet.STATE

<i>Object Type</i>	Property (read/write)
<i>Data Type</i>	Boolean
<i>Target</i>	Channel <i>Ch</i> , <i>Ch</i> : channel number 1–16 (see Table 1 on page 25)
<i>Description</i>	The ON/OFF state of the frequency offset feature.
<i>Allowable Values</i>	True: frequency offset feature is ON False: frequency offset feature is OFF
<i>Preset Value</i>	False
<i>Syntax</i>	<p>Dim <i>Status</i> As Boolean</p> <p><i>Status</i> = <i>app</i>.SCPI.SENSe(<i>Ch</i>).OFFSet.STATE</p> <p><i>app</i>.SCPI.SENSe(<i>Ch</i>).OFFSet.STATE = True</p>
<i>Equivalent Softkeys</i>	Stimulus > Frequency Offset > Frequency Offset

SCPI.SENSe(*Ch*).OFFSet.TYPE

<i>Object Type</i>	Property (read/write)
<i>Data Type</i>	String
<i>Target</i>	Channel <i>Ch</i> , <i>Ch</i> : channel number 1–16 (see Table 1 on page 25)
<i>Description</i>	The frequency offset type when the frequency offset feature is ON. There are two frequency offset types: "Port1/Port2" and "Source/Receivers". First offset type offsets ports against each other. Second offset type offsets source against receivers.
<i>Range</i>	"PORT" : Port1/Port2 "SRCRcv" : Source/Receivers
<i>Preset Value</i>	"PORT"
<i>Syntax</i>	Dim <i>Param</i> As String <i>Param</i> = app.SCPI.SENSe(<i>Ch</i>).OFFSet.TYPE app.SCPI.SENSe(<i>Ch</i>).OFFSet.TYPE = "PORT"
<i>Equivalent Softkeys</i>	Stimulus > Frequency Offset > Offset Type

SCPI.SENSE(*Ch*).ROSCillator.SOURce

<i>Object Type</i>	Property (read/write)
<i>Data Type</i>	String
<i>Target</i>	Channel <i>Ch</i> , <i>Ch</i> : channel number 1–16 (see Table 1 on page 25)
<i>Description</i>	Selects the internal or external source of the reference frequency of 10 MHz.
<i>Range</i>	"INTernal" : Internal source of the reference frequency "EXTernal" : External source of the reference frequency
<i>Notes</i>	The short form of the parameter is indicated by upper case letters. There is no distinction between upper and lower case letters when the property is written. When the property is read out, the short format is indicated by upper case letters.
<i>Out of Range</i>	An error occurs. Error code: 224.
<i>Preset Value</i>	"INT"
<i>Syntax</i>	<p>Dim <i>Param</i> As String</p> <pre>Param = app.SCPI.SENSE(<i>Ch</i>).ROSCillator.SOURce app.SCPI.SENSE(<i>Ch</i>).ROSCillator.SOURce = "EXT"</pre>
<i>Equivalent Softkeys</i>	System > Misc Setup > Ref Source

SCPI.SENSe(*Ch*).SEGMenT.DATa

<i>Object Type</i>	Property (read/write)
<i>Data Type</i>	Variant (array of double)
<i>Target</i>	Channel <i>Ch</i> , <i>Ch</i> : channel number 1–16 (see Table 1 on page 25)
<i>Description</i>	<p>The array of the segment sweep table.</p> <p>The array has the following format:</p> <pre>{ <Buf>, <Flag1>, <Flag2>, <Flag3>, <Flag4>, <Flag5>, <N>, <Start(1)>, <Stop(1)>, <NOP(1)> [,<IFBW(1)>] [,<Pow(1)>] [,<Del(1)>] [,<Time(1)>], <Start(2)>, <Stop(2)>, <NOP(2)> [,<IFBW(2)>] [,<Pow(2)>] [,<Del(2)>] [,<Time(2)>], ... <Start(N)>, <Stop(N)>, <NOP(N)> [,<IFBW(N)>] [,<Pow(N)>] [,<Del(N)>] [,<Time(N)>] }</pre> <p> <<i>Buf</i>> : Always 5, <<i>Flag1</i>> : Stimulus start setting (0 – start/stop, 1 – center/span), <<i>Flag2</i>> : Setting of the <<i>IFBW</i>> field (0 – disabled, 1 – enabled), <<i>Flag3</i>> : Setting of the <<i>Pow</i>> field (0 – disabled, 1 – enabled), <<i>Flag4</i>> : Setting of the <<i>Del</i>> field (0 – disabled, 1 – enabled), <<i>Flag5</i>> : Setting of the <<i>Time</i>> field (0 – disabled, 1 – enabled), <<i>N</i>> : Number of segments, <<i>Start n</i>> : Start value of the <i>n</i>-th segment, <<i>Stop n</i>> : Stop value of the <i>n</i>-th segment, <<i>NOP n</i>> : Number of points of the <i>n</i>-th segment, <<i>IFBW n</i>> : IF bandwidth of the <i>n</i>-th segment (if enabled), <<i>Pow n</i>> : Power of the <i>n</i>-th segment (if enabled), <<i>Del n</i>> : Measurement delay of the <i>n</i>-th segment (if enabled), <<i>Time n</i>> : Reserved for future use (if enabled). </p>
<i>Syntax</i>	<p>Dim <i>Data</i> As Variant</p> <p><i>Data</i> = <i>app</i>.SCPI.SENSe(<i>Ch</i>).SEGMenT.DATa</p> <p><i>app</i>.SCPI.SENSe(<i>Ch</i>).SEGMenT.DATa = <i>Data</i></p>
<i>Equivalent Softkeys</i>	Stimulus / Segment Table

SCPI.SENSe(*Ch*).SWEEP.POINT.TIME

<i>Object Type</i>	Property (read/write)
<i>Data Type</i>	Double
<i>Target</i>	Channel <i>Ch</i> , <i>Ch</i> : channel number 1–16 (see Table 1 on page 25)
<i>Description</i>	The value of the delay before measurement in each measurement point.
<i>Range</i>	from 0 to 0.3
<i>Resolution</i>	5E-6
<i>Out of Range</i>	Sets the value of the limit, which is closer to the specified value.
<i>Preset Value</i>	0
<i>Unit</i>	s (second)
<i>Syntax</i>	<p>Dim <i>Value</i> As Double</p> <p><i>Value</i> = <i>app</i>.SCPI.SENSe(<i>Ch</i>).SWEEP.POINT.TIME</p> <p><i>app</i>.SCPI.SENSe(<i>Ch</i>).SWEEP.POINT.TIME = 5E-6</p>
<i>Equivalent Softkeys</i>	Stimulus > Meas Delay

SCPI.SENSe(*Ch*).SWEEP.POINts

<i>Object Type</i>	Property (read/write)
<i>Data Type</i>	Long
<i>Target</i>	Channel <i>Ch</i> , <i>Ch</i> : channel number 1–16 (see Table 1 on page 25)
<i>Description</i>	The number of measurement points.
<i>Range</i>	2 to 500001 (PLANAR-804) 2 to 200001 (PLANAR-304, S5048, S7530)
<i>Out of Range</i>	Sets the value of the limit, which is closer to the specified value.
<i>Preset Value</i>	201
<i>Syntax</i>	Dim <i>Value</i> As Long <i>Value</i> = <i>app</i> .SCPI.SENSe(<i>Ch</i>).SWEEP.POINts <i>app</i> .SCPI.SENSe(<i>Ch</i>).SWEEP.POINts = 1001
<i>Equivalent Softkeys</i>	Stimulus > Points

SCPI.SENSe(*Ch*).SWEep.TYPE

<i>Object Type</i>	Property (read/write)
<i>Data Type</i>	String
<i>Target</i>	Channel <i>Ch</i> , <i>Ch</i> : channel number 1–16 (see Table 1 on page 25)
<i>Description</i>	Sets the sweep type.
<i>Range</i>	"LINear" : Linear frequency sweep "LOGarithmic" : Logarithmic frequency sweep "SEGMENT" : Segment frequency sweep "POWER" : Power sweep
<i>Notes</i>	The short form of the parameter is indicated by upper case letters. There is no distinction between upper and lower case letters when the property is written. When the property is read out, the short format is indicated by upper case letters.
<i>Out of Range</i>	An error occurs. Error code: 206.
<i>Preset Value</i>	"LIN"
<i>Syntax</i>	<pre>Dim Param As String Param = app.SCPI.SENSe(<i>Ch</i>).SWEep.TYPE app.SCPI.SENSe(<i>Ch</i>).SWEep.TYPE = "LOG"</pre>
<i>Equivalent Softkeys</i>	Stimulus > Sweep Type

SCPI.SERViCe.CHANnel(1).ACTive

<i>Object Type</i>	Property (read only)
<i>Data Type</i>	Long
<i>Target</i>	Analyzer
<i>Description</i>	The number of the active channel.
<i>Syntax</i>	Dim Value As Long Value = app.SCPI.SERViCe.CHANnel.ACTive
<i>Warning</i>	Object <i>CHANnel</i> has an index. In Visual Basic the index is 1 by default when it is omitted. The index can not be omitted in other programming languages.
<i>Equivalent Softkeys</i>	None

SCPI.SERVICE.CHANNEL(1).COUNT

<i>Object Type</i>	Property (read only)
<i>Data Type</i>	Long
<i>Target</i>	Analyzer
<i>Description</i>	The maximum number of channels.
<i>Syntax</i>	Dim Value As Long <code>Value = app.SCPI.SERVICE.CHANNEL.COUNT</code>
<i>Warning</i>	Object <i>CHANNEL</i> has an index. In Visual Basic the index is 1 by default when it is omitted. The index can not be omitted in other programming languages.
<i>Equivalent Softkeys</i>	None

SCPI.SERVICE.CHANNEL(*Ch*).TRACE(1).ACTIVE

<i>Object Type</i>	Property (read only)
<i>Data Type</i>	Long
<i>Target</i>	Channel <i>Ch</i> , <i>Ch</i> : channel number 1–16 (see Table 1 on page 25)
<i>Description</i>	The active trace number of the specified channel.
<i>Syntax</i>	Dim Value As Long <code>Value = app.SCPI.SERVICE.CHANNEL(<i>Ch</i>).TRACE.ACTIVE</code>
<i>Warning</i>	Object <i>TRACE</i> has an index. In Visual Basic the index is 1 by default when it is omitted. The index can not be omitted in other programming languages.
<i>Equivalent Softkeys</i>	None

SCPI.SERVICE.CHANNEL(1).TRACe.COUNT

<i>Object Type</i>	Property (read only)
<i>Data Type</i>	Long
<i>Target</i>	Analyzer
<i>Description</i>	The maximum number of the traces in the channel.
<i>Syntax</i>	<p>Dim <i>Value</i> As Long</p> <p><i>Value</i> = <i>app.SCPI.SERVICE.CHANNEL.TRACE.COUNT</i></p>
<i>Warning</i>	Objects <i>CHANNEL</i> and <i>TRACE</i> have an index. In Visual Basic the index is 1 by default when it is omitted. The index can not be omitted in other programming languages.
<i>Equivalent Softkeys</i>	None

SCPI.SERVICE.CHANNEL(*Ch*).TRACe(*Tr*).MARKer.ACTive

<i>Object Type</i>	Property (read only)
<i>Data Type</i>	Long
<i>Target</i>	<p>The specified trace <i>Tr</i> of channel <i>Ch</i>,</p> <p><i>Tr</i>: trace number 1–16 (see Table 3 on page 33)</p> <p><i>Ch</i>: channel number 1–16 (see Table 1 on page 25)</p>
<i>Description</i>	The active marker number of the specified trace of the specified channel.
<i>Syntax</i>	<p>Dim <i>Value</i> As Long</p> <p><i>Value</i> = <i>app.SCPI.SERVICE.CHANNEL(Ch).TRACe.Tr.MARKer.ACTive</i></p>
<i>Equivalent Softkeys</i>	None

SCPI.SERVICE.PORT.COUNT

<i>Object Type</i>	Property (read only)
<i>Data Type</i>	Long
<i>Target</i>	Analyzer
<i>Description</i>	The number of the ports.
<i>Syntax</i>	Dim <i>Value</i> As Long <i>Value</i> = <i>app.SCPI.SERVICE.PORT.COUNT</i>
<i>Equivalent Softkeys</i>	None

SCPI.SERVICE.SWEep.FREQency.MAXimum

<i>Object Type</i>	Property (read only)
<i>Data Type</i>	Double
<i>Target</i>	Analyzer
<i>Description</i>	The maximum value of the analyzer frequency range.
<i>Syntax</i>	Dim <i>Value</i> As Double <i>Value</i> = <i>app.SCPI.SERVICE.SWEep.FREQency.MAXimum</i>
<i>Equivalent Softkeys</i>	None

SCPI.SERVICE.SWEep.FREQency.MINimum

<i>Object Type</i>	Property (read only)
<i>Data Type</i>	Double
<i>Target</i>	Analyzer
<i>Description</i>	The minimum value of the analyzer frequency range.
<i>Syntax</i>	Dim <i>Value</i> As Double <i>Value</i> = <i>app.SCPI.SERVICE.SWEep.FREQency.MINimum</i>
<i>Equivalent Softkeys</i>	None

SCPI.SERVICE.SWEep.POINts

<i>Object Type</i>	Property (read only)
<i>Data Type</i>	Double
<i>Target</i>	Analyzer
<i>Description</i>	The maximum number of the measurement points.
<i>Syntax</i>	Dim <i>Value</i> As Long <i>Value</i> = <i>app.SCPI.SERVICE.SWEep.POINts</i>
<i>Equivalent Softkeys</i>	None

SCPI.SOURce(*Ch*).POWer.CENTer

<i>Object Type</i>	Property (read/write)
<i>Data Type</i>	Double
<i>Target</i>	Channel <i>Ch</i> , <i>Ch</i> : channel number 1–16 (see Table 1 on page 25)
<i>Description</i>	The center value of the power sweep range when sweep type is Power.
<i>Range</i>	From the minimum value to the maximum value of the analyzer power range.
<i>Resolution</i>	0.025
<i>Out of Range</i>	Sets the value of the limit, which is closer to the specified value.
<i>Preset Value</i>	Center value of the analyzer power range.
<i>Unit</i>	dBm (decibels above 1 milliwatt)
<i>Syntax</i>	<p>Dim <i>Value</i> As Double <i>Value</i> = <i>app</i>.SCPI.SOURce(<i>Ch</i>).POWer.CENTer <i>app</i>.SCPI.SOURce(<i>Ch</i>).POWer.CENTer = 5</p>
<i>Equivalent Softkeys</i>	Stimulus > Center

SCPI.SOURce(*Ch*).POWer.LEVel.IMMediate.AMPLitude

<i>Object Type</i>	Property (read/write)
<i>Data Type</i>	Double
<i>Target</i>	Channel <i>Ch</i> , <i>Ch</i> : channel number 1–16 (see Table 1 on page 25)
<i>Description</i>	The power level for the frequency sweep when port couple feature is set to ON by the command SCPI.SOURce(<i>Ch</i>).POWer.PORT(<i>Pt</i>).COUPle.
<i>Range</i>	From the minimum value to the maximum value of the analyzer power range.
<i>Resolution</i>	0.05
<i>Out of Range</i>	Sets the value of the limit, which is closer to the specified value.
<i>Preset Value</i>	0
<i>Unit</i>	dBm (decibels above 1 milliwatt)
<i>Syntax</i>	<p>Dim <i>Value</i> As Double</p> <p><i>Value</i> = <i>app</i>.SCPI.SOURce(<i>Ch</i>).POWer.LEVel.IMMediate.AMPLitude</p> <p><i>app</i>.SCPI.SOURce(<i>Ch</i>).POWer.LEVel.IMMediate.AMPLitude = 10</p>
<i>Related Commands</i>	SCPI.SOURce(<i>Ch</i>).POWer.PORT(<i>Pt</i>).COUPle
<i>Equivalent Softkeys</i>	Stimulus > Power > Power

SCPI.SOURce(*Ch*).POWer.LEVel.SLOPe.DATA

<i>Object Type</i>	Property (read/write)
<i>Data Type</i>	Double
<i>Target</i>	Channel <i>Ch</i> , <i>Ch</i> : channel number 1–16 (see Table 1 on page 25)
<i>Description</i>	The power slope value for the frequency sweep.
<i>Range</i>	from –2 to 2
<i>Resolution</i>	0.1
<i>Out of Range</i>	Sets the value of the limit, which is closer to the specified value.
<i>Preset Value</i>	0
<i>Unit</i>	dB/GHz (decibel/gigahertz)
<i>Syntax</i>	<p>Dim <i>Value</i> As Double</p> <pre>Value = app.SCPI.SOURce(<i>Ch</i>).POWer.LEVel.SLOPe.DATA app.SCPI.SOURce(<i>Ch</i>).POWer.LEVel.SLOPe.DATA = 0.2</pre>
<i>Equivalent Softkeys</i>	Stimulus > Power > Slope [dB/GHz]

SCPI.SOURce(*Ch*).POWer.LEVel.SLOPe.STATE

<i>Object Type</i>	Property (read/write)
<i>Data Type</i>	Boolean
<i>Target</i>	Channel <i>Ch</i> , <i>Ch</i> : channel number 1–16 (see Table 1 on page 25)
<i>Description</i>	The ON/OFF state of the power slope for the frequency sweep.
<i>Allowable Values</i>	True: Power slope ON False: Power slope OFF
<i>Preset Value</i>	False
<i>Syntax</i>	<p>Dim <i>Status</i> As Boolean</p> <pre>Status = app.SCPI.SOURce(<i>Ch</i>).POWer.LEVel.SLOPe.STATE app.SCPI.SOURce(<i>Ch</i>).POWer.LEVel.SLOPe.STATE = True</pre>
<i>Equivalent Softkeys</i>	Stimulus > Power > Slope [ON/OFF]

SCPI.SOURce(*Ch*).POWER.PORT(*Pt*).CORRection.COLLect.ACQuire

<i>Object Type</i>	Method
<i>Target</i>	Port <i>Pt</i> of channel <i>Ch</i> , <i>Ch</i> : channel number 1–16 (see Table 1 on page 25) <i>Pt</i> : port number 1–2 (see Table 2 on page 25)
<i>Description</i>	Measures the power calibration data for the specified port using the power meter controlled via USB or USB/GPIB. Calculates calibration coefficients on completion of the measurement, and turns ON the power correction for the port.
<i>Syntax</i>	<i>app</i> .SCPI.SOURce(<i>Ch</i>).POWER.PORT(<i>Pt</i>).CORRection.COLLect.ACQuire
<i>Equivalent Softkeys</i>	Calibration > Power Calibration > Calibrate

SCPI.SOURce(*Ch*).POWer.PO RT(*Pt*).CORRection.COLLect.TABLe. LOSS.DATa

<i>Object Type</i>	Property (read/write)
<i>Data Type</i>	Variant (array of double)
<i>Target</i>	Port <i>Pt</i> of channel <i>Ch</i> , <i>Ch</i> : channel number 1–16 (see Table 1 on page 25) <i>Pt</i> : port number 1–2 (see Table 2 on page 25)
<i>Description</i>	The loss compensation table used during the power calibration. The array size is 1+2N, where N is the number of the table rows. For the n-th point, where n from 1 to N: <i>Data(0)</i> the number of the table rows N integer from 0 to 100; <i>Data(2n-1)</i> the frequency of the n-th row of the table from 300 kHz to 8.0 GHz; <i>Data(2n)</i> the loss value of the n-th table row in dB.
<i>Syntax</i>	Dim <i>Data</i> As Variant <i>Data</i> = app.SCPI.SOURce(<i>Ch</i>).POWer.PO RT(<i>Pt</i>).CORRection.COLLect.TABLe.LOSS.DATa app.SCPI.SOURce(<i>Ch</i>).POWer.PO RT(<i>Pt</i>).CORRection.COLLect.TABLe.LOSS.DATa = <i>Data</i>
<i>Related Commands</i>	SCPI.SOURce(<i>Ch</i>).POWer.PO RT(<i>Pt</i>).CORRection.COLLect.ACQuire
<i>Equivalent Softkeys</i>	Calibration > Power Calibration > Loss Compen

SCPI.SOURce(*Ch*).POWeR.PO_t.CORRection.COLLect.TABLe. LOSS.STATe

<i>Object Type</i>	Property (read/write)
<i>Data Type</i>	Boolean
<i>Target</i>	Port <i>Pt</i> of channel <i>Ch</i> , <i>Ch</i> : channel number 1–16 (see Table 1 on page 25) <i>Pt</i> : port number 1–2 (see Table 2 on page 25)
<i>Description</i>	The ON/OFF state of the loss compensation function used during the power calibration.
<i>Allowable Values</i>	True: Loss compensation ON False: Loss compensation OFF
<i>Preset Value</i>	False
<i>Syntax</i>	<p>Dim <i>Status</i> As Boolean</p> <pre><i>Status</i> = app.SCPI.SOURce(<i>Ch</i>).POWeR.PO_t.CORRection.COLLect.TABLe. LOSS.STATe</pre> <pre>app.SCPI.SOURce(<i>Ch</i>).POWeR.PO_t.CORRection.COLLect.TABLe.LOSS.STATe = True</pre>
<i>Related Commands</i>	SCPI.SOURce(<i>Ch</i>).POWeR.PO _t .CORRection.COLLect.TABLe.LOSS.DATA
<i>Equivalent Softkeys</i>	Calibration > Power Calibration > Loss Compen > Compensation

SCPI.SOURce(*Ch*).POWer.PO RT(*Pt*).CORRection.DAT A

<i>Object Type</i>	Property (read/write)
<i>Data Type</i>	Variant (array of double)
<i>Target</i>	Port <i>Pt</i> of channel <i>Ch</i> , <i>Ch</i> : channel number 1–16 (see Table 1 on page 25) <i>Pt</i> : port number 1–2 (see Table 2 on page 25)
<i>Description</i>	The power correction table. The array size is 1+2N, where N is the number of the table rows. For the n-th point, where n from 1 to N: <i>Data(0)</i> the number of the table rows N integer from 0 to 10001; <i>Data(2n-1)</i> the frequency of the n-th row of the table from 300 kHz to 8.0 GHz; <i>Data(2n)</i> power correction value of the n-th row of the table from –10 to +10 dB.
<i>Syntax</i>	Dim <i>Data</i> As Variant <i>Data</i> = <i>app</i> .SCPI.SOURce(<i>Ch</i>).POWer.PO RT(<i>Pt</i>).CORRection.DAT A <i>app</i> .SCPI.SOURce(<i>Ch</i>).POWer.PO RT(<i>Pt</i>).CORRection.DAT A = <i>Data</i>
<i>Related Commands</i>	SCPI.SOURce(<i>Ch</i>).POWer.PO RT(<i>Pt</i>).CORRection.COLLect.ACQuire
<i>Equivalent Softkeys</i>	None

SCPI.SOURce(*Ch*).POWer.PORT(*Pt*).CORRection.STATE

<i>Object Type</i>	Property (read/write)
<i>Data Type</i>	Boolean
<i>Target</i>	Port <i>Pt</i> of channel <i>Ch</i> , <i>Ch</i> : channel number 1–16 (see Table 1 on page 25) <i>Pt</i> : port number 1–2 (see Table 2 on page 25)
<i>Description</i>	The ON/OFF state of the power correction function.
<i>Allowable Values</i>	True: Power correction ON False: Power correction OFF
<i>Preset Value</i>	False
<i>Syntax</i>	Dim <i>Status</i> As Boolean <i>Status</i> = <i>app</i> .SCPI.SOURce(<i>Ch</i>).POWer.PORT(<i>Pt</i>).CORRection.STATE <i>app</i> .SCPI.SOURce(<i>Ch</i>).POWer.PORT(<i>Pt</i>).CORRection.STATE = True
<i>Related Commands</i>	SCPI.SOURce(<i>Ch</i>).POWer.PORT(<i>Pt</i>).CORRection.COLLect.ACQuire
<i>Equivalent Softkeys</i>	Calibration > Power Calibration > Correction

SCPI.SOURce(*Ch*).POWer.PORT(1).COUPLE

<i>Object Type</i>	Property (read/write)
<i>Data Type</i>	Boolean
<i>Target</i>	Channel <i>Ch</i> , <i>Ch</i> : channel number 1–16 (see Table 1 on page 25)
<i>Description</i>	The ON/OFF state of the port power couple function. Setting the port power couple feature to OFF allows independent power level setting for each port.
<i>Allowable Values</i>	True: Port power couple ON False: Port power couple OFF
<i>Preset Value</i>	True
<i>Syntax</i>	Dim <i>Status</i> As Boolean <i>Status</i> = <i>app</i> .SCPI.SOURce(<i>Ch</i>).POWer.PORT.COUPLE <i>app</i> .SCPI.SOURce(<i>Ch</i>).POWer.PORT.COUPLE = True
<i>Warning</i>	Object <i>PORT</i> has an index. In Visual Basic the index is 1 by default when it is omitted. The index can not be omitted in other programming languages.
<i>Related Commands</i>	SCPI.SOURce(<i>Ch</i>).POWer.PORT(Pt).LEVel.IMMEDIATE.AMPLitude
<i>Equivalent Softkeys</i>	Stimulus > Power > Port Couple

SCPI.SOURce(*Ch*).POWer.PO RT(*Pt*).LEVel.I MMEDIATE.AMPLitude

<i>Object Type</i>	Property (read/write)
<i>Data Type</i>	Double
<i>Target</i>	Port <i>Pt</i> of channel <i>Ch</i> , <i>Ch</i> : channel number 1–16 (see Table 1 on page 25) <i>Pt</i> : port number 1–2 (see Table 2 on page 25)
<i>Description</i>	The power level of each port for the frequency sweep when port couple feature is set to OFF by the command SCPI.SOURce(<i>Ch</i>).POWer.PO RT(<i>Pt</i>).COUPlE.
<i>Range</i>	From the minimum value to the maximum value of the analyzer power range.
<i>Resolution</i>	0.05
<i>Out of Range</i>	Sets the value of the limit, which is closer to the specified value.
<i>Preset Value</i>	0
<i>Unit</i>	dBm (decibels above 1 milliwatt)
<i>Syntax</i>	Dim <i>Value</i> As Double <i>Value</i> = <i>app</i> .SCPI.SOURce(<i>Ch</i>).POWer.PO RT(<i>Pt</i>).LEVel.I MMEDIATE.AMPLitude <i>app</i> .SCPI.SOURce(<i>Ch</i>).POWer.PO RT(<i>Pt</i>).LEVel.I MMEDIATE.AMPLitude = 10
<i>Related Commands</i>	SCPI.SOURce(<i>Ch</i>).POWer.PO RT(<i>Pt</i>).COUPlE
<i>Equivalent Softkeys</i>	Stimulus > Power > Port Power > [Port1 Port2]

SCPI.SOURce(*Ch*).POWER.SPAN

<i>Object Type</i>	Property (read/write)
<i>Data Type</i>	Double
<i>Target</i>	Channel <i>Ch</i> , <i>Ch</i> : channel number 1–16 (see Table 1 on page 25)
<i>Description</i>	The power span for the power sweep.
<i>Range</i>	From 0 to the maximum span of the analyzer power range.
<i>Resolution</i>	0.05
<i>Out of Range</i>	Sets the value of the limit, which is closer to the specified value.
<i>Preset Value</i>	The maximum span of the analyzer power range.
<i>Unit</i>	dBm (decibels above 1 milliwatt)
<i>Syntax</i>	<p>Dim <i>Value</i> As Double <i>Value</i> = <i>app</i>.SCPI.SOURce(<i>Ch</i>).POWER.SPAN <i>app</i>.SCPI.SOURce(<i>Ch</i>).POWER.SPAN = 50</p>
<i>Equivalent Softkeys</i>	Stimulus > Span

SCPI.SOURce(*Ch*).POWeR.STARt

<i>Object Type</i>	Property (read/write)
<i>Data Type</i>	Double
<i>Target</i>	Channel <i>Ch</i> , <i>Ch</i> : channel number 1–16 (see Table 1 on page 25)
<i>Description</i>	The power sweep start for the power sweep.
<i>Range</i>	From the minimum value to the maximum value of the analyzer power range.
<i>Resolution</i>	0.05
<i>Out of Range</i>	Sets the value of the limit, which is closer to the specified value.
<i>Preset Value</i>	The minimum value of the analyzer power range.
<i>Unit</i>	dBm (decibels above 1 milliwatt)
<i>Syntax</i>	<p>Dim <i>Value</i> As Double</p> <p><i>Value</i> = <i>app</i>.SCPI.SOURce(<i>Ch</i>).POWeR.STARt</p> <p><i>app</i>.SCPI.SOURce(<i>Ch</i>).POWeR.STARt = 5</p>
<i>Equivalent Softkeys</i>	Stimulus > Start

SCPI.SOURce(*Ch*).POWER.STOP

<i>Object Type</i>	Property (read/write)
<i>Data Type</i>	Double
<i>Target</i>	Channel <i>Ch</i> , <i>Ch</i> : channel number 1–16 (see Table 1 on page 25)
<i>Description</i>	The power sweep stop for the power sweep.
<i>Range</i>	From the minimum value to the maximum value of the analyzer power range.
<i>Resolution</i>	0.05
<i>Out of Range</i>	Sets the value of the limit, which is closer to the specified value.
<i>Preset Value</i>	The maximum value of the analyzer power range.
<i>Unit</i>	dBm (decibels above 1 milliwatt)
<i>Syntax</i>	<p>Dim <i>Value</i> As Double <i>Value</i> = <i>app</i>.SCPI.SOURce(<i>Ch</i>).POWER.STOP <i>app</i>.SCPI.SOURce(<i>Ch</i>).POWER.STOP = 5</p>
<i>Equivalent Softkeys</i>	Stimulus > Stop

SCPI.STATUS.OPERATION.CONDITION

<i>Object Type</i>	Property (read only)
<i>Data Type</i>	Long
<i>Target</i>	IEE488.2 Status Reporting System (Appendix 1)
<i>Description</i>	Operation Status Condition Register.
<i>Syntax</i>	<p>Dim <i>Value</i> As Long</p> <p><i>Value</i> = <i>app.SCPI.STATUS.OPERATION.CONDITION</i></p>
<i>Equivalent Softkeys</i>	None

SCPI.STATUS.OPERATION.ENABLE

<i>Object Type</i>	Property (read/write)
<i>Data Type</i>	Long
<i>Target</i>	IEE488.2 Status Reporting System (Appendix 1)
<i>Description</i>	Operation Status Enable Register.
<i>Range</i>	from 0 to 65535
<i>Preset Value</i>	0
<i>Syntax</i>	<p>Dim <i>Value</i> As Long</p> <p><i>Value</i> = <i>app.SCPI.STATUS.OPERATION.ENABLE</i></p> <p><i>app.SCPI.STATUS.OPERATION.ENABLE</i> = <i>Value</i></p>
<i>Equivalent Softkeys</i>	None

SCPI.STATUS.OPERATION.EVENT

<i>Object Type</i>	Property (read only)
<i>Data Type</i>	Long
<i>Target</i>	IEE488.2 Status Reporting System (Appendix 1)
<i>Description</i>	Operation Status Event Register.
<i>Syntax</i>	<p>Dim <i>Value</i> As Long</p> <p><i>Value</i> = <i>app.SCPI.STATUS.OPERATION.EVENT</i></p>
<i>Equivalent Softkeys</i>	None

SCPI.STATUS.OPERATION.NTRANSITION

<i>Object Type</i>	Property (read/write)
<i>Data Type</i>	Long
<i>Target</i>	IEE488.2 Status Reporting System (Appendix 1)
<i>Description</i>	Negative Transition Filter of the Operation Status Register.
<i>Range</i>	from 0 to 65535
<i>Preset Value</i>	0
<i>Syntax</i>	<p>Dim <i>Value</i> As Long</p> <p><i>Value</i> = <i>app.SCPI.STATUS.OPERATION.NTRANSITION</i></p> <p><i>app.SCPI.STATUS.OPERATION.NTRANSITION</i> = <i>Value</i></p>
<i>Equivalent Softkeys</i>	None

SCPI.STATus.OPERation.PTRansition

<i>Object Type</i>	Property (read/write)
<i>Data Type</i>	Long
<i>Target</i>	IEE488.2 Status Reporting System (Appendix 1)
<i>Description</i>	Positive Transition Filter of the Operation Status Register.
<i>Range</i>	from 0 to 65535
<i>Preset Value</i>	65535
<i>Syntax</i>	<p>Dim <i>Value</i> As Long</p> <p><i>Value</i> = <i>app.SCPI.STATus.OPERation.PTRansition</i></p> <p><i>app.SCPI.STATus.OPERation.PTRansition</i> = <i>Value</i></p>
<i>Equivalent Softkeys</i>	None

SCPI.STATus.PRESet

<i>Object Type</i>	Method
<i>Target</i>	IEE488.2 Status Reporting System (Appendix 1)
<i>Description</i>	Initialization of all registers.
<i>Syntax</i>	<i>app.SCPI.STATus.PRESet</i>
<i>Equivalent Softkeys</i>	None

SCPI.STATUS.QUESTIONable.CONDITION

<i>Object Type</i>	Property (read only)
<i>Data Type</i>	Long
<i>Target</i>	IEE488.2 Status Reporting System (Appendix 1)
<i>Description</i>	Questionable Status Condition Register.
<i>Syntax</i>	<p>Dim <i>Value</i> As Long</p> <p><i>Value</i> = <i>app.SCPI.STATUS.QUESTIONable.CONDITION</i></p>
<i>Equivalent Softkeys</i>	None

SCPI.STATUS.QUESTIONable.ENABLE

<i>Object Type</i>	Property (read/write)
<i>Data Type</i>	Long
<i>Target</i>	IEE488.2 Status Reporting System (Appendix 1)
<i>Description</i>	Questionable Status Enable Register.
<i>Range</i>	from 0 to 65535
<i>Preset Value</i>	0
<i>Syntax</i>	<p>Dim <i>Value</i> As Long</p> <p><i>Value</i> = <i>app.SCPI.STATUS.QUESTIONable.ENABLE</i></p> <p><i>app.SCPI.STATUS.QUESTIONable.ENABLE</i> = <i>Value</i></p>
<i>Equivalent Softkeys</i>	None

SCPI.STATus.QUESTIONable.EVENT

<i>Object Type</i>	Property (read only)
<i>Data Type</i>	Long
<i>Target</i>	IEE488.2 Status Reporting System (Appendix 1)
<i>Description</i>	Questionable Status Event Register.
<i>Syntax</i>	<p>Dim <i>Value</i> As Long</p> <p><i>Value</i> = <i>app.SCPI.STATUS.QUESTIONable.EVENT</i></p>
<i>Equivalent Softkeys</i>	None

SCPI.STATus.QUESTIONable.LIMit.CHANnel(*Ch*).CONDITION

<i>Object Type</i>	Property (read only)
<i>Data Type</i>	Long
<i>Target</i>	IEE488.2 Status Reporting System (Appendix 1)
<i>Description</i>	Questionable Limit Channel Status Condition Register for channel <i>Ch</i> , <i>Ch</i> : channel number 1–16 (see Table 1 on page 25)
<i>Syntax</i>	<p>Dim <i>Value</i> As Long</p> <p><i>Value</i> = <i>app.SCPI.STATUS.QUESTIONable.LIMit.CHANnel(Ch).CONDITION</i></p>
<i>Equivalent Softkeys</i>	None

SCPI.STATUS.QUESTIONable.LIMit.CHANnel(*Ch*).ENABLE

<i>Object Type</i>	Property (read/write)
<i>Data Type</i>	Long
<i>Target</i>	IEE488.2 Status Reporting System (Appendix 1)
<i>Description</i>	Questionable Limit Channel Status Enable Register for channel <i>Ch</i> , <i>Ch</i> : channel number 1–16 (see Table 1 on page 25)
<i>Range</i>	from 0 to 65535
<i>Preset Value</i>	0
<i>Syntax</i>	<p>Dim <i>Value</i> As Long</p> <p><i>Value</i> = <i>app</i>.SCPI.STATUS.QUESTIONable.LIMit.CHANnel(<i>Ch</i>).ENABLE</p> <p><i>app</i>.SCPI.STATUS.QUESTIONable.LIMit.CHANnel(<i>Ch</i>).ENABLE = <i>Value</i></p>
<i>Equivalent Softkeys</i>	None

SCPI.STATUS.QUESTIONable.LIMit.CHANnel(*Ch*).EVENT

<i>Object Type</i>	Property (read only)
<i>Data Type</i>	Long
<i>Target</i>	IEE488.2 Status Reporting System (Appendix 1)
<i>Description</i>	Questionable Limit Channel Status Event Register for channel <i>Ch</i> , <i>Ch</i> : channel number 1–16 (see Table 1 on page 25)
<i>Syntax</i>	<p>Dim <i>Value</i> As Long</p> <p><i>Value</i> = <i>app</i>.SCPI.STATUS.QUESTIONable.LIMit.CHANnel(<i>Ch</i>).EVENT</p>
<i>Equivalent Softkeys</i>	None

SCPI.STATus.QUESTIONable.LIMit.CHANnel(*Ch*).NTRansition

<i>Object Type</i>	Property (read/write)
<i>Data Type</i>	Long
<i>Target</i>	IEE488.2 Status Reporting System (Appendix 1)
<i>Description</i>	Negative Transition Filter of the Questionable Limit Channel Status Register for channel <i>Ch</i> , <i>Ch</i> : channel number 1–16 (see Table 1 on page 25)
<i>Range</i>	from 0 to 65535
<i>Preset Value</i>	0
<i>Syntax</i>	<p>Dim <i>Value</i> As Long</p> <p><i>Value</i> = <i>app.SCPI.STATus.QUESTIONable.LIMit.CHANnel(Ch).NTRansition</i></p> <p><i>app.SCPI.STATus.QUESTIONable.LIMit.CHANnel(Ch).NTRansition</i> = <i>Value</i></p>
<i>Equivalent Softkeys</i>	None

SCPI.STATus.QUESTIONable.LIMit.CHANnel(*Ch*).PTRansition

<i>Object Type</i>	Property (read/write)
<i>Data Type</i>	Long
<i>Target</i>	IEE488.2 Status Reporting System (Appendix 1)
<i>Description</i>	Positive Transition Filter of the Questionable Limit Channel Status Register for channel <i>Ch</i> , <i>Ch</i> : channel number 1–16 (see Table 1 on page 25)
<i>Range</i>	from 0 to 65535
<i>Preset Value</i>	65535
<i>Syntax</i>	<p>Dim <i>Value</i> As Long</p> <p><i>Value</i> = <i>app.SCPI.STATus.QUESTIONable.LIMit.CHANnel(Ch).PTRansition</i></p> <p><i>app.SCPI.STATus.QUESTIONable.LIMit.CHANnel(Ch).PTRansition</i> = <i>Value</i></p>
<i>Equivalent Softkeys</i>	None

SCPI.STATus.QUESTIONable.LIMit.CONDITION

<i>Object Type</i>	Property (read only)
<i>Data Type</i>	Long
<i>Target</i>	IEE488.2 Status Reporting System (Appendix 1).
<i>Description</i>	Questionable Limit Status Condition Register.
<i>Syntax</i>	<p>Dim <i>Value</i> As Long</p> <p><i>Value</i> = <i>app.SCPI.STATus.QUESTIONable.LIMit.CONDITION</i></p>
<i>Equivalent Softkeys</i>	None

SCPI.STATUs.QUESTIONable.LIMit.ENABLE

<i>Object Type</i>	Property (read/write)
<i>Data Type</i>	Long
<i>Target</i>	IEE488.2 Status Reporting System (Appendix 1)
<i>Description</i>	Questionable Limit Status Enable Register.
<i>Range</i>	from 0 to 65535
<i>Preset Value</i>	0
<i>Syntax</i>	<p>Dim Value As Long</p> <p><i>Value</i> = <i>app.SCPI.STATUs.QUESTIONable.LIMit.ENABLE</i></p> <p><i>app.SCPI.STATUs.QUESTIONable.LIMit.ENABLE</i> = <i>Value</i></p>
<i>Equivalent Softkeys</i>	None

SCPI.STATUs.QUESTIONable.LIMit.EVENT

<i>Object Type</i>	Property (read only)
<i>Data Type</i>	Long
<i>Target</i>	IEE488.2 Status Reporting System (Appendix 1)
<i>Description</i>	Questionable Limit Status Event Register.
<i>Syntax</i>	<p>Dim Value As Long</p> <p><i>Value</i> = <i>app.SCPI.STATUs.QUESTIONable.LIMit.EVENT</i></p>
<i>Equivalent Softkeys</i>	None

SCPI.STATus.QUESTIONable.LIMit.NTRansition

<i>Object Type</i>	Property (read/write)
<i>Data Type</i>	Long
<i>Target</i>	IEE488.2 Status Reporting System (Appendix 1)
<i>Description</i>	Negative Transition Filter of the Questionable Limit Status Register.
<i>Range</i>	from 0 to 65535
<i>Preset Value</i>	0
<i>Syntax</i>	<p>Dim <i>Value</i> As Long</p> <p><i>Value</i> = <i>app.SCPI.STATus.QUESTIONable.LIMit.NTRansition</i></p> <p><i>app.SCPI.STATus.QUESTIONable.LIMit.NTRansition</i> = <i>Value</i></p>
<i>Equivalent Softkeys</i>	None

SCPI.STATus.QUESTIONable.LIMit.PTRansition

<i>Object Type</i>	Property (read/write)
<i>Data Type</i>	Long
<i>Target</i>	IEE488.2 Status Reporting System (Appendix 1)
<i>Description</i>	Positive Transition Filter of the Questionable Limit Status Register.
<i>Range</i>	from 0 to 65535
<i>Preset Value</i>	65535
<i>Syntax</i>	<p>Dim <i>Value</i> As Long</p> <p><i>Value</i> = <i>app.SCPI.STATus.QUESTIONable.LIMit.PTRansition</i></p> <p><i>app.SCPI.STATus.QUESTIONable.LIMit.PTRansition</i> = <i>Value</i></p>
<i>Equivalent Softkeys</i>	None

SCPI.STATUS.QUESTIONable.NTRansition

<i>Object Type</i>	Property (read/write)
<i>Data Type</i>	Long
<i>Target</i>	IEE488.2 Status Reporting System (Appendix 1)
<i>Description</i>	Negative Transition Filter of the Questionable Status Register.
<i>Range</i>	from 0 to 65535
<i>Preset Value</i>	0
<i>Syntax</i>	<p>Dim <i>Value</i> As Long</p> <p><i>Value</i> = <i>app.SCPI.STATUS.QUESTIONable.NTRansition</i></p> <p><i>app.SCPI.STATUS.QUESTIONable.NTRansition</i> = <i>Value</i></p>
<i>Equivalent Softkeys</i>	None

SCPI.STATUS.QUESTIONable.PTRansition

<i>Object Type</i>	Property (read/write)
<i>Data Type</i>	Long
<i>Target</i>	IEE488.2 Status Reporting System (Appendix 1)
<i>Description</i>	Positive Transition Filter of the Questionable Status Register.
<i>Range</i>	from 0 to 65535
<i>Preset Value</i>	65535
<i>Syntax</i>	<p>Dim <i>Value</i> As Long <i>Value</i> = <i>app.SCPI.STATUS.QUESTIONable.PTRansition</i> <i>app.SCPI.STATUS.QUESTIONable.PTRansition</i> = <i>Value</i></p>
<i>Equivalent Softkeys</i>	None

SCPI.STATUS.QUESTIONable.RLIMit.CHANnel(*Ch*).CONDition

<i>Object Type</i>	Property (read only)
<i>Data Type</i>	Long
<i>Target</i>	IEE488.2 Status Reporting System (Appendix 1)
<i>Description</i>	Questionable Ripple Limit Channel Status Condition Register for channel <i>Ch</i> , <i>Ch</i> : channel number 1–16 (see Table 1 on page 25)
<i>Syntax</i>	<p>Dim <i>Value</i> As Long <i>Value</i> = <i>app.SCPI.STATUS.QUESTIONable.RLIMit.CHANnel(Ch).CONDition</i></p>
<i>Equivalent Softkeys</i>	None

SCPI.STATus.QUESTIONable.RLIMit.CHANnel(*Ch*).ENABLE

<i>Object Type</i>	Property (read/write)
<i>Data Type</i>	Long
<i>Target</i>	IEE488.2 Status Reporting System (Appendix 1)
<i>Description</i>	Questionable Ripple Limit Channel Status Enable Register for channel <i>Ch</i> , <i>Ch</i> : channel number 1–16 (see Table 1 on page 25)
<i>Range</i>	from 0 to 65535
<i>Preset Value</i>	0
<i>Syntax</i>	<p>Dim <i>Value</i> As Long</p> <p><i>Value</i> = <i>app.SCPI.STATUS.QUESTIONable.RLIMit.CHANnel(Ch).ENABLE</i></p> <p><i>app.SCPI.STATUS.QUESTIONable.RLIMit.CHANnel(Ch).ENABLE</i> = <i>Value</i></p>
<i>Equivalent Softkeys</i>	None

SCPI.STATus.QUESTIONable.RLIMit.CHANnel(*Ch*).EVENT

<i>Object Type</i>	Property (read only)
<i>Data Type</i>	Long
<i>Target</i>	IEE488.2 Status Reporting System (Appendix 1)
<i>Description</i>	Questionable Ripple Limit Channel Status Event Register for channel <i>Ch</i> , <i>Ch</i> : channel number 1–16 (see Table 1 on page 25)
<i>Syntax</i>	<p>Dim <i>Value</i> As Long</p> <p><i>Value</i> = <i>app.SCPI.STATUS.QUESTIONable.RLIMit.CHANnel(Ch).EVENT</i></p>
<i>Equivalent Softkeys</i>	None

SCPI.STATus.QUESTIONable.RLIMit.CHANnel(*Ch*).NTRansition

<i>Object Type</i>	Property (read/write)
<i>Data Type</i>	Long
<i>Target</i>	IEE488.2 Status Reporting System (Appendix 1)
<i>Description</i>	Negative Transition Filter of the Questionable Ripple Limit Channel Status Register for channel <i>Ch</i> , <i>Ch</i> : channel number 1–16 (see Table 1 on page 25)
<i>Range</i>	from 0 to 65535
<i>Preset Value</i>	0
<i>Syntax</i>	<p>Dim <i>Value</i> As Long</p> <p><i>Value</i> = <i>app.SCPI.STATus.QUESTIONable.RLIMit.CHANnel(Ch).NTRansition</i></p> <p><i>app.SCPI.STATus.QUESTIONable.RLIMit.CHANnel(Ch).NTRansition</i> = <i>Value</i></p>
<i>Equivalent Softkeys</i>	None

SCPI.STATus.QUESTIONable.RLIMit.CHANnel(*Ch*).PTRansition

<i>Object Type</i>	Property (read/write)
<i>Data Type</i>	Long
<i>Target</i>	IEE488.2 Status Reporting System (Appendix 1)
<i>Description</i>	Positive Transition Filter of the Questionable Ripple Limit Channel Status Register for channel <i>Ch</i> , <i>Ch</i> : channel number 1–16 (see Table 1 on page 25)
<i>Range</i>	from 0 to 65535
<i>Preset Value</i>	65535
<i>Syntax</i>	<p>Dim <i>Value</i> As Long</p> <p><i>Value</i> = <i>app.SCPI.STATus.QUESTIONable.RLIMit.CHANnel(Ch).PTRansition</i></p> <p><i>app.SCPI.STATus.QUESTIONable.RLIMit.CHANnel(Ch).PTRansition</i> = <i>Value</i></p>
<i>Equivalent Softkeys</i>	None

SCPI.STATus.QUESTIONable.RLIMit.CONDition

<i>Object Type</i>	Property (read only)
<i>Data Type</i>	Long
<i>Target</i>	IEE488.2 Status Reporting System (Appendix 1)
<i>Description</i>	Questionable Ripple Limit Status Condition Register.
<i>Syntax</i>	<p>Dim <i>Value</i> As Long</p> <p><i>Value</i> = <i>app.SCPI.STATus.QUESTIONable.RLIMit.CONDition</i></p>
<i>Equivalent Softkeys</i>	None

SCPI.STATUs.QUESTIONable.RLIMit.ENABLE

<i>Object Type</i>	Property (read/write)
<i>Data Type</i>	Long
<i>Target</i>	IEE488.2 Status Reporting System (Appendix 1)
<i>Description</i>	Questionable Ripple Limit Status Enable Register.
<i>Range</i>	from 0 to 65535
<i>Preset Value</i>	0
<i>Syntax</i>	<p>Dim Value As Long</p> <p><i>Value</i> = <i>app.SCPI.STATUs.QUESTIONable.RLIMit.ENABLE</i></p> <p><i>app.SCPI.STATUs.QUESTIONable.RLIMit.ENABLE</i> = <i>Value</i></p>
<i>Equivalent Softkeys</i>	None

SCPI.STATUs.QUESTIONable.RLIMit.EVENT

<i>Object Type</i>	Property (read only)
<i>Data Type</i>	Long
<i>Target</i>	IEE488.2 Status Reporting System (Appendix 1)
<i>Description</i>	Questionable Ripple Limit Status Event Register.
<i>Syntax</i>	<p>Dim Value As Long</p> <p><i>Value</i> = <i>app.SCPI.STATUs.QUESTIONable.RLIMit.EVENT</i></p>
<i>Equivalent Softkeys</i>	None

SCPI.STATUS.QUESTIONable.RLIMit.NTRansition

<i>Object Type</i>	Property (read/write)
<i>Data Type</i>	Long
<i>Target</i>	IEE488.2 Status Reporting System (Appendix 1)
<i>Description</i>	Negative Transition Filter of the Questionable Ripple Limit Status Register.
<i>Range</i>	from 0 to 65535
<i>Preset Value</i>	0
<i>Syntax</i>	<p>Dim Value As Long</p> <p><i>Value</i> = <i>app</i>.SCPI.STATUS.QUESTIONable.RLIMit.NTRansition</p> <p><i>app</i>.SCPI.STATUS.QUESTIONable.RLIMit.NTRansition = <i>Value</i></p>
<i>Equivalent Softkeys</i>	None

SCPI.STATus.QUESTIONable.RLIMit.PTRansition

<i>Object Type</i>	Property (read/write)
<i>Data Type</i>	Long
<i>Target</i>	IEE488.2 Status Reporting System (Appendix 1)
<i>Description</i>	Positive Transition Filter of the Questionable Ripple Limit Status Register.
<i>Range</i>	from 0 to 65535
<i>Preset Value</i>	65535
<i>Syntax</i>	<p>Dim <i>Value</i> As Long</p> <p><i>Value</i> = <i>app.SCPI.STATus.QUESTIONable.RLIMit.PTRansition</i></p> <p><i>app.SCPI.STATus.QUESTIONable.RLIMit.PTRansition</i> = <i>Value</i></p>
<i>Equivalent Softkeys</i>	None

SCPI.SYSTem.BEEPer.COMplete.IMMEDIATE

<i>Object Type</i>	Method
<i>Target</i>	Analyzer
<i>Description</i>	Generates a beep to notify of the completion of the operation.
<i>Syntax</i>	<code>app.SCPI.SYSTem.BEEPer.COMplete.IMMEDIATE</code>
<i>Equivalent Softkeys</i>	System > Misc Setup > Beeper > Test Beep Complete

SCPI.SYSTem.BEEPer.COMplete.STATE

<i>Object Type</i>	Property (read/write)
<i>Data Type</i>	Boolean
<i>Target</i>	Analyzer
<i>Description</i>	The ON/OFF state of the beeper notifying of the completion of the operation.
<i>Allowable Values</i>	True: Completion beeper ON False: Completion beeper OFF
<i>Preset Value</i>	True
<i>Syntax</i>	<p>Dim Status As Boolean</p> <code>Status = app.SCPI.SYSTem.BEEPer.COMplete.STATE</code> <code>app.SCPI.SYSTem.BEEPer.COMplete.STATE = False</code>
<i>Equivalent Softkeys</i>	System > Misc Setup > Beeper > Beep complete

SCPI.SYSTem.BEEPer.WARNING.IMMEDIATE

<i>Object Type</i>	Method
<i>Target</i>	Analyzer
<i>Description</i>	Generates a beep to notify of warning.
<i>Syntax</i>	<code>app.SCPI.SYSTem.BEEPer.WARNING.IMMEDIATE</code>
<i>Equivalent Softkeys</i>	System > Misc Setup > Beeper > Test Beep Warning

SCPI.SYSTem.BEEPer.WARNING.STATE

<i>Object Type</i>	Property (read/write)
<i>Data Type</i>	Boolean
<i>Target</i>	Analyzer
<i>Description</i>	The ON/OFF state of the beeper notifying of warning.
<i>Allowable Values</i>	True: Warning beeper ON False: Warning beeper OFF
<i>Preset Value</i>	True
<i>Syntax</i>	<p>Dim Status As Boolean</p> <p><code>Status = app.SCPI.SYSTem.BEEPer.WARNING.STATE</code></p> <p><code>app.SCPI.SYSTem.BEEPer.WARNING.STATE = False</code></p>
<i>Equivalent Softkeys</i>	System > Misc Setup > Beeper > Beep Warning

SCPI.SYSTem.COMMunicate.ECAL.TEMPerature.SENSor

<i>Object Type</i>	Property (read only)
<i>Data Type</i>	Double
<i>Target</i>	AutoCal module
<i>Description</i>	Reads out the temperature of the AutoCal module connected to the analyzer.
<i>Unit</i>	°C (Celsius)
<i>Syntax</i>	<p>Dim Value As Double</p> <p>Value = app.SCPI.SYSTem.COMMunicate.ECAL.TEMPerature.SENSor</p>
<i>Equivalent Softkeys</i>	None

SCPI.SYSTem.COMMunicate.ECAL.IMPedance(*Pt*)

<i>Object Type</i>	Property (read/write)
<i>Data Type</i>	String
<i>Target</i>	AutoCal module
<i>Description</i>	Sets or reads out the impedance state of the specified port (<i>Pt</i>) of AutoCal module connected to the analyzer.
<i>Range</i>	"OPEN" : OPEN impedance state "SHORT" : SHORT impedance state "LOAD" : LOAD impedance state "LOAD2" : LOAD2 impedance state "OPEN2" : OPEN2 impedance state
<i>Preset Value</i>	"LOAD"
<i>Syntax</i>	Dim <i>Param</i> As String <i>Param</i> = <i>app</i> .SCPI.SYSTem.COMMunicate.ECAL.IMPedance(<i>Pt</i>) <i>app</i> .SCPI.SYSTem.COMMunicate.ECAL.IMPedance(<i>Pt</i>) = "OPEN"
<i>Equivalent Softkeys</i>	None

SCPI.SYSTem.COMMunicate.ECAL.CHECk()

<i>Object Type</i>	Method
<i>Target</i>	AutoCal module
<i>Description</i>	Sets the check state of AutoCal module connected to the analyzer.
<i>Syntax</i>	<i>app</i> .SCPI.SYSTem.COMMunicate.ECAL.CHECk()
<i>Equivalent Softkeys</i>	None

SCPI.SYSTem.COMMunicate.ECAL.THRU(*Pt1, Pt2*)

<i>Object Type</i>	Method
<i>Target</i>	AutoCal module
<i>Description</i>	Sets the thru state between the specified 2 ports (<i>Pt1, Ptr2</i>) of AutoCal module connected to the analyzer.
<i>Syntax</i>	<i>app.SCPI.SYSTem.COMMunicate.ECAL.THRU(1, 2)</i>
<i>Equivalent Softkeys</i>	None

SCPI.SYSTem.CORRection.STATe

<i>Object Type</i>	Property (read/write)
<i>Data Type</i>	Boolean
<i>Target</i>	Analyzer
<i>Description</i>	The ON/OFF state of the system error correction.
<i>Allowable Values</i>	True: System error correction ON False: System error correction OFF
<i>Preset Value</i>	True
<i>Syntax</i>	Dim <i>Status</i> As Boolean <i>Status</i> = <i>app.SCPI.SYSTem.CORRection.STATe</i> <i>app.SCPI.SYSTem.CORRection.STATe</i> = False
<i>Equivalent Softkeys</i>	System > Misc Setup > System Correction

SCPI.SYSTem.DATE

<i>Object Type</i>	Property (read/write)						
<i>Data Type</i>	Variant (array of long)						
<i>Target</i>	Analyzer						
<i>Description</i>	<p>The current date.</p> <p>The array consists of three elements:</p> <table style="margin-left: 40px;"> <tr> <td><i>Data(0)</i></td> <td>year from 1900 to 2100;</td> </tr> <tr> <td><i>Data(1)</i></td> <td>month from 1 to 12;</td> </tr> <tr> <td><i>Data(2)</i></td> <td>day from 1 to 31.</td> </tr> </table>	<i>Data(0)</i>	year from 1900 to 2100;	<i>Data(1)</i>	month from 1 to 12;	<i>Data(2)</i>	day from 1 to 31.
<i>Data(0)</i>	year from 1900 to 2100;						
<i>Data(1)</i>	month from 1 to 12;						
<i>Data(2)</i>	day from 1 to 31.						
<i>Syntax</i>	<pre>Dim Data As Variant Data = app.SCPI.SYSTem.DATE app.app.SCPI.SYSTem.DATE = Array(2009, 9, 9)</pre>						
<i>Equivalent Softkeys</i>	None						

SCPI.SYSTem.HIDE

<i>Object Type</i>	Method
<i>Target</i>	Analyzer GUI
<i>Description</i>	Minimizes the analyzer main window removing it from desktop.
<i>Syntax</i>	<code>app.SCPI.SYSTem.HIDE</code>
<i>Related Commands</i>	SCPI.SYSTem.SHOW
<i>Equivalent Softkeys</i>	None

SCPI.SYSTem.LOCal

<i>Object Type</i>	Method
<i>Target</i>	Analyzer
<i>Description</i>	Sets the analyzer to the local operation mode, when all the keys on the front panel, mouse and the touch screen are active.
<i>Syntax</i>	<code>app.SCPI.SYSTem.LOCal</code>
<i>Related Commands</i>	SCPI.SYSTem.REMote SCPI.SYSTem.RWLock
<i>Equivalent Softkeys</i>	None

SCPI.SYSTem.PRESet

<i>Object Type</i>	Method
<i>Target</i>	Analyzer
<i>Description</i>	Resets the analyzer to the factory settings. The difference from the SCPI.IEEE4882.RST: method is that the trigger is set to the <i>Continuous</i> trigger mode.
<i>Syntax</i>	<code>app.SCPI.SYSTem.PRESet</code>
<i>Equivalent Softkeys</i>	System > Preset > OK

SCPI.SYSTem.PORT.SWITchover.Delay.STATE

<i>Object Type</i>	Property (read/write)
<i>Data Type</i>	Boolean
<i>Target</i>	Analyzer
<i>Description</i>	<p>The ON/OFF state of the port switchover delay feature.</p> <p>Turn off the port switchover delay allows decreasing sweep time approximately by 10 ms providing that stimulus direction is changing.</p>
<i>Allowable Values</i>	<p>True: port switchover delay feature ON</p> <p>False: port switchover delay feature OFF</p>
<i>Preset Value</i>	True
<i>Syntax</i>	<p>Dim Status As Boolean</p> <p><i>Status</i> = app.SCPI.SYSTem.PORT.SWITchover.Delay.STATE</p> <p>app.SCPI.SYSTem.PORT.SWITchover.Delay.STATE = False</p>
<i>Equivalent Softkeys</i>	System > Misc Setup > Port Switchover Delay { On OFF }

SCPI.SYSTem.REMote

<i>Object Type</i>	Method
<i>Target</i>	Analyzer
<i>Description</i>	Sets the analyzer to the remote operation mode, when all the keys on the front panel, mouse and the touch screen are not active, except for one key labeled <i>Return to Local</i> . Pushing this button will reset the analyzer to the local operation mode.
<i>Syntax</i>	<code>app.SCPI.SYSTem.REMote</code>
<i>Related Commands</i>	SCPI.SYSTem.LOCal SCPI.SYSTem.RWLock
<i>Equivalent Softkeys</i>	None

SCPI.SYSTem.RWLock

<i>Object Type</i>	Method
<i>Target</i>	Analyzer
<i>Description</i>	Sets the analyzer to the remote operation mode, when all the keys on the front panel, mouse and the touch screen are not active. Only SCPI.SYSTem.LOCal or SCPI.SYSTem.REMote command can release this remote operation mode.
<i>Syntax</i>	<code>app.SCPI.SYSTem.RWLock</code>
<i>Related Commands</i>	SCPI.SYSTem.LOCal SCPI.SYSTem.REMote
<i>Equivalent Softkeys</i>	None

SCPI.SYSTem.TEMPerature.SENSor(*Idx*)

<i>Object Type</i>	Property (read only)
<i>Data Type</i>	Double
<i>Target</i>	Analyzer
<i>Description</i>	Reads out the specified sensor (<i>Idx</i>) temperature inside the analyzer.
<i>Unit</i>	°C (Celsius)
<i>Syntax</i>	<pre>Dim Value As Double Value = app.SCPI.SYSTem.TEMPerature.SENSor(1)</pre>
<i>Equivalent Softkeys</i>	None

SCPI.SYSTem.SHOW

<i>Object Type</i>	Method
<i>Target</i>	Analyzer GUI
<i>Description</i>	Restores the analyzer main window hidden by the SCPI.SYSTem.HIDE command.
<i>Syntax</i>	<code>app.SCPI.SYSTem.SHOW</code>
<i>Related Commands</i>	SCPI.SYSTem.HIDE
<i>Equivalent Softkeys</i>	None

SCPI.SYSTem.TERMinate

<i>Object Type</i>	Method
<i>Target</i>	Analyzer GUI
<i>Description</i>	Terminates the application.
<i>Syntax</i>	<code>app.SCPI.SYSTem.TERMinate</code>
<i>Equivalent Softkeys</i>	None

SCPI.SYSTem.TIME

<i>Object Type</i>	Property (read/write)						
<i>Data Type</i>	Variant (array of long)						
<i>Target</i>	Analyzer						
<i>Description</i>	<p>The current time. The array consists of three elements:</p> <table style="margin-left: 40px;"> <tr> <td><i>Data(0)</i></td> <td>hours from 0 to 23;</td> </tr> <tr> <td><i>Data(1)</i></td> <td>minutes from 0 to 59;</td> </tr> <tr> <td><i>Data(2)</i></td> <td>seconds from 0 to 59.</td> </tr> </table>	<i>Data(0)</i>	hours from 0 to 23;	<i>Data(1)</i>	minutes from 0 to 59;	<i>Data(2)</i>	seconds from 0 to 59.
<i>Data(0)</i>	hours from 0 to 23;						
<i>Data(1)</i>	minutes from 0 to 59;						
<i>Data(2)</i>	seconds from 0 to 59.						
<i>Syntax</i>	<pre>Dim Data As Variant Data = app.SCPI.SYSTem.TIME app.app.SCPI.SYSTem.TIME = Array(15, 20, 30)</pre>						
<i>Equivalent Softkeys</i>	None						

SCPI.TRIGger.SEQuence.EXternal.Delay

<i>Object Type</i>	Property (read/write)
<i>Data Type</i>	Double
<i>Target</i>	External trigger
<i>Description</i>	Sets the response delay with respect to the external trigger signal.
<i>Range</i>	from 0 to 100 sec
<i>Resolution</i>	0.1 μ sec
<i>Out of Range</i>	Sets to the nearest bound.
<i>Preset Value</i>	0
<i>Syntax</i>	<p>Dim <i>Param</i> As Double <i>Param</i> = <i>app</i>.SCPI.TRIGger.EXternal.Delay <i>app</i>.SCPI.TRIGger.INPut.EXternal.Delay = 0</p>
<i>Related Commands</i>	SCPI.TRIGger.SEQuence.SOURce
<i>Equivalent Softkeys</i>	Stimulus > Trigger > Ext Trigger > Delay

SCPI.TRIGger.SEQuence.EXternal.POSition

<i>Object Type</i>	Property (read/write)
<i>Data Type</i>	String
<i>Target</i>	External trigger
<i>Description</i>	<p>Selects the position of the external trigger. The Analyzer waits for external trigger:</p> <ul style="list-style-type: none"> • Before sampling, when the frequency of the stimulus port have been set. The frequency change of the stimulus port begins after sampling. • Before the frequency setup and subsequent measurement. The frequency change of the stimulus port begins when the external trigger arrives. <p>Depending on the command SCPI.TRIGger.SEQuence.POInt the external trigger wait occurs before each point or before the first point of the full sweep cycle.</p>
<i>Range</i>	<p>"BSAM" : Before sampling</p> <p>"BSET" : Before frequency setup</p>
<i>Out of Range</i>	An error occurs. Error code: 205.
<i>Preset Value</i>	"BSAM"
<i>Syntax</i>	<pre>Dim Param As String Param = app.SCPI.TRIGger.EXTERNAL.POSITION app.SCPI.TRIGger.INPUT.EXTERNAL.POSITION = "BSAM"</pre>
<i>Related Commands</i>	SCPI.TRIGger.SEQuence.SOURCE
<i>Equivalent Softkeys</i>	Stimulus > Trigger > Ext Trigger > POSition > { Before Samp Before Setup }

SCPI.TRIGger.SEQuence.EXternal.SLOPe

<i>Object Type</i>	Property (read/write)
<i>Data Type</i>	String
<i>Target</i>	External trigger
<i>Description</i>	Selects the edge of the external trigger.
<i>Range</i>	"POPositive" : Positive edge "NEGative" : Negative edge
<i>Notes</i>	The short form of the parameter is indicated by upper case letters. There is no distinction between upper and lower case letters when the property is written. When the property is read out, the short format is indicated by upper case letters.
<i>Out of Range</i>	An error occurs. Error code: 205.
<i>Preset Value</i>	"NEG"
<i>Syntax</i>	Dim Param As String <i>Param</i> = app.SCPI.TRIGger.EXTERNAL.SLOPe app.SCPI.TRIGger.INPUT.EXTERNAL.SLOPe = "POS"
<i>Related Commands</i>	SCPI.TRIGger.SEQuence.SOURce
<i>Equivalent Softkeys</i>	Stimulus > Trigger > Ext Trigger > Polarity

SCPI.TRIGger.SEQuence.IMMEDIATE

<i>Object Type</i>	Method
<i>Target</i>	Analyzer
<i>Description</i>	<p>Generates a trigger, independently of the trigger source setting (except for the <i>External</i>). If the trigger source is set to <i>External</i>, an error occurs (error code 221) and the command is ignored.</p> <p>If the analyzer is not in the waiting for a trigger state (sweep is in progress or all the channels are set to <i>Hold</i>), an error occurs (error code 211) and the command is ignored.</p> <p>The method returns control before the end of the sweep.</p>
<i>Syntax</i>	<code>app.SCPI.TRIGger.SEQuence.IMMEDIATE</code>
<i>Related Commands</i>	SCPI.TRIGger.SEQuence.SOURce SCPI.INITiate(Ch).CONTinuous SCPI.INITiate(Ch).IMMEDIATE
<i>Equivalent Softkeys</i>	None

SCPI.TRIGger.SEQuence.POINt

<i>Object Type</i>	Property (read/write)
<i>Data Type</i>	Boolean
<i>Target</i>	Analyzer
<i>Description</i>	<p>The ON/OFF state of the point trigger feature.</p> <p>When the point trigger feature is turned ON, the external trigger response is single point. When the point trigger feature is turned OFF, the external trigger response is single sweep.</p>
<i>Allowable Values</i>	True: point trigger feature ON False: point trigger feature OFF
<i>Preset Value</i>	False
<i>Syntax</i>	<p>Dim <i>Status</i> As Boolean</p> <p><i>Status</i> = <i>app.SCPI.TRIGger.SEQuence.POINt</i></p> <p><i>app.SCPI.TRIGger.SEQuence.POINt</i> = True</p>
<i>Related Commands</i>	SCPI.TRIGger.SEQuence.SOURce
<i>Equivalent Softkeys</i>	Stimulus > Trigger > Ext Trig Event > { On Sweep On Point }

SCPI.TRIGger.SEQuence.SINGle

<i>Object Type</i>	Method
<i>Target</i>	Analyzer
<i>Description</i>	<p>Generates a trigger, independently of the trigger source setting (except for the <i>External</i>). If the trigger source is set to <i>External</i>, an error occurs (error code 221) and the command is ignored.</p> <p>If the analyzer is not in the waiting for a trigger state (sweep is in progress or all the channels are set to <i>Hold</i>), an error occurs (error code 211) and the command is ignored.</p> <p>The method does not return control before the end of the sweep (waiting for the completion of the sweep of all the channels).</p>
<i>Syntax</i>	<code>app.SCPI.TRIGger.SEQuence.SINGle</code>
<i>Related Commands</i>	SCPI.TRIGger.SEQuence.SOURce SCPI.INITiate(Ch).CONTinuous SCPI.INITiate(Ch).IMMEDIATE
<i>Equivalent Softkeys</i>	None

SCPI.TRIGger.SEQuence.SCOPe

<i>Object Type</i>	Property (read/write)
<i>Data Type</i>	String
<i>Target</i>	Analyzer
<i>Description</i>	Selects the trigger scope.
<i>Range</i>	"ALL" : All Channels "ACTive" : Active Channel
<i>Notes</i>	The short form of the parameter is indicated by upper case letters. There is no distinction between upper and lower case letters when the property is written. When the property is read out, the short format is indicated by upper case letters.
<i>Out of Range</i>	An error occurs. Error code: 205.
<i>Preset Value</i>	"ALL"
<i>Syntax</i>	Dim Param As String <i>Param</i> = app.SCPI.TRIGger.SEQuence.SCOPe app.SCPI.TRIGger.SEQuence.SCOPe = "ACT"
<i>Related Commands</i>	SCPI.TRIGger.SEQuence.IMMediate SCPI.TRIGger.SEQuence.SINGle SCPI.IEEE4882.TRG
<i>Equivalent Softkeys</i>	Stimulus > Trigger > Trigger Scope > { All Channels Active Channel }

SCPI.TRIGger.SEQuence.SOURce

<i>Object Type</i>	Property (read/write)
<i>Data Type</i>	String
<i>Target</i>	Analyzer
<i>Description</i>	Selects the sweep trigger source.
<i>Range</i>	"INTernal" : Internal "EXTernal" : External "MANual" : Manual "BUS" : Bus
<i>Notes</i>	The short form of the parameter is indicated by upper case letters. There is no distinction between upper and lower case letters when the property is written. When the property is read out, the short format is indicated by upper case letters.
<i>Out of Range</i>	An error occurs. Error code: 205.
<i>Preset Value</i>	"INT"
<i>Syntax</i>	<pre>Dim Param As String Param = app.SCPI.TRIGger.SEQuence.SOURce app.SCPI.TRIGger.SEQuence.SOURce = "BUS"</pre>
<i>Related Commands</i>	SCPI.TRIGger.SEQuence.IMMEDIATE SCPI.TRIGger.SEQuence.SINGLE SCPI.IEEE4882.TRG
<i>Equivalent Softkeys</i>	Stimulus > Trigger > Trigger Source > { Internal External Manual Bus }

SCPI.TRIGger.SEQuence.STATUS

<i>Object Type</i>	Property (read only)
<i>Data Type</i>	String
<i>Target</i>	Analyzer
<i>Description</i>	Returns the trigger system status of the analyzer.
<i>Return value</i>	"HOLD" : Hold "WAIT" : Wait for Trigger "MEAS" : Measurement
<i>Syntax</i>	Dim <i>Param</i> As String <i>Param</i> = <i>app</i> .SCPI.TRIGger.SEQuence.STATUS
<i>Equivalent Softkeys</i>	None

SCPI.TRIGger.SEQuence.WAIT(STATUS)

<i>Object Type</i>	Method
<i>Target</i>	Analyzer
<i>Description</i>	Waits for the specified status of the trigger system. Blocks the programm execution until the specified status of the analyzer has been reached.
<i>Parameter</i>	<i>Status</i> of String type: "HOLD" : Hold "WAIT" : Wait for Trigger "MEAS" : Measurement
<i>Syntax</i>	<i>app</i> .SCPI.TRIGger.SEQuence.WAIT("HOLD")
<i>Equivalent Softkeys</i>	None

SCPI.TRIGger.OUTPUT.FUNCTION

<i>Object Type</i>	Property (read/write)
<i>Data Type</i>	String
<i>Target</i>	Trigger output
<i>Description</i>	Selects the trigger output function. The function defines the signal shape at the trigger output.
<i>Range</i>	<p>"BSET" : Before frequency setup pulse "BSAM" : Before sampling pulse "ASAM" : After sampling pulse "RTRG" : Ready for trigger signal "ESWP" : End of sweep pulse "MEAS" : Measurement sweep signal</p>
<i>Out of Range</i>	An error occurs. Error code: 205.
<i>Preset Value</i>	"RTRG"
<i>Syntax</i>	<pre>Dim Param As String Param = app.SCPI.TRIGger.OUTPUT.FUNCTION app.SCPI.TRIGger.INPUT.OUTPUT.FUNCTION = "ESWP"</pre>
<i>Related Commands</i>	SCPI.TRIGger.OUTPUT.POLarity SCPI.TRIGger.OUTPUT.STATE
<i>Equivalent Softkeys</i>	Stimulus > Trigger > Trigger Output > Function > { Before Setup Before Sampling After Sampling Ready for Trig Sweep End Measurement }

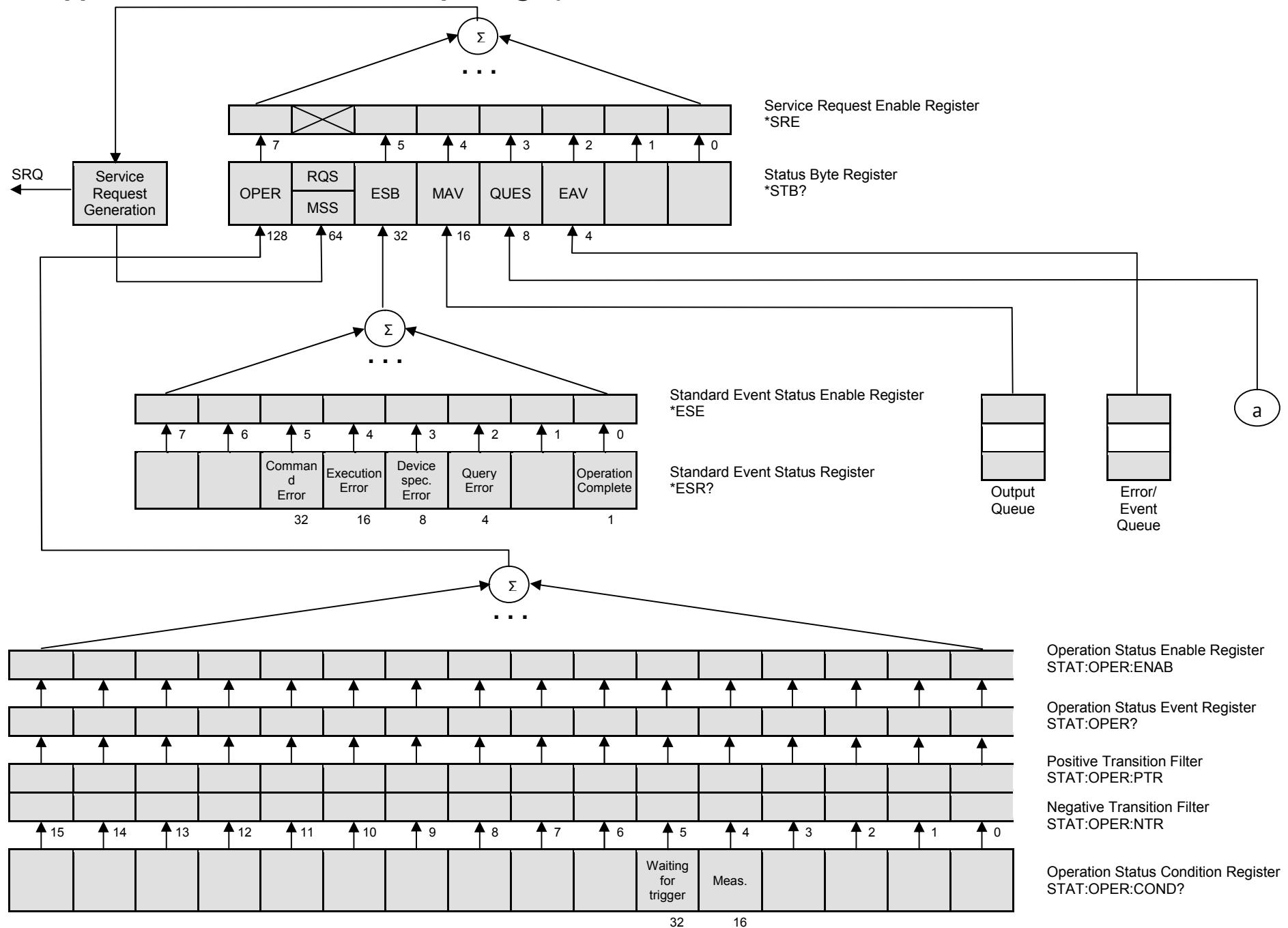
SCPI.TRIGger.OUTPut.POLarity

<i>Object Type</i>	Property (read/write)
<i>Data Type</i>	String
<i>Target</i>	Trigger output
<i>Description</i>	Selects the polarity of the trigger output function set by the SCPI.TRIGger.OUTPut.FUNCTion command.
<i>Range</i>	"POSitive" : Positive polarity "NEGative" : Negative polarity
<i>Notes</i>	The short form of the parameter is indicated by upper case letters. There is no distinction between upper and lower case letters when the property is written. When the property is read out, the short format is indicated by upper case letters.
<i>Out of Range</i>	An error occurs. Error code: 205.
<i>Preset Value</i>	"NEG"
<i>Syntax</i>	Dim <i>Param</i> As String <i>Param</i> = <i>app</i> .SCPI.TRIGger.OUTPut.POLarity <i>app</i> .SCPI.TRIGger.INPut.OUTPut.POLarity = "NEG"
<i>Related Commands</i>	SCPI.TRIGger.OUTPut.FUNCTion SCPI.TRIGger.OUTPut.STATE
<i>Equivalent Softkeys</i>	Stimulus > Trigger > Trigger Output > Polarity

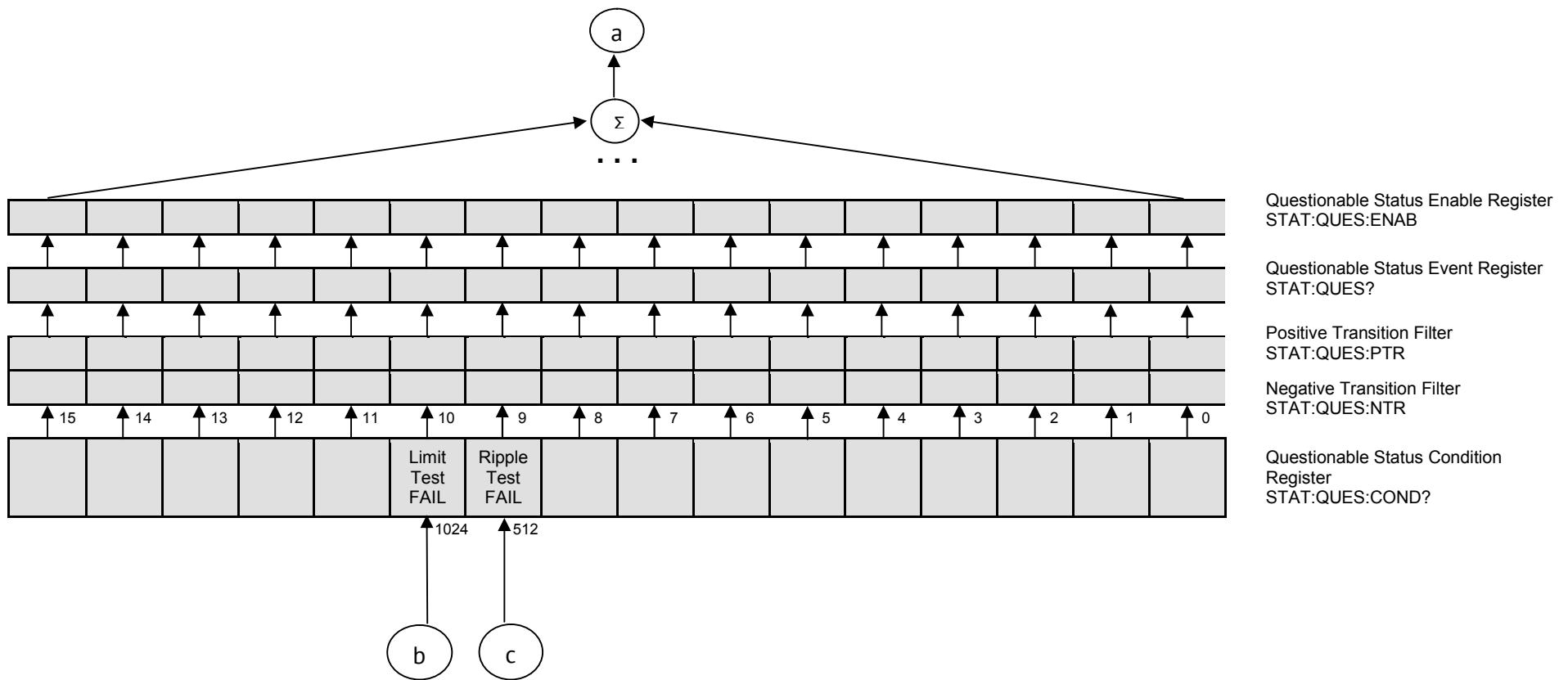
SCPI.TRIGger.OUTPUT.STATE

<i>Object Type</i>	Property (read/write)
<i>Data Type</i>	Boolean
<i>Target</i>	Trigger output
<i>Description</i>	The ON/OFF state of the trigger output.
<i>Allowable Values</i>	True: trigger output ON False: trigger output OFF
<i>Preset Value</i>	False
<i>Syntax</i>	<p>Dim <i>Status</i> As Boolean</p> <p><i>Status</i> = <i>app.SCPI.TRIGger.OUTPUT.STATE</i></p> <p><i>app.SCPI.TRIGger.OUTPUT.STATE</i> = True</p>
<i>Related Commands</i>	SCPI.TRIGger.OUTPUT.FUNCTION SCPI.TRIGger.OUTPUT.POLarity
<i>Equivalent Softkeys</i>	Stimulus > Trigger > Trigger Output > Trigger Output

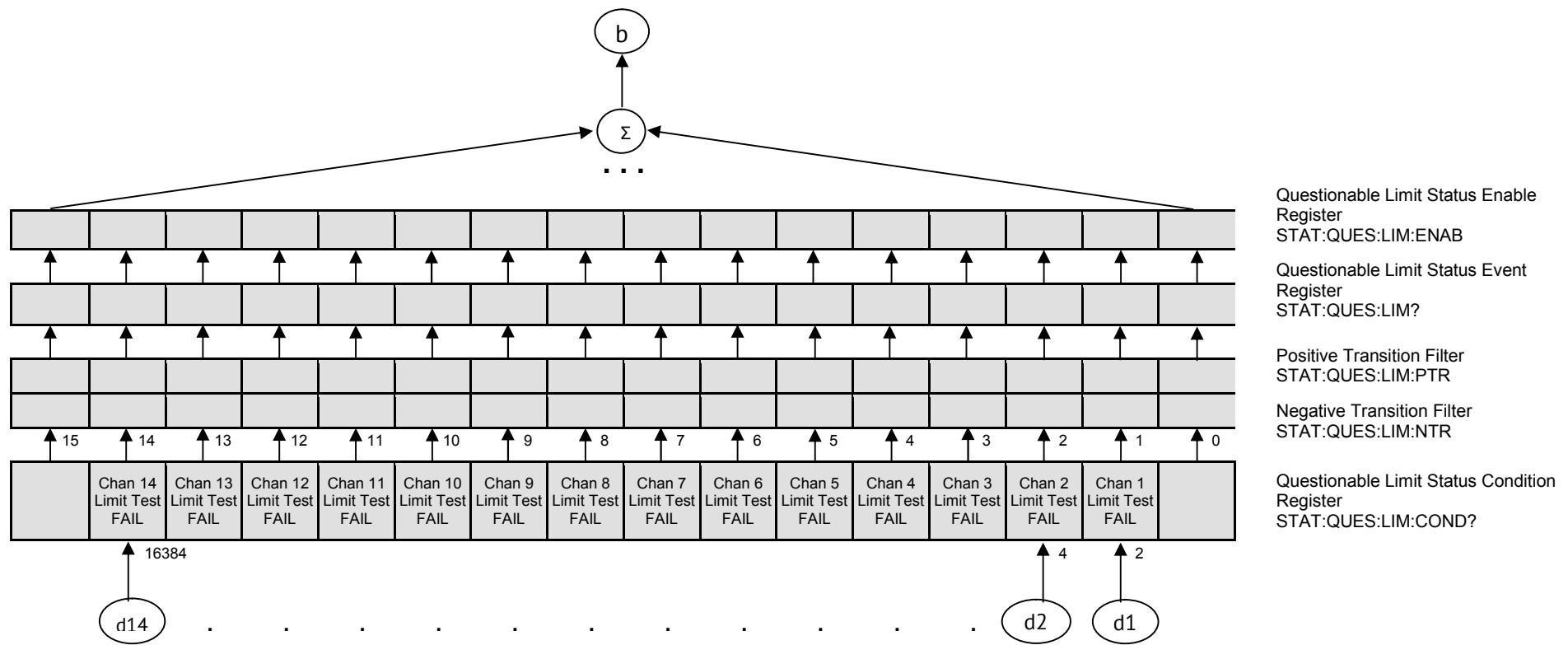
16 Appendix 1. IEE488.2 Status Reporting System



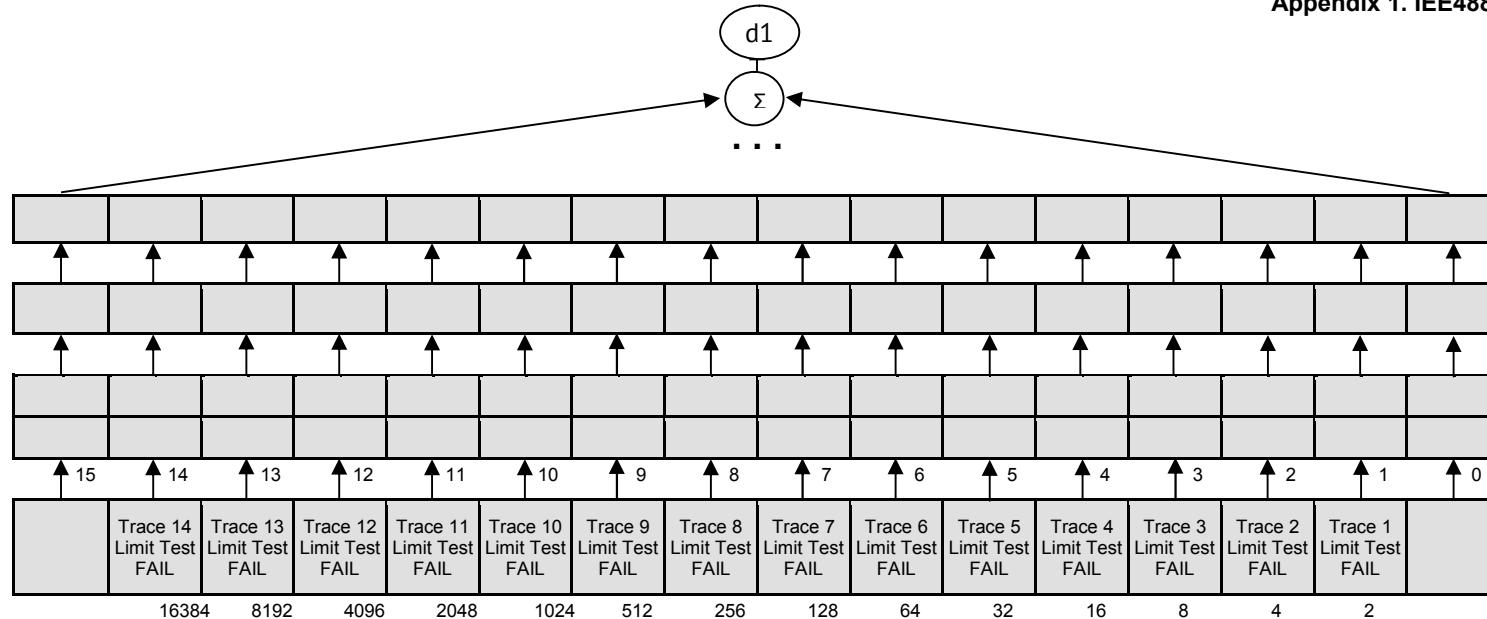
Appendix 1. IEE488.2 Status Reporting System



Appendix 1. IEE488.2 Status Reporting System



Appendix 1. IEE488.2 Status Reporting System



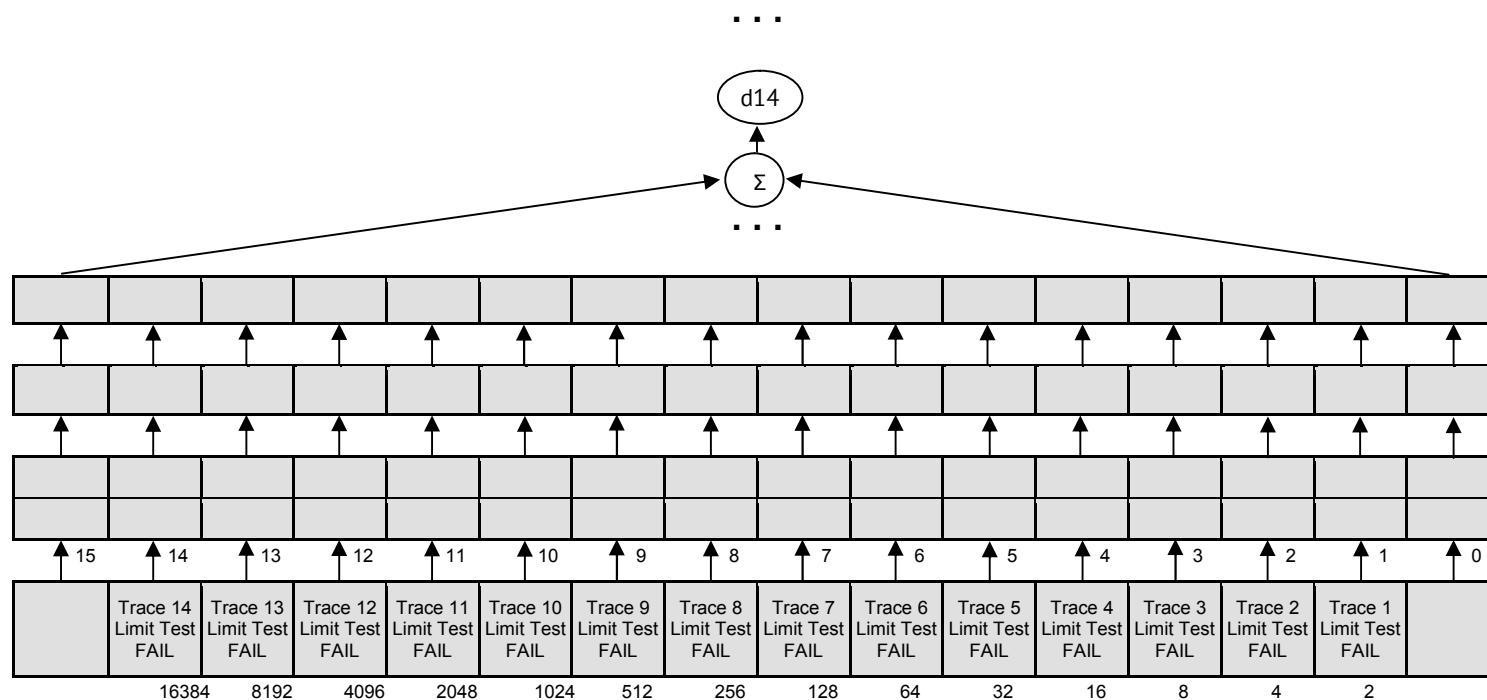
Questionable Limit Channel 1 Status
Enable Register
STAT:QUES:LIM:CHAN1:ENAB

Questionable Limit Channel 1 Status
Event Register
STAT:QUES:CHAN1:LIM?

Positive Transition Filter
STAT:QUES:LIM:CHAN1:PTR

Negative Transition Filter
STAT:QUES:LIM:CHAN1:NTR

Questionable Limit Channel 1 Status
Condition Register
STAT:QUES:LIM:CHAN1:COND?



Questionable Limit Channel 14 Status
Enable Register
STAT:QUES:LIM:CHAN14:ENAB

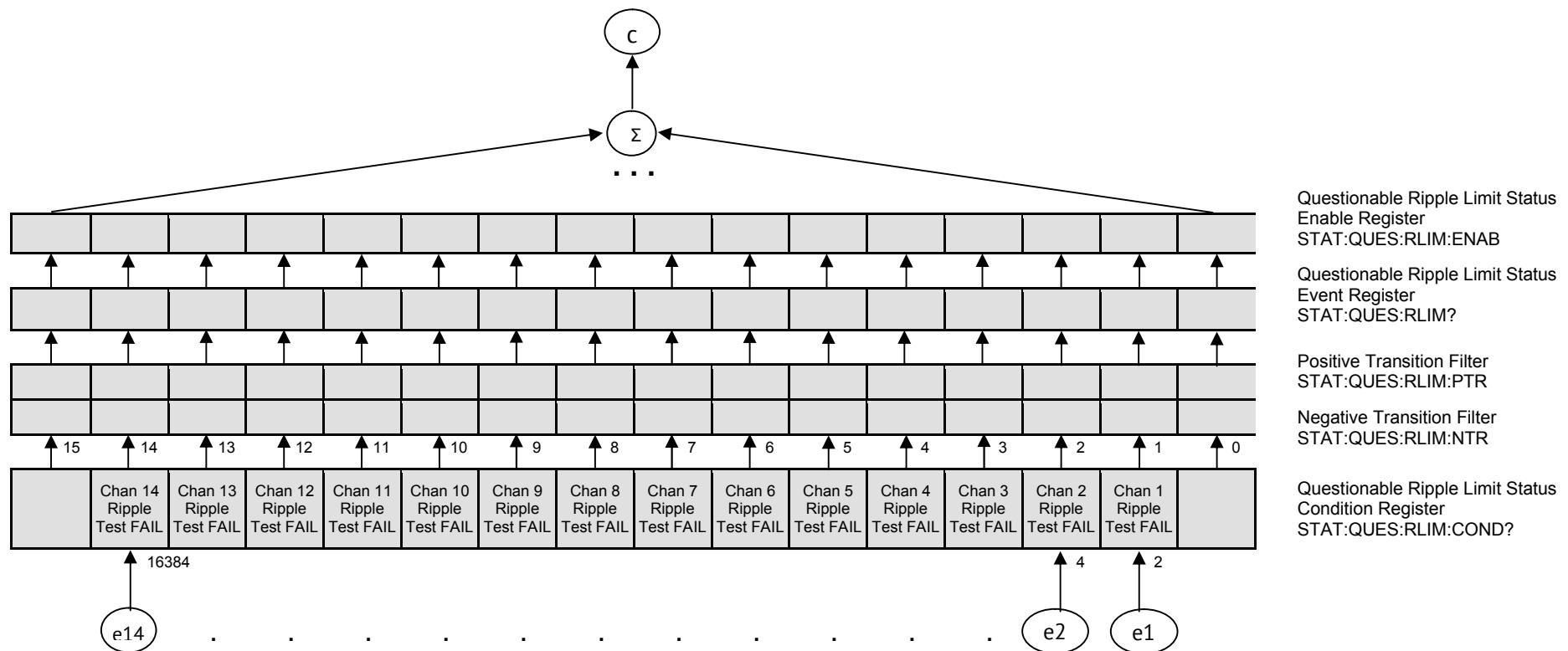
Questionable Limit Channel 14 Status
Event Register
STAT:QUES:CHAN14:LIM?

Positive Transition Filter
STAT:QUES:LIM:CHAN14:PTR

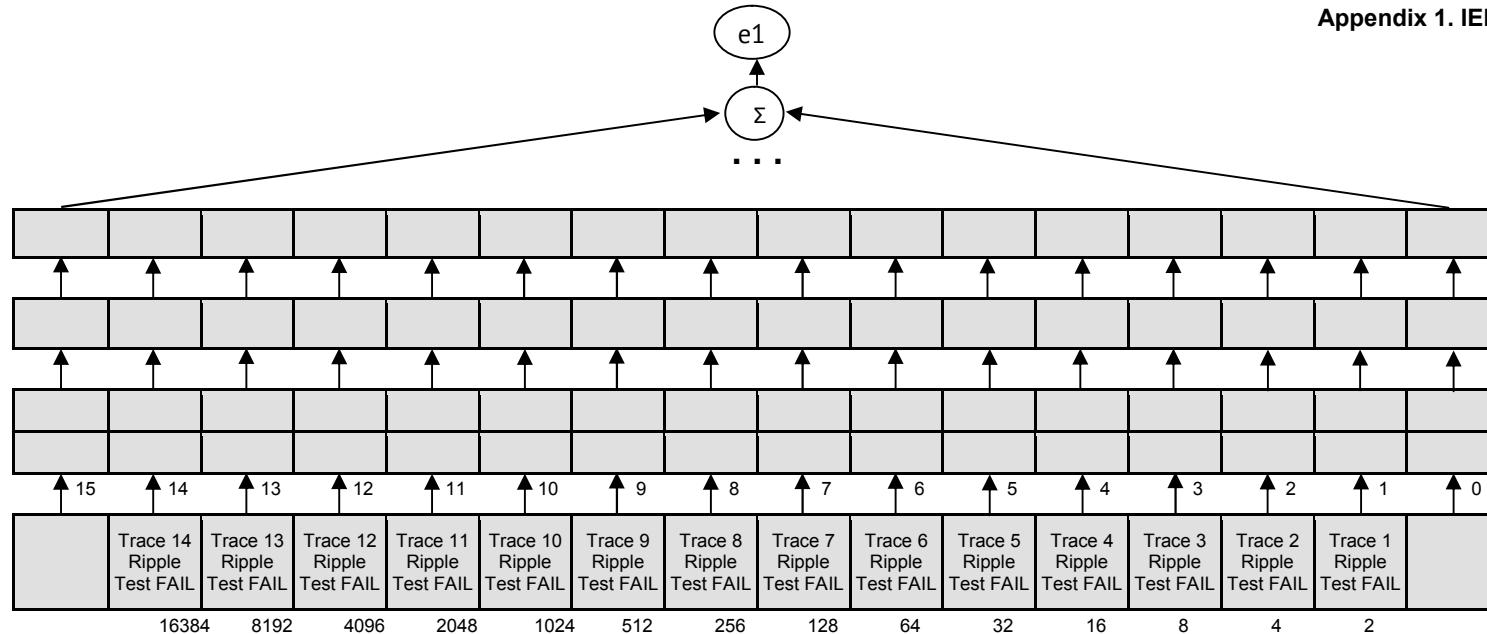
Negative Transition Filter
STAT:QUES:LIM:CHAN14:NTR

Questionable Limit Channel 14 Status
Condition Register
STAT:QUES:LIM:CHAN14:COND?

Appendix 1. IEE488.2 Status Reporting System



Appendix 1. IEE488.2 Status Reporting System



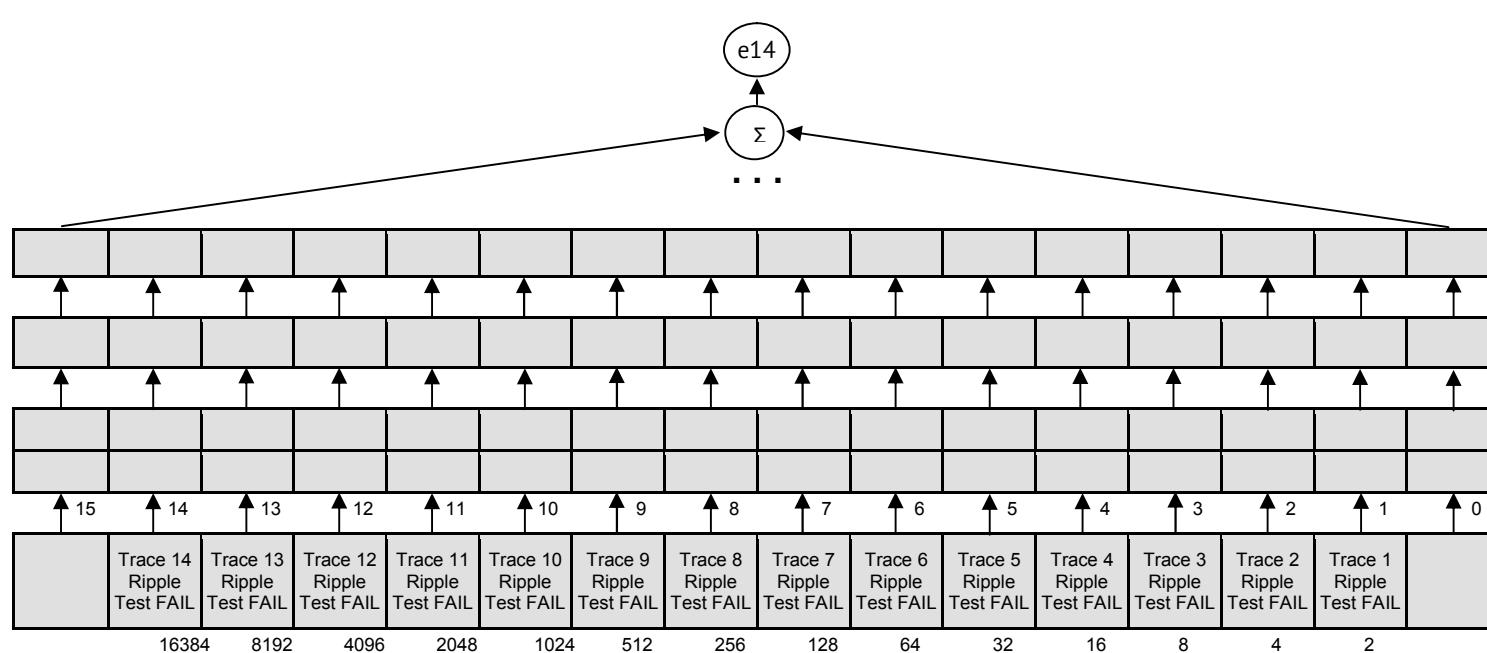
Questionable Ripple Limit Channel 1 Status Enable Register
STAT:QUES:RLIM:CHAN1:ENAB

Questionable Ripple Limit Channel 1 Status Event Register
STAT:QUES:CHAN1:RLIM?

Positive Transition Filter
STAT:QUES:RLIM:CHAN1:PTR

Negative Transition Filter
STAT:QUES:RLIM:CHAN1:NTR

Questionable Ripple Limit Channel 1 Status Condition Register
STAT:QUES:RLIM:CHAN1:COND?



Questionable Ripple Limit Channel 14 Status Enable Register
STAT:QUES:RLIM:CHAN14:ENAB

Questionable Ripple Limit Channel 14 Status Event Register
STAT:QUES:CHAN14:RLIM?

Positive Transition Filter
STAT:QUES:RLIM:CHAN14:PTR

Negative Transition Filter
STAT:QUES:RLIM:CHAN14:NTR

Questionable Ripple Limit Channel 14 Status Condition Register
STAT:QUES:RLIM:CHAN14:COND?

17 Appendix 2. Error Codes

114	"Header suffix out of range"
200	"Execution error"
211	"Trigger ignored"
213	"Init ignored"
220	"Parameter Error"
222	"Data out of range"
224	"Illegal parameter value"
201	"Invalid channel index"
202	"Invalid trace index"
203	"Invalid marker index"
204	"Marker is not active"
205	"Invalid save type specifier"
206	"Invalid sweep type specifier"
207	"Invalid trigger source specifier"
208	"Invalid measurement parameter specifier"
209	"Invalid format specifier"
210	"Invalid data math specifier"
214	"Invalid limit data"
215	"Invalid segment data"
216	"Invalid standard type specifier"
217	"Invalid conversion specifier"
218	"Invalid gating shape specifier"
219	"Invalid gating type specifier"
300	"Device-specific error"
302	"Status reporting system error"

