



COPPER MOUNTAIN
TECHNOLOGIES

S2 and S4 series Network Analyzer

COM/DCOM Programming Manual



Software version 19.3
2019

TABLE OF CONTENTS

1 Introduction.....	11
2 Scope of Manual	11
3 Registering COM Server	12
4 COM Technology Overview	12
5 Automation Server	12
6 Automation Controllers	13
7 Local and Remote Server	13
8 Structure of COM Objects	15
9 Accessing the Application Object	16
10 Object Methods.....	19
11 Object Properties.....	19
12 Error Handling	20
13 COM Automation Data Types	22
14 Measurement Data Arrays	23
15 COM Server Commands	24
NAME	24
READY	24
SCPI.ABORT	25
SCPI.CALCulate(<i>Ch</i>).FSIMulator.BALun.CZConversion.STATe	25
SCPI.CALCulate(<i>Ch</i>).FSIMulator.BALun.CZConversion.BPORT(<i>Bpt</i>).Z0.R	26
SCPI.CALCulate(<i>Ch</i>).FSIMulator.BALun.DEVice	27
SCPI.CALCulate(<i>Ch</i>).FSIMulator.BALun.DMCircuit.BPORT(<i>Bpt</i>).PARAMeters.C	28
SCPI.CALCulate(<i>Ch</i>).FSIMulator.BALun.DMCircuit.BPORT(<i>Bpt</i>).PARAMeters.G	29
SCPI.CALCulate(<i>Ch</i>).FSIMulator.BALun.DMCircuit.BPORT(<i>Bpt</i>).PARAMeters.L	30
SCPI.CALCulate(<i>Ch</i>).FSIMulator.BALun.DMCircuit.BPORT(<i>Bpt</i>).PARAMeters.R	31
SCPI.CALCulate(<i>Ch</i>).FSIMulator.BALun.DMCircuit.BPORT(<i>Bpt</i>).TYPE	32
SCPI.CALCulate(<i>Ch</i>).FSIMulator.BALun.DMCircuit.BPORT(<i>Bpt</i>).USER.FILEname	33
SCPI.CALCulate(<i>Ch</i>).FSIMulator.BALun.DMCircuit.STATe	34
SCPI.CALCulate(<i>Ch</i>).FSIMulator.BALun.DZConversion.STATe	34
SCPI.CALCulate(<i>Ch</i>).FSIMulator.BALun.DZConversion.BPORT(<i>Bpt</i>).Z0.R	35
SCPI.CALCulate(<i>Ch</i>).FSIMulator.BALun.PARAmeter(<i>Tr</i>).BALanced.DEFine	36
SCPI.CALCulate(<i>Ch</i>).FSIMulator.BALun.PARAmeter(<i>Tr</i>).BBALanced.DEFine	37
SCPI.CALCulate(<i>Ch</i>).FSIMulator.BALun.PARAmeter(<i>Tr</i>).SBALanced.DEFine	38
SCPI.CALCulate(<i>Ch</i>).FSIMulator.BALun.PARAmeter(<i>Tr</i>).SSBalanced.DEFine	39
SCPI.CALCulate(<i>Ch</i>).FSIMulator.BALun.PARAmeter(<i>Tr</i>).STATe	40
SCPI.CALCulate(<i>Ch</i>).FSIMulator.BALun.TOPology.PROPErty.STATe	41
SCPI.CALCulate(<i>Ch</i>).FSIMulator.BALun.TOPology.BALanced.PPORTs	42
SCPI.CALCulate(<i>Ch</i>).FSIMulator.BALun.TOPology.BBALanced.PPORTs.....	43
SCPI.CALCulate(<i>Ch</i>).FSIMulator.BALun.TOPology.SBALanced.PPORTs.....	44
SCPI.CALCulate(<i>Ch</i>).FSIMulator.BALun.TOPology.SSBalanced.PPORTs.....	45
SCPI.CALCulate(<i>Ch</i>).FSIMulator.EMBed.NETWork(<i>Nwk</i>).FILEname	46
SCPI.CALCulate(<i>Ch</i>).FSIMulator.EMBed.NETWork(<i>Nwk</i>).TYPE.....	47
SCPI.CALCulate(<i>Ch</i>).FSIMulator.EMBed.STATe	48
SCPI.CALCulate(<i>Ch</i>).FSIMulator.EMBed.TOPology.A.PORTs.....	49

SCPI.CALCulate(Ch).FSIMulator.EMBed.TOPology.B.PORTs	50
SCPI.CALCulate(Ch).FSIMulator.EMBed.TOPology.C.PORTs	51
SCPI.CALCulate(Ch).FSIMulator.EMBed.TYPE	52
SCPI.CALCulate(Ch).FSIMulator.SENDEd.DEEMbed.PORT(<i>Pt</i>).USER.FILEname	53
SCPI.CALCulate(Ch).FSIMulator.SENDEd.DEEMbed.PORT(<i>Pt</i>).STATE	54
SCPI.CALCulate(Ch).FSIMulator.SENDEd.DEEMbed.STATE	54
SCPI.CALCulate(Ch).FSIMulator.SENDEd.PMCircuit.PORT(<i>Pt</i>).USER.FILEname	55
SCPI.CALCulate(Ch).FSIMulator.SENDEd.PMCircuit.PORT(<i>Pt</i>).STATE	56
SCPI.CALCulate(Ch).FSIMulator.SENDEd.PMCircuit.STATE	56
SCPI.CALCulate(Ch).FSIMulator.SENDEd.ZCONversion.PORT(<i>Pt</i>).Z0.R	57
SCPI.CALCulate(Ch).FSIMulator.SENDEd.ZCONversion.STATE	58
SCPI.CALCulate(Ch).FSIMulator.STATE	58
SCPI.CALCulate(Ch).PARAmeter(1).COUNT	59
SCPI.CALCulate(Ch).PARAmeter(<i>Tr</i>).DEFine	60
SCPI.CALCulate(Ch).PARAmeter(<i>Tr</i>).SELEct	61
SCPI.CALCulate(Ch).PARAmeter(<i>Tr</i>).SPORT	62
SCPI.CALCulate(Ch).SELEcted.CONVersion.FUNcTION	63
SCPI.CALCulate(Ch).SELEcted.CONVersion.STATE	64
SCPI.CALCulate(Ch).SELEcted.CORRection.EDElay.TIME	65
SCPI.CALCulate(Ch).SELEcted.CORRection.OFFSet.PHASE	66
SCPI.CALCulate(Ch).SELEcted.CORRection.STATUS	67
SCPI.CALCulate(Ch).SELEcted.DATA.FDATA	68
SCPI.CALCulate(Ch).SELEcted.DATA.FMEMORY	69
SCPI.CALCulate(Ch).SELEcted.DATA.SDATA	70
SCPI.CALCulate(Ch).SELEcted.DATA.SMEMORY	71
SCPI.CALCulate(Ch).SELEcted.DATA.XAXis	71
SCPI.CALCulate(Ch).SELEcted.FILTer.GATE.TIME.CENter	72
SCPI.CALCulate(Ch).SELEcted.FILTer.GATE.TIME.SHAPe	73
SCPI.CALCulate(Ch).SELEcted.FILTer.GATE.TIME.SPAN	74
SCPI.CALCulate(Ch).SELEcted.FILTer.GATE.TIME.STARt	75
SCPI.CALCulate(Ch).SELEcted.FILTer.GATE.TIME.STATE	75
SCPI.CALCulate(Ch).SELEcted.FILTer.GATE.TIME.STOP	76
SCPI.CALCulate(Ch).SELEcted.FILTer.GATE.TIME.TYPE	76
SCPI.CALCulate(Ch).SELEcted.FORMat	77
SCPI.CALCulate(Ch).SELEcted.FUNcTION.DATA	78
SCPI.CALCulate(Ch).SELEcted.FUNcTION.DOMain.COUPle	79
SCPI.CALCulate(Ch).SELEcted.FUNcTION.DOMain.STARt	79
SCPI.CALCulate(Ch).SELEcted.FUNcTION.DOMain.STATE	80
SCPI.CALCulate(Ch).SELEcted.FUNcTION.DOMain.STOP	81
SCPI.CALCulate(Ch).SELEcted.FUNcTION.EXECute	81
SCPI.CALCulate(Ch).SELEcted.FUNcTION.PEXCursion	82
SCPI.CALCulate(Ch).SELEcted.FUNcTION.POINts	82
SCPI.CALCulate(Ch).SELEcted.FUNcTION.PPOLarity	83
SCPI.CALCulate(Ch).SELEcted.FUNcTION.TARGet	84
SCPI.CALCulate(Ch).SELEcted.FUNcTION.TTRansition	85
SCPI.CALCulate(Ch).SELEcted.FUNcTION.TYPE	86
SCPI.CALCulate(Ch).SELEcted.LIMit.DATA	87
SCPI.CALCulate(Ch).SELEcted.LIMit.DISPlay.STATE	88
SCPI.CALCulate(Ch).SELEcted.LIMit.FAIL	88
SCPI.CALCulate(Ch).SELEcted.LIMit.OFFSet.AMPLitude	89
SCPI.CALCulate(Ch).SELEcted.LIMit.OFFSet.MARKer	89
SCPI.CALCulate(Ch).SELEcted.LIMit.OFFSet.STIMulus	90
SCPI.CALCulate(Ch).SELEcted.LIMit.REPort.ALL	91
SCPI.CALCulate(Ch).SELEcted.LIMit.REPort.DATA	91
SCPI.CALCulate(Ch).SELEcted.LIMit.REPort.POINts	92
SCPI.CALCulate(Ch).SELEcted.LIMit.STATE	92

SCPI.CALCulate(<i>Ch</i>).SElected.MARKer(<i>Mk</i>).ACTivate	93
SCPI.CALCulate(<i>Ch</i>).SElected.MARKer(<i>Mk</i>).BWIDth.DATA	94
SCPI.CALCulate(<i>Ch</i>).SElected.MARKer(1).BWIDth.REFerence	95
SCPI.CALCulate(<i>Ch</i>).SElected.MARKer(1).BWIDth.STATe	96
SCPI.CALCulate(<i>Ch</i>).SElected.MARKer(1).BWIDth.THREshold	97
SCPI.CALCulate(<i>Ch</i>).SElected.MARKer(1).BWIDth.TYPE	97
SCPI.CALCulate(<i>Ch</i>).SElected.MARKer(1).COUPle	98
SCPI.CALCulate(<i>Ch</i>).SElected.MARKer(1).COUNT	99
SCPI.CALCulate(<i>Ch</i>).SElected.MARKer(1).DISCrete	100
SCPI.CALCulate(<i>Ch</i>).SElected.MARKer(<i>Mk</i>).DATA	101
SCPI.CALCulate(<i>Ch</i>).SElected.MARKer(1).DATA	102
SCPI.CALCulate(<i>Ch</i>).SElected.MARKer(1).FUNctIon.DOMain.COUPle	103
SCPI.CALCulate(<i>Ch</i>).SElected.MARKer(1).FUNctIon.DOMain.START	104
SCPI.CALCulate(<i>Ch</i>).SElected.MARKer(1).FUNctIon.DOMain.STATe	105
SCPI.CALCulate(<i>Ch</i>).SElected.MARKer(1).FUNctIon.DOMain.STOP	106
SCPI.CALCulate(<i>Ch</i>).SElected.MARKer(<i>Mk</i>).FUNctIon.EXECute	107
SCPI.CALCulate(<i>Ch</i>).SElected.MARKer(<i>Mk</i>).FUNctIon.PEXCursion	108
SCPI.CALCulate(<i>Ch</i>).SElected.MARKer(<i>Mk</i>).FUNctIon.PPOLarity	109
SCPI.CALCulate(<i>Ch</i>).SElected.MARKer(<i>Mk</i>).FUNctIon.TARGET	110
SCPI.CALCulate(<i>Ch</i>).SElected.MARKer(<i>Mk</i>).FUNctIon.TRACKing	111
SCPI.CALCulate(<i>Ch</i>).SElected.MARKer(<i>Mk</i>).FUNctIon.TTRansition	112
SCPI.CALCulate(<i>Ch</i>).SElected.MARKer(<i>Mk</i>).FUNctIon.TYPE	113
SCPI.CALCulate(<i>Ch</i>).SElected.MARKer(1).MATH.FLATness.DATA	114
SCPI.CALCulate(<i>Ch</i>).SElected.MARKer(1).MATH.FLATness.DOMain.START	115
SCPI.CALCulate(<i>Ch</i>).SElected.MARKer(1).MATH.FLATness.DOMain.STOP	116
SCPI.CALCulate(<i>Ch</i>).SElected.MARKer(1).MATH.FLATness.STATe	117
SCPI.CALCulate(<i>Ch</i>).SElected.MARKer(1).REFerence.STATe	118
SCPI.CALCulate(<i>Ch</i>).SElected.MARKer(<i>Mk</i>).SET	119
SCPI.CALCulate(<i>Ch</i>).SElected.MARKer(<i>Mk</i>).POSition	119
SCPI.CALCulate(<i>Ch</i>).SElected.MARKer(<i>Mk</i>).STATe	120
SCPI.CALCulate(<i>Ch</i>).SElected.MARKer(<i>Mk</i>).X	121
SCPI.CALCulate(<i>Ch</i>).SElected.MARKer(<i>Mk</i>).Y	122
SCPI.CALCulate(<i>Ch</i>).SElected.MATH.FUNctIon	123
SCPI.CALCulate(<i>Ch</i>).SElected.MATH.MEMorize	123
SCPI.CALCulate(<i>Ch</i>).SElected.MSTatistics.DATA	124
SCPI.CALCulate(<i>Ch</i>).SElected.MSTatistics.DOMain.MARKer.START	125
SCPI.CALCulate(<i>Ch</i>).SElected.MSTatistics.DOMain.MARKer.STOP	125
SCPI.CALCulate(<i>Ch</i>).SElected.MSTatistics.DOMain.STATe	126
SCPI.CALCulate(<i>Ch</i>).SElected.MSTatistics.STATe	126
SCPI.CALCulate(<i>Ch</i>).SElected.RLIMit.DATA	127
SCPI.CALCulate(<i>Ch</i>).SElected.RLIMit.DISPlay.LINE	128
SCPI.CALCulate(<i>Ch</i>).SElected.RLIMit.DISPlay.SELect	128
SCPI.CALCulate(<i>Ch</i>).SElected.RLIMit.DISPlay.VALue	129
SCPI.CALCulate(<i>Ch</i>).SElected.RLIMit.FAIL	129
SCPI.CALCulate(<i>Ch</i>).SElected.RLIMit.REPort.DATA	130
SCPI.CALCulate(<i>Ch</i>).SElected.RLIMit.STATe	130
SCPI.CALCulate(<i>Ch</i>).SElected.SMOothing.APERture	131
SCPI.CALCulate(<i>Ch</i>).SElected.SMOothing.STATe	131
SCPI.CALCulate(<i>Ch</i>).SElected.TRANsform.TIME.CENter	132
SCPI.CALCulate(<i>Ch</i>).SElected.TRANsform.TIME.IMPulse.WIDTh	133
SCPI.CALCulate(<i>Ch</i>).SElected.TRANsform.TIME.KBESsel	134
SCPI.CALCulate(<i>Ch</i>).SElected.TRANsform.TIME.LPFRequency	134
SCPI.CALCulate(<i>Ch</i>).SElected.TRANsform.TIME.REFlection.TYPE	135
SCPI.CALCulate(<i>Ch</i>).SElected.TRANsform.TIME.SPAN	136
SCPI.CALCulate(<i>Ch</i>).SElected.TRANsform.TIME.START	137
SCPI.CALCulate(<i>Ch</i>).SElected.TRANsform.TIME.STATe	138

SCPI.CALCulate(<i>Ch</i>).SElected.TRANSform.TIME.STEP.RTime.....	139
SCPI.CALCulate(<i>Ch</i>).SElected.TRANSform.TIME.STIMulus	140
SCPI.CALCulate(<i>Ch</i>).SElected.TRANSform.TIME.STOP	141
SCPI.CALCulate(<i>Ch</i>).SElected.TRANSform.TIME.TYPE	142
SCPI.CALCulate(<i>Ch</i>).SElected.TRANSform.TIME.UNIT	143
SCPI.CALCulate(<i>Ch</i>).TRACe(<i>Tr</i>).DATA.FDATA.....	144
SCPI.CALCulate(<i>Ch</i>).TRACe(<i>Tr</i>).DATA.FMEMory	145
SCPI.CALCulate(<i>Ch</i>). TRACe(<i>Tr</i>).DATA.SDATA	146
SCPI.CALCulate(<i>Ch</i>). TRACe(<i>Tr</i>).DATA.SMEMory.....	147
SCPI.CALCulate(<i>Ch</i>). TRACe(<i>Tr</i>).DATA.XAXis.....	147
SCPI.DISPlay.COLor.BACK.....	148
SCPI.DISPlay.COLor.GRATicule	149
SCPI.DISPlay.COLor.RESet	149
SCPI.DISPlay.COLor.TRACe(<i>Tr</i>).DATA	150
SCPI.DISPlay.COLor.TRACe(<i>Tr</i>).MEMory	151
SCPI.DISPlay.ENABLE	151
SCPI.DISPlay.FSIGn	152
SCPI.DISPlay.GLABel.....	152
SCPI.DISPlay.IMAGe.....	153
SCPI.DISPlay.HIDE	153
SCPI.DISPlay.MAXimize	154
SCPI.DISPlay.PARTition.FONT.SIZE(<i>Param</i>).....	154
SCPI.DISPlay.PARTition.VISible(<i>Param</i>).....	155
SCPI.DISPlay.POSition.....	156
SCPI.DISPlay.SHOW	157
SCPI.DISPlay.SPLit.....	157
SCPI.DISPlay.REFResh.IMMediate.....	158
SCPI.DISPlay.UPDate.IMMediate	158
SCPI.DISPlay.WINDow(<i>Ch</i>).ACTivate.....	159
SCPI.DISPlay.WINDow(<i>Ch</i>).ANNotation.MARKer.ALIGn.TYPE	159
SCPI.DISPlay.WINDow(<i>Ch</i>).ANNotation.MARKer.SINGLE.STATe	160
SCPI.DISPlay.WINDow(<i>Ch</i>).MAXimize	160
SCPI.DISPlay.WINDow(<i>Ch</i>).SPLit.....	161
SCPI.DISPlay.WINDow(<i>Ch</i>).TITLe.DATA.....	161
SCPI.DISPlay.WINDow(<i>Ch</i>).TITLe.STATe	162
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	166
SCPI.DISPlay.WINDow(<i>Ch</i>).TRACe(<i>Tr</i>).Y.SCALe.PDIVision	167
SCPI.DISPlay.WINDow(<i>Ch</i>).TRACe(<i>Tr</i>).Y.SCALe.RLEVel	168
SCPI.DISPlay.WINDow(<i>Ch</i>).TRACe(<i>Tr</i>).Y.SCALe.RPOSITION	169
SCPI.DISPlay.WINDow(<i>Ch</i>).X.SPACing.....	169
SCPI.DISPlay.WINDow(<i>Ch</i>).Y.SCALe.DIVisions.....	170
SCPI.HCOPy.ABORt.....	170
SCPI.HCOPy.DATE.STAMP.....	171
SCPI.HCOPy.IMAGe.....	171
SCPI.HCOPy.IMMediate	172
SCPI.HCOPy.PAINT	172
SCPI.IEEE4882.CLS.....	173
SCPI.IEEE4882.IDN.....	173
SCPI.IEEE4882.OPC	174
SCPI.IEEE4882.RST	174
SCPI.IEEE4882.TRG	175

SCPI.IEEE4882.WAI	176
SCPI.INITiate(Ch).CONTInuous	177
SCPI.INITiate(Ch).IMMEdiate	178
SCPI.MMEMory.CATalog(Dir)	179
SCPI.MMEMory.COPY(Src, Dst)	180
SCPI.MMEMory.DELeTe(File)	180
SCPI.MMEMory.LOAD.CHANnel.STATe	181
SCPI.MMEMory.LOAD.CKIT(Ck)	181
SCPI.MMEMory.LOAD.LIMit	182
SCPI.MMEMory.LOAD.PLOsS(Pt)	182
SCPI.MMEMory.LOAD.RLIMit	183
SCPI.MMEMory.LOAD.SEGMent	183
SCPI.MMEMory.LOAD.SNP.DATA	184
SCPI.MMEMory.LOAD.SNP.TRACe(Tr).MEMory	184
SCPI.MMEMory.LOAD.STATe	185
SCPI.MMEMory.MDIReCTory	185
SCPI.MMEMory.STORe.CHANnel.CLEar	186
SCPI.MMEMory.STORe.CHANnel.STATe	186
SCPI.MMEMory.STORe.CKIT(Ck)	187
SCPI.MMEMory.STORe.FDATa	187
SCPI.MMEMory.STORe.IMAGe	188
SCPI.MMEMory.STORe.LIMit	188
SCPI.MMEMory.STORe.PLOsS(Pt)	189
SCPI.MMEMory.STORe.RLIMit	189
SCPI.MMEMory.STORe.SEGMent	190
SCPI.MMEMory.STORe.SNP.DATA	190
SCPI.MMEMory.STORe.SNP.FORMat	191
SCPI.MMEMory.STORe.SNP.SEParator	191
SCPI.MMEMory.STORe.SNP.TYPE.S1P	192
SCPI.MMEMory.STORe.SNP.TYPE.S2P	192
SCPI.MMEMory.STORe.SNP.TYPE.S3P	193
SCPI.MMEMory.STORe.SNP.TYPE.S4P	194
SCPI.MMEMory.STORe.STATe	194
SCPI.MMEMory.STORe.STYPE	195
SCPI.OUTPut.STATe	195
SCPI.SENSE(Ch).AVERAge.CLEar	196
SCPI.SENSE(Ch).AVERAge.COUNT	196
SCPI.SENSE(Ch).AVERAge.STATe	197
SCPI.SENSE(Ch).BANDwidth.RESolution	198
SCPI.SENSE(Ch).CORRection.CLEar	198
SCPI.SENSE(Ch).CORRection.COEFFicient.DATA(Str, Pt_r, Pt_s)	199
SCPI.SENSE(Ch).CORRection.COEFFicient.METHod.ERESponse	200
SCPI.SENSE(Ch).CORRection.COEFFicient.METHod.RESPonse.OPEN	201
SCPI.SENSE(Ch).CORRection.COEFFicient.METHod.RESPonse.SHORT	201
SCPI.SENSE(Ch).CORRection.COEFFicient.METHod.RESPonse.THURU	202
SCPI.SENSE(Ch).CORRection.COEFFicient.METHod.SOLT1	203
SCPI.SENSE(Ch).CORRection.COEFFicient.METHod.SOLT2	204
SCPI.SENSE(Ch).CORRection.COEFFicient.METHod.SOLT3	205
SCPI.SENSE(Ch).CORRection.COEFFicient.METHod.SOLT4	206
SCPI.SENSE(Ch).CORRection.COEFFicient.SAVE	207
SCPI.SENSE(Ch).CORRection.COLLeCt.ACQuire.ISOLation	208
SCPI.SENSE(Ch).CORRection.COLLeCt.ACQuire.LOAD	209
SCPI.SENSE(Ch).CORRection.COLLeCt.ACQuire.OPEN	210
SCPI.SENSE(Ch).CORRection.COLLeCt.ACQuire.SHORT	211
SCPI.SENSE(Ch).CORRection.COLLeCt.ACQuire.SUBClass	212
SCPI.SENSE(Ch).CORRection.COLLeCt.ACQuire.THURU	213

SCPI.SENSE(Ch).CORRection.COLlect.ACQuire.TRLLine	214
SCPI.SENSE(Ch).CORRection.COLlect.ACQuire.TRLReflect	215
SCPI.SENSE(Ch).CORRection.COLlect.ACQuire.TRLThru	216
SCPI.SENSE(Ch).CORRection.COLlect.CKIT.LABel	217
SCPI.SENSE(Ch).CORRection.COLlect.CKIT.ORDER.LOAD(Pt)	217
SCPI.SENSE(Ch).CORRection.COLlect.CKIT.ORDER.OPEN(Pt)	218
SCPI.SENSE(Ch).CORRection.COLlect.CKIT.ORDER.SELect	218
SCPI.SENSE(Ch).CORRection.COLlect.CKIT.ORDER.SHORT(Pt)	219
SCPI.SENSE(Ch).CORRection.COLlect.CKIT.ORDER.THURU(Pt_m, Pt_n)	220
SCPI.SENSE(Ch).CORRection.COLlect.CKIT.ORDER.TRLLine(Pt_m, Pt_n)	221
SCPI.SENSE(Ch).CORRection.COLlect.CKIT.ORDER.TRLReflect(Pt)	222
SCPI.SENSE(Ch).CORRection.COLlect.CKIT.ORDER.TRLThru(Pt_m, Pt_n)	223
SCPI.SENSE(Ch).CORRection.COLlect.CKIT.RESet	223
SCPI.SENSE(Ch).CORRection.COLlect.CKIT.SELect	224
SCPI.SENSE(Ch).CORRection.COLlect.CKIT.STAN(Std).ARBitrary	225
SCPI.SENSE(Ch).CORRection.COLlect.CKIT.STAN(Std).C0	226
SCPI.SENSE(Ch).CORRection.COLlect.CKIT.STAN(Std).C1	226
SCPI.SENSE(Ch).CORRection.COLlect.CKIT.STAN(Std).C2	227
SCPI.SENSE(Ch).CORRection.COLlect.CKIT.STAN(Std).C3	227
SCPI.SENSE(Ch).CORRection.COLlect.CKIT.STAN(Std).DELay	228
SCPI.SENSE(Ch).CORRection.COLlect.CKIT.STAN(Std).FMAXimum	229
SCPI.SENSE(Ch).CORRection.COLlect.CKIT.STAN(Std).FMINimum	230
SCPI.SENSE(Ch).CORRection.COLlect.CKIT.STAN(Std).L0	231
SCPI.SENSE(Ch).CORRection.COLlect.CKIT.STAN(Std).L1	232
SCPI.SENSE(Ch).CORRection.COLlect.CKIT.STAN(Std).L2	232
SCPI.SENSE(Ch).CORRection.COLlect.CKIT.STAN(Std).L3	233
SCPI.SENSE(Ch).CORRection.COLlect.CKIT.STAN(Std).LABel	233
SCPI.SENSE(Ch).CORRection.COLlect.CKIT.STAN(Std).LOSS	234
SCPI.SENSE(Ch).CORRection.COLlect.CKIT.STAN(Std).TYPE	235
SCPI.SENSE(Ch).CORRection.COLlect.CKIT.STAN(Std).Z0	236
SCPI.SENSE(Ch).CORRection.COLlect.CLEAR	237
SCPI.SENSE(Ch).CORRection.COLlect.DATA.ISOLation(Pt_r, Pt_s)	238
SCPI.SENSE(Ch).CORRection.COLlect.DATA.LOAD(Pt)	239
SCPI.SENSE(Ch).CORRection.COLlect.DATA.OPEN(Pt)	240
SCPI.SENSE(Ch).CORRection.COLlect.DATA.SHORT(Pt)	241
SCPI.SENSE(Ch).CORRection.COLlect.DATA.THURU.MATCH(Pt_r, Pt_s)	242
SCPI.SENSE(Ch).CORRection.COLlect.DATA.THURU.TRANsmission(Pt_r, Pt_s)	243
SCPI.SENSE(Ch).CORRection.COLlect.ECAL.CCHeck.ACQuire	244
SCPI.SENSE(Ch).CORRection.COLlect.ECAL.ERESponse	244
SCPI.SENSE(1).CORRection.COLlect.ECAL.INFormation	245
SCPI.SENSE(1).CORRection.COLlect.ECAL.ORlenation.STATe	246
SCPI.SENSE(1).CORRection.COLlect.ECAL.PATH(Pt)	246
SCPI.SENSE(Ch).CORRection.COLlect.ECAL.SOLT1	247
SCPI.SENSE(Ch).CORRection.COLlect.ECAL.SOLT2	247
SCPI.SENSE(Ch).CORRection.COLlect.ECAL.SOLT3	248
SCPI.SENSE(Ch).CORRection.COLlect.ECAL.SOLT4	248
SCPI.SENSE(1).CORRection.COLlect.ECAL.UCHar	249
SCPI.SENSE(1).CORRection.COLlect.ECAL.UTHRU.STATe	250
SCPI.SENSE(Ch).CORRection.COLlect.METHod.ERESponse	251
SCPI.SENSE(Ch).CORRection.COLlect.METHod.RESPOse.OPEN	252
SCPI.SENSE(Ch).CORRection.COLlect.METHod.RESPOse.SHORT	252
SCPI.SENSE(Ch).CORRection.COLlect.METHod.RESPOse.THURU	253
SCPI.SENSE(Ch).CORRection.COLlect.METHod.SOLT1	254
SCPI.SENSE(Ch).CORRection.COLlect.METHod.SOLT2	255
SCPI.SENSE(Ch).CORRection.COLlect.METHod.SOLT3	256
SCPI.SENSE(Ch).CORRection.COLlect.METHod.SOLT4	257

SCPI.SENSE(Ch).CORRection.COLlect.METHod.TRL2	258
SCPI.SENSE(Ch).CORRection.COLlect.METHod.TRL3	259
SCPI.SENSE(Ch).CORRection.COLlect.METHod.TRL4	260
SCPI.SENSE(Ch).CORRection.COLlect.METHod.TYPE	261
SCPI.SENSE(Ch).CORRection.COLlect.SAVE	262
SCPI.SENSE(Ch).CORRection.COLlect.SIMPLified.SAVE	263
SCPI.SENSE(Ch).CORRection.EXTension.PORT(Pt).FREQuency(Ls)	264
SCPI.SENSE(Ch).CORRection.EXTension.PORT(Pt).INCLude(Ls).STATe	265
SCPI.SENSE(Ch).CORRection.EXTension.PORT(Pt).LDC	266
SCPI.SENSE(Ch).CORRection.EXTension.PORT(Pt).LOSS(Ls)	267
SCPI.SENSE(Ch).CORRection.EXTension.PORT(Pt).TIME	268
SCPI.SENSE(Ch).CORRection.EXTension.STATe	268
SCPI.SENSE(Ch).CORRection.INFOrmation(rPt, sPt)	269
SCPI.SENSE(1).CORRection.IMPedance.INPUt.MAGNItude	269
SCPI.SENSE(Ch).CORRection.OFFSet.CLEar	270
SCPI.SENSE(Ch).CORRection.OFFSet.COLlect.ACQuire.LOAD	270
SCPI.SENSE(Ch).CORRection.OFFSet.COLlect.ACQuire.OPEN	271
SCPI.SENSE(Ch).CORRection.OFFSet.COLlect.ACQuire.SHORT	272
SCPI.SENSE(Ch).CORRection.OFFSet.COLlect.ACQuire.THRU	273
SCPI.SENSE(Ch).CORRection.OFFSet.COLlect.ACQuire.PMETer	274
SCPI.SENSE(Ch).CORRection.OFFSet.COLlect.CLEar	275
SCPI.SENSE(Ch).CORRection.OFFSet.COLlect.METHod.SMIX2	275
SCPI.SENSE(Ch).CORRection.OFFSet.COLlect.SAVE	276
SCPI.SENSE(Ch).CORRection.RECeiver(Pt).COLlect.ACQuire	277
SCPI.SENSE(Ch).CORRection.RECeiver(Pt).COLlect.RCHannelACQuire	278
SCPI.SENSE(Ch).CORRection.RECeiver(Pt).COLlect.TCHannelACQuire	278
SCPI.SENSE(Ch).CORRection.RECeiver(Pt).OFFSet.AMPLitude	279
SCPI.SENSE(Ch).CORRection.RECeiver(Pt).STATe	279
SCPI.SENSE(Ch).CORRection.STATe	280
SCPI.SENSE(Ch).CORRection.TYPE(Tr)	281
SCPI.SENSE(Ch).CORRection.TRANSform.TIME.FREQuency	282
SCPI.SENSE(Ch).CORRection.TRANSform.TIME.LOSS	282
SCPI.SENSE(Ch).CORRection.TRANSform.TIME.RVELocity	283
SCPI.SENSE(Ch).CORRection.TRANSform.TIME.STATe	283
SCPI.SENSE(Ch).CORRection.TRIGger.FREE.STATe	284
SCPI.SENSE(Ch).DATA.CORRdata(Param)	285
SCPI.SENSE(Ch).DATA.RAWData(Param)	286
SCPI.SENSE(Ch).FREQuency.CENTer	287
SCPI.SENSE(Ch).FREQuency.CW	288
SCPI.SENSE(Ch).FREQuency.DATA	288
SCPI.SENSE(Ch).FREQuency.SPAN	289
SCPI.SENSE(Ch).FREQuency.START	290
SCPI.SENSE(Ch).FREQuency.STOP	291
SCPI.SENSE(Ch).OFFSet.ADJust.CONTInuous.PERiod	292
SCPI.SENSE(Ch).OFFSet.ADJust.CONTInuous.STATe	293
SCPI.SENSE(Ch).OFFSet.ADJust.EXECute	293
SCPI.SENSE(Ch).OFFSet.ADJust.PORT	294
SCPI.SENSE(Ch).OFFSet.ADJust.STATe	294
SCPI.SENSE(Ch).OFFSet.ADJust.Value	295
SCPI.SENSE(Ch).OFFSet.PORT(Pt).FREQuency.DATA	296
SCPI.SENSE(Ch).OFFSet.PORT(Pt).FREQuency.DIVisor	297
SCPI.SENSE(Ch).OFFSet.PORT(Pt).FREQuency. MULTiplier	298
SCPI.SENSE(Ch).OFFSet.PORT(Pt).FREQuency.OFFSet	299
SCPI.SENSE(Ch).OFFSet.PORT(Pt).FREQuency. START	300
SCPI.SENSE(Ch).OFFSet.PORT(Pt).FREQuency. STOP	300
SCPI.SENSE(Ch).OFFSet.RECeiver.FREQuency.DATA	301

SCPI.SENSE(Ch).OFFSet.RECeiver.FREQuency.DIVisor	302
SCPI.SENSE(Ch).OFFSet.RECeiver.FREQuency.MULTIplier	303
SCPI.SENSE(Ch).OFFSet.RECeiver.FREQuency.OFFSet	304
SCPI.SENSE(Ch).OFFSet.RECeiver.FREQuency.START	305
SCPI.SENSE(Ch).OFFSet.RECeiver.FREQuency.STOP	305
SCPI.SENSE(Ch).OFFSet.SOURce.FREQuency.DATA	306
SCPI.SENSE(Ch).OFFSet.SOURce.FREQuency.DIVisor	307
SCPI.SENSE(Ch).OFFSet.SOURce.FREQuency.MULTIplier	308
SCPI.SENSE(Ch).OFFSet.SOURce.FREQuency.OFFSet	309
SCPI.SENSE(Ch).OFFSet.SOURce.FREQuency.START	310
SCPI.SENSE(Ch).OFFSet.SOURce.FREQuency.STOP	310
SCPI.SENSE(Ch).OFFSet.STATe	311
SCPI.SENSE(Ch).OFFSet.TYPE	311
SCPI.SENSE(Ch).ROSCillator.SOURce	312
SCPI.SENSE(Ch).SEGMENT.DATA	313
SCPI.SENSE(Ch).SWEep.POINT.TIME	314
SCPI.SENSE(Ch).SWEep.POINTs	315
SCPI.SENSE(Ch).SWEep.TYPE	316
SCPI.SERVICE.CHANnel(1).ACTive	316
SCPI.SERVICE.CHANnel(1).COUNT	317
SCPI.SERVICE.CHANnel(Ch).TRACe(1).ACTive	317
SCPI.SERVICE.CHANnel(1).TRACe.COUNT	318
SCPI.SERVICE.CHANnel(Ch).TRACe(Tr).MARKer.ACTive	318
SCPI.SERVICE.PORT.COUNT	319
SCPI.SERVICE.SWEep.FREQuency.MAXimum	319
SCPI.SERVICE.SWEep.FREQuency.MINimum	319
SCPI.SERVICE.SWEep.POINTs	320
SCPI.SOURce(Ch).POWER.CENTer	320
SCPI.SOURce(Ch).POWER.LEVel.IMMEdiate.AMPLitude	321
SCPI.SOURce(Ch).POWER.LEVel.SLOPe.DATA	322
SCPI.SOURce(Ch).POWER.LEVel.SLOPe.STATe	322
SCPI.SOURce(Ch).POWER.PORT(Pt).CORRection.COLLect.ACQuire	323
SCPI.SOURce(Ch).POWER.PORT(Pt).CORRection.COLLect.TABLe.LOSS.DATA	324
SCPI.SOURce(Ch).POWER.PORT(Pt).CORRection.COLLect.TABLe.LOSS.STATe	325
SCPI.SOURce(Ch).POWER.PORT(Pt).CORRection.DATA	326
SCPI.SOURce(Ch).POWER.PORT(Pt).CORRection.STATe	327
SCPI.SOURce(Ch).POWER.PORT(Pt).LEVel.IMMEdiate.AMPLitude	328
SCPI.SOURce(Ch).POWER.PORT(1).COUPLe	329
SCPI.SOURce(Ch).POWER.SPAN	330
SCPI.SOURce(Ch).POWER.START	331
SCPI.SOURce(Ch).POWER.STOP	332
SCPI.STATUS.OPERation.CONDITION	332
SCPI.STATUS.OPERation.ENABLE	333
SCPI.STATUS.OPERation.EVENT	333
SCPI.STATUS.OPERation.NTRansition	334
SCPI.STATUS.OPERation.PTRansition	334
SCPI.STATUS.PRESet	335
SCPI.STATUS.QUESTionable.CONDITION	335
SCPI.STATUS.QUESTionable.ENABLE	336
SCPI.STATUS.QUESTionable.EVENT	336
SCPI.STATUS.QUESTionable.LIMit.CHANnel(Ch).CONDITION	337
SCPI.STATUS.QUESTionable.LIMit.CHANnel(Ch).ENABLE	337
SCPI.STATUS.QUESTionable.LIMit.CHANnel(Ch).EVENT	338
SCPI.STATUS.QUESTionable.LIMit.CHANnel(Ch).NTRansition	338
SCPI.STATUS.QUESTionable.LIMit.CHANnel(Ch).PTRansition	339
SCPI.STATUS.QUESTionable.LIMit.CONDITION	339

SCPI.STATus.QUEStionable.LIMit.ENABLE	340
SCPI.STATus.QUEStionable.LIMit.EVENT	340
SCPI.STATus.QUEStionable.LIMit.NTRansition	341
SCPI.STATus.QUEStionable.LIMit.PTRansition	341
SCPI.STATus.QUEStionable.NTRansition	342
SCPI.STATus.QUEStionable.PTRansition	342
SCPI.STATus.QUEStionable.RLIMit.CHANnel(<i>Ch</i>).CONDition	343
SCPI.STATus.QUEStionable.RLIMit.CHANnel(<i>Ch</i>).ENABLE	343
SCPI.STATus.QUEStionable.RLIMit.CHANnel(<i>Ch</i>).EVENT	344
SCPI.STATus.QUEStionable.RLIMit.CHANnel(<i>Ch</i>).NTRansition	344
SCPI.STATus.QUEStionable.RLIMit.CHANnel(<i>Ch</i>).PTRansition	345
SCPI.STATus.QUEStionable.RLIMit.CONDition	345
SCPI.STATus.QUEStionable.RLIMit.ENABLE	346
SCPI.STATus.QUEStionable.RLIMit.EVENT	346
SCPI.STATus.QUEStionable.RLIMit.NTRansition	347
SCPI.STATus.QUEStionable.RLIMit.PTRansition	347
SCPI.SYSTem.BEEPer.COMPLete.IMMEdiate	348
SCPI.SYSTem.BEEPer.COMPLete.STATe	348
SCPI.SYSTem.BEEPer.WARNing.IMMEdiate	349
SCPI.SYSTem.BEEPer.WARNing.STATe	349
SCPI.SYSTem.COMMunicate.ECAL.TEMPerature.SENSor	350
SCPI.SYSTem.COMMunicate.ECAL.IMPedance(<i>Pt</i>)	350
SCPI.SYSTem.COMMunicate.ECAL.CHECK	351
SCPI.SYSTem.COMMunicate.ECAL.THRU(<i>Pt1</i> , <i>Pt2</i>)	351
SCPI.SYSTem.CORRection.STATe	352
SCPI.SYSTem.CYCLe.TIME.MEASurement	352
SCPI.SYSTem.DATE	353
SCPI.SYSTem.HIDE	353
SCPI.SYSTem.LOCal	354
SCPI.SYSTem.PRESet	354
SCPI.SYSTem.PORT.SWITchover.Delay.STATe	355
SCPI.SYSTem.REMote	355
SCPI.SYSTem.RWLock	356
SCPI.SYSTem.TEMPerature.SENSor(<i>Idx</i>)	356
SCPI.SYSTem.SHOW	357
SCPI.SYSTem.TERMinate	357
SCPI.SYSTem.TIME	358
SCPI.TRIGger.SEQuence.AVERage	359
SCPI.TRIGger.SEQuence.EXTErnal.Delay	360
SCPI.TRIGger.SEQuence.EXTErnal.POSition	361
SCPI.TRIGger.SEQuence.EXTErnal.SLOPe	362
SCPI.TRIGger.SEQuence.IMMEdiate	363
SCPI.TRIGger.SEQuence.POINt	364
SCPI.TRIGger.SEQuence.SINGle	365
SCPI.TRIGger.SEQuence.SCOPE	366
SCPI.TRIGger.SEQuence.SOURce	367
SCPI.TRIGger.SEQuence.STATus	368
SCPI.TRIGger.SEQuence.WAIT(<i>STATus</i>)	368
SCPI.TRIGger.OUTPUT.FUNCTion	369
SCPI.TRIGger.OUTPUT.POLarity	370
SCPI.TRIGger.OUTPUT.STATe	371
16 Appendix 1. IEE488.2 Status Reporting System	372
17 Appendix 2. Error Codes	378

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 instrument program on one PC. DCOM technology is used when a user program runs on a PC connected with the measurement instrument 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 programmer's manual covers the 2-port and 4-port models of the CMT network analyzers listed below.

The 2-port network analyzers controlled by the S2VNA software are:

- Planar 304/1
- Planar 804/1
- Planar 814/1
- S5048
- S5065
- S5085
- S7530
- S5180
- C1209
- C1220
- C2220
- C2209
- C4209
- C4220

The 4-port network analyzers controlled by the S4VNA software are:

- Planar 808/1
- C1409

- C1420
- C2409
- C4409

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
```

```
S4VNA.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 Instruments 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.

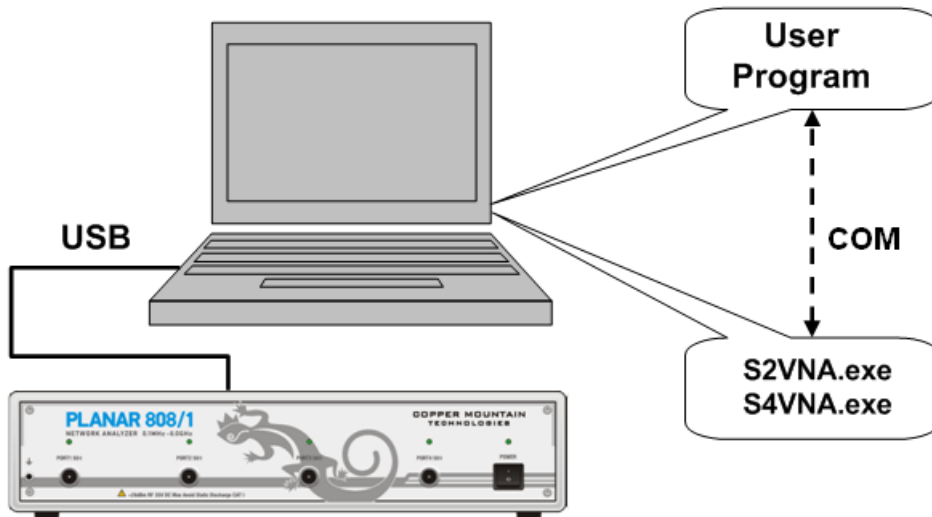


Figure 1. COM technology

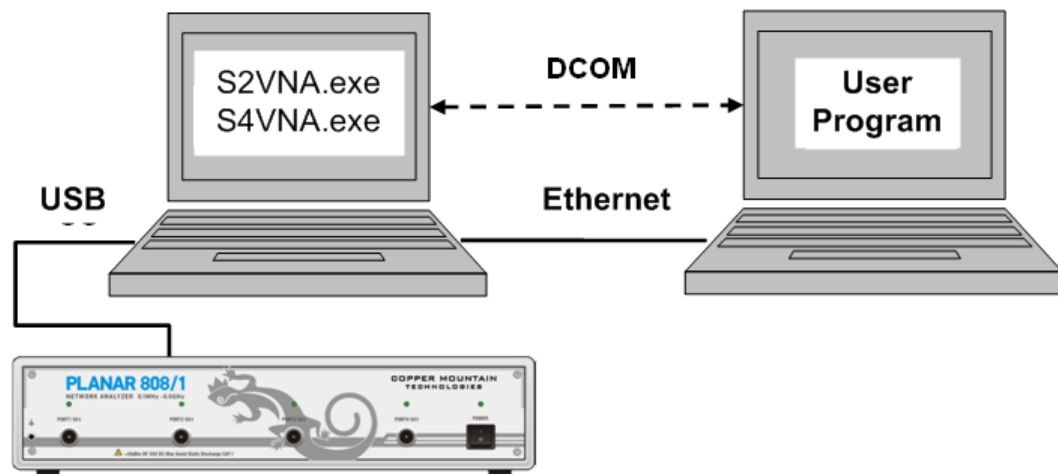


Figure 2. DCOM technology

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.

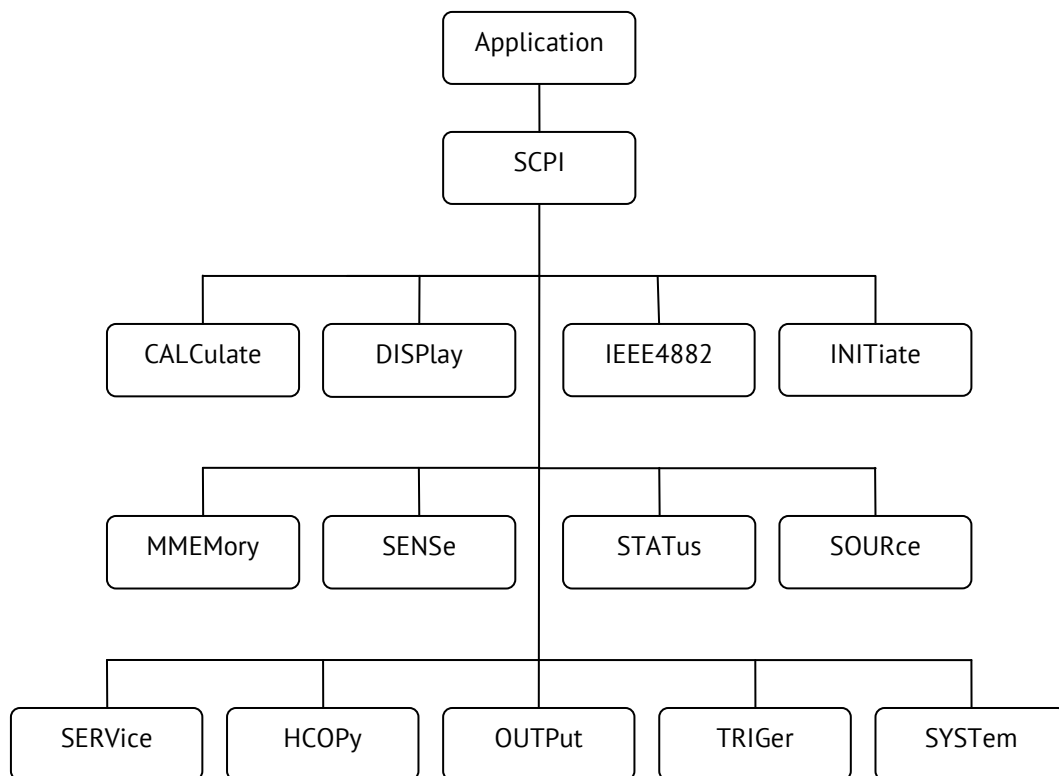


Figure 3. The structure of COM objects

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 borrowed from the SCPI command system, an alternative remote control technology of the analyzer. Commands in SCPI have a chain hierarchical structure, for example:

```
CALCulate:PARAmeter:DEFine S11
```

The same command in COM 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 perform the following:

- 1) Declare a variable as an object.
- 2) Create a COM Object and assign it 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.Applcation" or "S4VNA.Applcation"
<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.Applcation")
Set app = CreateObject("S2VNA.Applcation", "Analyzer_Name")
```

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

```
Set app = CreateObject("S4VNA.Application")
Set app = CreateObject("S4VNA.Application", "Host_Name")
Set app = CreateObject("S4VNA.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
```

Note: The models of vector network analyzers working with the S2VNA executable module share the same COM object. The name of COM object is *S2VNA.Application*

The models of vector network analyzers working with the S4VNA executable module share the same COM object. The name of COM object is *S4VNA.Application*

For example, the commands for creating a COM server for 2-port an analyzer is:

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

For the backward compatibility the old name is preserved for creating COM object for each model. The user can use the old and new name of the COM object interchangeably, since they all create the same COM object. For example: :

```
Set app = CreateObject("S2VNA.Application")  
Set app = CreateObject("Planar304.Application")  
Set app = CreateObject("Planar804.Application")  
Set app = CreateObject("S5048.Application")  
Set app = CreateObject("S7530.Application")
```

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 *SWEEp* object, i.e. the number of sweep points:

```
app.SCPI.SENSE.SWEEp.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("S4VNA.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("S4VNA.Application")
On Error GoTo ErrHandler
app.SCPI.PARameter.DEFine = "S13"
...
Exit Sub
ErrHandler:
    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 variant 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.

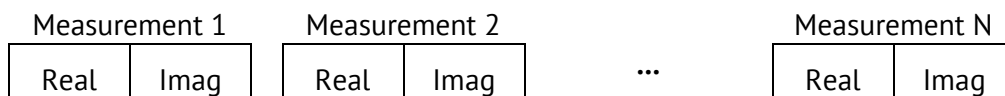


Figure 4. Array of complex measurements

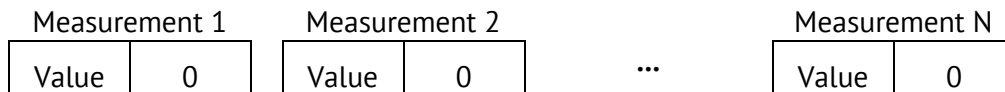


Figure 5. Array of real measurements

15 COM Server Commands

NAME

<i>Description</i>	Reads out the Analyzer identification string.
<i>Type</i>	String (read only)
<i>Syntax</i>	StrName = app.NAME
<i>Reply</i>	The identification string in format: <manufacturer>, <model>, <serial number>, <software version>/<hardware version>. For example: CMT, C1209, 08080188, 16.2/01
<i>Equivalent Softkeys</i>	None

READY

<i>Description</i>	Reads out the Ready state of the analyzer. The state is <i>True</i> when analyzer hardware is connected, powered and the boot process is completed (about 10 sec).
<i>Type</i>	Boolean (read only)
<i>Syntax</i>	<i>State</i> = app.READY
<i>Equivalent Softkeys</i>	None

SCPI.ABORT

<i>Description</i>	Aborts the sweep. The channels in the <i>Single</i> trigger initiation mode transit to the <i>Hold</i> state. The channels in the <i>Continuous</i> trigger initiation mode transit to the <i>trigger waiting</i> state, if the trigger source is set to <i>Internal</i> , the channel immediately starts a new sweep.
<i>Type</i>	Method
<i>Syntax</i>	<i>app</i> .SCPI.ABORT
<i>Equivalent Softkeys</i>	Stimulus > Trigger > Restart

SCPI.CALCulate(Ch).FSIMulator.BALun.CZConversion.STATe

<i>Description</i>	Turns ON/OFF the common port impedance conversion function when the fixture simulator function is ON, for all the balance ports of selected channel (<i>Ch</i>). (S4VNA only)
<i>Type</i>	Boolean (read/write)
<i>Parameter</i>	True: common port impedance conversion function is ON False: common port impedance conversion function is OFF
<i>Preset Value</i>	False
<i>Target</i>	Channel <i>Ch</i> from 1 to 16
<i>Syntax</i>	<i>Status</i> = <i>app</i> .SCPI.CALCulate(<i>Ch</i>).FSIMulator.BALun.CZConversion.STATe <i>app</i> .SCPI.CALCulate(<i>Ch</i>).FSIMulator.BALun.CZConversion.STATe = True
<i>Equivalent Softkeys</i>	Analysis > Fixture Simulator > Cmn ZConversion > Cmn ZConversion [ON OFF]

SCPI.CALCulate(*Ch*).FSIMulator.BALun.CZConversion.BPORT(*Bpt*).Z0.R

<i>Description</i>	Sets the impedance value for the common port impedance conversion function, for the balance ports 1 or 2 (<i>Bpt</i>) of the selected channel (<i>Ch</i>). (S4VNA only)
<i>Type</i>	Double (read/write)
<i>Range</i>	from 1e-3 to 1e7
<i>Out of Range</i>	Sets the value of the limit, which is closer to the specified value.
<i>Preset Value</i>	25
<i>Unit</i>	Ω (Ohm)
<i>Target</i>	Balanced Port <i>Bpt</i> of channel <i>Ch</i> , <i>Ch</i> : channel number 1–16 <i>Bpt</i> : balanced port number 1 or 2 for the Bal-Bal topology, always 1 for the SE-Bal, SE-SE-Bal and Bal topology.
<i>Syntax</i>	<i>Value</i> = app.SCPI.CALCulate(<i>Ch</i>).FSIMulator.BALun.CZConversion.BPORT(<i>Bpt</i>).Z0.R app.SCPI.CALCulate(<i>Ch</i>).FSIMulator.BALun.CZConversion.BPORT(<i>Bpt</i>).Z0.R = 20
<i>Equivalent Softkeys</i>	Analysis > Fixture Simulator > Cmn ZConversion > Bal Port 1 or 2

SCPI.CALCulate(Ch).FSIMulator.BALun.DEVICE

<i>Description</i>	Selects the type of the balanced device for the fixture simulation function. (S4VNA only)
<i>Type</i>	String (read/write)
<i>Parameter</i>	"SBALanced" : Specifies the unbalance-balance (3 ports) "BBALanced" : Specifies the balance-balance (4 ports) "SSBalanced" : Specifies the unbalance-unbalance-balance (4 ports) "BALanced" : Specifies the unbalance (2 ports)
<i>Preset Value</i>	"BBAL "
<i>Target</i>	Channel <i>Ch</i> from 1 to 16
<i>Syntax</i>	<i>Param</i> = app.SCPI.CALCulate(<i>Ch</i>).FSIMulator.BALun.DEVICE <i>app.SCPI.CALCulate(Ch).FSIMulator.BALun.DEVICE</i> = "SBAL "
<i>Equivalent Softkeys</i>	Analysis > Fixture Simulator > Topolgy > Device [SE-Bal Bal-Bal SE-SE-Bal Bal]

SCPI.CALCulate(*Ch*).FSIMulator.BALun.DMCircuit.BPORT(*Bpt*).PARameters.C

<i>Description</i>	Sets the capacitance value of the C element of the differential matching circuit of the fixture simulation function. (S4VNA only)
<i>Type</i>	Double (read/write)
<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
<i>Unit</i>	F (Farad)
<i>Target</i>	Balanced Port <i>Bpt</i> of channel <i>Ch</i> , <i>Ch</i> : channel number 1–16 <i>Bpt</i> : balanced port number 1 or 2 for the Bal-Bal topology, always 1 for the SE-Bal, SE-SE-Bal and Bal topology.
<i>Syntax</i>	<i>Value</i> = app.SCPI.CALCulate(<i>Ch</i>).FSIMulator.BALun.DMCircuit.BPORT(<i>Bpt</i>).PARameters.C app.SCPI.CALCulate(<i>Ch</i>).FSIMulator.BALun.DMCircuit.BPORT(<i>Bpt</i>).PARameters.C = 1e-12
<i>Equivalent Softkeys</i>	Analysis > Fixture Simulator > Diff Matching > Bal Port n > C

SCPI.CALCulate(*Ch*).FSIMulator.BALun.DMCircuit.BPORT(*Bpt*).PARameters.G

<i>Description</i>	Sets the conductance value of the G element of the differential matching circuit of the fixture simulation function. (S4VNA only)
<i>Type</i>	Double (read/write)
<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
<i>Unit</i>	S (Siemens)
<i>Target</i>	Balanced Port <i>Bpt</i> of channel <i>Ch</i> , <i>Ch</i> : channel number 1–16 <i>Bpt</i> : balanced port number 1 or 2 for the Bal-Bal topology, always 1 for the SE-Bal, SE-SE-Bal and Bal topology.
<i>Syntax</i>	<i>Value</i> = app.SCPI.CALCulate(<i>Ch</i>).FSIMulator.BALun.DMCircuit.BPORT(<i>Bpt</i>).PARameters.G app.SCPI.CALCulate(<i>Ch</i>).FSIMulator.BALun.DMCircuit.BPORT(<i>Bpt</i>).PARameters.G = 0.1
<i>Equivalent Softkeys</i>	Analysis > Fixture Simulator > Diff Matching > Bal Port n > G

SCPI.CALCulate(*Ch*).FSIMulator.BALun.DMCircuit.BPORT(*Bpt*).PARameters.L

<i>Description</i>	Sets the inductance value of the L element of the differential matching circuit of the fixture simulation function. (S4VNA only)
<i>Type</i>	Double (read/write)
<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
<i>Unit</i>	H (Henry)
<i>Note</i>	If both elements L and R are equal to zero, then L and R elements are omitted in the scheme. If any element L or R is not zero, then zero value of the rest element means short circuit.
<i>Target</i>	Balanced Port <i>Bpt</i> of channel <i>Ch</i> , <i>Ch</i> : channel number 1–16 <i>Bpt</i> : balanced port number 1 or 2 for the Bal-Bal topology, always 1 for the SE-Bal, SE-SE-Bal and Bal topology.
<i>Syntax</i>	<i>Value</i> = app.SCPI.CALCulate(<i>Ch</i>).FSIMulator.BALun.DMCircuit.BPORT(<i>Bpt</i>).PARameters.L app.SCPI.CALCulate(<i>Ch</i>).FSIMulator.BALun.DMCircuit.BPORT(<i>Bpt</i>).PARameters.L = 12e-9
<i>Equivalent Softkeys</i>	Analysis > Fixture Simulator > Diff Matching > Bal Port n > L

SCPI.CALCulate(*Ch*).FSIMulator.BALun.DMCircuit.BPORT(*Bpt*).PARameters.R

<i>Description</i>	Sets the resistance value of the R element of the differential matching circuit of the fixture simulation function. (S4VNA only)
<i>Type</i>	Double (read/write)
<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
<i>Unit</i>	Ω (Ohm)
<i>Note</i>	If both elements L and R are equal to zero, then L and R elements are omitted in the scheme. If any element L or R is not zero, then zero value of the rest element means short circuit.
<i>Target</i>	Balanced Port <i>Bpt</i> of channel <i>Ch</i> , <i>Ch</i> : channel number 1–16 <i>Bpt</i> : balanced port number 1 or 2 for the Bal-Bal topology, always 1 for the SE-Bal, SE-SE-Bal and Bal topology.
<i>Syntax</i>	<i>Value</i> = app.SCPI.CALCulate(<i>Ch</i>).FSIMulator.BALun.DMCircuit.BPORT(<i>Bpt</i>).PARameters.R app.SCPI.CALCulate(<i>Ch</i>).FSIMulator.BALun.DMCircuit.BPORT(<i>Bpt</i>).PARameters.R = 100
<i>Equivalent Softkeys</i>	Analysis > Fixture Simulator > Diff Matching > Bal Port n > L

SCPI.CALCulate(Ch).FSIMulator.BALun.DMCircuit.BPORT(Bpt).TYPE

<i>Description</i>	Selects the type of the differential matching circuit for the specified balanced port number <i>Bpt</i> of the channel <i>Ch</i> . (S4VNA only)
<i>Type</i>	String (read/write)
<i>Parameter</i>	"NONE" : Specifies no-circuit "PLPC" : Specifies Shunt L – Shunt C circuit "USER" : Specifies user defined circuit by touchstone file
<i>Preset Value</i>	"NONE"
<i>Target</i>	Balanced Port <i>Bpt</i> of channel <i>Ch</i> , <i>Ch</i> : channel number 1–16 <i>Bpt</i> : balanced port number 1 or 2 for the Bal-Bal topology, always 1 for the SE-Bal, SE-SE-Bal and Bal topology.
<i>Syntax</i>	<i>Param</i> = app.SCPI.CALCulate(<i>Ch</i>).FSIMulator.BALun.DMCircuit.BPORT(<i>Bpt</i>).TYPE app.SCPI.CALCulate(<i>Ch</i>).FSIMulator.BALun.DMCircuit.BPORT(<i>Bpt</i>).TYPE = "PLPC"
<i>Equivalent Softkeys</i>	Analysis > Fixture Simulator > Diff Matching > Bal Port n > { None Shunt L – Shunt C User }

SCPI.CALCulate(*Ch*).FSIMulator.BALun.DMCircuit.BPORT(*Bpt*).USER.FILEname

<i>Description</i>	Specifies a file defining the 2-port network which is used in the differential matching circuit, for the specified balanced port number <i>Bpt</i> of the channel <i>Ch</i>. The *.s2p file contains the circuit S-parameters in Touchstone format. (S4VNA only)
<i>Type</i>	String (read/write)
<i>Parameter</i>	File Name
<i>Target</i>	Balanced Port <i>Bpt</i> of channel <i>Ch</i>, <i>Ch</i>: channel number 1–16 <i>Bpt</i>: balanced port number 1 or 2 for the Bal-Bal topology, always 1 for the SE-Bal, SE-SE-Bal and Bal topology.
<i>Syntax</i>	<i>File</i> = <i>app</i>.SCPI.CALCulate(<i>Ch</i>).FSIMulator.BALun.DMCircuit.BPORT(<i>Bpt</i>).USER.FILEname <i>app</i>.SCPI.CALCulate(<i>Ch</i>).FSIMulator.BALun.DMCircuit.BPORT(<i>Bpt</i>).USER.FILEname = "circuit.s2p"
<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>Related Commands</i>	SCPI.CALCulate(<i>Ch</i>).FSIMulator.BALun.DMCircuit.BPORT(<i>Bpt</i>).TYPE
<i>Equivalent Softkeys</i>	Analysis > Fixture Simulator > Diff Matching > Bal Port n > User File

SCPI.CALCulate(Ch).FSIMulator.BALun.DMCircuit.STATe

<i>Description</i>	Turns ON/OFF the differential matching circuit function. (S4VNA only)
<i>Type</i>	Boolean (read/write)
<i>Parameter</i>	True: differential matching ON False: differential matching OFF
<i>Preset Value</i>	False
<i>Target</i>	Channel <i>Ch</i> from 1 to 16
<i>Syntax</i>	<i>Status</i> = app.SCPI.CALCulate(<i>Ch</i>).FSIMulator.BALun.DMCircuit.STATe app.SCPI.CALCulate(<i>Ch</i>).FSIMulator.BALun.DMCircuit.STATe = True
<i>Equivalent Softkeys</i>	Analysis > Fixture Simulator > Diff Matching > Diff Matching [ON OFF]

SCPI.CALCulate(Ch).FSIMulator.BALun.DZConversion.STATe

<i>Description</i>	Turns ON/OFF the differential port impedance conversion function when the fixture simulator function is ON, for all the balance ports of selected channel (<i>Ch</i>). (S4VNA only)
<i>Type</i>	Boolean (read/write)
<i>Parameter</i>	True: differential port impedance conversion function is ON False: differential port impedance conversion function is OFF
<i>Preset Value</i>	False
<i>Target</i>	Channel <i>Ch</i> from 1 to 16
<i>Syntax</i>	<i>Status</i> = app.SCPI.CALCulate(<i>Ch</i>).FSIMulator.BALun.DZConversion.STATe app.SCPI.CALCulate(<i>Ch</i>).FSIMulator.BALun.DZConversion.STATe = True
<i>Equivalent Softkeys</i>	Analysis > Fixture Simulator > Diff ZConversion > Diff ZConversion [ON OFF]

SCPI.CALCulate(*Ch*).FSIMulator.BALun.DZConversion.BPORT(*Bpt*).Z0.R

<i>Description</i>	Sets the impedance value for the differential port impedance conversion function, for the balance ports 1 or 2 (<i>Bpt</i>) of the selected channel (<i>Ch</i>). (S4VNA only)
<i>Type</i>	Double (read/write)
<i>Range</i>	from 1e-3 to 1e7
<i>Out of Range</i>	Sets the value of the limit, which is closer to the specified value.
<i>Preset Value</i>	100
<i>Unit</i>	Ω (Ohm)
<i>Target</i>	Balanced Port <i>Bpt</i> of channel <i>Ch</i> , <i>Ch</i> : channel number 1–16 <i>Bpt</i> : balanced port number 1 or 2 for the Bal-Bal topology, always 1 for the SE-Bal, SE-SE-Bal and Bal topology.
<i>Syntax</i>	<i>Value</i> = app.SCPI.CALCulate(<i>Ch</i>).FSIMulator.BALun.DZConversion.BPORT(<i>Bpt</i>).Z0.R app.SCPI.CALCulate(<i>Ch</i>).FSIMulator.BALun.DZConversion.BPORT(<i>Bpt</i>).Z0.R = 200
<i>Equivalent Softkeys</i>	Analysis > Fixture Simulator > Diff ZConversion > Bal Port 1 or 2

SCPI.CALCulate(*Ch*).FSIMulator.BALun.PARAmeter(*Tr*).BALanced.DEFine

<i>Description</i>	Sets/gets the measurement parameter when the balance-unbalance conversion function is ON and the balance device type is "BALanced", for the selected trace (<i>Tr</i>) of the selected channel (<i>Ch</i>). (S4VNA only)
<i>Type</i>	String (read/write)
<i>Parameter</i>	"SDD11" : Specifies Sdd11 "SCD11" : Specifies Scd11 "SDC11" : Specifies Sdc11 "SCC11" : Specifies Scc11
<i>Out of Range</i>	An error occurs. Error code: 214.
<i>Preset Value</i>	" SDD11"
<i>Target</i>	Trace <i>Tr</i> of channel <i>Ch</i> , <i>Ch</i> : channel number 1–16 <i>Tr</i> : trace number 1–16
<i>Syntax</i>	<i>Param</i> = app.SCPI.CALCulate(<i>Ch</i>).FSIMulator.BALun.PARAmeter(<i>Tr</i>).BALanced.DEFine app.SCPI.CALCulate(<i>Ch</i>).FSIMulator.BALun.PARAmeter(<i>Tr</i>).BALanced.DEFine= "SDD11"
<i>Equivalent Softkeys</i>	Analysis > Fixture Simulator > Measurement [SDD11 SCD11 SDC11 SCC11]

SCPI.CALCulate(*Ch*).FSIMulator.BALun.PARAmeter(*Tr*).BBALanced.DEFine

<i>Description</i>	Sets/gets the measurement parameter when the balance-unbalance conversion function is ON and the balance device type is "BBALanced", for the selected trace (<i>Tr</i>) of the selected channel (<i>Ch</i>). (S4VNA only)	
<i>Type</i>	String (read/write)	
<i>Parameter</i>	"SDD11" : Specifies Sdd11 "SDD21" : Specifies Sdd21 "SDD12" : Specifies Sdd12 "SDD22" : Specifies Sdd22 "SCD11" : Specifies Scd11 "SCD21" : Specifies Scd21 "SCD12" : Specifies Scd12 "SCD22" : Specifies Scd22 "SDC11" : Specifies Sdc11 "SDC21" : Specifies Sdc21	"SDC12" : Specifies Sdc12 "SDC22" : Specifies Sdc22 "SCC11" : Specifies Scc11 "SCC21" : Specifies Scc21 "SCC12" : Specifies Scc12 "SCC22" : Specifies Scc22 "IMB1" : Specifies Imbalance1 "IMB2" : Specifies Imbalance2 "CMRR" : Specifies CMRR (Sdd21/Scc21)
<i>Preset Value</i>	" SDD11"	
<i>Target</i>	Trace <i>Tr</i> of channel <i>Ch</i> , <i>Ch</i> : channel number 1–16 <i>Tr</i> : trace number 1–16	
<i>Syntax</i>	Param = app.SCPI.CALCulate(<i>Ch</i>).FSIMulator.BALun.PARAmeter(<i>Tr</i>).BBALanced.DEFine app.SCPI.CALCulate(<i>Ch</i>).FSIMulator.BALun.PARAmeter(<i>Tr</i>).BBALanced.DEFine = "SDD11"	
<i>Equivalent Softkeys</i>	Analysis > Fixture Simulator > Measurement [SDD11 SDD21 ... CMRR]	

SCPI.CALCulate(*Ch*).FSIMulator.BALun.PARAmeter(*Tr*).SBALanced.DEFine

<i>Description</i>	Sets/gets the measurement parameter when the balance-unbalance conversion function is ON and the balance device type is "SBALanced", for the selected trace (<i>Tr</i>) of the selected channel (<i>Ch</i>). (S4VNA only)	
<i>Type</i>	String (read/write)	
<i>Parameter</i>	"SSS11" : Specifies Sss11 "SDS21" : Specifies Sds21 "SSD12" : Specifies Ssd12 "SCS21" : Specifies Scs21 "SSC12" : Specifies Ssc12 "SDD22" : Specifies Sdd22 "SCD22" : Specifies Scd22	"SDC22" : Specifies Sdc22 "SCC22" : Specifies Scs22 "IMB" : Specifies Imbalance "CMRR1" : Specifies CMRR1 (Sds21/Scs21) "CMRR2" : Specifies CMRR2 (Ssd12/Ssc12)
<i>Preset Value</i>	" SSS11 "	
<i>Target</i>	Trace <i>Tr</i> of channel <i>Ch</i> , <i>Ch</i> : channel number 1–16 <i>Tr</i> : trace number 1–16	
<i>Syntax</i>	<i>Param</i> = app.SCPI.CALCulate(<i>Ch</i>).FSIMulator.BALun.PARAmeter(<i>Tr</i>).SBALanced.DEFine app.SCPI.CALCulate(<i>Ch</i>).FSIMulator.BALun.PARAmeter(<i>Tr</i>).SBALanced.DEFine = "SSS1"	
<i>Equivalent Softkeys</i>	Analysis > Fixture Simulator > Measurement [SSS11 SDS21 ... CMRR2]	

SCPI.CALCulate(*Ch*).FSIMulator.BALun.PARAmeter(*Tr*).SSBalanced.DEFine

<i>Description</i>	Sets/gets the measurement parameter when the balance-unbalance conversion function is ON and the balance device type is "SSBalanced", for the selected trace (<i>Tr</i>) of the selected channel (<i>Ch</i>). (S4VNA only)	
<i>Type</i>	String (read/write)	
<i>Parameter</i>	"SSS11" : Specifies Sss11 "SSS21" : Specifies Sss21 "SSS12" : Specifies Sss12 "SSS22" : Specifies Sss22 "SDS31" : Specifies Sds31 "SDS32" : Specifies Sds32 "SSD13" : Specifies Ssd13 "SSD23" : Specifies Ssd23 "SCS31" : Specifies Scs31 "SCS32" : Specifies Scs32 "SSC13" : Specifies Ssc13 "SSC23" : Specifies Ssc23	"SDD33" : Specifies Sdd33 "SCD33" : Specifies Scd33 "SDC33" : Specifies Sdc33 "SCC33" : Specifies Scs33 "IMB1" : Specifies Imbalance1 "IMB2" : Specifies Imbalance2 "IMB3" : Specifies Imbalance3 "IMB4" : Specifies Imbalance4 "CMRR1" : Specifies CMRR1 (Sds31/Scs31) "CMRR2" : Specifies CMRR2 (Sds32/Scs32)
<i>Preset Value</i>	" SSS11 "	
<i>Target</i>	Trace <i>Tr</i> of channel <i>Ch</i> , <i>Ch</i> : channel number 1–16 <i>Tr</i> : trace number 1–16	
<i>Syntax</i>	<i>Param</i> = app.SCPI.CALCulate(<i>Ch</i>).FSIMulator.BALun.PARAmeter(<i>Tr</i>).SSBalanced.DEFine app.SCPI.CALCulate(<i>Ch</i>).FSIMulator.BALun.PARAmeter(<i>Tr</i>).SSBalanced.DEFine = "SSS1"	
<i>Equivalent Softkeys</i>	Analysis > Fixture Simulator > Measurement [SSS11 SSS21 ... CMRR2]	

SCPI.CALCulate(*Ch*).FSIMulator.BALun.PARAmeter(*Tr*).STATe

<i>Description</i>	Turns ON/OFF the state of the balance-unbalance conversion function. (S4VNA only)
<i>Type</i>	Boolean (read/write)
<i>Parameter</i>	True: balance-unbalance conversion function ON False: balance-unbalance conversion function OFF
<i>Preset Value</i>	False
<i>Target</i>	Trace <i>Tr</i> of channel <i>Ch</i> , <i>Ch</i> : channel number 1–16 <i>Tr</i> : trace number 1–16
<i>Syntax</i>	Status = app.SCPI.CALCulate(<i>Ch</i>).FSIMulator.BALun.PARAmeter(<i>Tr</i>).STATe app.SCPI.CALCulate(<i>Ch</i>).FSIMulator.BALun.PARAmeter(<i>Tr</i>).STATe = True
<i>Equivalent Softkeys</i>	Analysis > Fixture Simulator > Balun [ON OFF]

SCPI.CALCulate(Ch).FSIMulator.BALun.TOPology.PROPerTy.STATe

<i>Description</i>	Turns ON/OFF the state of the property display for the topology setting when using the balance-unbalance conversion. (S4VNA only)
<i>Type</i>	Boolean (read/write)
<i>Parameter</i>	True: property display ON False: property display OFF
<i>Preset Value</i>	False
<i>Target</i>	Channel <i>Ch</i> from 1 to 16
<i>Syntax</i>	Status = app.SCPI.CALCulate(Ch).FSIMulator.BALun.TOPology.PROPerTy.STATe app.SCPI.CALCulate(Ch).FSIMulator.BALun.TOPology.PROPerTy.STATe = True
<i>Equivalent Softkeys</i>	Analysis > Fixture Simulator > Topolgy > Property [ON OFF]

SCPI.CALCulate(Ch).FSIMulator.BALun.TOPology.BALanced.PPORTs

<i>Description</i>	Assigns each port when the balance device type is "BALanced" and the balance-unbalance conversion function is ON. The array contains 2 elements: <i>Data(0)</i> specifies port number assigned to port a of balanced device; <i>Data(1)</i> specifies port number assigned to port b of balanced device. (S4VNA only)
<i>Type</i>	Variant: array of long (read/write)
<i>Parameter</i>	Port number from 1 to 4. 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>Target</i>	Channel <i>Ch</i> from 1 to 16
<i>Syntax</i>	app.SCPI.CALCulate(<i>Ch</i>).FSIMulator.BALun.TOPology.BALanced.PPORTs = Array(1,2) Ports = app.SCPI.CALCulate(<i>Ch</i>).FSIMulator.BALun.TOPology.BALanced.PPORTs
<i>Related Commands</i>	SCPI.CALCulate(<i>Ch</i>).FSIMulator.BALun.DEVICE
<i>Equivalent Softkeys</i>	Analysis > Fixture Simulator > Topolgy > Port 1 (bal)

SCPI.CALCulate(Ch).FSIMulator.BALun.TOPology.BBALanced.PPORTs

<i>Description</i>	Assigns each port when the balance device type is "BBALanced" and the balance-unbalance conversion function is ON. The array contains 4 elements: <i>Data(0)</i> specifies port number assigned to port a of balanced device; <i>Data(1)</i> specifies port number assigned to port b of balanced device. <i>Data(2)</i> specifies port number assigned to port c of balanced device. <i>Data(3)</i> specifies port number assigned to port d of balanced device. (S4VNA only)
<i>Type</i>	Variant: array of long (read/write)
<i>Parameter</i>	Port number is 1 to 4. 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>Target</i>	Channel <i>Ch</i> from 1 to 16
<i>Syntax</i>	app.SCPI.CALCulate(<i>Ch</i>).FSIMulator.BALun.TOPology.BBALanced.PPORTs = Array(1,2,3,4) Ports = app.SCPI.CALCulate(<i>Ch</i>).FSIMulator.BALun.TOPology.BBALanced.PPORTs
<i>Related Commands</i>	SCPI.CALCulate(<i>Ch</i>).FSIMulator.BALun.DEVICE
<i>Equivalent Softkeys</i>	Analysis > Fixture Simulator > Topolgy > Port 1 (bal), Port 2 (bal)

SCPI.CALCulate(Ch).FSIMulator.BALun.TOPology.SBALanced.PPORTs

<i>Description</i>	Assigns each port when the balance device type is "SBALanced" and the balance-unbalance conversion function is ON. The array contains 3 elements: <i>Data(0)</i> specifies port number assigned to port a of balanced device; <i>Data(1)</i> specifies port number assigned to port b of balanced device. <i>Data(2)</i> specifies port number assigned to port c of balanced device. (S4VNA only)
<i>Type</i>	Variant: array of long (read/write)
<i>Parameter</i>	Port number is 1 to 4. 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>Target</i>	Channel <i>Ch</i> from 1 to 16
<i>Syntax</i>	app.SCPI.CALCulate(<i>Ch</i>).FSIMulator.BALun.TOPology.SBALanced.PPORTs = Array(1,2,3) Ports = app.SCPI.CALCulate(<i>Ch</i>).FSIMulator.BALun.TOPology.SBALanced.PPORTs
<i>Related Commands</i>	SCPI.CALCulate(<i>Ch</i>).FSIMulator.BALun.DEVICE
<i>Equivalent Softkeys</i>	Analysis > Fixture Simulator > Topolgy > Port 1 (se), Port 2 (bal)

SCPI.CALCulate(Ch).FSIMulator.BALun.TOPology.SSBalanced.PPORTs

<i>Description</i>	Assigns each port when the balance device type is "SSBalanced" and the balance-unbalance conversion function is ON. The array contains 3 elements: <i>Data(0)</i> specifies port number assigned to port a of balanced device; <i>Data(1)</i> specifies port number assigned to port b of balanced device. <i>Data(2)</i> specifies port number assigned to port c of balanced device. <i>Data(3)</i> specifies port number assigned to port d of balanced device. (S4VNA only)
<i>Type</i>	Variant: array of long (read/write)
<i>Parameter</i>	Port number is 1 to 4. 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>Target</i>	Channel <i>Ch</i> from 1 to 16
<i>Syntax</i>	app.SCPI.CALCulate(<i>Ch</i>).FSIMulator.BALun.TOPology.SSBalanced.PPORTs = Array(1,2,3,4) Ports = app.SCPI.CALCulate(<i>Ch</i>).FSIMulator.BALun.TOPology.SSBalanced.PPORTs
<i>Related Commands</i>	SCPI.CALCulate(<i>Ch</i>).FSIMulator.BALun.DEVICE
<i>Equivalent Softkeys</i>	Analysis > Fixture Simulator > Topolgy > Port 1 (se), Port 2 (se) , Port 3 (bal)

SCPI.CALCulate(*Ch*).FSIMulator.EMBed.NETWork(*Nwk*).FILename

<i>Description</i>	Specifies a file defining the 4-port network which is used in the 4-port network embedding/de-embedding function. The file is a 4-port touchstone file with the ".s4p" extension. The network number (<i>Nwk</i>) is 1 or 2 depending on the selected topology: SCPI.CALCulate(<i>Ch</i>).FSIMulator.EMBed.TYPE (S4VNA only)
<i>Type</i>	String (read/write)
<i>Parameter</i>	File Name
<i>Target</i>	Channel <i>Ch</i> from 1 to 16
<i>Syntax</i>	File = app.SCPI.CALCulate(<i>Ch</i>).FSIMulator.EMBed.NETWork(<i>Nwk</i>).FILename app.SCPI.CALCulate(<i>Ch</i>).FSIMulator.EMBed.NETWork(<i>Nwk</i>).FILename="network.s4p"
<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>Related Commands</i>	SCPI.CALCulate(<i>Ch</i>).FSIMulator.EMBed.TYPE
<i>Equivalent Softkeys</i>	Analysis > Fixture Simulator > De-Embedding s4p > File (Nwk1), File (Nwk2)

SCPI.CALCulate(Ch).FSIMulator.EMBed.NETWork(Nwk).TYPE

<i>Description</i>	Sets/gets the processing type for network (<i>Nwk</i>), for the 4-port network embedding/de-embedding feature. The network number (<i>Nwk</i>) is 1 or 2 depending on the selected topology: SCPI.CALCulate(<i>Ch</i>).FSIMulator.EMBed.TYPE (S4VNA only)
<i>Type</i>	String (read/write)
<i>Parameter</i>	"NONE" : Specifies no-processing "EMBed" : Specifies embedding "DEEMbed" : Specifies de-embedding
<i>Preset Value</i>	"NONE"
<i>Target</i>	Channel <i>Ch</i> from 1 to 16
<i>Syntax</i>	Param = app.SCPI.CALCulate(<i>Ch</i>).FSIMulator.EMBed.NETWork(<i>Nwk</i>).TYPE app.SCPI.CALCulate(<i>Ch</i>).FSIMulator.EMBed.NETWork(<i>Nwk</i>).TYPE = "EMBed"
<i>Related Commands</i>	SCPI.CALCulate(<i>Ch</i>).FSIMulator.EMBed.TYPE
<i>Equivalent Softkeys</i>	Analysis > Fixture Simulator > De-Embedding s4p > Type (Nwk1), Type (Nwk2)

SCPI.CALCulate(Ch).FSIMulator.EMBed.STATe

<i>Description</i>	Turns ON/OFF the state of the 4-port network embedding/de-embedding feature when the fixture simulator feature is ON. (S4VNA only)
<i>Type</i>	Boolean (read/write)
<i>Parameter</i>	True: 4-port network embedding/de-embedding feature ON False: 4-port network embedding/de-embedding feature OFF
<i>Preset Value</i>	False
<i>Target</i>	Channel <i>Ch</i> from 1 to 16
<i>Syntax</i>	Status = app.SCPI.CALCulate(<i>Ch</i>).FSIMulator.EMBed.STATe app.SCPI.CALCulate(<i>Ch</i>).FSIMulator.EMBed.STATe = True
<i>Related Commands</i>	SCPI.CALCulate(<i>Ch</i>).FSIMulator.STATe
<i>Equivalent Softkeys</i>	Analysis > Fixture Simulator > De-Embedding s4p > De-Embedding s4p [ON OFF]

SCPI.CALCulate(*Ch*).FSIMulator.EMBed.TOPology.A.PORTs

<i>Description</i>	Sets/gets the test port assignment when the connection type (Topology) is set to A, for the 4-port network embedding/de-embedding feature. The array contains 2 elements: <i>Data(0)</i> Port number assigned to port a; <i>Data(1)</i> Port number assigned to port b. (S4VNA only)
<i>Type</i>	Variant: array of long (read/write)
<i>Parameter</i>	Port number is 1 to 4. 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>Target</i>	Channel <i>Ch</i> from 1 to 16
<i>Syntax</i>	app.SCPI.CALCulate(<i>Ch</i>).FSIMulator.EMBed.TOPology.A.PORTs = Array(1,2) Ports = app.SCPI.CALCulate(<i>Ch</i>).FSIMulator.EMBed.TOPology.A.PORTs
<i>Related Commands</i>	SCPI.CALCulate(<i>Ch</i>).FSIMulator.EMBed.TYPE
<i>Equivalent Softkeys</i>	Analysis > Fixture Simulator > De–Embedding s4p > Ports

SCPI.CALCulate(Ch).FSIMulator.EMBed.TOPology.B.PORTs

<i>Description</i>	Sets/gets the test port assignment when the connection type (Topology) is set to B, for the 4-port network embedding/de-embedding feature. The array contains 3 elements: <i>Data(0)</i> Port number assigned to port a; <i>Data(1)</i> Port number assigned to port b; <i>Data(2)</i> Port number assigned to port c. (S4VNA only)
<i>Type</i>	Variant: array of long (read/write)
<i>Parameter</i>	Port number is 1 to 4. 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>Target</i>	Channel <i>Ch</i> from 1 to 16
<i>Syntax</i>	app.SCPI.CALCulate(<i>Ch</i>).FSIMulator.EMBed.TOPology.B.PORTs = Array(1,2,3) Ports = app.SCPI.CALCulate(<i>Ch</i>).FSIMulator.EMBed.TOPology.B.PORTs
<i>Related Commands</i>	SCPI.CALCulate(<i>Ch</i>).FSIMulator.EMBed.TYPE
<i>Equivalent Softkeys</i>	Analysis > Fixture Simulator > De-Embedding s4p > Ports

SCPI.CALCulate(Ch).FSIMulator.EMBed.TOPology.C.PORTs

<i>Description</i>	Sets/gets the test port assignment when the connection type (Topology) is set to C, for the 4-port network embedding/de-embedding feature. The array contains 3 elements: <i>Data(0)</i> Port number assigned to port a; <i>Data(1)</i> Port number assigned to port a; <i>Data(2)</i> Port number assigned to port b; <i>Data(3)</i> Port number assigned to port c. (S4VNA only)
<i>Type</i>	Variant: array of long (read/write)
<i>Parameter</i>	Port number is 1 to 4. 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>Target</i>	Channel <i>Ch</i> from 1 to 16
<i>Syntax</i>	app.SCPI.CALCulate(<i>Ch</i>).FSIMulator.EMBed.TOPology.C.PORTs = Array(1,2,3,4) Ports = app.SCPI.CALCulate(<i>Ch</i>).FSIMulator.EMBed.TOPology.C.PORTs
<i>Related Commands</i>	SCPI.CALCulate(<i>Ch</i>).FSIMulator.EMBed.TYPE
<i>Equivalent Softkeys</i>	Analysis > Fixture Simulator > De-Embedding s4p > Ports

SCPI.CALCulate(Ch).FSIMulator.EMBed.TYPE

<i>Description</i>	Sets/gets the connection type (Topology), for the 4-port network embedding/de-embedding feature (S4VNA only)
<i>Type</i>	String (read/write)
<i>Parameter</i>	" A " : Specifies connection type A " B " : Specifies connection type B " C " : Specifies connection type C
<i>Preset Value</i>	"A"
<i>Target</i>	Channel <i>Ch</i> from 1 to 16
<i>Syntax</i>	Param = app.SCPI.CALCulate(<i>Ch</i>).FSIMulator.EMBed.TYPE app.SCPI.CALCulate(<i>Ch</i>).FSIMulator.EMBed.TYPE = "A"
<i>Equivalent Softkeys</i>	Analysis > Fixture Simulator > De-Embedding s4p > Topology [A B C]

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

<i>Description</i>	Sets/gets the de-embedding network file (*.s2p). The file contains the circuit S-parameters in Touchstone format.
<i>Type</i>	String (read/write)
<i>Parameter</i>	File Name
<i>Target</i>	Port <i>Pt</i> of channel <i>Ch</i> , <i>Ch</i> : channel number 1–16 <i>Pt</i> : port number 1–2 (4 for S4VNA)
<i>Syntax</i>	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"
<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 > User File

SCPI.CALCulate(Ch).FSIMulator.SENDEd.DEEMbed.PORT(Pt).STATe

<i>Description</i>	Turns ON/OFF the state of the 2-port network de-embedding function for specified port (s2p).
<i>Type</i>	Boolean (read/write)
<i>Parameter</i>	True: De-embedding function ON False: De-embedding function OFF
<i>Preset Value</i>	False
<i>Target</i>	Port <i>Pt</i> of channel <i>Ch</i> , <i>Ch</i> : channel number 1–16 <i>Pt</i> : port number 1–2 (4 for S4VNA)
<i>Syntax</i>	Status = app.SCPI.CALCulate(Ch).FSIMulator.SENDEd.DEEMbed.PORT(Pt).STATe app.SCPI.CALCulate(Ch).FSIMulator.SENDEd.DEEMbed.PORT(Pt).STATe = True
<i>Equivalent Softkeys</i>	Analysis > Fixture Simulator > De–Embedding > Port n

SCPI.CALCulate(Ch).FSIMulator.SENDEd.DEEMbed.STATe

<i>Description</i>	Turns ON/OFF the state of the 2-port network de-embedding function.
<i>Type</i>	Boolean (read/write)
<i>Parameter</i>	True: De-embedding function ON False: De-embedding function OFF
<i>Preset Value</i>	False
<i>Target</i>	Channel <i>Ch</i> from 1 to 16
<i>Syntax</i>	Status = app.SCPI.CALCulate(Ch).FSIMulator.SENDEd.DEEMbed.STATe app.SCPI.CALCulate(Ch).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>Description</i>	Sets/gets the embedding network file (*.s2p). The file contains the circuit S-parameters in Touchstone format.
<i>Type</i>	String (read/write)
<i>Parameter</i>	File Name
<i>Target</i>	Port <i>Pt</i> of channel <i>Ch</i> , <i>Ch</i> : channel number 1–16 <i>Pt</i> : port number 1–2 (4 for S4VNA)
<i>Syntax</i>	<i>File</i> = 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"
<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 > User File

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

<i>Description</i>	Turns ON/OFF the state of the 2-port network embedding function for specified port.
<i>Type</i>	Boolean (read/write)
<i>Parameter</i>	True: Embedding function ON False: Embedding function OFF
<i>Preset Value</i>	False
<i>Target</i>	Port <i>Pt</i> of channel <i>Ch</i> , <i>Ch</i> : channel number 1–16 <i>Pt</i> : port number 1–2 (4 for S4VNA)
<i>Syntax</i>	Status = 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 > Port n

SCPI.CALCulate(*Ch*).FSIMulator.SENDEd.PMCircuit.STATe

<i>Description</i>	Turns ON/OFF the state of the 2-port network embedding function.
<i>Type</i>	Boolean (read/write)
<i>Parameter</i>	True: Embedding function ON False: Embedding function OFF
<i>Preset Value</i>	False
<i>Target</i>	Channel <i>Ch</i> from 1 to 16
<i>Syntax</i>	Status = 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>Description</i>	Sets/gets the value of the impedance for port impedance conversion function.
<i>Type</i>	Double (read/write)
<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>Target</i>	Port <i>Pt</i> of channel <i>Ch</i> , <i>Ch</i> : channel number 1–16 <i>Pt</i> : port number 1–2 (4 for S4VNA)
<i>Syntax</i>	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 = 50
<i>Equivalent Softkeys</i>	Analysis > Fixture Simulator > Port Z Conversion > Port n Z0

SCPI.CALCulate(Ch).FSIMulator.SENDeD.ZCONversion.STATe

<i>Description</i>	Turns ON/OFF the state of the port impedance conversion function.
<i>Type</i>	Boolean (read/write)
<i>Parameter</i>	True: Port Z conversion function ON False: Port Z conversion function OFF
<i>Preset Value</i>	False
<i>Target</i>	Channel <i>Ch</i> from 1 to 16
<i>Syntax</i>	Status = app.SCPI.CALCulate(<i>Ch</i>).FSIMulator.SENDeD.ZCONversion.STATe app.SCPI.CALCulate(<i>Ch</i>).FSIMulator.SENDeD.ZCONversion.STATe = True
<i>Equivalent Softkeys</i>	Analysis > Fixture Simulator > Port Z Conversion > Port Z Conversion

SCPI.CALCulate(Ch).FSIMulator.STATe

<i>Description</i>	Turns ON/OFF the state of the fixture simulator function for specified channel.
<i>Type</i>	Boolean (read/write)
<i>Parameter</i>	True: fixture simulator function ON False: fixture simulator function OFF
<i>Preset Value</i>	False
<i>Target</i>	Channel <i>Ch</i> from 1 to 16
<i>Syntax</i>	Status = app.SCPI.CALCulate(<i>Ch</i>).FSIMulator.STATe app.SCPI.CALCulate(<i>Ch</i>).FSIMulator.STATe = True
<i>Equivalent Softkeys</i>	Analysis > Fixture Simulator > Fixture Simulator

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

<i>Description</i>	Sets/gets the number of traces in the channel.
<i>Type</i>	Long (read/write)
<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>Target</i>	Channel <i>Ch</i> from 1 to 16
<i>Syntax</i>	<i>TraceNum</i> = app.SCPI.CALCulate(<i>Ch</i>).PARAmeter.COUNT app.SCPI.CALCulate(<i>Ch</i>).PARAmeter.COUNT = 2
<i>Warning</i>	Object <i>PARAmeter</i> has an index of 1, which can be omitted in Visual Basic but it cannot be omitted in other programming languages.
<i>Equivalent Softkeys</i>	Display > Num of Traces

SCPI.CALCulate(Ch).PARAmeter(Tr).DEFine

<i>Description</i>	Sets/gets the measurement parameter of the trace.
<i>Type</i>	String (read/write)
<i>Parameter</i>	"S11", "S12", "S13", "S14", "S21", "S22", "S23", "S24", "S31", "S32", "S33", "S34", "S41", "S42", "S43", "S44" : S – parameter "T1", "T2", "T3", "T4" : Test receiver "R1", "R2", "R3", "R4" : Reference receiver "A", "B", "C", "D" : Alias for T1, T2, T3, T4
<i>Related Commands</i>	SCPI.CALCulate(Ch).PARAmeter(Tr).SPORT
<i>Target</i>	Trace <i>Tr</i> of channel <i>Ch</i> , <i>Ch</i> : channel number 1–16 <i>Tr</i> : trace number 1–16
<i>Syntax</i>	<code>StrMeas = app.SCPI.CALCulate(Ch).PARAmeter(Tr).DEFine</code> <code>app.SCPI.CALCulate(Ch).PARAmeter(Tr).DEFine = "S11"</code>
<i>Equivalent Softkeys</i>	(S2VNA) Measurement > S11 S21 S12 S22 Measurement > Absolute > { A(1) B(1) R1(1) A(2) B(2) R2(2) } (S4VNA) Measurement > S-parameter > S11 S12 ... S44 Measurement > Test Receiver > T1(1) T1(2) ... T4(4) Measurement > Reference Receiver > R1(1) R1(2) ... R4(4)

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

<i>Description</i>	Sets the active channel and trace.
<i>Type</i>	Method
<i>Target</i>	Trace <i>Tr</i> of channel <i>Ch</i> , <i>Ch</i> : channel number 1–16 <i>Tr</i> : trace number 1–16
<i>Syntax</i>	app.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>Description</i>	Sets/gets the number of the source port for absolute measurements.
<i>Type</i>	Long (read/write)
<i>Parameter</i>	Port number from 1 to 2 (4 for S4VNA)
<i>Out of Range</i>	An error occurs. Error code: 208.
<i>Preset Value</i>	1
<i>Target</i>	Trace <i>Tr</i> of channel <i>Ch</i> , <i>Ch</i> : channel number 1–16 <i>Tr</i> : trace number 1–16
<i>Syntax</i>	<i>StimPort</i> = app.SCPI.CALCulate(<i>Ch</i>).PARAmeter(<i>Tr</i>).SPORT app.SCPI.CALCulate(<i>Ch</i>).PARAmeter(<i>Tr</i>).SPORT = 1
<i>Equivalent Softkeys</i>	(S2VNA) Measurement > Measurement > Absolute > A(1) B(1) R1(1) A(2) B(2) R2(2) (S4VNA) Measurement > Test Receiver > T1(1) T1(2) ... T4(4) Measurement > Reference Receiver > R1(1) R1(2) ... R4(4)

SCPI.CALCulate(Ch).SElected.CONVersion.FUNction

<i>Description</i>	Sets/gets the S-parameter conversion function type.
<i>Type</i>	String (read/write)
<i>Parameter</i>	"ZREFlection" : Reflection equivalent impedance "ZTRansmit" : Transmission equivalent impedance "YREFlection" : Reflection equivalent admittance "YTRansmit" : Transmission equivalent admittance "INVersion" : Inverse S-parameter "ZTSHunt" : Shunt equivalent impedance "YTSHunt" : Shunt equivalent admittance "CONJugation" : S-parameter conjugate
<i>Preset Value</i>	"ZREF"
<i>Target</i>	The active trace of channel <i>Ch</i> , <i>Ch</i> : channel number 1–16
<i>Syntax</i>	Param = app.SCPICALCulate(Ch).SElected.CONVersion.FUNction app.SCPICALCulate(Ch).SElected.CONVersion.FUNction = "ZTR"
<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>Description</i>	Turns ON/OFF the state of the S-parameter conversion function.
<i>Type</i>	Boolean (read/write)
<i>Parameter</i>	True: S-parameter conversion function ON False: S-parameter conversion function OFF
<i>Preset Value</i>	False
<i>Target</i>	The active trace of channel <i>Ch</i> , <i>Ch</i> : channel number 1–16
<i>Syntax</i>	<i>Status</i> = app.SCPi.CALCulate(<i>Ch</i>).SElected.CONVersion.STATe app.SCPi.CALCulate(<i>Ch</i>).SElected.CONVersion.STATe = true
<i>Related Commands</i>	SCPI.CALCulate(<i>Ch</i>).SElected.CONVersion.FUNction
<i>Equivalent Softkeys</i>	Analysis > Conversion > Conversion

SCPI.CALCulate(Ch).SElected.CORRection.EDELay.TIME

<i>Description</i>	Sets/gets the value of the electrical delay.
<i>Type</i>	Double (read/write)
<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>Target</i>	The active trace of channel <i>Ch</i> , <i>Ch</i> : channel number 1–16
<i>Syntax</i>	<i>Value</i> = app.SCPI.CALCulate(<i>Ch</i>).SElected.CORRection.EDELay.TIME app.SCPI.CALCulate(<i>Ch</i>).SElected.CORRection.EDELay.TIME = 1e–9
<i>Equivalent Softkeys</i>	Scale > Electrical Delay

SCPI.CALCulate(Ch).SElected.CORRection.OFFSet.PHASE

<i>Description</i>	Sets/gets the value of the phase offset.
<i>Type</i>	Double (read/write)
<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>Target</i>	The active trace of channel <i>Ch</i> , <i>Ch</i> : channel number 1–16
<i>Syntax</i>	<i>Value</i> = app.SCPi.CALCulate(<i>Ch</i>).SElected.CORRection.OFFSet.PHASE app.SCPi.CALCulate(<i>Ch</i>).SElected.CORRection.OFFSet.PHASE = 360
<i>Equivalent Softkeys</i>	Scale > Phase Offset

SCPI.CALCulate(Ch).SElected.CORRection.STATus

<i>Description</i>	Gets the active trace correction status.
<i>Type</i>	String (read only)
<i>Parameter</i>	If active trace represents a S-parameter: "COR" : Correction is enabled. "C?" : Correction is enabled. Interpolation is applied. "C!" : Correction is enabled. Extrapolation is applied. " " : Correction is disabled. If active trace represents an Absolute measurement: "RC" : Correction is enabled. "RC?" : Correction is enabled. Interpolation is applied. "RC!" : Correction is enabled. Extrapolation is applied. " " : Correction is disabled.
<i>Target</i>	The active trace of channel <i>Ch</i> , <i>Ch</i> : channel number 1–16
<i>Syntax</i>	<i>Param</i> = <i>app</i> .SCPI.CALCulate(<i>Ch</i>).SElected.CORRection.STATus
<i>Equivalent Softkeys</i>	None

SCPI.CALCulate(Ch).SElected.DATA.FDATa

<i>Description</i>	Gets/sets the formatted data array. The array elements contain measurements in the current format, for example, in logarithmic magnitude format (Log Mag). Also, see section 14 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(2n-2)</i> real number in rectangular format, real part in polar and Smith chart formats; <i>Data(2n-1)</i> 0 in rectangular format, imaginary part in polar and Smith chart formats.
<i>Type</i>	Variant: array of double (read/write)
<i>Target</i>	The active trace of channel <i>Ch</i>, <i>Ch</i>: channel number 1–16
<i>Syntax</i>	<i>Data</i> = app.SCPICALCulate(<i>Ch</i>).SElected.DATA.FDATa app.SCPICALCulate(<i>Ch</i>).SElected.DATA.FDATa = <i>Data</i>
<i>Equivalent Softkeys</i>	None

SCPI.CALCulate(Ch).SElected.DATA.FMEMory

<i>Description</i>	Gets/sets 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 14. 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>Type</i>	Variant: array of double (read/write)
<i>Target</i>	The active trace of channel <i>Ch</i>, <i>Ch</i>: channel number 1–16
<i>Syntax</i>	<i>Data</i> = app.SCPI.CALCulate(<i>Ch</i>).SElected.DATA.FMEMory app.SCPI.CALCulate(<i>Ch</i>).SElected.DATA.FMEMory = <i>Data</i>
<i>Equivalent Softkeys</i>	None

SCPI.CALCulate(*Ch*).SElected.DATA.SDATa

<i>Description</i>	Gets/Sets the corrected data array. The corrected measurements are complex numbers. Also, see section 14. 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>(2<i>n</i>–2) the real part of corrected measurement; <i>Data</i>(2<i>n</i>–1) the imaginary part of corrected measurement.
<i>Type</i>	Variant: array of double (read/write)
<i>Target</i>	The active trace of channel <i>Ch</i>, <i>Ch</i>: channel number 1–16
<i>Syntax</i>	<i>Data</i> = app.SCPI.CALCulate(<i>Ch</i>).SElected.DATA.SDATa app.SCPI.CALCulate(<i>Ch</i>).SElected.DATA.SDATa = <i>Data</i>
<i>Equivalent Softkeys</i>	None

SCPI.CALCulate(Ch).SElected.DATA.SMEMory

<i>Description</i>	Gets/sets the corrected memory array. The corrected measurements are complex numbers. Also, see section 14. The array size is $2N$, where N is the number of measurement points. For the n-th point, where n from 1 to N: $Data(2n-2)$ the real part of corrected measurement memory; $Data(2n-1)$ the imaginary part of corrected measurement memory.
<i>Type</i>	Variant: array of double (read/write)
<i>Target</i>	The active trace of channel Ch , Ch : channel number 1–16
<i>Syntax</i>	$Data = \text{app.SCPICALCulate}(Ch).\text{SElected.DATA.SMEMory}$ $\text{app.SCPICALCulate}(Ch).\text{SElected.DATA.SMEMory} = Data$
<i>Equivalent Softkeys</i>	None

SCPI.CALCulate(Ch).SElected.DATA.XAXis

<i>Description</i>	Gets the X axis data array. The array size is N, where N is the number of measurement points. For the n-th point, where n from 0 to $N-1$: $Data(n)$ the X axis value;
<i>Type</i>	Variant: array of double (read only)
<i>Target</i>	The active trace of channel Ch , Ch : channel number 1–16
<i>Syntax</i>	$Data = \text{app.SCPICALCulate}(Ch).\text{SElected.DATA.XAXis}$
<i>Equivalent Softkeys</i>	None

SCPI.CALCulate(Ch).SElected.FILTer.GATE.TIME.CENTer

<i>Description</i>	Sets/gets the gate center value of the gating function.
<i>Type</i>	Double (read/write)
<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>Target</i>	The active trace of channel <i>Ch</i> , <i>Ch</i> : channel number 1–16
<i>Syntax</i>	Value = app.SCPI.CALCulate(<i>Ch</i>).SElected.FILTer.GATE.TIME.CENTer app.SCPI.CALCulate(<i>Ch</i>).SElected.FILTer.GATE.TIME.CENTer = 1e–8
<i>Equivalent Softkeys</i>	Analysis > Gating > Center

SCPI.CALCulate(Ch).SElected.FILTer.GATE.TIME.SHAPe

<i>Description</i>	Sets/gets the gate shape of the gating function.
<i>Type</i>	String (read/write)
<i>Parameter</i>	"MAXimum" : Maximum shape "WIDE" : Wide shape "NORMAl" : Normal shape "MINimum" : Minimum shape
<i>Preset Value</i>	"NORM"
<i>Target</i>	The active trace of channel <i>Ch</i> , <i>Ch</i> : channel number 1–16
<i>Syntax</i>	<i>Param</i> = app.SCPi.CALCulate(<i>Ch</i>).SElected.FILTer.GATE.TIME.SHAPe app.SCPi.CALCulate(<i>Ch</i>).SElected.FILTer.GATE.TIME.SHAPe = "MAX"
<i>Equivalent Softkeys</i>	Analysis > Gating > Shape > Maximum Wide Normal Minimum

SCPI.CALCulate(Ch).SElected.FILTer.GATE.TIME.SPAN

<i>Description</i>	Sets/gets the gate span value of the gating function.
<i>Type</i>	Double (read/write)
<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>Target</i>	The active trace of channel <i>Ch</i> , <i>Ch</i> : channel number 1-16
<i>Syntax</i>	<i>Value</i> = app.SCPI.CALCulate(<i>Ch</i>).SElected.FILTer.GATE.TIME.SPAN app.SCPI.CALCulate(<i>Ch</i>).SElected.FILTer.GATE.TIME.SPAN = 1e-8
<i>Equivalent Softkeys</i>	Analysis > Gating > Span

SCPI.CALCulate(Ch).SElected.FILTer.GATE.TIME.STARTt

<i>Description</i>	Sets/gets the gate start value of the gating function.
<i>Type</i>	Double (read/write)
<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>Target</i>	The active trace of channel <i>Ch</i> , <i>Ch</i> : channel number 1–16
<i>Syntax</i>	<i>Value</i> = app.SCPi.CALCulate(<i>Ch</i>).SElected.FILTer.GATE.TIME.STARTt app.SCPi.CALCulate(<i>Ch</i>).SElected.FILTer.GATE.TIME.STARTt = 1e-7
<i>Equivalent Softkeys</i>	Analysis > Gating > Start

SCPI.CALCulate(Ch).SElected.FILTer.GATE.TIME.STATe

<i>Description</i>	Turns ON/OFF the state of the gating function.
<i>Type</i>	Boolean (read/write)
<i>Parameter</i>	True: Gating function ON False: Gating function OFF
<i>Preset Value</i>	False
<i>Target</i>	The active trace of channel <i>Ch</i> , <i>Ch</i> : channel number 1–16
<i>Syntax</i>	<i>Status</i> = app.SCPi.CALCulate(<i>Ch</i>).SElected.FILTer.GATE.TIME.STATe app.SCPi.CALCulate(<i>Ch</i>).SElected.FILTer.GATE.TIME.STATe = <i>Status</i>
<i>Equivalent Softkeys</i>	Analysis > Gating > Gating

SCPI.CALCulate(Ch).SElected.FILTer.GATE.TIME.STOP

<i>Description</i>	Sets/gets the gate stop value of the gating function.
<i>Type</i>	Double (read/write)
<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>Target</i>	The active trace of channel <i>Ch</i> , <i>Ch</i> : channel number 1-16
<i>Syntax</i>	<i>Value</i> = app.SCPi.CALCulate(<i>Ch</i>).SElected.FILTer.GATE.TIME.STOP app.SCPi.CALCulate(<i>Ch</i>).SElected.FILTer.GATE.TIME.STOP = 1e-7
<i>Equivalent Softkeys</i>	Analysis > Gating > Stop

SCPI.CALCulate(Ch).SElected.FILTer.GATE.TIME.TYPE

<i>Description</i>	Sets/gets the gate type of the gating function.
<i>Type</i>	String (read/write)
<i>Parameter</i>	"BPASs" : Bandpass type "NOTCh" : Notch type
<i>Preset Value</i>	"BPAS"
<i>Target</i>	The active trace of channel <i>Ch</i> , <i>Ch</i> : channel number 1-16
<i>Syntax</i>	<i>Param</i> = app.SCPi.CALCulate(<i>Ch</i>).SElected.FILTer.GATE.TIME.TYPE app.SCPi.CALCulate(<i>Ch</i>).SElected.FILTer.GATE.TIME.TYPE = "bpas"
<i>Equivalent Softkeys</i>	Analysis > Gating > Type

SCPI.CALCulate(Ch).SElected.FORMat

<i>Description</i>	Sets/gets the data format for the trace.
<i>Type</i>	String (read/write)
<i>Parameter</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>Preset Value</i>	"MLOG"
<i>Target</i>	The active trace of channel <i>Ch</i> , <i>Ch</i> : channel number 1–16
<i>Syntax</i>	<i>Param</i> = app.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 Phase > 1 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>Description</i>	Gets the data array - result of analysis executed by the SCPI.CALCulate(<i>Ch</i>).SElected.FUNction.EXECute method. The array size is 2N, where N is the number of points defined by the SCPI.CALCulate(<i>Ch</i>).SElected.FUNction.POINts command. For the n-th point, where n from 1 to N: <i>Data</i>(2<i>n</i>-2) the response value in the n-th measurement point; <i>Data</i>(2<i>n</i>-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>Type</i>	Variant: array of double (read only)
<i>Target</i>	The active trace of channel <i>Ch</i> , <i>Ch</i> : channel number 1-16
<i>Syntax</i>	<i>Data</i> = app.SCPI.CALCulate(<i>Ch</i>).SElected.FUNction.DATA
<i>Equivalent Softkeys</i>	None

SCPI.CALCulate(Ch).SElected.FUNCTION.DOMain.COUPle

<i>Description</i>	Turns ON/OFF the coupling state of the analysis range for the SCPI.CALCulate(Ch).SElected.FUNCTION.EXECute method.
<i>Type</i>	Boolean (read/write)
<i>Parameter</i>	True: Coupling state ON False: Coupling state OFF
<i>Preset Value</i>	True
<i>Target</i>	All traces of channel <i>Ch</i> , <i>Ch</i> : channel number 1–16
<i>Syntax</i>	<i>Status</i> = app.SCPI.CALCulate(Ch).SElected.FUNCTION.DOMain.COUPle app.SCPI.CALCulate(Ch).SElected.FUNCTION.DOMain.COUPle = <i>Status</i>
<i>Equivalent Softkeys</i>	None

SCPI.CALCulate(Ch).SElected.FUNCTION.DOMain.START

<i>Description</i>	Sets/gets the start value of the analysis range set by the SCPI.CALCulate(Ch).SElected.FUNCTION.EXECute method.
<i>Type</i>	Double (read/write)
<i>Range</i>	From the stimulus current start value to the stimulus current stop value.
<i>Preset Value</i>	0
<i>Unit</i>	Hz s dBm
<i>Target</i>	All traces of channel <i>Ch</i> (if the coupling is set to ON by the SCPI.CALCulate(Ch).SElected.FUNCTION.DOMain.COUPle command), the active trace of channel <i>Ch</i> (if otherwise), <i>Ch</i> : channel number 1–16
<i>Syntax</i>	<i>Value</i> = app.SCPI.CALCulate(Ch).SElected.FUNCTION.DOMain.START app.SCPI.CALCulate(Ch).SElected.FUNCTION.DOMain.START = 1e9
<i>Equivalent Softkeys</i>	None

SCPI.CALCulate(Ch).SElected.FUNction.DOMain.STATe

<i>Description</i>	Turns ON/OFF the state of the arbitrary range when executing the analysis by the SCPI.CALCulate(Ch).SElected.FUNction.EXECute method.
<i>Type</i>	Boolean (read/write)
<i>Parameter</i>	True: Arbitrary range ON False: Arbitrary range OFF (entire sweep range)
<i>Preset Value</i>	False
<i>Target</i>	All traces of channel <i>Ch</i> (if the coupling is set to ON by the SCPI.CALCulate(Ch).SElected.FUNction.DOMain.COUPle command), the active trace of channel <i>Ch</i> (if otherwise), <i>Ch</i> : channel number 1–16
<i>Syntax</i>	<i>Status</i> = app.SCPI.CALCulate(Ch).SElected.FUNction.DOMain.STATe <i>app.SCPI.CALCulate(Ch).SElected.FUNction.DOMain.STATe</i> = <i>true</i>
<i>Equivalent Softkeys</i>	None

SCPI.CALCulate(Ch).SElected.FUNction.DOMain.STOP

<i>Description</i>	Sets/gets the stop value of the analysis range set by the SCPI.CALCulate(Ch).SElected.FUNction.EXECute method.
<i>Type</i>	Double (read/write)
<i>Range</i>	From the stimulus current start value to the stimulus current stop value.
<i>Preset Value</i>	0
<i>Unit</i>	Hz s dBm
<i>Target</i>	All traces of channel <i>Ch</i> (if the coupling is set to ON by the SCPI.CALCulate(Ch).SElected.FUNction.DOMain.COUPle command), the active trace of channel <i>Ch</i> (if otherwise), <i>Ch</i> : channel number 1–16
<i>Syntax</i>	<i>Value</i> = app.SCPI.CALCulate(Ch).SElected.FUNction.DOMain.STOP <i>app</i> .SCPI.CALCulate(Ch).SElected.FUNction.DOMain.STOP = 2e9
<i>Equivalent Softkeys</i>	None

SCPI.CALCulate(Ch).SElected.FUNction.EXECute

<i>Description</i>	Executes the analysis specified with the SCPI.CALCulate(Ch).SElected.FUNction.TYPE command. The analysis result can then be read out with the SCPI.CALCulate(Ch).SElected.FUNction.DATA command.
<i>Type</i>	Method
<i>Target</i>	The active trace of channel <i>Ch</i> , <i>Ch</i> : channel number 1–16
<i>Syntax</i>	<i>app</i> .SCPI.CALCulate(Ch).SElected.FUNction.EXECute
<i>Equivalent Softkeys</i>	None

SCPI.CALCulate(Ch).SElected.FUNCtion.PEXCursion

<i>Description</i>	Sets/gets the lower limit for the peak excursion value when executing the peak search by the SCPI.CALCulate(Ch).SElected.FUNCtion.EXECute method.
<i>Type</i>	Double (read/write)
<i>Range</i>	Varies depending on the trace format.
<i>Preset Value</i>	3
<i>Unit</i>	dB ° s
<i>Target</i>	The active trace of channel <i>Ch</i> , <i>Ch</i> : channel number 1–16
<i>Syntax</i>	<i>Value</i> = app.SCPI.CALCulate(Ch).SElected.FUNCtion.PEXCursion <i>app.SCPI.CALCulate(Ch).SElected.FUNCtion.PEXCursion</i> = 1.5
<i>Equivalent Softkeys</i>	None

SCPI.CALCulate(Ch).SElected.FUNCtion.POINTs

<i>Description</i>	Gets the number of points (data pairs) of the analysis result by the SCPI.CALCulate(Ch).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>Type</i>	Long (read only)
<i>Target</i>	The active trace of channel <i>Ch</i> , <i>Ch</i> : channel number 1–16
<i>Syntax</i>	<i>Value</i> = app.SCPI.CALCulate(Ch).SElected.FUNCtion.POINTs
<i>Equivalent Softkeys</i>	None

SCPI.CALCulate(Ch).SElected.FUNction.PPOLarity

<i>Description</i>	Sets/gets the polarity selection when performing the peak search by the SCPI.CALCulate(Ch).SElected.FUNction.EXECute method.
<i>Type</i>	String (read/write)
<i>Parameter</i>	"POSitive" : Positive peaks "NEGative" : Negative peaks "BOTH" : Both positive peaks and negative peaks
<i>Preset Value</i>	"POS"
<i>Target</i>	The active trace of channel <i>Ch</i> , <i>Ch</i> : channel number 1–16
<i>Syntax</i>	<i>Param</i> = app.SCPI.CALCulate(Ch).SElected.FUNction.PPOLarity app.SCPI.CALCulate(Ch).SElected.FUNction.PPOLarity = "NEG"
<i>Equivalent Softkeys</i>	None

SCPI.CALCulate(Ch).SElected.FUNCtion.TARGet

<i>Description</i>	Sets/gets the target level when performing the search for the trace and the target level crosspoints by the SCPI.CALCulate(Ch).SElected.FUNCtion.EXECute method.
<i>Type</i>	Double (read/write)
<i>Range</i>	Varies depending on the trace format.
<i>Preset Value</i>	0
<i>Unit</i>	dB ° s
<i>Target</i>	The active trace of channel <i>Ch</i> , <i>Ch</i> : channel number 1–16
<i>Syntax</i>	<i>Value</i> = app.SCPI.CALCulate(Ch).SElected.FUNCtion.TARGet app.SCPI.CALCulate(Ch).SElected.FUNCtion.TARGet = –10
<i>Equivalent Softkeys</i>	None

SCPI.CALCulate(Ch).SElected.FUNction.TTRansition

<i>Description</i>	Sets/gets the transition type selection when performing the search for the trace and the target level crosspoints by the SCPI.CALCulate(Ch).SElected.FUNction.EXECute method.
<i>Type</i>	String (read/write)
<i>Parameter</i>	"POSitive" : Positive peaks "NEGative" : Negative peaks "BOTH" : Both positive peaks and negative peaks
<i>Preset Value</i>	"POS"
<i>Target</i>	The active trace of channel <i>Ch</i> , <i>Ch</i> : channel number 1–16
<i>Syntax</i>	<i>Param</i> = app.SCPi.CALCulate(<i>Ch</i>).SElected.FUNction.TTRansition app.SCPi.CALCulate(<i>Ch</i>).SElected.FUNction.TTRansition = "both"
<i>Equivalent Softkeys</i>	None

SCPI.CALCulate(Ch).SElected.FUNction.TYPE

<i>Description</i>	Sets/gets the selection of the type of analysis executed by the SCPI.CALCulate(Ch).SElected.FUNction.EXECute method.
<i>Type</i>	String (read/write)
<i>Parameter</i>	"PTPeak" : Peak-to-peak (difference between the maximum value and the minimum value) "STDEV" : Standard deviation "MEAN" : Mean value "MAXimum" : Maximum value "MINimum" : Minimum value "PEAK" : Search for the peak "APEak" : Search for all the peaks "ATARget" : Search for all targets
<i>Preset Value</i>	"PTP"
<i>Target</i>	The active trace of channel <i>Ch</i> , <i>Ch</i> : channel number 1–16
<i>Syntax</i>	<i>Param</i> = app.SCPI.CALCulate(Ch).SElected.FUNction.TYPE <i>app</i>.SCPI.CALCulate(Ch).SElected.FUNction.TYPE = "STDEV"
<i>Equivalent Softkeys</i>	None

SCPI.CALCulate(Ch).SElected.LIMit.DATA

<i>Description</i>	Sets/gets 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. For the n-th point, where n from 1 to N: <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>Type</i>	Variant: array of double (read/write)
<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>Target</i>	The active trace of channel <i>Ch</i>, <i>Ch</i>: channel number 1–16
<i>Syntax</i>	<i>Data</i> = 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)
<i>Equivalent Softkeys</i>	Analysis > Limit Test > Edit Limit Line

SCPI.CALCulate(Ch).SElected.LIMit.DISPlay.STATe

<i>Description</i>	Turns ON/OFF the state of the limit line display of the limit test function.
<i>Type</i>	Boolean (read/write)
<i>Parameter</i>	True: Limit line display ON False: Limit line display OFF
<i>Preset Value</i>	False
<i>Target</i>	The active trace of channel <i>Ch</i> , <i>Ch</i> : channel number 1–16
<i>Syntax</i>	<i>Status</i> = app.SCPi.CALCulate(<i>Ch</i>).SElected.LIMit.DISPlay.STATe app.SCPi.CALCulate(<i>Ch</i>).SElected.LIMit.DISPlay.STATe = true
<i>Equivalent Softkeys</i>	Analysis > Limit Test > Limit Line

SCPI.CALCulate(Ch).SElected.LIMit.FAIL

<i>Description</i>	Gets the limit test result.
<i>Type</i>	Boolean (read only)
<i>Parameter</i>	True: Fail False: Pass
<i>Target</i>	The active trace of channel <i>Ch</i> , <i>Ch</i> : channel number 1–16
<i>Syntax</i>	<i>Status</i> = app.SCPi.CALCulate(<i>Ch</i>).SElected.LIMit.FAIL
<i>Equivalent Softkeys</i>	None

SCPI.CALCulate(Ch).SElected.LIMit.OFFSet.AMPLitude

<i>Description</i>	Sets/gets the value of the limit line offset along Y–axis.
<i>Type</i>	Double (read/write)
<i>Range</i>	Varies depending on the trace format.
<i>Preset Value</i>	0
<i>Unit</i>	dB ° s
<i>Target</i>	The active trace of channel <i>Ch</i> , <i>Ch</i> : channel number 1–16
<i>Syntax</i>	<i>Value</i> = app.SCPi.CALCulate(<i>Ch</i>).SElected.LIMit.OFFSet.AMPLitude app.SCPi.CALCulate(<i>Ch</i>).SElected.LIMit.OFFSet.AMPLitude = –10
<i>Equivalent Softkeys</i>	Analysis > Limit Test > Limit Line Offsets > Response Offset

SCPI.CALCulate(Ch).SElected.LIMit.OFFSet.MARKer

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

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

<i>Description</i>	Sets/gets the value of the limit line offset along X–axis.
<i>Type</i>	Double (read/write)
<i>Range</i>	From the stimulus current start value to the stimulus current stop value.
<i>Preset Value</i>	0
<i>Unit</i>	Hz s dBm
<i>Target</i>	The active trace of channel <i>Ch</i> , <i>Ch</i> : channel number 1–16
<i>Syntax</i>	<i>Value</i> = app.SCPI.CALCulate(<i>Ch</i>).SElected.LIMit.OFFSet.STIMulus app.SCPI.CALCulate(<i>Ch</i>).SElected.LIMit.OFFSet.STIMulus = 1e6
<i>Equivalent Softkeys</i>	Analysis > Limit Test > Limit Lines Offsets > Stimulus Offset

SCPI.CALCulate(Ch).SElected.LIMit.REPort.ALL

<i>Description</i>	Gets the data array, which is the limit test results. The array size is $4N$, where N is the number of measurement points. For the n-th point, where n from 1 to N: $Data(4n-3)$ the stimulus value in the n-th point $Data(4n-2)$ the limit test result in the n-th point -1: No limit 0: Fail 1: Pass $Data(4n-1)$ the upper limit value in the n-th point (0 – if there is no limit) $Data(4n-0)$ the lower limit value in the n-th point (0 – if there is no limit)
<i>Type</i>	Variant: array of double (read only)
<i>Target</i>	The active trace of channel Ch , Ch : channel number 1–16
<i>Syntax</i>	$Data = \text{app.SCPi.CALCulate}(Ch).\text{SElected.LIMit.REPort.ALL}$
<i>Equivalent Softkeys</i>	None

SCPI.CALCulate(Ch).SElected.LIMit.REPort.DATA

<i>Description</i>	Gets 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(Ch).SElected.LIMit.REPort.POINts command.
<i>Type</i>	Variant: array of double (read only)
<i>Target</i>	The active trace of channel Ch , Ch : channel number 1–16
<i>Syntax</i>	$Data = \text{app.SCPi.CALCulate}(Ch).\text{SElected.LIMit.REPort.DATA}$
<i>Equivalent Softkeys</i>	None

SCPI.CALCulate(Ch).SElected.LIMit.REPort.POINts

<i>Description</i>	Gets 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(Ch).SElected.LIMit.REPort.DATA command.
<i>Type</i>	Long (read only)
<i>Target</i>	The active trace of channel <i>Ch</i> , <i>Ch</i> : channel number 1–16
<i>Syntax</i>	<i>Cnt</i> = app.SCPI.CALCulate(Ch).SElected.LIMit.REPort.POINts
<i>Equivalent Softkeys</i>	None

SCPI.CALCulate(Ch).SElected.LIMit.STATe

<i>Description</i>	Turns ON/OFF the state of the limit test function.
<i>Type</i>	Boolean (read/write)
<i>Parameter</i>	True: Limit test function ON False: Limit test function OFF
<i>Preset Value</i>	False
<i>Target</i>	The active trace of channel <i>Ch</i> , <i>Ch</i> : channel number 1–16
<i>Syntax</i>	<i>Status</i> = app.SCPI.CALCulate(Ch).SElected.LIMit.STATe <i>app.SCPI.CALCulate(Ch).SElected.LIMit.STATe</i> = true
<i>Equivalent Softkeys</i>	Analysis > Limit Test > Limit Test

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

<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>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 <i>Mk</i>: marker number 1–15, or reference marker number 16
<i>Syntax</i>	<i>app</i> .SCPI.CALCulate(<i>Ch</i>).SELected.MARKer(<i>Mk</i>).ACTivate
<i>Equivalent Softkeys</i>	Markers > Select > Marker n Markers > Reference Marker

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

<i>Description</i>	Gets 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 command. The array contains 4 elements: <i>Data</i>(0) Bandwidth; <i>Data</i>(1) Center frequency; <i>Data</i>(2) Q value; <i>Data</i>(3) Loss.
<i>Type</i>	Variant: array of double (read only)
<i>Notes</i>	If the bandwidth search is impossible, all the read out values are 0. If the search is performed relatively to a marker, which is OFF, an error occurs (error code 204).
<i>Target</i>	Marker <i>Mk</i> of the active trace of channel <i>Ch</i>, <i>Ch</i>: channel number 1–16 <i>Mk</i>: marker number 1–15, or reference marker number 16
<i>Syntax</i>	<i>Data</i> = app.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>Description</i>	Sets/gets the selection of the reference point for the bandwidth search function: reference marker or absolute maximum value of the trace.
<i>Type</i>	String (read/write)
<i>Parameter</i>	"MARKer" : Bandwidth search relative to the reference marker "MAXimum" : Bandwidth search relative to the absolute maximum of the trace
<i>Preset Value</i>	"MAX"
<i>Target</i>	The active trace of channel <i>Ch</i> , <i>Ch</i> : channel number 1–16
<i>Syntax</i>	<i>Param</i> = app.SCPI.CALCulate(<i>Ch</i>).SElected.MARKer.BWIDth.REFerence app.SCPI.CALCulate(<i>Ch</i>).SElected.MARKer.BWIDth.REFerence = "marker"
<i>Warning</i>	Object <i>MARKer</i> has an index of 1, which can be omitted in Visual Basic but it cannot 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>Description</i>	Turns ON/OFF the state of the bandwidth search function.
<i>Type</i>	Boolean (read/write)
<i>Parameter</i>	True: Bandwidth search function ON False: Bandwidth search function OFF
<i>Preset Value</i>	False
<i>Target</i>	The active trace of channel <i>Ch</i> , <i>Ch</i> : channel number 1–16
<i>Syntax</i>	<i>Status</i> = app.SCPi.CALCulate(<i>Ch</i>).SElected.MARKer.BWIDth.STATe <i>app.SCPi.CALCulate(<i>Ch</i>).SElected.MARKer.BWIDth.STATe</i> = true
<i>Warning</i>	Object <i>MARKer</i> has an index of 1, which can be omitted in Visual Basic but it cannot 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>Description</i>	Sets/gets the bandwidth definition value for the bandwidth search function.
<i>Type</i>	Double (read/write)
<i>Range</i>	Varies depending on the trace format.
<i>Preset Value</i>	– 3
<i>Unit</i>	dB ° s
<i>Target</i>	The active trace of channel <i>Ch</i> , <i>Ch</i> : channel number 1–16
<i>Syntax</i>	<i>Value</i> = app.SCPI.CALCulate(<i>Ch</i>).SElected.MARKer(<i>Mk</i>).BWIDth.THReshold app.SCPI.CALCulate(<i>Ch</i>).SElected.MARKer(<i>Mk</i>).BWIDth.THReshold = –6.0
<i>Warning</i>	Object <i>MARKer</i> has an index of 1, which can be omitted in Visual Basic but it cannot 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>Description</i>	Sets/gets the type of the bandwidth search function.
<i>Type</i>	String (read/write)
<i>Parameter</i>	"BPASs" : Bandpass "NOTCh" : Notch
<i>Preset Value</i>	"BPAS"
<i>Target</i>	The active trace of channel <i>Ch</i> , <i>Ch</i> : channel number 1–16
<i>Syntax</i>	<i>Param</i> = app.SCPI.CALCulate(<i>Ch</i>).SElected.MARKer.BWIDth.TYPE app.SCPI.CALCulate(<i>Ch</i>).SElected.MARKer.BWIDth.TYPE = "notc"
<i>Equivalent Softkeys</i>	Markers > Marker Math > Bandwidth Search > Type

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

<i>Description</i>	Turns ON/OFF the state of the marker coupling between traces.
<i>Type</i>	Boolean (read/write)
<i>Parameter</i>	True: Marker coupling ON False: Marker coupling OFF
<i>Preset Value</i>	True
<i>Target</i>	All traces of channel <i>Ch</i> , <i>Ch</i> : channel number 1–16
<i>Syntax</i>	<i>Status</i> = app.SCPI.CALCulate(<i>Ch</i>).SElected.MARKer.COUPle <i>app.SCPI.CALCulate(<i>Ch</i>).SElected.MARKer.COUPle</i> = false
<i>Equivalent Softkeys</i>	Marker > Properties > Marker Couple

SCPI.CALCulate(Ch).SElected.MARKer(1).COUNT

<i>Description</i>	Sets/gets 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>Type</i>	Long (read/write)
<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>Target</i>	The active trace of channel <i>Ch</i> , <i>Ch</i> : channel number 1–16
<i>Syntax</i>	<i>MarkerCnt</i> = app.SCPI.CALCulate(<i>Ch</i>).SElected.MARKer.COUNT app.SCPI.CALCulate(<i>Ch</i>).SElected.MARKer.COUNT = 5
<i>Warning</i>	Object <i>MARKer</i> has an index of 1, which can be omitted in Visual Basic but it cannot be omitted in other programming languages.
<i>Equivalent Softkeys</i>	None

SCPI.CALCulate(*Ch*).SElected.MARKer(1).DIScrete

<i>Description</i>	Turns ON/OFF the state of the marker discrete mode.
<i>Type</i>	Boolean (read/write)
<i>Parameter</i>	True: Marker discrete mode ON False: Marker discrete mode OFF
<i>Preset Value</i>	False
<i>Target</i>	All traces of channel <i>Ch</i> (if the coupling is set to ON by the SCPI.CALCulate(<i>Ch</i>).SElected.MARKer(1).COUPle command), the active trace of channel <i>Ch</i> (if otherwise), <i>Ch</i> : channel number 1–16
<i>Syntax</i>	<i>Status</i> = app.SCPI.CALCulate(<i>Ch</i>).SElected.MARKer.DIScrete <i>app</i> .SCPI.CALCulate(<i>Ch</i>).SElected.MARKer.DIScrete = false
<i>Warning</i>	Object <i>MARKer</i> has an index of 1, which can be omitted in Visual Basic but it cannot be omitted in other programming languages.
<i>Equivalent Softkeys</i>	Marker > Properties > Discrete

SCPI.CALCulate(Ch).SElected.MARKer(Mk).DATA

<i>Description</i>	Gets the response and stimulus 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 3 elements: <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(2)</i> Stimulus value at the marker position. (S4VNA Only)
<i>Type</i>	Variant: array of double (read only)
<i>Target</i>	Marker <i>Mk</i> of the active trace of channel <i>Ch</i>, <i>Ch</i>: channel number 1–16 <i>Mk</i>: marker number 1–15, or reference marker number 16
<i>Syntax</i>	<i>Data</i> = app.SCPI.CALCulate(<i>Ch</i>).SElected.MARKer(<i>Mk</i>).DATA
<i>Equivalent Softkeys</i>	None

SCPI.CALCulate(Ch).SElected.MARKer(1).DATA

<i>Description</i>	Gets 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. For the n-th marker, where n from 1 to N: <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; (S2VNA only)
<i>Type</i>	Variant: array of double (read only)
<i>Target</i>	The active trace of channel <i>Ch</i> , <i>Ch</i> : channel number 1–16
<i>Syntax</i>	<i>Data</i> = <i>app</i> .SCPI.CALCulate(<i>Ch</i>).SElected.MARKer.DATA
<i>Warning</i>	Object <i>MARKer</i> has an index of 1, which can be omitted in Visual Basic but it cannot be omitted in other programming languages.
<i>Equivalent Softkeys</i>	None

SCPI.CALCulate(Ch).SElected.MARKer(1).FUNction.DOMain.COUPle

<i>Description</i>	Turns ON/OFF the state of the marker search range coupling for different traces.
<i>Type</i>	Boolean (read/write)
<i>Parameter</i>	True: Marker search range coupling ON False: Marker search range coupling OFF
<i>Preset Value</i>	True
<i>Target</i>	All traces of channel <i>Ch</i> , <i>Ch</i> : channel number 1–16
<i>Syntax</i>	<i>Status</i> = app.SCPI.CALCulate(<i>Ch</i>).SElected.MARKer.FUNction.DOMain. COUPle <i>app.SCPI.CALCulate(Ch).SElected.MARKer.FUNction.DOMain.COUPle</i> = false
<i>Warning</i>	Object <i>MARKer</i> has an index of 1, which can be omitted in Visual Basic but it cannot be omitted in other programming languages.
<i>Equivalent Softkeys</i>	Markers > Marker Search > Couple

SCPI.CALCulate(*Ch*).SElected.MARKer(1).FUNction.DOMain.STARt

<i>Description</i>	Sets/gets the start value of the marker search range.
<i>Type</i>	Double (read/write)
<i>Range</i>	From the stimulus current start value to the stimulus current stop value.
<i>Preset Value</i>	Minimum frequency
<i>Unit</i>	Hz s dBm
<i>Target</i>	All traces of channel <i>Ch</i> (if the marker search range coupling is set to ON by the SCPI.CALCulate(<i>Ch</i>).SElected.FUNction.DOMain.COUPLE command), the active trace of channel <i>Ch</i> (if otherwise), <i>Ch</i> : channel number 1–16
<i>Syntax</i>	<i>Value</i> = app.SCPI.CALCulate(<i>Ch</i>).SElected.MARKer.FUNction.DOMain.STARt <i>app.SCPI.CALCulate(<i>Ch</i>).SElected.MARKer.FUNction.DOMain.STARt</i> = 1e6
<i>Warning</i>	Object <i>MARKer</i> has an index of 1, which can be omitted in Visual Basic but it cannot 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>Description</i>	Turns ON/OFF the state of the arbitrary range when executing the marker search.
<i>Type</i>	Boolean (read/write)
<i>Parameter</i>	True: Marker search range ON False: Marker search range OFF (entire sweep range)
<i>Preset Value</i>	False
<i>Target</i>	All traces of channel <i>Ch</i> (if the marker search range coupling is set to ON by the SCPI.CALCulate(<i>Ch</i>).SElected.MARKer.FUNction.DOMain.COUPle command), the active trace of channel <i>Ch</i> (if otherwise), <i>Ch</i> : channel number 1–16
<i>Syntax</i>	<i>Status</i> = app.SCPI.CALCulate(<i>Ch</i>).SElected.MARKer.FUNction.DOMain.STATe <i>app</i> .SCPI.CALCulate(<i>Ch</i>).SElected.MARKer.FUNction.DOMain.STATe = true
<i>Warning</i>	Object <i>MARKer</i> has an index of 1, which can be omitted in Visual Basic but it cannot 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>Description</i>	Sets/gets the stop value of the marker search range.
<i>Type</i>	Double (read/write)
<i>Range</i>	From the stimulus current start value to the stimulus current stop value.
<i>Preset value</i>	Maximum frequency
<i>Unit</i>	Hz s dBm
<i>Target</i>	All traces of channel <i>Ch</i> (if the marker search range coupling is set to ON by the SCPI.CALCulate(Ch).SElected.MARKer.FUNction.DOMain.COUPLE command), the active trace of channel <i>Ch</i> (if otherwise), <i>Ch</i> : channel number 1–16
<i>Syntax</i>	<i>Value</i> = app.SCPI.CALCulate(Ch).SElected.MARKer.FUNction.DOMain.STOP <i>app.SCPI.CALCulate(Ch).SElected.MARKer.FUNction.DOMain.STOP</i> = 1e6
<i>Warning</i>	Object <i>MARKer</i> has an index of 1, which can be omitted in Visual Basic but it cannot be omitted in other programming languages.
<i>Equivalent Softkeys</i>	Markers > Marker Search > Search Stop

SCPI.CALCulate(*Ch*).SElected.MARKer(*Mk*).FUNction.EXECute

<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 command.
<i>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 <i>Mk</i> : marker number 1–15, or reference marker number 16
<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>Description</i>	Sets/gets 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>Type</i>	Double (read/write)
<i>Target</i>	Marker <i>Mk</i> of the active trace of channel <i>Ch</i> , <i>Ch</i> : channel number 1–16 <i>Mk</i> : marker number 1–15, or reference marker number 16
<i>Range</i>	Varies depending on the trace format.
<i>Preset Value</i>	1
<i>Unit</i>	dB (decibel) ° (degree) s (second)
<i>Syntax</i>	<i>Value</i> = app.SCPI.CALCulate(<i>Ch</i>).SElected.MARKer(<i>Mk</i>).FUNction.PEXCursion app.SCPI.CALCulate(<i>Ch</i>).SElected.MARKer(<i>Mk</i>).FUNction.PEXCursion = 3.0
<i>Equivalent Softkeys</i>	Markers > Marker Search > Peak > Peak Excursion

SCPI.CALCulate(*Ch*).SElected.MARKer(*Mk*).FUNction.PPOLarity

<i>Description</i>	Sets/gets 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>Type</i>	String (read/write)
<i>Parameter</i>	"POSitive" : Positive polarity "NEGative" : Negative polarity "BOTH" : Both positive polarity and negative polarity
<i>Preset Value</i>	"POS"
<i>Target</i>	Marker <i>Mk</i> of the active trace of channel <i>Ch</i> , <i>Ch</i> : channel number 1–16 <i>Mk</i> : marker number 1–15, or reference marker number 16
<i>Syntax</i>	<i>Param</i> = 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"
<i>Equivalent Softkeys</i>	Markers > Marker Search > Peak > Peak Polarity > Positive Negative Both

SCPI.CALCulate(*Ch*).SElected.MARKer(*Mk*).FUNction.TARGet

<i>Description</i>	Sets/gets 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>Type</i>	Double (read/write)
<i>Range</i>	Varies depending on the trace format.
<i>Preset Value</i>	0
<i>Unit</i>	dB (decibel) ° (degree) s (second)
<i>Target</i>	Marker <i>Mk</i> of the active trace of channel <i>Ch</i> , <i>Ch</i> : channel number 1–16 <i>Mk</i> : marker number 1–15, or reference marker number 16
<i>Syntax</i>	<i>Value</i> = app.SCPI.CALCulate(<i>Ch</i>).SElected.MARKer(<i>Mk</i>).FUNction.TARGet app.SCPI.CALCulate(<i>Ch</i>).SElected.MARKer(<i>Mk</i>).FUNction.TARGet = –10
<i>Equivalent Softkeys</i>	Markers > Marker Search > Target > Target Value

SCPI.CALCulate(*Ch*).SElected.MARKer(*Mk*).FUNction.TRACking

<i>Description</i>	Turns ON/OFF the state of the marker search tracking function.
<i>Type</i>	Boolean (read/write)
<i>Parameter</i>	True: Marker search tracking ON False: Marker search tracking OFF
<i>Preset Value</i>	False
<i>Target</i>	Marker <i>Mk</i> of the active trace of channel <i>Ch</i> , <i>Ch</i> : channel number 1–16 <i>Mk</i> : marker number 1–15, or reference marker number 16
<i>Syntax</i>	<i>Status</i> = app.SCPi.CALCulate(<i>Ch</i>).SElected.MARKer(<i>Mk</i>).FUNction.TRACking <i>app.SCPi.CALCulate(<i>Ch</i>).SElected.MARKer(<i>Mk</i>).FUNction.TRACking</i> = true
<i>Equivalent Softkeys</i>	Markers > Marker Search > Tracking

SCPI.CALCulate(Ch).SElected.MARKer(Mk).FUNCtion.TTRansition

<i>Description</i>	Sets/gets the selection of the type of the target transition, when the marker search for transition is performed by the SCPI.CALCulate(Ch).SElected.MARKer(Mk).FUNCtion.EXECute method.
<i>Type</i>	String (read/write)
<i>Parameter</i>	"POSitive" : Positive target transition "NEGative" : Negative target transition "BOTH" : Both positive target transition and negative target transition
<i>Preset Value</i>	"POS"
<i>Target</i>	Marker <i>Mk</i> of the active trace of channel <i>Ch</i> , <i>Ch</i> : channel number 1–16 <i>Mk</i> : marker number 1–15, or reference marker number 16
<i>Syntax</i>	<i>Param</i> = app.SCPI.CALCulate(Ch).SElected.MARKer(Mk).FUNCtion.TTRansition app.SCPI.CALCulate(Ch).SElected.MARKer(Mk).FUNCtion.TTRansition = "neg"
<i>Equivalent Softkeys</i>	Marker > Marker Search > Target > Target Transition

SCPI.CALCulate(Ch).SElected.MARKer(Mk).FUNction.TYPE

<i>Description</i>	Sets/gets the selection of the type of the marker search, which is performed by the SCPI.CALCulate(Ch).SElected.MARKer(Mk).FUNction.EXECute method.
<i>Type</i>	String (read/write)
<i>Parameter</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>Preset Value</i>	"MAX"
<i>Target</i>	Marker <i>Mk</i> of the active trace of channel <i>Ch</i> , <i>Ch</i> : channel number 1–16 <i>Mk</i> : marker number 1–15, or reference marker number 16
<i>Syntax</i>	<i>Param</i> = app.SCPI.CALCulate(Ch).SElected.MARKer(Mk).FUNction.TYPE app.SCPI.CALCulate(Ch).SElected.MARKer(Mk).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>Description</i>	Reads out the FLATNESS function data array. The FLATNESS function is applied within the range determined by two markers (see SCPI.CALCulate(Ch).SElected.MARKer.MATH.FLATness.DOMain.START and SCPI.CALCulate(Ch).SElected.MARKer.MATH.FLATness.DOMain.STOP properties). The array includes 4 elements: <i>Data(0)</i> Span; <i>Data(1)</i> Gain; <i>Data(2)</i> Slope; <i>Data(3)</i> Flatness.
<i>Type</i>	Variant: array of double (read only)
<i>Target</i>	The active trace of channel <i>Ch</i>, <i>Ch</i>: channel number 1–16
<i>Syntax</i>	<i>Data</i> = app.SCPI.CALCulate(Ch).SElected.MARKer.MATH.FLATness.DATA
<i>Warning</i>	Object <i>MARKer</i> has an index of 1, which can be omitted in Visual Basic but it cannot be omitted in other programming languages.
<i>Equivalent Softkeys</i>	None

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

<i>Description</i>	Sets/gets the number of the marker, which specifies the start frequency of the FLATNESS function range.
<i>Type</i>	Long (read/write)
<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>Target</i>	The active trace of channel <i>Ch</i> , <i>Ch</i> : channel number 1–16
<i>Syntax</i>	<i>MkrNum</i> = app.SCPI.CALCulate(<i>Ch</i>).SElected.MARKer.MATH.FLATness.DOMain.START <i>app.SCPI.CALCulate(Ch).SElected.MARKer.MATH.FLATness.DOMain.START</i> = 1
<i>Warning</i>	Object <i>MARKer</i> has an index of 1, which can be omitted in Visual Basic but it cannot 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>Description</i>	Sets/gets the number of the marker, which specifies the stop frequency of the FLATNESS function range.
<i>Type</i>	Long (read/write)
<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>Target</i>	The active trace of channel <i>Ch</i> , <i>Ch</i> : channel number 1–16
<i>Syntax</i>	<i>MkrNum</i> = app.SCPI.CALCulate(<i>Ch</i>).SElected.MARKer.MATH.FLATness.DOMain.STOP <i>app.SCPI.CALCulate(Ch).SElected.MARKer.MATH.FLATness.DOMain.STOP</i> = 1
<i>Warning</i>	Object <i>MARKer</i> has an index of 1, which can be omitted in Visual Basic but it cannot 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>Description</i>	Turns ON/OFF the state of the FLATNESS function.
<i>Type</i>	Boolean (read/write)
<i>Parameter</i>	True: FLATNESS function ON False: FLATNESS function OFF
<i>Preset Value</i>	False
<i>Target</i>	The active trace of channel <i>Ch</i> , <i>Ch</i> : channel number 1–16
<i>Syntax</i>	<i>Status</i> = app.SCPI.CALCulate(<i>Ch</i>).SElected.MARKer.MATH.FLATness.STATe app.SCPI.CALCulate(<i>Ch</i>).SElected.MARKer.MATH.FLATness.STATe = true
<i>Warning</i>	Object <i>MARKer</i> has an index of 1, which can be omitted in Visual Basic but it cannot be omitted in other programming languages.
<i>Equivalent Softkeys</i>	Markers > Marker Math > Flatness > Flatness

SCPI.CALCulate(*Ch*).SElected.MARKer(1).REFerence.STATe

<i>Description</i>	Turns ON/OFF the state of the reference marker. When the reference marker is turned ON, all the values of the other markers turn to relative values.
<i>Type</i>	Boolean (read/write)
<i>Parameter</i>	True: Reference marker ON False: Reference marker OFF
<i>Preset Value</i>	False
<i>Target</i>	The active trace of channel <i>Ch</i> , <i>Ch</i> : channel number 1–16
<i>Syntax</i>	<i>Status</i> = app.SCPI.CALCulate(<i>Ch</i>).SElected.MARKer.REFerence.STATe app.SCPI.CALCulate(<i>Ch</i>).SElected.MARKer.REFerence.STATe = true
<i>Warning</i>	Object <i>MARKer</i> has an index of 1, which can be omitted in Visual Basic but it cannot be omitted in other programming languages.
<i>Equivalent Softkeys</i>	Markers > Reference Marker

SCPI.CALCulate(Ch).SElected.MARKer(Mk).SET**SCPI.CALCulate(Ch).SElected.MARKer(Mk).POSition**

<i>Description</i>	Sets the value of the specified item to the value of the position of the marker.
<i>Type</i>	String (write only)
<i>Parameter</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>Target</i>	Marker <i>Mk</i> of the active trace of channel <i>Ch</i>, <i>Ch</i>: channel number 1–16 <i>Mk</i>: marker number 1–15, or reference marker number 16
<i>Syntax</i>	<i>app</i>.SCPI.CALCulate(<i>Ch</i>).SElected.MARKer(<i>Mk</i>).POSition = "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>Description</i>	Turns ON/OFF the 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>Type</i>	Boolean (read/write)
<i>Parameter</i>	True: Marker ON False: Marker OFF
<i>Preset Value</i>	False
<i>Target</i>	Marker <i>Mk</i> of the active trace of channel <i>Ch</i> , <i>Ch</i> : channel number 1–16 <i>Mk</i> : marker number 1–15, or reference marker number 16
<i>Syntax</i>	<i>Status</i> = app.SCPi.CALCulate(<i>Ch</i>).SElected.MARKer(<i>Mk</i>).STATe app.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>Description</i>	Sets/gets the stimulus value of the marker.
<i>Type</i>	Double (read/write)
<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 s dBm
<i>Target</i>	Marker <i>Mk</i> of the active trace of channel <i>Ch</i> , <i>Ch</i> : channel number 1–16 <i>Mk</i> : marker number 1–15, or reference marker number 16
<i>Syntax</i>	<i>Value</i> = app.SCPi.CALCulate(<i>Ch</i>).SElected.MARKer(<i>Mk</i>).X app.SCPi.CALCulate(<i>Ch</i>).SElected.MARKer(<i>Mk</i>).X = 1e9
<i>Equivalent Softkeys</i>	Markers > Edit Stimulus

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

<i>Description</i>	Reads out 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: <i>Data</i>(0) real number in rectangular format, real part in polar and Smith chart formats; <i>Data</i>(1) 0 in rectangular format, imaginary part in polar and Smith chart formats.
<i>Type</i>	Variant: array of double (read only)
<i>Target</i>	Marker <i>Mk</i> of the active trace of channel <i>Ch</i>, <i>Ch</i>: channel number 1–16 <i>Mk</i>: marker number 1–15, or reference marker number 16
<i>Syntax</i>	<i>Data</i> = app.SCPI.CALCulate(<i>Ch</i>).SElected.MARKer(<i>Mk</i>).Y
<i>Equivalent Softkeys</i>	None

SCPI.CALCulate(Ch).SElected.MATH.FUNcTion

<i>Description</i>	Sets/gets 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>Type</i>	String (read/write)
<i>Parameter</i>	"DIVide" : Division $Data / Mem$. "MULTipty" : Multiplication $Data \times Mem$. "ADD" : Addition $Data + Mem$. "SUBTract" : Subtraction $Data - Mem$. "NORMal" : No math
<i>Preset Value</i>	"NORM"
<i>Target</i>	The active trace of channel <i>Ch</i> , <i>Ch</i> : channel number 1–16
<i>Syntax</i>	<i>Param</i> = app.SCPI.CALCulate(Ch).SElected.MATH.FUNcTion app.SCPI.CALCulate(Ch).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>Description</i>	Saves the measurement data to the memory trace. Automatically turns on the display of the memory trace.
<i>Type</i>	Method
<i>Target</i>	The active trace of channel <i>Ch</i> , <i>Ch</i> : channel number 1–16
<i>Syntax</i>	app.SCPI.CALCulate(Ch).SElected.MATH.MEMorize
<i>Equivalent Softkeys</i>	Display > Data–>Memory

SCPI.CALCulate(Ch).SElected.MStatisticks.DATA

<i>Description</i>	Read out 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(Ch).SElected.MStatisticks.DOMain.STATe command (the range limits are determined by two markers). The array includes 3 elements: <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>Type</i>	Variant: array of double (read only)
<i>Target</i>	The active trace of channel <i>Ch</i> , <i>Ch</i> : channel number 1–16
<i>Syntax</i>	<i>Data</i> = app.SCPI.CALCulate(Ch).SElected.MStatisticks.DATA
<i>Equivalent Softkeys</i>	None

SCPI.CALCulate(Ch).SElected.MSTatistics.DOMain.MARKer.START

<i>Description</i>	Sets/gets the number of the marker, which specifies the start frequency of the math statistics range.
<i>Type</i>	Long (read/write)
<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>Target</i>	The active trace of channel <i>Ch</i> , <i>Ch:</i> channel number 1–16
<i>Syntax</i>	<i>MkrNum</i> = app.SCPi.CALCulate(<i>Ch</i>).SElected.MSTatistics.DOMain.MARKer.START <i>app.SCPi.CALCulate(Ch).SElected.MSTatistics.DOMain.MARKer.START</i> = 3
<i>Equivalent Softkeys</i>	Markers > Marker Math > Statistics > Statistics Start

SCPI.CALCulate(Ch).SElected.MSTatistics.DOMain.MARKer.STOP

<i>Description</i>	Sets/gets the number of the marker, which specifies the stop frequency of the math statistics range.
<i>Type</i>	Long (read/write)
<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>Target</i>	The active trace of channel <i>Ch</i> , <i>Ch:</i> channel number 1–16
<i>Syntax</i>	<i>MarkerNum</i> = <i>app.SCPi.CALCulate(Ch).SElected.MSTatistics.DOMain.MARKer.STOP</i> <i>app.SCPi.CALCulate(Ch).SElected.MSTatistics.DOMain.MARKer.STOP</i> = 4
<i>Equivalent Softkeys</i>	Markers > Marker Math > Statistics > Statistics Stop

SCPI.CALCulate(Ch).SElected.MStatisticks.DOMain.STATe

<i>Description</i>	Turns ON/OFF the state of the math statistics range.
<i>Type</i>	Boolean (read/write)
<i>Parameter</i>	True: Statistics range ON False: Statistics range OFF
<i>Preset Value</i>	False
<i>Target</i>	The active trace of channel <i>Ch</i> , <i>Ch</i> : channel number 1–16
<i>Syntax</i>	<i>Status</i> = app.SCPi.CALCulate(<i>Ch</i>).SElected.MStatisticks.DOMain.STATe <i>app.SCPi.CALCulate(Ch).SElected. MStatisticks.DOMain.STATe</i> = true
<i>Equivalent Softkeys</i>	Markers > Marker Math > Statistics > Statistics Range

SCPI.CALCulate(Ch).SElected.MStatisticks.STATe

<i>Description</i>	Turns ON/OFF the state of the math statistics display.
<i>Type</i>	Boolean (read/write)
<i>Parameter</i>	True: Statistics display ON False: Statistics display OFF
<i>Preset Value</i>	False
<i>Target</i>	The active trace of channel <i>Ch</i> , <i>Ch</i> : channel number 1–16
<i>Syntax</i>	<i>Status</i> = app.SCPi.CALCulate(<i>Ch</i>).SElected.MStatisticks.STATe <i>app.SCPi.CALCulate(Ch).SElected.MStatisticks.STATe</i> = true
<i>Equivalent Softkeys</i>	Markers > Marker Math > Statistics > Statistics

SCPI.CALCulate(Ch).SElected.RLIMit.DATA

<i>Description</i>	Sets/gets 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. For the n-th point, where n from 1 to N: <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-2)</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>Type</i>	Variant: array of double (read/write)
<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>Target</i>	The active trace of channel <i>Ch</i>, <i>Ch</i>: channel number 1-16
<i>Syntax</i>	<i>Data</i> = app.SCPI.CALCulate(<i>Ch</i>).SElected.RLIMit.DATA app.SCPI.CALCulate(<i>Ch</i>).SElected.RLIMit.DATA = Array(1,1,800,900,10)
<i>Equivalent Softkeys</i>	Analysis > Ripple Limit > Edit Ripple Limit

SCPI.CALCulate(Ch).SElected.RLIMit.DISPlay.LINE

<i>Description</i>	Turns ON/OFF the state of the ripple limit line display.
<i>Type</i>	Boolean (read/write)
<i>Parameter</i>	True: Ripple limit line ON False: Ripple limit line OFF
<i>Preset Value</i>	False
<i>Target</i>	The active trace of channel <i>Ch</i> , <i>Ch</i> : channel number 1–16
<i>Syntax</i>	<i>Status</i> = app.SCPi.CALCulate(<i>Ch</i>).SElected.RLIMit.DISPlay.LINE <i>app.SCPi.CALCulate(Ch).SElected.RLIMit.DISPlay.LINE</i> = true
<i>Equivalent Softkeys</i>	Analysis > Ripple Limit > Ripple Limit

SCPI.CALCulate(Ch).SElected.RLIMit.DISPlay.SELect

<i>Description</i>	Sets/gets the number of the ripple limit test band selected for the ripple value display.
<i>Type</i>	Long (read/write)
<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>Target</i>	The active trace of channel <i>Ch</i> , <i>Ch</i> : channel number 1–16
<i>Syntax</i>	<i>Value</i> = app.SCPi.CALCulate(<i>Ch</i>).SElected.RLIMit.DISPlay.SELect <i>app.SCPi.CALCulate(Ch).SElected.RLIMit.DISPlay.SELect</i> = 2
<i>Equivalent Softkeys</i>	Analysis > Ripple Limit > Ripple Value Band

SCPI.CALCulate(Ch).SElected.RLIMit.DISPlay.VALue

<i>Description</i>	Sets/gets the selection of the display type of the ripple value in the specified band.
<i>Type</i>	String (read/write)
<i>Parameter</i>	"OFF" : Ripple value display OFF "ABSolute" : Absolute value "MARgin" : Margin (difference between the ripple limit and the absolute value)
<i>Preset Value</i>	"OFF"
<i>Target</i>	The active trace of channel <i>Ch</i> , <i>Ch</i> : channel number 1–16
<i>Syntax</i>	<i>Param</i> = app.SCPi.CALCulate(<i>Ch</i>).SElected.RLIMit.DISPlay.VALue app.SCPi.CALCulate(<i>Ch</i>).SElected.RLIMit.DISPlay.VALue = "ABS"
<i>Equivalent Softkeys</i>	Analysis > Ripple Limit > Ripple Value

SCPI.CALCulate(Ch).SElected.RLIMit.FAIL

<i>Description</i>	Reads out the Ripple limit test result.
<i>Type</i>	Boolean (read only)
<i>Parameter</i>	True: Fail False: Pass
<i>Target</i>	The active trace of channel <i>Ch</i> , <i>Ch</i> : channel number 1–16
<i>Syntax</i>	<i>Status</i> = app.SCPi.CALCulate(<i>Ch</i>).SElected.RLIMit.FAIL
<i>Equivalent Softkeys</i>	None

SCPI.CALCulate(Ch).SElected.RLIMit.REPort.DATA

<i>Description</i>	Reads out 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. For the n-th point, where n from 1 to N: $Data(0)$ N total number of the bands; $Data(3n-2)$ n number of the band; $Data(3n-1)$ $Ripple$ value in the n-th band; $Data(3n-0)$ $Ripple$ limit test result in the n-th band: 0: Pass 1: Fail
<i>Type</i>	Variant: array of double (read only)
<i>Target</i>	The active trace of channel Ch , Ch : channel number 1–16
<i>Syntax</i>	$Data = app.SCPICALCulate(Ch).SElected.RLIMit.REPort.DATA$
<i>Equivalent Softkeys</i>	None

SCPI.CALCulate(Ch).SElected.RLIMit.STATe

<i>Description</i>	Turns ON/OFF the state of the ripple limit test.
<i>Type</i>	Boolean (read/write)
<i>Parameter</i>	True: Ripple limit test ON False: Ripple limit test OFF
<i>Preset Value</i>	False
<i>Target</i>	The active trace of channel Ch , Ch : channel number 1–16
<i>Syntax</i>	$Status = app.SCPICALCulate(Ch).SElected.RLIMit.STATe$ $app.SCPICALCulate(Ch).SElected.RLIMit.STATe = true$
<i>Equivalent Softkeys</i>	Analysis > Ripple Limit > Ripple Test

SCPI.CALCulate(Ch).SElected.SMOothing.APERture

<i>Description</i>	Sets/gets the smoothing aperture for the smoothing function.
<i>Type</i>	Double (read/write)
<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>Target</i>	The active trace of channel <i>Ch</i> , <i>Ch</i> : channel number 1–16
<i>Syntax</i>	<i>Value</i> = app.SCPi.CALCulate(<i>Ch</i>).SElected.SMOothing.APERture app.SCPi.CALCulate(<i>Ch</i>).SElected.SMOothing.APERture = 1.5
<i>Equivalent Softkeys</i>	Average > Smo Aperture

SCPI.CALCulate(Ch).SElected.SMOothing.STATe

<i>Description</i>	Turns ON/OFF the state of the trace smoothing function.
<i>Type</i>	Boolean (read/write)
<i>Parameter</i>	True: Trace smoothing ON False: Trace smoothing OFF
<i>Preset Value</i>	False
<i>Target</i>	The active trace of channel <i>Ch</i> , <i>Ch</i> : channel number 1–16
<i>Syntax</i>	<i>Status</i> = app.SCPi.CALCulate(<i>Ch</i>).SElected.SMOothing.STATe app.SCPi.CALCulate(<i>Ch</i>).SElected.SMOothing.STATe = true
<i>Equivalent Softkeys</i>	Average > Smoothing

SCPI.CALCulate(Ch).SElected.TRANSform.TIME.CENter

<i>Description</i>	Sets/gets the time (distance) domain center value, when the time domain transformation function is turned ON.
<i>Type</i>	Double (read/write)
<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 (second) m (meters) or ft (feet)
<i>Target</i>	The active trace of channel <i>Ch</i> , <i>Ch</i> : channel number 1–16
<i>Syntax</i>	<i>Value</i> = app.SCPI.CALCulate(Ch).SElected.TRANSform.TIME.CENter app.SCPI.CALCulate(Ch).SElected.TRANSform.TIME.CENter = 1e–8
<i>Related Commands</i>	SCPI.CALCulate(Ch).SElected.TRANSform.TIME.UNIT
<i>Equivalent Softkeys</i>	Analysis > Time Domain > Center

SCPI.CALCulate(Ch).SElected.TRANSform.TIME.IMPulse.WIDTH

<i>Description</i>	Sets/gets 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>Type</i>	Double (read/write)
<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 (second)
<i>Target</i>	The active trace of channel <i>Ch</i> , <i>Ch</i> : channel number 1–16
<i>Syntax</i>	<i>Value</i> = app.SCPI.CALCulate(<i>Ch</i>).SElected.TRANSform.TIME.IMPulse.WIDTH app.SCPI.CALCulate(<i>Ch</i>).SElected.TRANSform.TIME.IMPulse.WIDTH = 1e–8
<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>Description</i>	Sets/gets the β parameter, which controls the Kaiser–Bessel window shape, when performing time domain transformation.
<i>Type</i>	Double (read/write)
<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>Target</i>	The active trace of channel <i>Ch</i> , <i>Ch</i> : channel number 1–16
<i>Syntax</i>	<i>Value</i> = app.SCPi.CALCulate(<i>Ch</i>).SElected.TRANSform.TIME.KBESsel app.SCPi.CALCulate(<i>Ch</i>).SElected.TRANSform.TIME.KBESsel = 13
<i>Equivalent Softkeys</i>	Analysis > Time Domain > Window > Kaiser Beta

SCPI.CALCulate(Ch).SElected.TRANSform.TIME.LPFRequency

<i>Description</i>	Changes the frequency range to match with the lowpass type of the time domain transformation function.
<i>Type</i>	Method
<i>Target</i>	The active trace of channel <i>Ch</i> , <i>Ch</i> : channel number 1–16
<i>Syntax</i>	app.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>Description</i>	Sets/gets the selection of the reflection distance either one way or round trip for the time domain transformation function.
<i>Type</i>	String (read/write)
<i>Parameter</i>	"RTRip" Round Trip "OWAY" One Way
<i>Preset Value</i>	"RTRip"
<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>Syntax</i>	<i>Param</i> = <i>app</i> .SCPI.CALCulate(<i>Ch</i>).SElected.TRANSform.TIME.REFlection.TYPE <i>app</i> .SCPI.CALCulate(<i>Ch</i>).SElected.TRANSform.TIME.REFlection.TYPE = "RTR"
<i>Equivalent Softkeys</i>	Analysis > Time Domain > Reflection Type > Round Trip One Way

SCPI.CALCulate(Ch).SElected.TRANSform.TIME.SPAN

<i>Description</i>	Sets/gets the time (distance) domain span value, when the time domain transformation function is turned ON.
<i>Type</i>	Double (read/write)
<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>	2e-8
<i>Unit</i>	s (second) m (meters) or ft (feet)
<i>Target</i>	The active trace of channel <i>Ch</i> , <i>Ch</i> : channel number 1-16
<i>Syntax</i>	<i>Value</i> = app.SCPI.CALCulate(<i>Ch</i>).SElected.TRANSform.TIME.SPAN <i>app.</i> SCPI.CALCulate(<i>Ch</i>).SElected.TRANSform.TIME.SPAN = 1e-8
<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.STARTt

<i>Description</i>	Sets/gets the start value used for the transformation function of the time domain function. The time (distance) domain start value, when the time domain transformation function is turned ON.
<i>Type</i>	Double (read/write)
<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>	-1e-8
<i>Unit</i>	s (second) m (meters) or ft (feet)
<i>Target</i>	The active trace of channel <i>Ch</i> , <i>Ch</i> : channel number 1-16
<i>Syntax</i>	<i>Value</i> = app.SCPi.CALCulate(<i>Ch</i>).SElected.TRANSform.TIME.STARTt app.SCPi.CALCulate(<i>Ch</i>).SElected.TRANSform.TIME.STARTt = 1e-8
<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>Description</i>	Turns ON/OFF the state of the time domain transformation function.
<i>Type</i>	Boolean (read/write)
<i>Parameter</i>	True: Time domain transformation ON False: Time domain transformation OFF
<i>Preset Value</i>	False
<i>Target</i>	The active trace of channel <i>Ch</i> , <i>Ch</i> : channel number 1–16
<i>Syntax</i>	<i>Status</i> = app.SCPi.CALCulate(<i>Ch</i>).SElected.TRANSform.TIME.STATe <i>app.SCPi.CALCulate(Ch).SElected.TRANSform.TIME.STATe</i> = true
<i>Equivalent Softkeys</i>	Analysis > Time Domain > Time Domain

SCPI.CALCulate(Ch).SElected.TRANSform.TIME.STEP.RTIme

<i>Description</i>	Sets/gets 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>Type</i>	Double (read/write)
<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 (second)
<i>Target</i>	The active trace of channel <i>Ch</i> , <i>Ch</i> : channel number 1–16
<i>Syntax</i>	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
<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>Description</i>	Sets/gets the selection of the stimulus type for the time domain transformation function: impulse or step.
<i>Type</i>	String (read/write)
<i>Parameter</i>	"IMPulse" Impulse "STEP" Step
<i>Preset Value</i>	"IMP"
<i>Target</i>	The active trace of channel <i>Ch</i> , <i>Ch</i> : channel number 1–16
<i>Syntax</i>	<i>Param</i> = app.SCPi.CALCulate(<i>Ch</i>).SElected.TRANSform.TIME.STIMulus app.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>Description</i>	Sets/gets the time (distance) domain stop value, when the time domain transformation function is turned ON.
<i>Type</i>	Double (read/write)
<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>	1e-8
<i>Unit</i>	s (second) m (meters) or ft (feet)
<i>Target</i>	The active trace of channel <i>Ch</i> , <i>Ch</i> : channel number 1-16
<i>Syntax</i>	<i>Value</i> = app.SCPI.CALCulate(<i>Ch</i>).SElected.TRANSform.TIME.STOP app.SCPI.CALCulate(<i>Ch</i>).SElected.TRANSform.TIME.STOP = 2e-8
<i>Equivalent Softkeys</i>	Analysis > Time Domain > Stop

SCPI.CALCulate(Ch).SElected.TRANSform.TIME.TYPE

<i>Description</i>	Sets/gets the selection of the transformation type for the time domain transformation function: bandpass response or direct current circuit.
<i>Type</i>	String (read/write)
<i>Parameter</i>	"BPASs" Bandpass "LPASs" Lowpass
<i>Preset Value</i>	"BPASs"
<i>Target</i>	The active trace of channel <i>Ch</i> , <i>Ch</i> : channel number 1–16
<i>Syntax</i>	<i>Param</i> = app.SCPI.CALCulate(<i>Ch</i>).SElected.TRANSform.TIME.TYPE app.SCPI.CALCulate(<i>Ch</i>).SElected.TRANSform.TIME.TYPE = "STEP"
<i>Equivalent Softkeys</i>	Analysis > Time Domain > Type > Bandpass Lowpass Step Lowpass Impulse

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

<i>Description</i>	Sets/gets the selection of the transformation unit for the time domain transformation function: seconds, meters, feet.
<i>Type</i>	String (read/write)
<i>Parameter</i>	"SEConds" Seconds "METers" Meters "FEET" Feet
<i>Preset Value</i>	"SEConds"
<i>Target</i>	The active trace of channel <i>Ch</i> , <i>Ch</i> : channel number 1–16
<i>Syntax</i>	<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>Description</i>	Reads out or writes the formatted data array. The array elements contain measurements in the current format, for example, in logarithmic magnitude format (Log Mag). Also, see section 14. 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>Type</i>	Variant: array of double (read/write)
<i>Target</i>	Trace <i>Tr</i> of channel <i>Ch</i>, <i>Ch</i>: channel number 1-16 <i>Tr</i>: trace number 1-16
<i>Syntax</i>	Data = app.SCPI.CALCulate(<i>Ch</i>).Trace(<i>Tr</i>).DATA.FDATa app.SCPI.CALCulate(<i>Ch</i>).Trace(<i>Tr</i>).DATA.FDATa = Data
<i>Equivalent Softkeys</i>	None

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

<i>Description</i>	Reads out or writes 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 14. 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>Type</i>	Variant: array of double (read/write)
<i>Target</i>	Trace <i>Tr</i> of channel <i>Ch</i>, <i>Ch</i>: channel number 1–16 <i>Tr</i>: trace number 1–16
<i>Syntax</i>	<i>Data</i> = app.SCPI.CALCulate(<i>Ch</i>).Trace(<i>Tr</i>).DATA.FMEMory app.SCPI.CALCulate(<i>Ch</i>).Trace(<i>Tr</i>).DATA.FMEMory = <i>Data</i>
<i>Equivalent Softkeys</i>	None

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

<i>Description</i>	Reads out or writes the corrected data array. The corrected measurements are complex numbers. Also, see section 14. The array size is $2N$, where N is the number of measurement points. For the n-th point, where n from 1 to N: $Data(2n-2)$ the real part of corrected measurement; $Data(2n-1)$ the imaginary part of corrected measurement.
<i>Type</i>	Variant: array of double (read/write)
<i>Target</i>	Trace Tr of channel Ch, Ch: channel number 1–16 Tr: trace number 1–16
<i>Syntax</i>	$Data = app.SCPICALCulate(Ch).TRACE(Tr).DATA.SDATA$ $app.SCPICALCulate(Ch).TRACE(Tr).DATA.SDATA = Data$
<i>Equivalent Softkeys</i>	None

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

<i>Description</i>	Reads out or writes the corrected memory array. The corrected measurements are complex numbers. Also, see section 14. 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>Type</i>	Variant: array of double (read/write)
<i>Target</i>	Trace <i>Tr</i> of channel <i>Ch</i>, <i>Ch</i>: channel number 1–16 <i>Tr</i>: trace number 1–16
<i>Syntax</i>	<i>Data</i> = app.SCPI.CALCulate(<i>Ch</i>).Trace(<i>Tr</i>).DATA.SMEMory app.SCPI.CALCulate(<i>Ch</i>).Trace(<i>Tr</i>).DATA.SMEMory = <i>Data</i>
<i>Equivalent Softkeys</i>	None

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

<i>Description</i>	Reads out the trace X axis data array. The array size is N, where N is the number of measurement points. For the n-th point, where n from 0 to $N-1$: <i>Data</i>(n) the X axis value;
<i>Type</i>	Variant: array of double (read only)
<i>Target</i>	Trace <i>Tr</i> of channel <i>Ch</i>, <i>Ch</i>: channel number 1–16 <i>Tr</i>: trace number 1–16
<i>Syntax</i>	<i>Data</i> = app.SCPI.CALCulate(<i>Ch</i>).Trace(<i>Tr</i>).DATA.XAXis
<i>Equivalent Softkeys</i>	None

SCPI.DISPlay.COLOr.BACK

<i>Description</i>	Sets/gets the background color for trace display. 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>Type</i>	Variant: array of long (read/write)
<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>	<i>Data</i> = app.SCPI.DISPlay.COLOr.BACK <i>app.SCPI.DISPlay.COLOr.BACK</i> = Array(255, 255, 255)
<i>Equivalent Softkeys</i>	Display > Properties > Color > Background > Red Green Blue

SCPI.DISPlay.COLOr.GRATicule

<i>Description</i>	Sets/gets the grid and the graticule label color for trace display. 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>Type</i>	Variant: array of long (read/write)
<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>	<i>Data(0)</i> 160; <i>Data(1)</i> 160; <i>Data(2)</i> 164.
<i>Syntax</i>	<i>Data</i> = app.SCPI.DISPlay.COLOr.GRATicule <i>app.SCPI.DISPlay.COLOr. GRATicule</i> = Array(128, 128, 128)
<i>Equivalent Softkeys</i>	Display > Properties > Color > Grid > Red Green Blue

SCPI.DISPlay.COLOr.RESet

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

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

<i>Description</i>	Sets/gets the data trace color. The array contains 3 elements: <i>Data</i>(0) Red value R; <i>Data</i>(1) Green value G; <i>Data</i>(2) Blue value B.
<i>Type</i>	Variant: array of long (read/write)
<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>Target</i>	Trace number <i>Tr</i> in all channels, <i>Tr</i> : trace number 1–16
<i>Syntax</i>	<i>Data</i> = app.SCPI.DISPlay.COLor.TRACe(<i>Tr</i>).DATA app.SCPI.DISPlay.COLor.TRACe(<i>Tr</i>).DATA = Array(255, 255, 0)
<i>Equivalent Softkeys</i>	Display > Properties > Color > Data Trace > Red Green Blue

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

<i>Description</i>	Sets/gets the memory trace color. The array contains 3 elements: <i>Data</i>(0) Red value R; <i>Data</i>(1) Green value G; <i>Data</i>(2) Blue value B.
<i>Type</i>	Variant: array of long (read/write)
<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>Target</i>	Trace number <i>Tr</i> in all channels <i>Tr</i>: trace number 1–16
<i>Syntax</i>	<i>Data</i> = app.SCPI.DISPlay.COLOr.TRACe(<i>Tr</i>).MEMory app.SCPI.DISPlay.COLOr.TRACe(<i>Tr</i>).MEMory = Array(255, 255, 0)
<i>Equivalent Softkeys</i>	Display > Properties > Color > Data Trace > Red Green Blue

SCPI.DISPlay.ENABLE

<i>Description</i>	Turns ON/OFF the display update.
<i>Type</i>	Boolean (read/write)
<i>Parameter</i>	True: Display update ON False: Display update OFF
<i>Preset Value</i>	True
<i>Syntax</i>	<i>Status</i> = app.SCPI.DISPlay.ENABLE app.SCPI.DISPlay.ENABLE = true
<i>Equivalent Softkeys</i>	Display > Update

SCPI.DISPlay.FSIGN

<i>Description</i>	Turns ON/OFF the state of the <i>Fail</i> sign display, when performing limit test or ripple limit test.
<i>Type</i>	Boolean (read/write)
<i>Parameter</i>	True: <i>Fail</i> sign display ON False: <i>Fail</i> sign display OFF
<i>Preset Value</i>	False
<i>Syntax</i>	<i>Status</i> = app.SCPI.DISPlay.FSIGN <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>Description</i>	Sets/gets the Graticule Label state. (S2VNA only)
<i>Type</i>	String (read/write)
<i>Parameter</i>	"OFF" : Graticule label is OFF "ACTive" : Only active trace has graticule label "ALL" : All traces have graticule label
<i>Preset Value</i>	"ACTive"
<i>Syntax</i>	<i>Param</i> = app.SCPI.DISPlay.GLABEL <i>app.SCPI.DISPlay.GLABEL</i> = "OFF"
<i>Equivalent Softkeys</i>	Display > Properties > Graticule Label

SCPI.DISPlay.IMAGe

<i>Description</i>	Sets/gets the inverted color display of the data traces.
<i>Type</i>	String (read/write)
<i>Parameter</i>	"NORMal" : Normal display "INVert" : Inverted color display
<i>Preset Value</i>	"NORM"
<i>Syntax</i>	Param = app.SCPI.DISPlay.IMAGe app.SCPI.DISPlay.IMAGe = "INV"
<i>Equivalent Softkeys</i>	Display > Properties > Invert Color

SCPI.DISPlay.HIDE

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

SCPI.DISPlay.MAXimize

<i>Description</i>	Turns ON/OFF the state of the maximization of the active channel window.
<i>Type</i>	Boolean (read/write)
<i>Parameter</i>	True: maximization ON False: maximization OFF
<i>Preset Value</i>	False
<i>Target</i>	The active channel window
<i>Syntax</i>	<i>Status</i> = app.SCPI.DISPlay.MAXimize <i>app</i> .SCPI.DISPlay.MAXimize = true
<i>Equivalent Softkeys</i>	Display > Active Trace/Channel > Maximize Channel

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

<i>Description</i>	Sets/gets the font size of the item specified by Parameter. (S2VNA only)
<i>Type</i>	Long (read/write)
<i>Parameter</i>	"CHANnel" : Channel window "BUTTon" : Soft buttons "MENU" : Menu bar "CStatus" : Channel status "AStatus" : Analyzer status
<i>Range</i>	From 10 to 22.
<i>Preset Value</i>	"11"
<i>Syntax</i>	<i>Size</i> = <i>app</i> .SCPI.DISPlay.PARTition.FONT.SIZE("CHAN") <i>app</i> .SCPI.DISPlay.PARTition.FONT.SIZE("CHAN") = 20
<i>Equivalent Softkeys</i>	Display > Properties > Font > Size

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

<i>Description</i>	Shows or hides the display partition specified by Parameter. (S2VNA only)
<i>Type</i>	Boolean (read/write)
<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>Syntax</i>	<i>State</i> = <i>app</i> .SCPI.DISPlay.PARTition.VISible("MENU") <i>app</i> .SCPI.DISPlay.PARTition.VISible("MENU") = true
<i>Equivalent Softkeys</i>	Display > Properties > Menu Bar Display > Display Properties > Frequency Label Markers > Properties > Marker Table or None

SCPI.DISPlay.POSition

<i>Description</i>	Sets/gets the application window position on the screen and its dimension. The array contains 4 elements: 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. (S2VNA only)
<i>Type</i>	Variant: array of long (read/write)
<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>	Data(0) – (screen width – 800) / 2, Data(1) – (screen height – 600) / 2, Data(2) – 800, Data(3) – 600, Preset: Display > Properties > Set Defaults
<i>Syntax</i>	<i>Pos</i> = <i>app</i>.SCPI.DISPlay.POSition <i>app</i>.SCPI.DISPlay.POSition = Array(0, 0, 800, 600)
<i>Equivalent Softkeys</i>	None

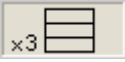
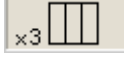
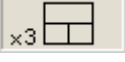
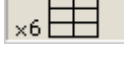
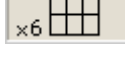
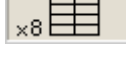
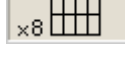
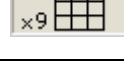
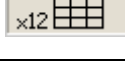
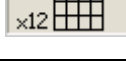
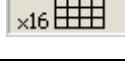
SCPI.DISPlay.SHOW

<i>Description</i>	Shows the analyzer GUI hidden by the SCPI.DISPlay.HIDE command.
<i>Type</i>	Method
<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>Description</i>	Sets/gets the number and layout of the channel windows on the screen. The channel window layout is in Table 1 below.
<i>Type</i>	Long (read/write)
<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>	<i>Value</i> = <i>app</i> .SCPI.DISPlay.SPLit <i>app</i> .SCPI.DISPlay.SPLit = 2
<i>Equivalent Softkeys</i>	Display > Allocate channels

Table 1. 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>Description</i>	Updates the display once, when the display update is set to OFF (SCPI.DISPlay.ENABLE is set to <i>False</i>).
<i>Type</i>	Method
<i>Syntax</i>	<i>app</i> .SCPI.DISPlay.REFresh.IMMediate <i>app</i> .SCPI.DISPlay.UPDate.IMMediate
<i>Equivalent Softkeys</i>	None

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

<i>Description</i>	Selects the active channel.
<i>Type</i>	Method
<i>Notes</i>	The channel window must be displayed. In attempt to set to the active channel the channel, which is not displayed, an error occurs.
<i>Target</i>	Channel <i>Ch</i> from 1 to 16
<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>Description</i>	Sets/gets 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 <i>False</i>).
<i>Type</i>	String (read/write)
<i>Parameter</i>	"VERTical" : Vertical alignment "HORizontal" : Horizontal alignment "NONE" : No alignment
<i>Preset Value</i>	"NONE"
<i>Target</i>	Channel <i>Ch</i> from 1 to 16
<i>Syntax</i>	<code>Param = app.SCPi.DISPlay.WINDow(<i>Ch</i>).ANNotation.MARKer.ALIGn.TYPE</code> <code>app.SCPi.DISPlay.WINDow(<i>Ch</i>).ANNotation.MARKer.ALIGn.TYPE = "VERT"</code>
<i>Equivalent Softkeys</i>	Markers > Properties > Align > Vertical Horizontal OFF

SCPI.DISPlay.WINDow(*Ch*).ANNotation.MARKer.SINGle.STATe

<i>Description</i>	Turns ON/OFF the state of the marker display for the active trace only.
<i>Type</i>	Boolean (read/write)
<i>Parameter</i>	True: Only active trace markers display False: Markers of All traces display
<i>Preset Value</i>	True
<i>Target</i>	Channel <i>Ch</i> from 1 to 16
<i>Syntax</i>	<i>Status</i> = app.SCPI SCPI.DISPlay.WINDow(<i>Ch</i>).ANNotation.MARKer.SINGle.STATe <i>app.SCPI SCPI.DISPlay.WINDow(<i>Ch</i>).ANNotation.MARKer.SINGle.STATe</i> = true
<i>Equivalent Softkeys</i>	Markers > Properties > Active Only

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

<i>Description</i>	Turns ON/OFF the state of the active trace maximization of the specified channel
<i>Type</i>	Boolean (read/write)
<i>Parameter</i>	True: maximization ON False: maximization OFF
<i>Preset Value</i>	False
<i>Target</i>	The active trace of channel <i>Ch</i> , <i>Ch</i> : channel number 1–16
<i>Syntax</i>	<i>Status</i> = app.SCPI.DISPlay.WINDow(<i>Ch</i>).MAXimize <i>app.SCPI.DISPlay.WINDow(<i>Ch</i>).MAXimize</i> = true
<i>Equivalent Softkeys</i>	Display > Active Trace/Channel > Maximize Trace

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

<i>Description</i>	Sets/gets the code of the graph layout of channels. The channel window layout is in Table 1 above.
<i>Type</i>	Long (read/write)
<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 command 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 command.
<i>Target</i>	Channel <i>Ch</i> from 1 to 16
<i>Syntax</i>	<i>Value</i> = app.SCPI.DISPlay.WINDow(<i>Ch</i>).SPLit app.SCPI.DISPlay.WINDow(<i>Ch</i>).SPLit = 2
<i>Equivalent Softkeys</i>	Display > Allocate Traces

SCPI.DISPlay.WINDow(*Ch*).TITLe.DATA

<i>Description</i>	Sets/gets the channel title label.
<i>Type</i>	String (read/write)
<i>Target</i>	Channel <i>Ch</i> from 1 to 16
<i>Syntax</i>	<i>Text</i> = app.SCPI.DISPlay.WINDow(<i>Ch</i>).TITLe.DATA app.SCPI.DISPlay.WINDow(<i>Ch</i>).TITLe.DATA = "Network 1"
<i>Equivalent Softkeys</i>	Display > Edit Title Label

SCPI.DISPlay.WINDow(*Ch*).TITLe.STATe

<i>Description</i>	Turns ON/OFF the state of the title label display.
<i>Type</i>	Boolean (read/write)
<i>Parameter</i>	True: Title label display ON False: Title label display OFF
<i>Preset Value</i>	False
<i>Target</i>	Channel <i>Ch</i> from 1 to 16
<i>Syntax</i>	<i>Status</i> = app.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>Description</i>	Turns ON/OFF the state of the memory value display on the marker.
<i>Type</i>	Boolean (read/write)
<i>Parameter</i>	True: Memory value display ON False: Memory value display OFF
<i>Target</i>	Trace <i>Tr</i> of channel <i>Ch</i> , <i>Ch</i> : channel number 1–16 <i>Tr</i> : trace number 1–16
<i>Syntax</i>	<i>Status</i> = <i>app</i> .SCPI.DISPlay.WINDow(<i>Ch</i>).TRACe(<i>Tr</i>).ANNotation.MARKer.MEMory <i>app</i> .SCPI.DISPlay.WINDow(<i>Ch</i>).TRACe(<i>Tr</i>).ANNotation.MARKer.MEMory = true
<i>Equivalent Softkeys</i>	Markers > Properties > Memory Value

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

<i>Description</i>	Sets/gets the display position of the marker value on the X-axis by a percentage of the display width.
<i>Type</i>	Double (read/write)
<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>Target</i>	Trace <i>Tr</i> of channel <i>Ch</i> , <i>Ch</i> : channel number 1–16 <i>Tr</i> : trace number 1–16
<i>Syntax</i>	<i>Value</i> = app.SCPI.DISPlay.WINDow(<i>Ch</i>).TRACe(<i>Tr</i>).ANNotation.MARKer.POSition.X app.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>Description</i>	Sets/gets the display position of the marker value on the Y-axis by a percentage of the display height.
<i>Type</i>	Double (read/write)
<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>Target</i>	Trace <i>Tr</i> of channel <i>Ch</i> , <i>Ch</i> : channel number 1–16 <i>Tr</i> : trace number 1–16
<i>Syntax</i>	<i>Value</i> = app.SCPI.DISPlay.WINDow(<i>Ch</i>).TRACe(<i>Tr</i>).ANNotation.MARKer.POSition.Y app.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>Description</i>	Turns ON/OFF the state of the memory trace display.
<i>Type</i>	Boolean (read/write)
<i>Parameter</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>Target</i>	Trace <i>Tr</i> of channel <i>Ch</i> , <i>Ch</i> : channel number 1–16 <i>Tr</i> : trace number 1–16
<i>Syntax</i>	<i>Status</i> = app.SCPI.DISPlay.WINDow(<i>Ch</i>).TRACe(<i>Tr</i>).MEMory.STATe app.SCPI.DISPlay.WINDow(<i>Ch</i>).TRACe(<i>Tr</i>).MEMory.STATe = true
<i>Equivalent Softkeys</i>	Display > Display > Memory Data & Memory (ON) Display > Display > Data OFF (OFF)

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

<i>Description</i>	Turns ON/OFF the state of the data trace display.
<i>Type</i>	Boolean (read/write)
<i>Parameter</i>	True: Data trace display ON False: Data trace display OFF
<i>Preset Value</i>	True
<i>Target</i>	Trace <i>Tr</i> of channel <i>Ch</i> , <i>Ch</i> : channel number 1–16 <i>Tr</i> : trace number 1–16
<i>Syntax</i>	<i>Status</i> = app.SCPI.DISPlay.WINDow(<i>Ch</i>).TRACe(<i>Tr</i>).STATe app.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>Description</i>	Executes the auto scale function for the trace.
<i>Type</i>	Method
<i>Target</i>	Trace <i>Tr</i> of channel <i>Ch</i> , <i>Ch</i> : channel number 1–16 <i>Tr</i> : trace number 1–16
<i>Syntax</i>	app.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>Description</i>	Sets/gets 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>Type</i>	Double (read/write)
<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>Target</i>	Trace <i>Tr</i> of channel <i>Ch</i> , <i>Ch</i> : channel number 1-16 <i>Tr</i> : trace number 1-16
<i>Syntax</i>	<i>Value</i> = 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
<i>Equivalent Softkeys</i>	Scale > Scale

SCPI.DISPlay.WINDow(*Ch*).TRACe(*Tr*).Y.SCALe.RLEVel

<i>Description</i>	Sets/gets the value of the reference line (response value on the reference line). For the rectangular format only.
<i>Type</i>	Double (read/write)
<i>Range</i>	from $-1\text{E}-18$ to $1\text{E}18$
<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>Target</i>	Trace <i>Tr</i> of channel <i>Ch</i> , <i>Ch</i> : channel number 1–16 <i>Tr</i> : trace number 1–16
<i>Syntax</i>	<i>Value</i> = app.SCPI.DISPlay.WINDow(<i>Ch</i>).TRACe(<i>Tr</i>).Y.SCALe.RLEVel app.SCPI.DISPlay.WINDow(<i>Ch</i>).TRACe(<i>Tr</i>).Y.SCALe.RLEVel = 10
<i>Equivalent Softkeys</i>	Scale > Ref Value

SCPI.DISPlay.WINDow(Ch).TRACe(Tr).Y.SCALe.RPOSition

<i>Description</i>	Sets/gets the position of the reference line. For the rectangular format only.
<i>Type</i>	Long (read/write)
<i>Range</i>	From 0 to the number of the scale divisions (set by the SCPI.DISPlay.WINDow(Ch).Y.SCALe.DIVisions command, 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>Target</i>	Trace <i>Tr</i> of channel <i>Ch</i> , <i>Ch</i> : channel number 1–16 <i>Tr</i> : trace number 1–16
<i>Syntax</i>	<i>Value</i> = app.SCPI.DISPlay.WINDow(Ch).TRACe(Tr).Y.SCALe.RPOSition app.SCPI.DISPlay.WINDow(Ch).TRACe(Tr).Y.SCALe.RPOSition = 10
<i>Equivalent Softkeys</i>	Scale > Ref Position

SCPI.DISPlay.WINDow(Ch).X.SPACing

<i>Description</i>	Sets/gets the selection of the display method of the graph horizontal axis for the segment sweep.
<i>Type</i>	String (read/write)
<i>Parameter</i>	"LINear" : Frequency base (linear frequency axis) "OBASe" : Order base (linear axis of the point numbers)
<i>Preset Value</i>	"LIN"
<i>Target</i>	Channel <i>Ch</i> from 1 to 16
<i>Syntax</i>	<i>Param</i> = app.SCPI.DISPlay.WINDow(Ch).X.SPACing app.SCPI.DISPlay.WINDow(Ch).X.SPACing = "OBAS"
<i>Equivalent Softkeys</i>	Stimulus > Segment Table > Segment Display

SCPI.DISPlay.WINDow(*Ch*).Y.SCALe.DIVisions

<i>Description</i>	Sets/gets the number of the vertical scale divisions. For the rectangular format only.
<i>Type</i>	Long (read/write)
<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>Target</i>	Channel <i>Ch</i> from 1 to 16
<i>Syntax</i>	<i>Value</i> = app.SCPI.DISPlay.WINDow(<i>Ch</i>).Y.SCALe.DIVisions app.SCPI.DISPlay.WINDow(<i>Ch</i>).Y.SCALe.DIVisions = 12
<i>Equivalent Softkeys</i>	Scale > Divisions

SCPI.HCOPy.ABORt

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

SCPI.HCOPy.DATE.STAMp

<i>Description</i>	Sets/gets the ON/OFF state of the current date and time printout in the upper right corner.
<i>Type</i>	Boolean (read/write)
<i>Parameter</i>	True: Date & time printout ON False: Date & time printout OFF
<i>Preset Value</i>	True
<i>Syntax</i>	Status = app.SCPI.HCOPy.DATE.STAMp app.SCPI.HCOPy.DATE.STAMp = False
<i>Equivalent Softkeys</i>	System > Print > Print Date & Time

SCPI.HCOPy.IMAGe

<i>Description</i>	Sets/gets the inverted color image printout.
<i>Type</i>	String (read/write)
<i>Parameter</i>	" NORM al" : Normal printout "IN Ver t" : Inverted color printout
<i>Preset Value</i>	"NORM"
<i>Syntax</i>	Param = app.SCPI.HCOPy.IMAGe app.SCPI.HCOPy.IMAGe = "INV"
<i>Equivalent Softkeys</i>	System > Print > Invert Image

SCPI.HCOPy.IMMediate

<i>Description</i>	Prints out the image displayed on the screen without previewing.
<i>Type</i>	Method
<i>Syntax</i>	<code>app.SCPi.HCOPy.IMMediate</code>
<i>Equivalent Softkeys</i>	System > Print > Print Embedded

SCPI.HCOPy.PAINt

<i>Description</i>	Sets/gets the color chart for the image printout.
<i>Type</i>	String (read/write)
<i>Parameter</i>	"COLor" : Color printout "GRAY" : Grayscale printout "BW" : Black&white printout
<i>Preset Value</i>	"BW"
<i>Syntax</i>	<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>Description</i>	Clears the following: • Error Queue • Status Byte Register • Standard Event Status Register • Operation Status Event Register • Questionable Status Event Register • Questionable Limit Status Event Register • Questionable Limit Channel Status Event Register
<i>Type</i>	Method
<i>Target</i>	Status Reporting System
<i>Syntax</i>	<i>app</i> .SCPI.IEEE4882.CLS
<i>Equivalent Softkeys</i>	None

SCPI.IEEE4882.IDN

<i>Description</i>	Reads out the Analyzer identification string.
<i>Type</i>	String (read only)
<i>Syntax</i>	StrName = <i>app</i> .NAME
<i>Reply</i>	The identification string in format: <manufacturer>, <model>, <serial number>, <software version>/<hardware version>. For example: CMT, C1209, 08080188, 16.2/01
<i>Equivalent Softkeys</i>	None

SCPI.IEEE4882.OPC

<i>Description</i>	Reads out the "1" at the completion of all pending operations. Write form of the command sets 1 the OPC bit (bit 0) of the Standard Event Status Register when all of pending operations complete. <i>Note:</i> since COM server executes commands sequentially and any operation is complete before COM server returns control the OPC read command doesn't wait anything.
<i>Type</i>	Long (read/write)
<i>Target</i>	Status Reporting System
<i>Syntax</i>	<i>Value</i> = <i>app</i>.SCPI.IEEE4882.OPC <i>app</i>.SCPI.IEEE4882.OPC = Dummy
<i>Equivalent Softkeys</i>	None

SCPI.IEEE4882.RST

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

SCPI.IEEE4882.TRG

<i>Description</i>	Generates a trigger signal and initiates a sweep under the following conditions. • Trigger source is set to the <i>BUS</i> (set by the command SCPI.TRIGger.SEQuence.SOURce = BUS), otherwise an error occurs and the command is ignored. • Analyzer must be in the <i>trigger waiting</i> state, otherwise (the analyzer is in the <i>measurement</i> state or <i>hold</i> state) an error occurs and the command is ignored. The command is completed immediately after the generation of the trigger signal (does not wait the end of a sweep).
<i>Type</i>	Method
<i>Target</i>	Analyzer
<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>Description</i>	Waits for the execution of all commands sent before this command. <i>Note:</i> since COM server executes commands sequentially and any operation is complete before COM server returns control the WAI command doesn't wait anything.
<i>Type</i>	Method
<i>Target</i>	Analyzer
<i>Syntax</i>	<i>app</i> .SCPI.IEEE4882.WAI
<i>Equivalent Softkeys</i>	None

SCPI.INITiate(Ch).CONTinuous

<i>Description</i>	Turns ON/OFF the <i>continuous trigger initiation</i> mode. When the <i>continuous initiation</i> mode turned ON: • If the <i>Internal</i> trigger source is selected by the command SCPI.TRIGger.SEQuence.SOURce = "INT", then the channel continuously sweeps; • If the trigger source other than the internal is selected, then the channel goes to the <i>trigger waiting</i> state. Upon receipt of a trigger from the selected source, the sweep starts for the channels awaiting trigger. On completion of the sweep the channel goes to the <i>trigger waiting</i> state. When the <i>continuous trigger initiation</i> mode turned OFF the channel is in the <i>Hold</i> state, to initiate a sweep one should use the INIT command.
<i>Type</i>	Boolean (read/write)
<i>Parameter</i>	True: Continuous trigger initiation mode ON False: Continuous trigger initiation mode OFF
<i>Preset Value</i>	True
<i>Target</i>	Channel <i>Ch</i> from 1 to 16
<i>Syntax</i>	<i>Status</i> = app.SCPI.INITiate(<i>Ch</i>).CONTinuous <i>app</i>.SCPI.INITiate(<i>Ch</i>).CONTinuous = 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>Description</i>	Puts the channel to the <i>Trigger Waiting</i> state for the one trigger event. The channel should be in the <i>hold</i> state, otherwise an error occurs and the command is ignored. The channel goes into the <i>Hold</i> as a result of the command SCPI.INITiate(<i>Ch</i>).CONTinuous = False. If the <i>Internal</i> trigger source is selected by the command TRIG:SOUR INT, then the command initiates a sweep in the single channel, otherwise the channel goes to the <i>Waiting for a Single Trigger</i> mode. Upon receipt of a trigger from the selected source, the sweep starts for the channels awaiting trigger. On completion of the sweep the channel goes to the <i>Hold</i> state. The method returns control immediately (does not wait the end of the sweep).
<i>Type</i>	Method
<i>Target</i>	Channel <i>Ch</i> from 1 to 16
<i>Syntax</i>	<i>app</i> .SCPI.INITiate(<i>Ch</i>).IMMEDIATE
<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.MMEMemory.CATalog(*Dir*)

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

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

<i>Description</i>	Copies a file.
<i>Type</i>	Method
<i>Syntax</i>	<i>app</i> .SCPI.MMEMemory.COPY(<i>Src, Dst</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.MMEMemory.DELEte(*File*)

<i>Description</i>	Deletes a file.
<i>Type</i>	Method
<i>Syntax</i>	<i>app</i> .SCPI.MMEMemory.DELEte(<i>File</i>)
<i>Parameter</i>	<i>File</i> – File name. String data type.
<i>Equivalent Softkeys</i>	None

SCPI.MMEMemory.LOAD.CHANnel.STATe

<i>Description</i>	Recalls the analyzer state for the active channel, saved in one of the four memory registers by the SCPI.MMEMemory.STORE.CHANnel.STATe command.
<i>Type</i>	String (write only)
<i>Parameter</i>	"A" : Recall from register A "B" : Recall from register B "C" : Recall from register C "D" : Recall from register D
<i>Target</i>	Active channel
<i>Syntax</i>	<i>app</i> .SCPI.MMEMemory.LOAD.CHANnel.STATe = "A"
<i>Equivalent Softkeys</i>	Save/Recall > Recall Channel > State A B C D

SCPI.MMEMemory.LOAD.CKIT(Ck)

<i>Description</i>	Recalls the definition file for the calibration kit. The file must be saved by the SCPI.MMEMemory.STORE.CKIT(Ck) command.
<i>Type</i>	String (write only)
<i>Parameter</i>	File Name
<i>Target</i>	Calibration kit Ck, Ck: calibration kit number 1–50
<i>Syntax</i>	<i>app</i> .SCPI.MMEMemory.LOAD.CKIT(Ck) = <i>File</i>
<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

SCPI.MMEMemory.LOAD.LIMit

<i>Description</i>	Recalls the specified limit table file. The file must be saved by the SCPI.MMEMemory.STORE.LIMit command.
<i>Type</i>	String (write only)
<i>Parameter</i>	File Name
<i>Target</i>	Active trace of the active channel.
<i>Syntax</i>	<i>app</i> .SCPI.MMEMemory.LOAD.LIMit = <i>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.MMEMemory.LOAD.PLOSs(*Pt*)

<i>Description</i>	Recalls the specified loss compensation table file. The file must be saved by the SCPI.MMEMemory.STORE.PLOSs(<i>Pt</i>) command.
<i>Type</i>	String (write only)
<i>Parameter</i>	File Name
<i>Target</i>	Port <i>Pt</i> of the active channel, <i>Pt:</i> port number 1–2 (4 for S4VNA) .
<i>Syntax</i>	<i>app</i> .SCPI.MMEMemory.LOAD.PLOSs(<i>Pt</i>) = <i>File</i>
<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 Compens > Import Loss Table

SCPI.MMEemory.LOAD.RLIMit

<i>Description</i>	Recalls the ripple limit table file. The file must be saved by the SCPI.MMEemory.STORE.RLIMit command.
<i>Type</i>	String (write only)
<i>Parameter</i>	File Name
<i>Target</i>	Active trace of the active channel.
<i>Syntax</i>	<i>app</i> .SCPI.MMEemory.LOAD.RLIMit = <i>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.MMEemory.LOAD.SEGMent

<i>Description</i>	Recalls the segment table file. The file must be saved by the SCPI.MMEemory.STORE.SEGMent command.
<i>Type</i>	String (write only)
<i>Parameter</i>	File Name
<i>Target</i>	Active channel
<i>Syntax</i>	<i>app</i> .SCPI.MMEemory.LOAD.SEGMent = <i>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>Description</i>	Loads the Touchstone file with the specified name to the measured S-parameters of the active channel. The Touchstone file types s1p, s2p, s3p and s4p are supported. The *.s1p file loads the S11 parameter only. *.s2p file loads all S11, S21, S12 and S22 parameters and so on. On completion of the command, the channel goes to the hold state.
<i>Type</i>	String (write only)
<i>Parameter</i>	File Name
<i>Target</i>	Active channel
<i>Syntax</i>	<i>app</i> .SCPI.MMEMory.LOAD.SNP.DATA = <i>File</i>
<i>Equivalent Softkeys</i>	Save/Recall > Load Data From Touchstone File > To S-parameters...

SCPI.MMEMory.LOAD.SNP.TRACe(*Tr*).MEMory

<i>Description</i>	Loads the Touchstone file with the specified name to the memory trace. The Touchstone file types s1p, s2p, s3p and s4p 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>Type</i>	String (write only)
<i>Parameter</i>	File Name
<i>Target</i>	Trace <i>Tr</i> of active channel, <i>Tr</i> : trace number 1–16 Active channel set by command SCPI.DISPlay.WINDow(<i>Ch</i>).ACTivate
<i>Syntax</i>	<i>app</i> .SCPI.MMEMory.LOAD.SNP.TRACe(<i>Tr</i>).MEMory = <i>File</i>
<i>Equivalent Softkeys</i>	Save/Recall > Load Data From Touchstone File > To Active Trace Memory

SCPI.MMEemory.LOAD.STATe

<i>Description</i>	Recalls the specified analyzer state file. The file must be saved by the SCPI.MMEemory.STORE.STATe command.
<i>Type</i>	String (write only)
<i>Parameter</i>	File Name
<i>Syntax</i>	<i>app</i> .SCPI.MMEemory.LOAD.STATe = <i>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.MMEemory.MDIReactory

<i>Description</i>	Creates a new directory (folder). Contains the full path to the folder being created.
<i>Type</i>	String (write only)
<i>Parameter</i>	Directory Name
<i>Syntax</i>	<i>app</i> .SCPI.MMEemory.MDIReactory = <i>Path</i>
<i>Equivalent Softkeys</i>	None

SCPI.MMEMory.STORE.CHANnel.CLEAr

<i>Description</i>	Clears the memory of the channel state saved by the SCPI.MMEMory.STORE.CHANnel.STATe command.
<i>Type</i>	Method
<i>Syntax</i>	<i>app</i> .SCPI.MMEMory.STORE.CHANnel.CLEAr
<i>Equivalent Softkeys</i>	Save/Recall > Save Channel > Clear States

SCPI.MMEMory.STORE.CHANnel.STATe

<i>Description</i>	Saves the analyzer state of the items set for the active channel into one of the four memory registers.
<i>Type</i>	String (write only)
<i>Parameter</i>	"A" : Save to register A "B" : Save to register B "C" : Save to register C "D" : Save to register D
<i>Target</i>	Active channel
<i>Syntax</i>	<i>app</i> .SCPI.MMEMory.STORE.CHANnel.STATe = "A"
<i>Equivalent Softkeys</i>	Save/Recall > Save Channel > State A B C D

SCPI.MMEMemory.STORE.CKIT(Ck)

<i>Description</i>	Saves the definition file for the calibration kit parameters.
<i>Type</i>	String (write only)
<i>Parameter</i>	File Name
<i>Target</i>	Calibration kit Ck, Ck: calibration kit number 1–50
<i>Syntax</i>	<i>app.SCPi.MMEMemory.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.MMEMemory.STORE.FDATA

<i>Description</i>	Saves the CSV formatted data into a file.
<i>Type</i>	String (write only)
<i>Parameter</i>	File Name
<i>Target</i>	Active trace of the active channel
<i>Syntax</i>	<i>app.SCPi.MMEMemory.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.MMEMemory.STORE.IMAGE

<i>Description</i>	Saves the display image in BMP or PNG format into a file.
<i>Type</i>	String (write only)
<i>Parameter</i>	File Name
<i>Syntax</i>	<i>app.SCPi.MMEMemory.STORE.IMAGE = File</i>
<i>Notes</i>	If the full path to the file is not specified, the file will be saved to the <i>\Image</i> subdirectory of the main directory. If the file has <i>.png</i> 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.MMEMemory.STORE.LIMit

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

SCPI.MMEMory.STORe.PLOsS(*Pt*)

<i>Description</i>	Saves the loss compensation table into a file with the specified name.
<i>Type</i>	String (write only)
<i>Parameter</i>	File Name
<i>Target</i>	Port <i>Pt</i> of the active channel, <i>Pt:</i> port number 1–2 (4 for S4VNA)
<i>Syntax</i>	<i>app</i> .SCPI.MMEMory.STORe.PLOsS(<i>Pt</i>) = <i>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 Compens > Export Loss Table

SCPI.MMEMory.STORe.RLIMit

<i>Description</i>	Saves the ripple limit table into a file with the specified name.
<i>Type</i>	String (write only)
<i>Parameter</i>	File Name
<i>Target</i>	Active trace of the active channel
<i>Syntax</i>	<i>app</i> .SCPI.MMEMory.STORe.RLIMit = <i>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.STORe.SEGMent

<i>Description</i>	Saves the segment table in a file with the specified name.
<i>Type</i>	String (write only)
<i>Parameter</i>	File Name
<i>Target</i>	Active channel
<i>Syntax</i>	<i>app</i> .SCPI.MMEMory.STORe.SEGMent = <i>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>Description</i>	Saves the measured S-parameters of the active channel into a Touchstone file with the specified name. The file type (s1p, s2p, s3p or s4p) is set by the SCPI.MMEMory.STORe.SNP.TYPE.S1P to SCPI.MMEMory.STORe.SNP.TYPE.S4P properties. The 1-port type file saves one reflection parameter: S11 or S22. The 2-port type file saves all the four parameters: S11, S21, S12, S22 and so on.
<i>Type</i>	String (write only)
<i>Parameter</i>	File Name
<i>Target</i>	Active channel
<i>Syntax</i>	<i>app</i> .SCPI.MMEMory.STORe.SNP.DATA = <i>File</i>
<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.
<i>Equivalent Softkeys</i>	Save/Recall > Save Data to Touchstone File > Save File...

SCPI.MMEMemory.STORE.SNP.FORMat

<i>Description</i>	Sets/gets the data format for the S-parameters saving by the SCPI.MMEMemory.STORE.SNP.DATA command.
<i>Type</i>	String (read/write)
<i>Parameter</i>	" MA" : Logarithmic Magnitude / Angle format " DB" : Linear Magnitude / Angle format " RI" : Real part /Imaginary part format
<i>Preset Value</i>	"RI"
<i>Syntax</i>	Param = app.SCPI.MMEMemory.STORE.SNP.FORMat app.SCPI.MMEMemory.STORE.SNP.FORMat = "DB"
<i>Equivalent Softkeys</i>	Save/Recall > Save Data to Touchstone File > Format

SCPI.MMEMemory.STORE.SNP.SEParator

<i>Description</i>	Sets/gets the Touchstone file separator symbol when the S-parameters are saved by the SCPI.MMEMemory.STORE.SNP.DATA command.
<i>Type</i>	String (read/write)
<i>Parameter</i>	" TAB" : Tab symbol (0x09) " SPACe" : Space symbol (0x20)
<i>Preset Value</i>	"TAB"
<i>Syntax</i>	Param = app.SCPI.MMEMemory.STORE.SNP.SEParator app.SCPI.MMEMemory.STORE.SNP.SEParator = "SPACe"
<i>Equivalent Softkeys</i>	Save/Recall > Save Data to Touchstone File > Separator

SCPI.MMEMemory.STORE.SNP.TYPE.S1P

<i>Description</i>	Selects s1p save type (1-port measurements) and sets port numbers (selects S11, S22, S33 or S44 parameter) when saving the measured S-parameters by the SCPI.MMEMemory.STORE.SNP.DATA command.
<i>Type</i>	Long (read/write)
<i>Range</i>	Port number from 1 to 2 (4 for S4VNA)
<i>Out of Range</i>	An error occurs. Error code 222.
<i>Preset Value</i>	1
<i>Syntax</i>	<i>Value</i> = app.SCPI.MMEMemory.STORE.SNP.TYPE.S1P <i>app.SCPI.MMEMemory.STORE.SNP.TYPE.S1P</i> = 2
<i>Equivalent Softkeys</i>	Save/Recall > Save Data to Touchstone File > Type > 1–Port (s1p) Save/Recall > Save Data to Touchstone File > Select Port (s1p)

SCPI.MMEMemory.STORE.SNP.TYPE.S2P

<i>Description</i>	Selects s2p save type (2-port measurements) and sets port numbers when saving the measured S-parameters by the SCPI.MMEMemory.STORE.SNP.DATA command. The array contains 2 elements: <i>Data(0)</i> First port number from 1 to 2 (4 for S4VNA); <i>Data(1)</i> Second port number from 1 to 2 (4 for S4VNA).
<i>Type</i>	Variant: array of long (read/write)
<i>Syntax</i>	<i>Data</i> = app.SCPI.MMEMemory.STORE.SNP.TYPE.S2P <i>app.SCPI.MMEMemory.STORE.SNP.TYPE.S2P</i> = Array(1, 2)
<i>Equivalent Softkeys</i>	Save/Recall > Save Data to Touchstone File > Type > 2–Port (s2p) Save/Recall > Save Data to Touchstone File > Select Ports (s2p)

SCPI.MMEMemory.STORE.SNP.TYPE.S3P

<i>Description</i>	Selects s3p save type and sets port numbers when saving the measured S-parameters by the SCPI.MMEMemory.STORE.SNP.DATA command. The array contains 3 elements: <i>Data(0)</i> First port number; <i>Data(1)</i> Second port number; <i>Data(2)</i> Third port number. (S4VNA only)
<i>Type</i>	Variant: array of long (read/write)
<i>Syntax</i>	<i>Data</i> = app.SCPI.MMEMemory.STORE.SNP.TYPE.S3P <i>app.SCPI.MMEMemory.STORE.SNP.TYPE.S3P</i> = Array(1, 2, 3)
<i>Equivalent Softkeys</i>	Save/Recall > Save Data to Touchstone File > Type > 3–Port (s3p) Save/Recall > Save Data to Touchstone File > Select Ports (s3p)

SCPI.MMEMemory.STORE.SNP.TYPE.S4P

<i>Description</i>	Selects s4p save type and sets port numbers when saving the measured S-parameters by the SCPI.MMEMemory.STORE.SNP.DATA command. The array contains 3 elements: <i>Data(0)</i> First port number; <i>Data(1)</i> Second port number; <i>Data(2)</i> Third port number; <i>Data(3)</i> Forth port number. (S4VNA only)
<i>Type</i>	Variant: array of long (read/write)
<i>Syntax</i>	<i>Data</i> = app.SCPi.MMEMemory.STORE.SNP.TYPE.S4P <i>app.SCPi.MMEMemory.STORE.SNP.TYPE.S4P</i> = Array(1, 2, 3, 4)
<i>Equivalent Softkeys</i>	Save/Recall > Save Data to Touchstone File > Type > 4–Port (s4p)

SCPI.MMEMemory.STORE.STATE

<i>Description</i>	Saves the analyzer state into a file with the specified name.
<i>Type</i>	String (write only)
<i>Parameter</i>	File Name
<i>Syntax</i>	<i>app.SCPi.MMEMemory.STORE.STATE</i> = <i>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>Description</i>	Selects the type of the analyzer or channel state saving by the SCPI.MMEMory.STORe.STATe or SCPI.MMEMory.STORe.CHANnel.STATe command.
<i>Type</i>	String (read/write)
<i>Parameter</i>	"STATe": Measurement conditions "CSTate": Measurement conditions and calibration tables "DSTate": Measurement conditions and data traces "CDSTate": Measurement conditions, calibration tables and data traces "CMSTate": Measurement conditions, calibration and memory
<i>Preset Value</i>	"CST"
<i>Syntax</i>	Param = app.SCPi.MMEMory.STORe.STYPE app.SCPi.MMEMory.STORe.STYPE = "STATe"
<i>Equivalent Softkeys</i>	Save/Recall > Save Type

SCPI.OUTPut.STATe

<i>Description</i>	Turns ON/OFF the stimulus signal output. Measurements cannot be performed when the stimulus signal output is set to OFF.
<i>Type</i>	Boolean (read/write)
<i>Parameter</i>	True: Stimulus signal output ON False: Stimulus signal output OFF
<i>Preset Value</i>	True
<i>Target</i>	Analyzer
<i>Syntax</i>	Status = app.SCPi.OUTPut.STATe app.SCPi.OUTPut.STATe = False
<i>Equivalent Softkeys</i>	Stimulus > Power > RF Out

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

<i>Description</i>	Resets the averaging count to 0. Restarts the averaging process.
<i>Type</i>	Method
<i>Target</i>	Channel <i>Ch</i> from 1 to 16
<i>Syntax</i>	<i>app</i> .SCPI.SENSE(<i>Ch</i>).AVERage.CLEar
<i>Equivalent Softkeys</i>	None

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

<i>Description</i>	Sets/gets the averaging factor, when the averaging function is set to ON by the SCPI.SENSE(<i>Ch</i>).AVERage.STATe command.
<i>Type</i>	Long (read/write)
<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>Target</i>	Channel <i>Ch</i> from 1 to 16
<i>Syntax</i>	<i>Value</i> = <i>app</i> .SCPI.SENSE(<i>Ch</i>).AVERage.COUNT <i>app</i> .SCPI.SENSE(<i>Ch</i>).AVERage.COUNT = 2
<i>Equivalent Softkeys</i>	Average > Avg Factor

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

<i>Description</i>	Turns ON/OFF the state of the averaging function.
<i>Type</i>	Boolean (read/write)
<i>Parameter</i>	True: Averaging ON False: Averaging OFF
<i>Preset Value</i>	False
<i>Target</i>	Channel <i>Ch</i> from 1 to 16
<i>Syntax</i>	<i>Status</i> = app.SCPi.SENSE(<i>Ch</i>).AVERage.STATe app.SCPi.SENSE(<i>Ch</i>).AVERage.STATe = False
<i>Equivalent Softkeys</i>	Average > Averaging

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

<i>Description</i>	Sets/gets the IF bandwidth value. Less IF bandwidth reduces noise and increases measurement time.
<i>Type</i>	Double (read/write)
<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>Target</i>	Channel <i>Ch</i> from 1 to 16
<i>Syntax</i>	<i>Value</i> = app.SCPi.SENSE(<i>Ch</i>).BANDwidth.RESolution app.SCPi.SENSE(<i>Ch</i>).BANDwidth.RESolution = 100
<i>Equivalent Softkeys</i>	Average > IF Bandwidth

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

<i>Description</i>	Clears the calibration coefficient table.
<i>Type</i>	Method
<i>Target</i>	Channel <i>Ch</i> from 1 to 16
<i>Syntax</i>	app.SCPi.SENSE(<i>Ch</i>).CORRection.CLEar
<i>Equivalent Softkeys</i>	None

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

<i>Description</i>	Sets/gets the calibration coefficient data array set by the type of the corrected error term <i>Str</i>, the number of the receiver port <i>Pt_r</i> and the number of the source port <i>Pt_s</i>, <i>Str</i> : error term (see below) <i>Pt_r</i>: the number of the receiver port 1–2 (4 for S4VNA) <i>Pt_s</i>: the number of the source port 1–2 (4 for S4VNA) 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>(2<i>n</i>–2) real part of the calibration coefficients <i>Data</i>(2<i>n</i>–1) imaginary part of the calibration coefficients
<i>Type</i>	Variant: array of double (read/write)
<i>Parameter</i>	String <i>Str</i> – corrected error term: "ES": Source match "ER": Reflection tracking "ED": Directivity "EL": Load match "ET": Transmission tracking "EX": Isolation 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.
<i>Target</i>	Channel <i>Ch</i> from 1 to 16
<i>Syntax</i>	<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>
<i>Notes</i>	When sets, this command writes the calibration coefficient data array to a temporary storage. Written calibration coefficients become effective 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>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(<i>Ch</i>).CORRection.COEFficient.SAVE method. The array contains 2 elements: <i>Data</i>(0) the number of the receiver port; <i>Data</i>(1) the number of the source port.
<i>Type</i>	Variant: array of long (write only)
<i>Range</i>	Port number is 1–2 (4 for S4VNA). 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>Target</i>	Channel <i>Ch</i> from 1 to 16
<i>Syntax</i>	<i>Ports</i> = Array(2, 1) <i>app</i>.SCPI.SENSE(<i>Ch</i>).CORRection.COEFficient.METHod.ERESponse = <i>Ports</i>
<i>Equivalent Softkeys</i>	None

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

<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>Type</i>	Long (write only)
<i>Range</i>	Port number from 1 to 2 (4 for S4VNA)
<i>Out of Range</i>	An error occurs. Error code: 222.
<i>Target</i>	Channel <i>Ch</i> from 1 to 16
<i>Syntax</i>	<i>Port</i> = 1 <i>app</i> .SCPI.SENSE(<i>Ch</i>).CORRection.COEFficient.METHod.RESPOnse.OPEN = <i>Port</i>
<i>Equivalent Softkeys</i>	None

SCPI.SENSE(*Ch*).CORRection.COEFficient.METHod.RESPOnse.SHORt

<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>Type</i>	Long (write only)
<i>Range</i>	from 1 – 2 (4 for S4VNA)
<i>Out of Range</i>	An error occurs. Error code: 222.
<i>Target</i>	Channel <i>Ch</i> from 1 to 16
<i>Syntax</i>	<i>Port</i> = 1 <i>app</i> .SCPI.SENSE(<i>Ch</i>).CORRection.COEFficient.METHod.RESPOnse.SHORt = <i>Port</i>
<i>Equivalent Softkeys</i>	None

SCPI.SENSE(*Ch*).CORRection.COEFficient.METHod.RESPOnse.THru

<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(<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>Type</i>	Variant: array of long (write only)
<i>Range</i>	Port number is 1 to 2 (or 4 for S4VNA). 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>Target</i>	Channel <i>Ch</i> from 1 to 16
<i>Syntax</i>	<i>Ports</i> = Array(2, 1) <i>app</i>.SCPI.SENSE(<i>Ch</i>).CORRection.COEFficient.METHod.RESPOnse.THru = <i>Ports</i>
<i>Equivalent Softkeys</i>	None

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

<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>Type</i>	Long (write only)
<i>Range</i>	Port number from 1 to 2 (4 for S4VNA)
<i>Out of Range</i>	An error occurs. Error code: 222.
<i>Target</i>	Channel <i>Ch</i> from 1 to 16
<i>Syntax</i>	<i>Port</i> = 1 <i>app</i> .SCPI.SENSE(<i>Ch</i>).CORRection.COEFficient.METHod.SOLT1= <i>Port</i>
<i>Equivalent Softkeys</i>	None

SCPI.SENSE(*Ch*).CORRection.COEFFicient.METHod.SOLT2

<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</i>(0) specifies a port for full 2-port calibration 1-4; <i>Data</i>(1) specifies a port for full 2-port calibration 1-4.
<i>Type</i>	Variant: array of long (write only)
<i>Range</i>	Port number is 1 to 2 (or 4 for S4VNA). 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>Target</i>	Channel <i>Ch</i> from 1 to 16
<i>Syntax</i>	<i>Ports</i> = Array(1,2) <i>app</i>.SCPI.SENSE(<i>Ch</i>).CORRection.COEFFicient.METHod.SOLT2 = <i>Ports</i>
<i>Equivalent Softkeys</i>	None

SCPI.SENSE(*Ch*).CORRection.COEFficient.METHod.SOLT3

<i>Description</i>	Selects the ports and sets the <i>full 3-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 3 elements: <i>Data</i>(0) specifies a port for full 3-port calibration; <i>Data</i>(1) specifies a port for full 3-port calibration; <i>Data</i>(2) specifies a port for full 3-port calibration. (S4VNA only)
<i>Type</i>	Variant: array of long (write only)
<i>Range</i>	Port number is 1 to 4. 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>Target</i>	Channel <i>Ch</i> from 1 to 16
<i>Syntax</i>	<i>Ports</i> = Array(1,2,4) <i>app</i>.SCPI.SENSE(<i>Ch</i>).CORRection.COEFficient.METHod.SOLT2 = <i>Ports</i>
<i>Equivalent Softkeys</i>	None

SCPI.SENSE(*Ch*).CORRection.COEFficient.METHod.SOLT4

<i>Description</i>	Selects the ports and sets the <i>full 4-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 4 elements: <i>Data</i>(0) specifies a port for full 4-port calibration; <i>Data</i>(1) specifies a port for full 4-port calibration; <i>Data</i>(2) specifies a port for full 4-port calibration; <i>Data</i>(3) specifies a port for full 4-port calibration. (S4VNA only)
<i>Type</i>	Variant: array of long (write only)
<i>Range</i>	Port number is 1 to 4. 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>Target</i>	Channel <i>Ch</i> from 1 to 16
<i>Syntax</i>	<i>Ports</i> = Array(1,2,3,4) <i>app</i>.SCPI.SENSE(<i>Ch</i>).CORRection.COEFficient.METHod.SOLT2 = <i>Ports</i>
<i>Equivalent Softkeys</i>	None

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

<i>Description</i>	Enables the written calibration coefficients depending on the selected calibration type. On completion of the method the written calibration coefficients are cleared, the error correction automatically turns ON. At the attempt to execute this method before all the needed calibration coefficients are written, an error occurs and the method is ignored.
<i>Type</i>	Method
<i>Target</i>	Channel <i>Ch</i> from 1 to 16
<i>Syntax</i>	<i>app</i> .SCPI.SENSE(<i>Ch</i>).CORRection.COEFficient.SAVE
<i>Related Commands</i>	Calibration type selection: 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 SCPI.SENSE(<i>Ch</i>).CORRection.COEFficient.METHod.SOLT3 SCPI.SENSE(<i>Ch</i>).CORRection.COEFficient.METHod.SOLT4 Calibration coefficient writing: SCPI.SENSE(<i>Ch</i>).CORRection.COEFficient.DATA (<i>Str</i>, <i>Pt_r</i>, <i>Pt_s</i>)
<i>Equivalent Softkeys</i>	None

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

<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>Type</i>	Variant: array of long (write only)
<i>Range</i>	Port number is 1 to 2 (or 4 for S4VNA). 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>Target</i>	Channel <i>Ch</i> from 1 to 16
<i>Syntax</i>	<code>app.SCPI.SENSE(<i>Ch</i>).CORRection.COLlect.ACQuire.ISOLation = Array(1, 2)</code>
<i>Notes</i>	The command starts the measurement immediately if the trigger source for calibration set to the "Internal" by the command SENS:CORR:TRIG:FREE, otherwise waits for the trigger signal. The command 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 > 2–Port SOLT Cal > Port x-y Isol (Optional) Calibration > Calibrate > 3–Port SOLT Cal > Port x-y Isol (Optional) Calibration > Calibrate > 4–Port SOLT Cal > Port x-y Isol (Optional)

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

<i>Description</i>	Measures the calibration data of the <i>load</i> standard for the specified port.
<i>Type</i>	Long (write only)
<i>Range</i>	Port number is 1 to 2 (4 for S4VNA).
<i>Out of Range</i>	An error occurs (error code: 222).
<i>Target</i>	Channel <i>Ch</i> from 1 to 16
<i>Syntax</i>	<i>app</i> .SCPI.SENSE(<i>Ch</i>).CORRection.COLLection.ACQuire.LOAD = 1
<i>Notes</i>	The command starts the measurement immediately if the trigger source for calibration set to the "Internal" by the command SENS:CORR:TRIG:FREE, otherwise waits for the trigger signal. The command 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 > 1–Port SOL Cal > Load Calibration > Calibrate > One Path 2–Port Cal > Load Calibration > Calibrate > 2–Port SOLT Cal > Port n Load Calibration > Calibrate > 3–Port SOLT Cal > Reflection Port n > Load Calibration > Calibrate > 4–Port SOLT Cal > Reflection Port n > Load

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

<i>Description</i>	Measures the calibration data of the <i>open</i> standard for the specified port.
<i>Type</i>	Long (write only)
<i>Range</i>	Port number is 1 to 2 (4 for S4VNA).
<i>Out of Range</i>	An error occurs (error code: 222).
<i>Target</i>	Channel <i>Ch</i> from 1 to 16
<i>Syntax</i>	<i>app</i> .SCPI.SENSE(<i>Ch</i>).CORRection.COLLect.ACQuire.OPEN= 1
<i>Notes</i>	The command starts the measurement immediately if the trigger source for calibration set to the "Internal" by the command SENS:CORR:TRIG:FREE, otherwise waits for the trigger signal. The command waits for the completion of the measurement.
<i>Equivalent Softkeys</i>	Calibration > Calibrate > Response (Open) > Open Calibration > Calibrate > 1–Port SOL Cal > Open Calibration > Calibrate > One Path 2–Port Cal > Open Calibration > Calibrate > 2–Port SOLT Cal > Port n Open Calibration > Calibrate > 3–Port SOLT Cal > Reflection Port n > Open Calibration > Calibrate > 4–Port SOLT Cal > Reflection Port n > Open

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

<i>Description</i>	Measures the calibration data of the <i>short</i> standard for the specified port.
<i>Type</i>	Long (write only)
<i>Range</i>	Port number is 1 to 2 (4 for S4VNA).
<i>Out of Range</i>	An error occurs (error code: 222).
<i>Target</i>	Channel <i>Ch</i> from 1 to 16
<i>Syntax</i>	<code>app.SCPI.SENSE(<i>Ch</i>).CORRection.COLLection.ACQuire.SHORt = 1</code>
<i>Notes</i>	The command starts the measurement immediately if the trigger source for calibration set to the "Internal" by the command SENS:CORR:TRIG:FREE, otherwise waits for the trigger signal. The command waits for the completion of the measurement.
<i>Equivalent Softkeys</i>	Calibration > Calibrate > Response (Short) > Short Calibration > Calibrate > 1–Port SOL Cal > Short Calibration > Calibrate > One Path 2–Port Cal > Short Calibration > Calibrate > 2–Port SOLT Cal > Port n Short Calibration > Calibrate > 3–Port SOLT Cal > Reflection Port n > Short Calibration > Calibrate > 4–Port SOLT Cal > Reflection Port n > Short

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

<i>Description</i>	Sets/gets the subclass number which is used for the calibration of the selected channel (<i>Ch</i>). 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>Type</i>	Long (read/write)
<i>Range</i>	1 to 8
<i>Target</i>	Channel <i>Ch</i> from 1 to 16
<i>Syntax</i>	<i>app</i> .SCPI.SENSE(<i>Ch</i>).CORRection.COLLect.ACQuire.SUBClass = 2 <i>Subclass</i> = <i>app</i> .SCPI.SENSE(<i>Ch</i>).CORRection.COLLect.ACQuire.SUBClass
<i>Equivalent Softkeys</i>	None

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

<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>Type</i>	Variant: array of long (write only)
<i>Range</i>	Port number is 1 to 2 (or 4 for S4VNA). 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>Target</i>	Channel <i>Ch</i> from 1 to 16
<i>Syntax</i>	<i>app</i> .SCPI.SENSE(<i>Ch</i>).CORRection.COLlect.ACQuire.THRU= Array(1, 2)
<i>Notes</i>	The command starts the measurement immediately if the trigger source for calibration set to the "Internal" by the command SENS:CORR:TRIG:FREE, otherwise waits for the trigger signal. The command 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 > 2–Port SOLT Cal > Port x–y Thru Calibration > Calibrate > 3–Port SOLT Cal > x–y Thru Calibration > Calibrate > 4–Port SOLT Cal > Transmission > x–y Thru

SCPI.SENSE(*Ch*).CORRection.COLLection.ACQuire.TRLLine

<i>Description</i>	Measures the calibration data of the <i>TRL Line/Match</i> standard between the source port and the receiver port. Line standard is measured in both directions. Match standard requires two commands to measure forward and reverse directions. Match is connected to the stimulus port (second port). The array contains 2 elements: <i>Data(0)</i> the number of the receiver port; <i>Data(1)</i> the <i>number</i> of the source port.
<i>Type</i>	Variant: array of long (write only)
<i>Range</i>	Port number is 1 to 4. 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>Target</i>	Channel <i>Ch</i> from 1 to 16
<i>Syntax</i>	<i>app</i> .SCPI.SENSE(<i>Ch</i>).CORRection.COLLection.ACQuire.TRLLine= Array(1, 2)
<i>Notes</i>	The command starts the measurement immediately if the trigger source for calibration set to the "Internal" by the command SENS:CORR:TRIG:FREE, otherwise waits for the trigger signal. The command waits for the completion of the measurement.
<i>Equivalent Softkeys</i>	Calibration > Calibrate > 2–Port TRL Cal > x–y Line/Match Calibration > Calibrate > 3–Port TRL Cal > Line/Match > x–y Line/Match Calibration > Calibrate > 4–Port TRL Cal > Line/Match > x–y Line/Match

SCPI.SENSE(*Ch*).CORRection.COLLection.ACQuire.TRLReflect

<i>Description</i>	Measures the calibration data of the <i>TRL Reflect</i> standard for the specified port.
<i>Type</i>	Long (write only)
<i>Range</i>	Port number is 1 to 4.
<i>Out of Range</i>	An error occurs (error code: 222).
<i>Target</i>	Channel <i>Ch</i> from 1 to 16
<i>Syntax</i>	<code>app.SCPi.SENSE(<i>Ch</i>).CORRection.COLLection.ACQuire.TRLReflect = 1</code>
<i>Notes</i>	The command starts the measurement immediately if the trigger source for calibration set to the "Internal" by the command SENS:CORR:TRIG:FREE, otherwise waits for the trigger signal. The command waits for the completion of the measurement.
<i>Equivalent Softkeys</i>	Calibration > Calibrate > 2–Port TRL Cal > Port x Reflect Calibration > Calibrate > 3–Port TRL Cal > Reflect > Port x Calibration > Calibrate > 4–Port TRL Cal > Reflect > Port x

SCPI.SENSE(*Ch*).CORRection.COLLection.ACQuire.TRLThru

<i>Description</i>	Measures the calibration data of the <i>TRL Thru/Line</i> standard between two ports. Thru/line standard is measured in both directions. The array contains 2 elements: <i>Data(0)</i> the number of the first port; <i>Data(1)</i> the number of the second port.
<i>Type</i>	Variant: array of long (write only)
<i>Range</i>	Port number is 1 to 4. 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>Target</i>	Channel <i>Ch</i> from 1 to 16
<i>Syntax</i>	<i>app</i> .SCPI.SENSE(<i>Ch</i>).CORRection.COLLection.ACQuire.TRLThru= Array(1, 2)
<i>Notes</i>	The command starts the measurement immediately if the trigger source for calibration set to the "Internal" by the command SENS:CORR:TRIG:FREE, otherwise waits for the trigger signal. The command waits for the completion of the measurement.
<i>Equivalent Softkeys</i>	Calibration > Calibrate > 2–Port TRL Cal > x–y Thru/Line Calibration > Calibrate > 3–Port TRL Cal > Thru/Line > x–y Thru/Line Calibration > Calibrate > 4–Port TRL Cal > Thru/Line > x–y Thru/Line

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

<i>Description</i>	Sets/gets the calibration kit label.
<i>Type</i>	String (read/write)
<i>Parameter</i>	Label
<i>Target</i>	Calibration kit, selected for channel <i>Ch</i> , <i>Ch</i> : channel number 1–16
<i>Syntax</i>	<i>Lab</i> = app.SCPI.SENSE(<i>Ch</i>).CORRection.COLLect.CKIT.LABel <i>app.SCPI.SENSE(<i>Ch</i>).CORRection.COLLect.CKIT.LABel</i> = "User1"
<i>Equivalent Softkeys</i>	Calibration > Cal Kit > Label

SCPI.SENSE(*Ch*).CORRection.COLLect.CKIT.ORDER.LOAD(*Pt*)

<i>Description</i>	Sets/gets 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 (4 for S4VNA)
<i>Type</i>	Long (read/write)
<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>Target</i>	Calibration kit, selected for channel <i>Ch</i> , <i>Ch</i> : channel number 1–16
<i>Syntax</i>	<i>Num</i> = app.SCPI.SENSE(<i>Ch</i>).CORRection.COLLect.CKIT.ORDER.LOAD(<i>Pt</i>) <i>app.SCPI.SENSE(<i>Ch</i>).CORRection.COLLect.CKIT.ORDER.LOAD(<i>Pt</i>)</i> = 1
<i>Equivalent Softkeys</i>	Calibration > Cal Kit > Specify CLSs > Load Port n (Row)

SCPI.SENSE(*Ch*).CORRection.COLlect.CKIT.ORDER.OPEN(*Pt*)

<i>Description</i>	Sets/gets 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 (4 for S4VNA)
<i>Type</i>	Long (read/write)
<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>Target</i>	Calibration kit, selected for channel <i>Ch</i> , <i>Ch</i> : channel number 1–16
<i>Syntax</i>	<i>Num</i> = app.SCPI.SENSE(<i>Ch</i>).CORRection.COLlect.CKIT.ORDER.OPEN(<i>Pt</i>) app.SCPI.SENSE(<i>Ch</i>).CORRection.COLlect.CKIT.ORDER.OPEN(<i>Pt</i>) = 1
<i>Equivalent Softkeys</i>	Calibration > Cal Kit > Specify CLSs > Open Port x (Row)

SCPI.SENSE(*Ch*).CORRection.COLlect.CKIT.ORDER.SELECT

<i>Description</i>	Sets/gets the subclass used to specify classes of calibration standards by the commands SCPI.SENSE(<i>Ch</i>).CORRection.COLlect.CKIT.ORDER.XXXX.
<i>Type</i>	Long (read/write)
<i>Range</i>	1 to 8.
<i>Target</i>	Calibration kit, selected for channel <i>Ch</i> , <i>Ch</i> : channel number 1–16
<i>Syntax</i>	<i>Num</i> = app.SCPI.SENSE(<i>Ch</i>).CORRection.COLlect.CKIT.ORDER.SELECT app.SCPI.SENSE(<i>Ch</i>).CORRection.COLlect.CKIT.ORDER.SELECT = 1
<i>Equivalent Softkeys</i>	Calibration > Cal Kit > Specify CLSs > Subclass n (Column)

SCPI.SENSE(*Ch*).CORRection.COLLection.CKIT.ORDer.SHORt(*Pt*)

<i>Description</i>	Sets/gets 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 (4 for S4VNA)
<i>Type</i>	Long (read/write)
<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>Target</i>	Calibration kit, selected for channel <i>Ch</i> , <i>Ch</i> : channel number 1–16
<i>Syntax</i>	<i>Num</i> = app.SCPI.SENSE(<i>Ch</i>).CORRection.COLLection.CKIT.ORDer.SHORt(<i>Pt</i>) app.SCPI.SENSE(<i>Ch</i>).CORRection.COLLection.CKIT.ORDer.SHORt(<i>Pt</i>) = 1
<i>Equivalent Softkeys</i>	Calibration > Cal Kit > Specify CLSs > Short Port x (Row)

SCPI.SENSE(*Ch*).CORRection.COLLect.CKIT.ORDER.THURU(*Pt_m*, *Pt_n*)

<i>Description</i>	Sets/gets 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 (4 for S4VNA) <i>Pt_n</i> : port number 1–2 (4 for S4VNA)
<i>Type</i>	Long (read/write)
<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>Target</i>	Calibration kit, selected for channel <i>Ch</i> , <i>Ch</i> : channel number 1–16
<i>Syntax</i>	<i>Num</i> = app.SCPi.SENSE(<i>Ch</i>).CORRection.COLLect.CKIT.ORDER.THURU(1, 2) app.SCPi.SENSE(<i>Ch</i>).CORRection.COLLect.CKIT.ORDER.THURU(1, 2) = 1
<i>Equivalent Softkeys</i>	Calibration > Cal Kit > Specify CLSs > Thru Port x-y (Row)

SCPI.SENSE(*Ch*).CORRection.COLLect.CKIT.ORDER.TRLLine(*Pt_m*, *Pt_n*)

<i>Description</i>	Sets/gets the number of the calibration standard of the TRL Line/Match 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 (4 for S4VNA) <i>Pt_n</i> : port number 1–2 (4 for S4VNA)
<i>Type</i>	Long (read/write)
<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>Target</i>	Calibration kit, selected for channel <i>Ch</i> , <i>Ch</i> : channel number 1–16
<i>Syntax</i>	<i>Num</i> = app.SCPi.SENSE(<i>Ch</i>).CORRection.COLLect.CKIT.ORDER.TRLLine(1, 2) app.SCPi.SENSE(<i>Ch</i>).CORRection.COLLect.CKIT.ORDER.TRLLine(1, 2) = 1
<i>Equivalent Softkeys</i>	Calibration > Cal Kit > Specify CLSs > TRL Line/Match Port x-y (Row)

SCPI.SENSE(*Ch*).CORRection.COLLection.CKIT.ORDER.TRLReflect(*Pt*)

<i>Description</i>	Sets/gets the number of the calibration standard of the TRL Reflect type, used for the measurement of the specified port <i>Pt</i> , <i>Pt</i> : port number 1–2 (4 for S4VNA)
<i>Type</i>	Long (read/write)
<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>Target</i>	Calibration kit, selected for channel <i>Ch</i> , <i>Ch</i> : channel number 1–16
<i>Syntax</i>	<i>Num</i> = app.SCPI.SENSE(<i>Ch</i>).CORRection.COLLection.CKIT.ORDER.TRLReflect(<i>Pt</i>) app.SCPI.SENSE(<i>Ch</i>).CORRection.COLLection.CKIT.ORDER.TRLReflect(<i>Pt</i>) = 1
<i>Equivalent Softkeys</i>	Calibration > Cal Kit > Specify CLSs > TRL Reflect Port x (Row)

SCPI.SENSE(*Ch*).CORRection.COLLect.CKIT.ORDER.TRLThru(*Pt_m*, *Pt_n*)

<i>Description</i>	Sets/gets the number of the calibration standard of the TRL 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 (4 for S4VNA) <i>Pt_n</i> : port number 1–2 (4 for S4VNA)
<i>Type</i>	Long (read/write)
<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>Target</i>	Calibration kit, selected for channel <i>Ch</i> , <i>Ch</i> : channel number 1–16
<i>Syntax</i>	<i>Num</i> = app.SCPi.SENSE(<i>Ch</i>).CORRection.COLLect.CKIT.ORDER.TRLThru(1, 2) app.SCPi.SENSE(<i>Ch</i>).CORRection.COLLect.CKIT.ORDER.TRLThru(1, 2) = 1
<i>Equivalent Softkeys</i>	Calibration > Cal Kit > Specify CLSs > TRL Thru Port x-y (Row)

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

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

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

<i>Description</i>	Sets/gets the selected calibration kit for the channel.
<i>Type</i>	Long (read/write)
<i>Range</i>	from 1 to 50
<i>Out of Range</i>	An error occurs. Error code: 222.
<i>Preset Value</i>	1
<i>Target</i>	Channel <i>Ch</i> from 1 to 16
<i>Syntax</i>	<i>Value</i> = app.SCPi.SENSE(<i>Ch</i>).CORRection.COLLect.CKIT.SELect app.SCPi.SENSE(<i>Ch</i>).CORRection.COLLect.CKIT.SELect = 3
<i>Equivalent Softkeys</i>	Calibration > Cal Kit > Select n

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

<i>Description</i>	Sets/gets the value of the arbitrary impedance for the load standard.
<i>Type</i>	Double (read/write)
<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>Target</i>	Standard <i>Std</i> of the calibration kit specified for channel <i>Ch</i> , <i>Ch</i> : channel number 1–16 <i>Std</i> : calibration standard number 1-30
<i>Syntax</i>	<i>Value</i> = app.SCPI.SENSE(<i>Ch</i>).CORRection.COLlect.CKIT.STAN(<i>Std</i>).ARBitrary app.SCPI.SENSE(<i>Ch</i>).CORRection.COLlect.CKIT.STAN(<i>Std</i>).ARBitrary = 50
<i>Equivalent Softkeys</i>	Calibration > Cal Kit > Define STDs > Terminal Impedance

SCPI.SENSE(*Ch*).CORRection.COLlect.CKIT.STAN(*Std*).C0

<i>Description</i>	Sets/gets the C0 value of the open calibration standard.
<i>Type</i>	Double (read/write)
<i>Range</i>	from $-1\text{E}18$ to $1\text{E}18$
<i>Out of Range</i>	Sets the value of the limit, which is closer to the specified value.
<i>Unit</i>	$1\text{E}-15$ F (Farad)
<i>Target</i>	Standard <i>Std</i> of the calibration kit specified for channel <i>Ch</i> , <i>Ch</i> : channel number 1–16 <i>Std</i> : standard number 1–30
<i>Syntax</i>	<i>Value</i> = app.SCPI.SENSE(<i>Ch</i>).CORRection.COLlect.CKIT.STAN(<i>Std</i>).C0 app.SCPI.SENSE(<i>Ch</i>).CORRection.COLlect.CKIT.STAN(<i>Std</i>).C0 = 100
<i>Equivalent Softkeys</i>	Calibration > Cal Kit > Define STDs > C0 10^{-15} F

SCPI.SENSE(*Ch*).CORRection.COLlect.CKIT.STAN(*Std*).C1

<i>Description</i>	Sets/gets the C1 value of the open calibration standard.
<i>Type</i>	Double (read/write)
<i>Range</i>	from $-1\text{E}18$ to $1\text{E}18$
<i>Out of Range</i>	Sets the value of the limit, which is closer to the specified value.
<i>Unit</i>	$1\text{E}-27$ F/Hz (Farad/Hertz)
<i>Target</i>	Standard <i>Std</i> of the calibration kit specified for channel <i>Ch</i> , <i>Ch</i> : channel number 1–16 <i>Std</i> : standard number 1–30
<i>Syntax</i>	<i>Value</i> = 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
<i>Equivalent Softkeys</i>	Calibration > Cal Kit > Define STDs > C1 10^{-27} F/Hz

SCPI.SENSE(*Ch*).CORRection.COLlect.CKIT.STAN(*Std*).C2

<i>Description</i>	Sets/gets the C2 value of the open calibration standard.
<i>Type</i>	Double (read/write)
<i>Range</i>	from $-1\text{E}18$ to $1\text{E}18$
<i>Out of Range</i>	Sets the value of the limit, which is closer to the specified value.
<i>Unit</i>	$1\text{E}-36 \text{ F/Hz}^2$ (Farad/Hertz ²)
<i>Target</i>	Standard <i>Std</i> of the calibration kit specified for channel <i>Ch</i> , <i>Ch</i> : channel number 1–16 <i>Std</i> : standard number 1-30
<i>Syntax</i>	<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
<i>Equivalent Softkeys</i>	Calibration > Cal Kit > Define STDs > C2 10^{-36} F/Hz^2

SCPI.SENSE(*Ch*).CORRection.COLlect.CKIT.STAN(*Std*).C3

<i>Description</i>	Sets/gets the C3 value of the open calibration standard.
<i>Type</i>	Double (read/write)
<i>Range</i>	from $-1\text{E}18$ to $1\text{E}18$
<i>Out of Range</i>	Sets the value of the limit, which is closer to the specified value.
<i>Unit</i>	$1\text{E}-45 \text{ F/Hz}^3$ (Farad/Hertz ³)
<i>Target</i>	Standard <i>Std</i> of the calibration kit specified for channel <i>Ch</i> , <i>Ch</i> : channel number 1–16 <i>Std</i> : standard number 1-30
<i>Syntax</i>	<i>Value</i> = app.SCPI.SENSE(<i>Ch</i>).CORRection.COLlect.CKIT.STAN(<i>Std</i>).C3 app.SCPI.SENSE(<i>Ch</i>).CORRection.COLlect.CKIT.STAN(<i>Std</i>).C3 = 100
<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>Description</i>	Sets/gets the offset delay value of the calibration standard.
<i>Type</i>	Double (read/write)
<i>Range</i>	from $-1\text{E}18$ to $1\text{E}18$
<i>Out of Range</i>	Sets the value of the limit, which is closer to the specified value.
<i>Unit</i>	s (second)
<i>Target</i>	Standard <i>Std</i> of the calibration kit specified for channel <i>Ch</i> , <i>Ch</i> : channel number 1–16 <i>Std</i> : standard number 1-30
<i>Syntax</i>	<i>Value</i> = app.SCPI.SENSE(<i>Ch</i>).CORRection.COLlect.CKIT.STAN(<i>Std</i>).DELay app.SCPI.SENSE(<i>Ch</i>).CORRection.COLlect.CKIT.STAN(<i>Std</i>).DELay = $93\text{E}-12$
<i>Equivalent Softkeys</i>	Calibration > Cal Kit > Define STDs > Offset Delay

SCPI.SENSE(*Ch*).CORRection.COLLection.CKIT.STAN(*Std*).FMAXimum

<i>Description</i>	Sets/gets the maximum frequency value of the calibration standard.
<i>Type</i>	Double (read/write)
<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>Target</i>	Standard <i>Std</i> of the calibration kit specified for channel <i>Ch</i> , <i>Ch</i> : channel number 1–16 <i>Std</i> : standard number 1-30
<i>Syntax</i>	<i>Value</i> = app.SCPI.SENSE(<i>Ch</i>).CORRection.COLLection.CKIT.STAN(<i>Std</i>).FMAXimum app.SCPI.SENSE(<i>Ch</i>).CORRection.COLLection.CKIT.STAN(<i>Std</i>).FMAXimum = 3E9
<i>Equivalent Softkeys</i>	Calibration > Cal Kit > Define STDs > Frequency max

SCPI.SENSE(*Ch*).CORRection.COLLection.CKIT.STAN(*Std*).FMINimum

<i>Description</i>	Sets/gets the minimum frequency value of the calibration standard.
<i>Type</i>	Double (read/write)
<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>Target</i>	Standard <i>Std</i> of the calibration kit specified for channel <i>Ch</i> , <i>Ch</i> : channel number 1–16 <i>Std</i> : standard number 1-30
<i>Syntax</i>	<i>Value</i> = app.SCPI.SENSE(<i>Ch</i>).CORRection.COLLection.CKIT.STAN(<i>Std</i>).FMINimum app.SCPI.SENSE(<i>Ch</i>).CORRection.COLLection.CKIT.STAN(<i>Std</i>).FMINimum = 3E9
<i>Equivalent Softkeys</i>	Calibration > Cal Kit > Define STDs > Frequency min

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

<i>Description</i>	Sets/gets the L0 value of the short calibration standard.
<i>Type</i>	Double (read/write)
<i>Range</i>	from $-1\text{E}18$ to $1\text{E}18$
<i>Out of Range</i>	Sets the value of the limit, which is closer to the specified value.
<i>Unit</i>	$1\text{E}-12$ H (Henry)
<i>Target</i>	Standard <i>Std</i> of the calibration kit specified for channel <i>Ch</i> , <i>Ch</i> : channel number 1–16 <i>Std</i> : standard number 1–30
<i>Syntax</i>	<i>Value</i> = app.SCPI.SENSE(<i>Ch</i>).CORRection.COLlect.CKIT.STAN(<i>Std</i>).L0 app.SCPI.SENSE(<i>Ch</i>).CORRection.COLlect.CKIT.STAN(<i>Std</i>).L0 = 100
<i>Equivalent Softkeys</i>	Calibration > Cal Kit > Define STDs > L0 10^{-12} H

SCPI.SENSE(*Ch*).CORRection.COLlect.CKIT.STAN(*Std*).L1

<i>Description</i>	Sets/gets the L1 value of the short calibration standard.
<i>Type</i>	Double (read/write)
<i>Range</i>	from $-1\text{E}18$ to $1\text{E}18$
<i>Out of Range</i>	Sets the value of the limit, which is closer to the specified value.
<i>Unit</i>	$1\text{E}-24$ H/Hz (Henry/Hertz)
<i>Target</i>	Standard <i>Std</i> of the calibration kit specified for channel <i>Ch</i> , <i>Ch</i> : channel number 1–16 <i>Std</i> : standard number 1–30
<i>Syntax</i>	<i>Value</i> = app.SCPI.SENSE(<i>Ch</i>).CORRection.COLlect.CKIT.STAN(<i>Std</i>).L1 app.SCPI.SENSE(<i>Ch</i>).CORRection.COLlect.CKIT.STAN(<i>Std</i>).L1 = 100
<i>Equivalent Softkeys</i>	Calibration > Cal Kit > Define STDs > L1 10^{-24} H/Hz

SCPI.SENSE(*Ch*).CORRection.COLlect.CKIT.STAN(*Std*).L2

<i>Description</i>	Sets/gets the L2 value of the short calibration standard.
<i>Type</i>	Double (read/write)
<i>Range</i>	from $-1\text{E}18$ to $1\text{E}18$
<i>Out of Range</i>	Sets the value of the limit, which is closer to the specified value.
<i>Unit</i>	$1\text{E}-33$ H/Hz ² (Henry/Hertz ²)
<i>Target</i>	Standard <i>Std</i> of the calibration kit specified for channel <i>Ch</i> , <i>Ch</i> : channel number 1–16 <i>Std</i> : standard number
<i>Syntax</i>	<i>Value</i> = app.SCPI.SENSE(<i>Ch</i>).CORRection.COLlect.CKIT.STAN(<i>Std</i>).L2 app.SCPI.SENSE(<i>Ch</i>).CORRection.COLlect.CKIT.STAN(<i>Std</i>).L2 = 100
<i>Equivalent Softkeys</i>	Calibration > Cal Kit > Define STDs > L2 10^{-33} H/Hz²

SCPI.SENSE(*Ch*).CORRection.COLlect.CKIT.STAN(*Std*).L3

<i>Description</i>	Sets/gets the L3 value of the short calibration standard.
<i>Type</i>	Double (read/write)
<i>Range</i>	from $-1\text{E}18$ to $1\text{E}18$
<i>Out of Range</i>	Sets the value of the limit, which is closer to the specified value.
<i>Unit</i>	$1\text{E}-42 \text{ H/Hz}^3$ (Henry/Hertz ³)
<i>Target</i>	Standard <i>Std</i> of the calibration kit specified for channel <i>Ch</i> , <i>Ch</i> : channel number 1–16 <i>Std</i> : standard number 1-30
<i>Syntax</i>	<i>Value</i> = app.SCPI.SENSE(<i>Ch</i>).CORRection.COLlect.CKIT.STAN(<i>Std</i>).L3 app.SCPI.SENSE(<i>Ch</i>).CORRection.COLlect.CKIT.STAN(<i>Std</i>).L3 = 100
<i>Equivalent Softkeys</i>	Calibration > Cal Kit > Define STDs > L3 10^{-42} H/Hz^3

SCPI.SENSE(*Ch*).CORRection.COLlect.CKIT.STAN(*Std*).LABel

<i>Description</i>	Sets/gets the label of the calibration standard.
<i>Type</i>	String (read/write)
<i>Parameter</i>	Label
<i>Target</i>	Standard <i>Std</i> of the calibration kit specified for channel <i>Ch</i> , <i>Ch</i> : channel number 1–16 <i>Std</i> : standard number 1-30
<i>Syntax</i>	<i>Lab</i> = app.SCPI.SENSE(<i>Ch</i>).CORRection.COLlect.CKIT.STAN(<i>Std</i>).LABel app.SCPI.SENSE(<i>Ch</i>).CORRection.COLlect.CKIT.STAN(<i>Std</i>).LABel = "Open"
<i>Equivalent Softkeys</i>	Calibration > Cal Kit > Define STDs > Standard Label

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

<i>Description</i>	Sets/gets the offset loss value of the calibration standard.
<i>Type</i>	Double (read/write)
<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>Unit</i>	Ω/s (Ohm/second)
<i>Target</i>	Standard <i>Std</i> of the calibration kit specified for channel <i>Ch</i> , <i>Ch</i> : channel number 1–16 <i>Std</i> : standard number 1-30
<i>Syntax</i>	<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
<i>Equivalent Softkeys</i>	Calibration > Cal Kit > Define STDs > Offset Loss

SCPI.SENSE(*Ch*).CORRection.COLLection.CKIT.STAN(*Std*).TYPE

<i>Description</i>	Sets/gets the type of the calibration standard.
<i>Type</i>	String (read/write)
<i>Parameter</i>	"OPEN" : Open "SHORT" : Short "LOAD" : Load "THRU" : Thru "UTHRu" : Unknown Thru "SLID" : Sliding load "DATA" : Data based standard "NONE" : Not defined
<i>Target</i>	Standard <i>Std</i> of the calibration kit specified for channel <i>Ch</i> , <i>Ch</i> : channel number 1–16 <i>Std</i> : standard number 1-30
<i>Syntax</i>	<i>Param</i> = app.SCPI.SENSE(<i>Ch</i>).CORRection.COLLection.CKIT.STAN(<i>Std</i>).TYPE app.SCPI.SENSE(<i>Ch</i>).CORRection.COLLection.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>Description</i>	Sets/gets the offset Z0 value of the calibration standard.
<i>Type</i>	Double (read/write)
<i>Range</i>	from 0 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>Target</i>	Standard <i>Std</i> of the calibration kit specified for channel <i>Ch</i> , <i>Ch</i> : channel number 1–16 <i>Std</i> : standard number 1-30
<i>Syntax</i>	<i>Value</i> = app.SCPI.SENSE(<i>Ch</i>).CORRection.COLlect.CKIT.STAN(<i>Std</i>).Z0 app.SCPI.SENSE(<i>Ch</i>).CORRection.COLlect.CKIT.STAN(<i>Std</i>).Z0 = 50
<i>Equivalent Softkeys</i>	Calibration > Cal Kit > Define STDs > Offset Z0

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

<i>Description</i>	Clears the measurement values of the calibration standards.
<i>Type</i>	Method
<i>Target</i>	Channel <i>Ch</i> from 1 to 16
<i>Syntax</i>	<i>app</i> .SCPI.SENSE(<i>Ch</i>).CORRection.COLLect.CLEar
<i>Equivalent Softkeys</i>	Calibration > Calibrate > Response (Open) Response (Short) Response (Thru) One Path 2–Port Cal 1–Port SOL Cal 2–Port SOLT Cal 3–Port SOLT Cal 4–Port SOLT Cal 2–Port TRL Cal 3–Port TRL Cal 4–Port TRL Cal > Cancel > OK

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

<i>Description</i>	Sets/gets the array of the isolation calibration measurements performed between the receiver port <i>Pt_r</i> and the source port <i>Pt_s</i>, <i>Pt_r</i>: the number of the receiver port 1–2 (4 for S4VNA) <i>Pt_s</i>: the number of the source port 1–2 (4 for S4VNA) The array elements are complex numbers. Also see section 14. 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 part of the measurement; <i>Data</i>(2n–1) imaginary part of the measurement.
<i>Type</i>	Variant: array of double (read/write)
<i>Target</i>	Channel <i>Ch</i> from 1 to 16
<i>Syntax</i>	<i>Data</i> = app.SCPi.SENSE(<i>Ch</i>).CORRection.COLlect.DATA.ISOLation(<i>Pt_r</i>, <i>Pt_s</i>) app.SCPi.SENSE(<i>Ch</i>).CORRection.COLlect.DATA.ISOLation(<i>Pt_r</i>, <i>Pt_s</i>) = <i>Data</i>
<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>Description</i>	Sets/gets the array of the <i>load</i> calibration standard measurements for the port <i>Pt</i>, <i>Pt</i>: port number 1–2 (4 for S4VNA) The array elements are complex numbers. Also see section 14. 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 part of the measurement; <i>Data</i>(2n–1) imaginary part of the measurement.
<i>Type</i>	Variant: array of double (read/write)
<i>Target</i>	Channel <i>Ch</i> from 1 to 16
<i>Syntax</i>	<i>Data</i> = app.SCPI.SENSE(<i>Ch</i>).CORRection.COLLect.DATA.LOAD(<i>Pt</i>) app.SCPI.SENSE(<i>Ch</i>).CORRection.COLLect.DATA.LOAD(<i>Pt</i>) = <i>Data</i>
<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>Description</i>	Sets/gets the array of the <i>open</i> calibration standard measurements for the port <i>Pt</i>, <i>Pt</i>: port number 1–2 (4 for S4VNA) The array elements are complex numbers. Also see section 14. 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>(2<i>n</i>–2) real part of the measurement; <i>Data</i>(2<i>n</i>–1) imaginary part of the measurement.
<i>Type</i>	Variant: array of double (read/write)
<i>Target</i>	Channel <i>Ch</i> from 1 to 16
<i>Syntax</i>	<i>Data</i> = app.SCPI.SENSE(<i>Ch</i>).CORRection.COLLect.DATA.OPEN(<i>Pt</i>) app.SCPI.SENSE(<i>Ch</i>).CORRection.COLLect.DATA.OPEN(<i>Pt</i>) = <i>Data</i>
<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>Description</i>	Sets/gets the array of the <i>short</i> calibration standard measurements for the port <i>Pt</i>, <i>Pt</i>: port number 1–2 (4 for S4VNA) The array elements are complex numbers. Also see section 14. 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 part of the measurement; <i>Data</i>(2n–1) imaginary part of the measurement.
<i>Type</i>	Variant: array of double (read/write)
<i>Target</i>	Channel <i>Ch</i> from 1 to 16
<i>Syntax</i>	<i>Data</i> = app.SCPI.SENSE(<i>Ch</i>).CORRection.COLlect.DATA.SHORt(<i>Pt</i>) app.SCPI.SENSE(<i>Ch</i>).CORRection.COLlect.DATA.SHORt(<i>Pt</i>) = <i>Data</i>
<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>Description</i>	Sets/gets 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>, <i>Pt_r</i>: the number of the receiver port 1–2 (4 for S4VNA) <i>Pt_s</i>: the number of the source port 1–2 (4 for S4VNA) The array elements are complex numbers. Also see section 14. 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 part of the measurement; <i>Data</i>(2n–1) imaginary part of the measurement.
<i>Type</i>	Variant: array of double (read/write)
<i>Target</i>	Channel <i>Ch</i> from 1 to 16
<i>Syntax</i>	<i>Data</i> = app.SCPi.SENSE(<i>Ch</i>).CORRection.COLlect.DATA.THRU.MATCh(<i>Pt_r</i>, <i>Pt_s</i>) app.SCPi.SENSE(<i>Ch</i>).CORRection.COLlect.DATA.THRU.MATCh(<i>Pt_r</i>, <i>Pt_s</i>) = <i>Data</i>
<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>Description</i>	Sets/gets 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, <i>Pt_r</i>: the number of the receiver port 1–2 (4 for S4VNA) <i>Pt_s</i>: the number of the source port 1–2 (4 for S4VNA) The array elements are complex numbers. Also see section 14. 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 part of the measurement; <i>Data</i>(2n–1) imaginary part of the measurement.
<i>Type</i>	Variant: array of double (read/write)
<i>Target</i>	Channel <i>Ch</i> from 1 to 16
<i>Syntax</i>	<i>Data</i> = app.SCPI.SENSE(<i>Ch</i>).CORRection.COLLect.DATA.THru.TRANsmission(<i>Pt_r</i>, <i>Pt_s</i>) app.SCPI.SENSE(<i>Ch</i>).CORRection.COLLect.DATA.THru.TRANsmission(<i>Pt_r</i>, <i>Pt_s</i>) = <i>Data</i>
<i>Related Commands</i>	SCPI.SENSE(<i>Ch</i>).CORRection.COLLect.ACQuire.THru
<i>Equivalent Softkeys</i>	None

SCPI.SENSE(*Ch*).CORRection.COLlect.ECAL.CCHeck.ACQuire

<i>Description</i>	Executes the confidence check of the calibration coefficients of specified channel using the AutoCal module. The command sets the AutoCal Module to the special internal state, reads the S-parameters of this state from the AutoCal Module and sets memory traces so that they can be compared with actual measured data. Comparison is carried out visually by the user.
<i>Type</i>	Method
<i>Target</i>	Channel <i>Ch</i> from 1 to 16
<i>Syntax</i>	<code>app.SCPi.SENSE(<i>Ch</i>).CORRection.COLlect.ECAL.CCHeck.ACQuire</code>
<i>Equivalent Softkeys</i>	Calibration > AutoCal > Confidence Check

SCPI.SENSE(*Ch*).CORRection.COLlect.ECAL.ERESponse

<i>Description</i>	Executes one path 2-port calibration between the specified 2 ports of selected channel (<i>Ch</i>) using the AutoCal module.
<i>Type</i>	Variant: array of long (write only)
<i>Target</i>	Channel <i>Ch</i> from 1 to 16
<i>Syntax</i>	<code>app.SCPi.SENSE(<i>Ch</i>).CORRection.COLlect.ECAL.ERESponse = Array(2, 1)</code>
<i>Equivalent Softkeys</i>	Calibration > AutoCal > One Path 2-Port Cal > Port x-y

SCPI.SENSE(1).CORRection.COLLect.ECAL.INFOrmation

<i>Description</i>	Gets information about the AutoCal Module connected to the Network Analyzer in a string with comma separated fields. Autocal Module Information: - Model Name, - Serial Number, - Current Temperature of AutoCal Module, Selected Characterization Information: - 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>Type</i>	String (read only)
<i>Target</i>	AutoCal module
<i>Syntax</i>	ID = app.SCPI.SENSE(1).CORRection.COLLect.ECAL.INFOrmation
<i>Equivalent Softkeys</i>	Calibration > AutoCal > Characterization Info...

SCPI.SENSE(1).CORRection.COLLect.ECAL.ORIentation.STATe

<i>Description</i>	Turns ON/OFF the state of the Auto-Orientation function used when executing AutoCal.
<i>Type</i>	Boolean (read/write)
<i>Parameter</i>	True: Auto-Orientation function ON False: Auto-Orientation function OFF
<i>Preset Value</i>	False
<i>Target</i>	AutoCal
<i>Syntax</i>	<i>Status</i> = app.SCPI.SENSE(1).CORRection.COLLect.ECAL.ORIentation.STATe <i>app.SCPI.SENSE(1).CORRection.COLLect.ECAL.ORIentation.STATe</i> = False
<i>Equivalent Softkeys</i>	Calibration > AutoCal > Orientation > Auto-Orientation

SCPI.SENSE(1).CORRection.COLLect.ECAL.PATH(*Pt*)

<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>Type</i>	Long (read/write)
<i>Parameter</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>Target</i>	AutoCal
<i>Syntax</i>	<i>Value</i> = app.SCPI.SENSE(1).CORRection.COLLect.ECAL.PATH(<i>Pt</i>) <i>app.SCPI.SENSE(1).CORRection.COLLect.ECAL.PATH(<i>Pt</i>)</i> = 2
<i>Equivalent Softkeys</i>	Calibration > AutoCal > Orientation > Port x

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

<i>Description</i>	Executes 1-port calibration of the specified port of selected channel (<i>Ch</i>) using the AutoCal module.
<i>Type</i>	Long (write only)
<i>Target</i>	Channel <i>Ch</i> from 1 to 16
<i>Syntax</i>	<i>app</i> .SCPI.SENSE(<i>Ch</i>).CORRection.COLlect.ECAL.SOLT1 = <i>Port</i>
<i>Equivalent Softkeys</i>	Calibration > AutoCal > 1-Port Cal > Port x

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

<i>Description</i>	Executes full 2-port calibration between the specified 2 ports of selected channel (<i>Ch</i>) using the AutoCal module.
<i>Type</i>	Variant: array of long (write only)
<i>Target</i>	Channel <i>Ch</i> from 1 to 16
<i>Syntax</i>	<i>app</i> .SCPI.SENSE(<i>Ch</i>).CORRection.COLlect.ECAL.SOLT2 = Array(2, 1)
<i>Equivalent Softkeys</i>	Calibration > AutoCal > 2-Port Cal > Port x-y

SCPI.SENSE(*Ch*).CORRection.COLLect.ECAL.SOLT3

<i>Description</i>	Executes full 3-port calibration between the specified 3 ports of selected channel (<i>Ch</i>) using the AutoCal module (4-port AutoCal module only).
<i>Type</i>	Variant: array of long (write only)
<i>Target</i>	Channel <i>Ch</i> from 1 to 16
<i>Syntax</i>	<code>app.SCPI.SENSE(<i>Ch</i>).CORRection.COLLect.ECAL.SOLT3 = Array(1, 2, 3)</code>
<i>Equivalent Softkeys</i>	Calibration > AutoCal > 3-Port Cal > Select Ports [x-y-z] Calibration > AutoCal > 3-Port Cal > 4-Port Autocal Module

SCPI.SENSE(*Ch*).CORRection.COLLect.ECAL.SOLT4

<i>Description</i>	Executes full 4-port calibration between the specified 4 ports of selected channel (<i>Ch</i>) using the AutoCal module (4-port AutoCal module only).
<i>Type</i>	Variant: array of long (write only)
<i>Target</i>	Channel <i>Ch</i> from 1 to 16
<i>Syntax</i>	<code>app.SCPI.SENSE(<i>Ch</i>).CORRection.COLLect.ECAL.SOLT4 = Array(1, 2, 3, 4)</code>
<i>Equivalent Softkeys</i>	Calibration > AutoCal > 3-Port Cal > 4-Port Autocal Module

SCPI.SENSE(1).CORRection.COLLect.ECAL.UCHar

<i>Description</i>	Sets or reads out the Characteristic number used when executing AutoCal (factory or user characterization).
<i>Type</i>	String (read/write)
<i>Parameter</i>	"CHAR0" : Factory characterization "CHAR1" : User characterization 1 "CHAR2" : User characterization 2 "CHAR3" : User characterization 3
<i>Preset Value</i>	CHAR0
<i>Target</i>	AutoCal
<i>Syntax</i>	Param = app.SCPI.SENSE(1).CORRection.COLLect.ECAL.UCHar app.SCPI.SENSE(1).CORRection.COLLect.ECAL.UCHar = "CHAR0"
<i>Equivalent Softkeys</i>	Calibration > AutoCal > Characterization

SCPI.SENSE(1).CORRection.COLlect.ECAL.UTHRu.STATe

<i>Description</i>	Turns ON/OFF the state of the Unknown Thru feature used when executing AutoCal.
<i>Type</i>	Boolean (read/write)
<i>Parameter</i>	True: Unknown Thru feature ON False: Unknown Thru feature OFF
<i>Preset Value</i>	False
<i>Target</i>	AutoCal
<i>Syntax</i>	<i>Status</i> = app.SCPi.SENSE(1).CORRection.COLlect.ECAL.UTHRu.STATe <i>app.SCPi.SENSE(1).CORRection.COLlect.ECAL.UTHRu.STATe</i> = 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>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</i>(0) the number of the receiver port; <i>Data</i>(1) the number of the source port.
<i>Type</i>	Variant: array of long (write only)
<i>Range</i>	Port number is 1 to 2 (4 for S4VNA). 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>Target</i>	Channel <i>Ch</i> from 1 to 16
<i>Syntax</i>	<i>app</i> .SCPI.SENSE(<i>Ch</i>).CORRection.COLlect.METHod.ERESponse = Array(2, 1)
<i>Equivalent Softkeys</i>	Calibration > Calibrate > One Path 2-Port Cal > Select Ports

SCPI.SENSE(*Ch*).CORRection.COLlect.METHod.RESPOnse.OPEN

<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>Type</i>	Long (write only)
<i>Range</i>	Port number from 1 to 2 (4 for S4VNA)
<i>Out of Range</i>	An error occurs. Error code: 222.
<i>Target</i>	Channel <i>Ch</i> from 1 to 16
<i>Syntax</i>	<code>app.SCPI.SENSE(<i>Ch</i>).CORRection.COLlect.METHod.RESPOnse.OPEN = 1</code>
<i>Equivalent Softkeys</i>	Calibration > Calibrate > Response (Open) > Select Port

SCPI.SENSE(*Ch*).CORRection.COLlect.METHod.RESPOnse.SHORt

<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>Type</i>	Long (write only)
<i>Range</i>	Port number from 1 to 2 (4 for S4VNA)
<i>Out of Range</i>	An error occurs. Error code: 222.
<i>Target</i>	Channel <i>Ch</i> from 1 to 16
<i>Syntax</i>	<code>app.SCPI.SENSE(<i>Ch</i>).CORRection.COLlect.METHod.RESPOnse.HORt = 1</code>
<i>Equivalent Softkeys</i>	Calibration > Calibrate > Response (Short) > Select Port

SCPI.SENSE(*Ch*).CORRection.COLlect.METHod.RESPOnse.THRU

<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</i>(0) the number of the receiver port; <i>Data</i>(1) the number of the source port.
<i>Type</i>	Variant: array of long (write only)
<i>Range</i>	Port number is 1 to 2 (4 for S4VNA). 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>Target</i>	Channel <i>Ch</i> from 1 to 16
<i>Syntax</i>	<i>app</i> .SCPI.SENSE(<i>Ch</i>).CORRection.COLlect.METHod.RESPOnse.THRU = Array(2, 1)
<i>Equivalent Softkeys</i>	Calibration > Calibrate > Response (Thru) > Select Ports

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

<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>Type</i>	Long (write only)
<i>Range</i>	Port number from 1 to 2 (4 for S4VNA)
<i>Out of Range</i>	An error occurs. Error code: 222.
<i>Target</i>	Channel <i>Ch</i> from 1 to 16
<i>Syntax</i>	<i>app</i> .SCPI.SENSE(<i>Ch</i>).CORRection.COLLect.METHod.SOLT1 = 1
<i>Equivalent Softkeys</i>	Calibration > Calibrate > 1–Port SOL Cal > Select Port

SCPI.SENSE(*Ch*).CORRection.COLlect.METHod.SOLT2

<i>Description</i>	Selects the port and sets the <i>2-port SOLT 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</i>(0) specifies a port for full 2-port calibration 1-4; <i>Data</i>(1) specifies a port for full 2-port calibration 1-4.
<i>Type</i>	Variant: array of long (write only)
<i>Range</i>	Port number is 1 to 2 (4 for S4VNA). 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>Target</i>	Channel <i>Ch</i> from 1 to 16
<i>Syntax</i>	<i>app</i> .SCPI.SENSE(<i>Ch</i>).CORRection.COLlect.METHod.SOLT2 = Array(2, 1)
<i>Equivalent Softkeys</i>	Calibration > Calibrate > 2-Port SOLT Cal > Select Ports

SCPI.SENSE(*Ch*).CORRection.COLlect.METHod.SOLT3

<i>Description</i>	Selects the port and sets the <i>3-port SOLT 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 3 elements: <i>Data</i>(0) specifies a port for full 3-port calibration; <i>Data</i>(1) specifies a port for full 3-port calibration; <i>Data</i>(2) specifies a port for full 3-port calibration. (S4VNA only)
<i>Type</i>	Variant: array of long (write only)
<i>Range</i>	Port number is 1 to 4. 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>Target</i>	Channel <i>Ch</i> from 1 to 16
<i>Syntax</i>	<i>app</i> .SCPI.SENSE(<i>Ch</i>).CORRection.COLlect.METHod.SOLT3 = Array(1, 2, 4)
<i>Equivalent Softkeys</i>	Calibration > Calibrate > 3-Port SOLT Cal > Select Ports

SCPI.SENSE(*Ch*).CORRection.COLlect.METHod.SOLT4

<i>Description</i>	Selects the port and sets the <i>4-port SOLT 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 4 elements: <i>Data</i>(0) specifies a port for full 4-port calibration; <i>Data</i>(1) specifies a port for full 4-port calibration; <i>Data</i>(2) specifies a port for full 4-port calibration; <i>Data</i>(3) specifies a port for full 4-port calibration. (S4VNA only)
<i>Type</i>	Variant: array of long (write only)
<i>Range</i>	Port number is 1 to 4. 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>Target</i>	Channel <i>Ch</i> from 1 to 16
<i>Syntax</i>	<i>app</i> .SCPI.SENSE(<i>Ch</i>).CORRection.COLlect.METHod.SOLT4 = Array(1, 2, 3, 4)
<i>Equivalent Softkeys</i>	Calibration > Calibrate > 4-Port SOLT Cal

SCPI.SENSE(*Ch*).CORRection.COLLection.METHod.TRL2

<i>Description</i>	Selects the port and sets the <i>2-port TRL calibration</i> type for the calculation of the calibration coefficients on completion of the calibration executed by the SCPI.SENSE(<i>Ch</i>).CORRection.COLLection.SAVE method. The array contains 2 elements: <i>Data</i>(0) specifies a port for 2-port TRL calibration; <i>Data</i>(1) specifies a port for 2-port TRL calibration.
<i>Type</i>	Variant: array of long (write only)
<i>Range</i>	Port number is 1 to 4. 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>Target</i>	Channel <i>Ch</i> from 1 to 16
<i>Syntax</i>	<code>app.SCPI.SENSE(<i>Ch</i>).CORRection.COLLection.METHod.TRL2 = Array(1, 2)</code>
<i>Equivalent Softkeys</i>	Calibration > Calibrate > 2-Port TRL Cal > Select Ports

SCPI.SENSE(*Ch*).CORRection.COLlect.METHod.TRL3

<i>Description</i>	Selects the port and sets the <i>3-port TRL 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 3 elements: <i>Data</i>(0) specifies a port for 3-port TRL calibration; <i>Data</i>(1) specifies a port for 3-port TRL calibration; <i>Data</i>(2) specifies a port for 3-port TRL calibration. (S4VNA only)
<i>Type</i>	Variant: array of long (write only)
<i>Range</i>	Port number is 1 to 4. 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>Target</i>	Channel <i>Ch</i> from 1 to 16
<i>Syntax</i>	<i>app</i> .SCPI.SENSE(<i>Ch</i>).CORRection.COLlect.METHod.TRL3 = Array(1, 2, 3)
<i>Equivalent Softkeys</i>	Calibration > Calibrate > 3-Port TRL Cal > Select Ports

SCPI.SENSE(*Ch*).CORRection.COLlect.METHod.TRL4

<i>Description</i>	Selects the port and sets the <i>4-port TRL 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 4 elements: <i>Data</i>(0) specifies a port for 4-port TRL calibration; <i>Data</i>(1) specifies a port for 4-port TRL calibration; <i>Data</i>(2) specifies a port for 4-port TRL calibration; <i>Data</i>(3) specifies a port for 4-port TRL calibration. (S4VNA only)
<i>Type</i>	Variant: array of long (write only)
<i>Range</i>	Port number is 1 to 4. 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>Target</i>	Channel <i>Ch</i> from 1 to 16
<i>Syntax</i>	<i>app</i> .SCPI.SENSE(<i>Ch</i>).CORRection.COLlect.METHod.TRL4 = Array(1, 2, 3, 4)
<i>Equivalent Softkeys</i>	Calibration > Calibrate > 4-Port TRL Cal

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

<i>Description</i>	Reads out 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>Type</i>	String (read only)
<i>Parameter</i>	"RESPO" : Response (Open) "RESPS" : Response (Short) "RESPT" : Response (Thru) "1PATH" : One path 2–port calibration "SOLT1" : Full 1–port calibration "SOLT2" : Full 2–port calibration "SOLT3" : Full 3–port calibration "SOLT4" : Full 4–port calibration "TRL2" : 2–port TRL calibration "TRL3" : 3–port TRL calibration "TRL4" : 4–port TRL calibration "NONE" : Not defined
<i>Target</i>	Channel <i>Ch</i> from 1 to 16
<i>Syntax</i>	<i>Param</i> = app.SCPI.SENSE(<i>Ch</i>).CORRection.COLlect.METHod.TYPE
<i>Equivalent Softkeys</i>	None

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

<i>Description</i>	Calculates the calibration coefficients from the calibration standards measurements depending on the selected calibration type. On completion of the method, all the calibration standards measurements are cleared and the error correction automatically turns ON. At the attempt to execute this method before all the needed standards are measured, an error occurs and the method is ignored. Before executing this command it is necessary to select calibration type by one of commands SCPI.SENSE(<i>Ch</i>).CORRection.COLlect.METHod.XXXX.
<i>Type</i>	Method
<i>Target</i>	Channel <i>Ch</i> from 1 to 16
<i>Syntax</i>	<i>app</i> .SCPI.SENSE(<i>Ch</i>).CORRection.COLlect.SAVE
<i>Related Commands</i>	Calibration type selection: 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.THURU 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 SCPI.SENSE(<i>Ch</i>).CORRection.COLlect.METHod.SOLT3 SCPI.SENSE(<i>Ch</i>).CORRection.COLlect.METHod.SOLT4 SCPI.SENSE(<i>Ch</i>).CORRection.COLlect.METHod.TRL2 SCPI.SENSE(<i>Ch</i>).CORRection.COLlect.METHod.TRL3 SCPI.SENSE(<i>Ch</i>).CORRection.COLlect.METHod.TRL4 Calibration standards measurement: 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.THURU SCPI.SENSE(<i>Ch</i>).CORRection.COLlect.ACQuire.TRLLine SCPI.SENSE(<i>Ch</i>).CORRection.COLlect.ACQuire.TRLReflect SCPI.SENSE(<i>Ch</i>).CORRection.COLlect.ACQuire.TRLThru
<i>Equivalent Softkeys</i>	Calibration > Calibrate > Response (Open) Response (Short) Response (Thru) One Path 2–Port Cal 1–Port SOL Cal 2–Port SOLT Cal 3–Port SOLT Cal 4–Port SOLT Cal 2–Port TRL Cal 3–Port TRL Cal 4–Port TRL Cal > Apply

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

<i>Description</i>	Same as SCPI.SENSE(<i>Ch</i>).CORRection.COLLect.SAVE method. Allows to accomplish 3 and 4 port calibration with non complete set of standard measurements. About omissible calibration standards see user guide chapter "Simplified calibration".
<i>Type</i>	Method
<i>Target</i>	Channel <i>Ch</i> from 1 to 16
<i>Syntax</i>	<i>app</i> .SCPI.SENSE(<i>Ch</i>).CORRection.COLLect.SIMPLified.SAVE
<i>Related Commands</i>	Calibration type selection: SCPI.SENSE(<i>Ch</i>).CORRection.COLLect.METHod.SOLT3 SCPI.SENSE(<i>Ch</i>).CORRection.COLLect.METHod.SOLT4 SCPI.SENSE(<i>Ch</i>).CORRection.COLLect.METHod.TRL3 SCPI.SENSE(<i>Ch</i>).CORRection.COLLect.METHod.TRL4 Calibration standards measurement: SCPI.SENSE(<i>Ch</i>).CORRection.COLLect.ACQuire.ISOLation SCPI.SENSE(<i>Ch</i>).CORRection.COLLect.ACQuire.THURU SCPI.SENSE(<i>Ch</i>).CORRection.COLLect.ACQuire.TRLLine
<i>Equivalent Softkeys</i>	Calibration > Calibrate > 3–Port SOLT Cal 4–Port SOLT Cal 3–Port TRL Cal 4–Port TRL Cal > Apply

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

<i>Description</i>	Sets/gets 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
<i>Type</i>	Double (read/write)
<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>Target</i>	Port <i>Pt</i> of channel <i>Ch</i> , <i>Ch</i> : channel number 1–16 <i>Pt</i> : port number 1–2 (4 for S4VNA)
<i>Syntax</i>	<i>Value</i> = app.SCPI.SENSE(<i>Ch</i>).CORRection.EXTension.PORT(<i>Pt</i>).FREQuency(<i>Ls</i>) app.SCPI.SENSE(<i>Ch</i>).CORRection.EXTension.PORT(<i>Pt</i>).FREQuency(<i>Ls</i>) = 100E6
<i>Equivalent Softkeys</i>	Calibration > Port Extensions > Loss Port n > Freq1 Freq2

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

<i>Description</i>	Turns ON/OFF the 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
<i>Type</i>	Boolean (read/write)
<i>Parameter</i>	True: Loss compensation ON False: Loss compensation OFF
<i>Preset Value</i>	False
<i>Target</i>	Port <i>Pt</i> of channel <i>Ch</i> , <i>Ch</i> : channel number 1–16 <i>Pt</i> : port number 1–2 (4 for S4VNA)
<i>Syntax</i>	<i>Status</i> = app.SCPI.SENSE(<i>Ch</i>).CORRection.EXTension.PORT(<i>Pt</i>).INCLude(<i>Ls</i>).STATe app.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 Port n > Loss1 Loss2

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

<i>Description</i>	Sets/gets the loss value at DC for the loss calculation of the port extension function.
<i>Type</i>	Double (read/write)
<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>Target</i>	Port <i>Pt</i> of channel <i>Ch</i> , <i>Ch</i> : channel number 1–16 <i>Pt</i> : port number 1–2 (4 for S4VNA)
<i>Syntax</i>	<i>Value</i> = app.SCPI.SENSE(<i>Ch</i>).CORRection.EXTension.PORT(<i>Pt</i>).LDC app.SCPI.SENSE(<i>Ch</i>).CORRection.EXTension.PORT(<i>Pt</i>).LDC = 10
<i>Equivalent Softkeys</i>	Calibration > Port Extensions > Loss Port n > Loss at DC

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

<i>Description</i>	Sets/gets 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
<i>Type</i>	Double (read/write)
<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>Target</i>	Port <i>Pt</i> of channel <i>Ch</i> , <i>Ch</i> : channel number 1–16 <i>Pt</i> : port number 1–2 (4 for S4VNA)
<i>Syntax</i>	<i>Value</i> = app.SCPi.SENSE(<i>Ch</i>).CORRection.EXTension.PORT(<i>Pt</i>).LOSS(<i>Ls</i>) app.SCPi.SENSE(<i>Ch</i>).CORRection.EXTension.PORT(<i>Pt</i>).LOSS(<i>Ls</i>) = 10
<i>Equivalent Softkeys</i>	Calibration > Port Extensions > Loss Port n > Loss1 Loss2

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

<i>Description</i>	Sets/gets the electrical delay value for the port extension function.
<i>Type</i>	Double (read/write)
<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>Target</i>	Port <i>Pt</i> of channel <i>Ch</i> , <i>Ch</i> : channel number 1–16 <i>Pt</i> : port number 1–2 (4 for S4VNA)
<i>Syntax</i>	<i>Value</i> = app.SCPi.SENSE(<i>Ch</i>).CORRection.EXTension.PORT(<i>Pt</i>).TIME app.SCPi.SENSE(<i>Ch</i>).CORRection.EXTension.PORT(<i>Pt</i>).TIME = 10E-9
<i>Equivalent Softkeys</i>	Calibration > Port Extensions > Extension Port n

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

<i>Description</i>	Turns ON/OFF the state of the port extension function.
<i>Type</i>	Boolean (read/write)
<i>Parameter</i>	True: Port extension function ON False: Port extension function OFF
<i>Preset Value</i>	False
<i>Target</i>	Channel <i>Ch</i> from 1 to 16
<i>Syntax</i>	<i>Status</i> = app.SCPi.SENSE(<i>Ch</i>).CORRection.EXTension.STATe app.SCPi.SENSE(<i>Ch</i>).CORRection.EXTension.STATe = True
<i>Equivalent Softkeys</i>	Calibration > Port Extensions > Extensions

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

<i>Description</i>	Reads out the calibration information string for port pair.
<i>Type</i>	String (read only)
<i>Target</i>	Channel <i>Ch</i> from 1 to 16
<i>Syntax</i>	<i>CalInfo</i> = app.SCPI.SENSE(<i>Ch</i>).CORRection.INFoRmation(<i>rPt</i> , <i>sPt</i>)
<i>Equivalent Softkeys</i>	None

SCPI.SENSE(1).CORRection.IMPedance.INPut.MAGNitude

<i>Description</i>	Sets/gets the system characteristic impedance (<i>Z0</i>) value.
<i>Type</i>	Double (read/write)
<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>	<i>Value</i> = app.SCPI.SENSE.CORRection.IMPedance.INPut.MAGNitude <i>app.SCPI.SENSE.CORRection.IMPedance.INPut.MAGNitude</i> = 50
<i>Warning</i>	Object <i>SENSe</i> has an index of 1, which can be omitted in Visual Basic but it cannot be omitted in other programming languages.
<i>Equivalent Softkeys</i>	Calibration > System Z0

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

<i>Description</i>	Clears the scalar mixer calibration coefficient table.
<i>Type</i>	Method
<i>Target</i>	Channel <i>Ch</i> from 1 to 16
<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>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: <i>Port</i>(0) Measurement port number; <i>Port</i>(1) Frequency port number.
<i>Type</i>	Variant: array of long (write only)
<i>Range</i>	Port number is 1 to 2 (4 for S4VNA). 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>Target</i>	Channel <i>Ch</i> from 1 to 16
<i>Syntax</i>	<i>app</i> .SCPI.SENSE(<i>Ch</i>).CORRection.OFFSet.COLlect.ACQuire.LOAD = Array(1, 2)
<i>Notes</i>	The command starts the measurement immediately if the trigger source for calibration set to the "Internal" by the command SENS:CORR:TRIG:FREE, otherwise waits for the trigger signal. The command 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>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: <i>Port(0)</i> Measurement port number; <i>Port(1)</i> Frequency port number.
<i>Type</i>	Variant: array of long (write only)
<i>Range</i>	Port number is 1 to 2 (4 for S4VNA). 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>Target</i>	Channel <i>Ch</i> from 1 to 16
<i>Syntax</i>	<code>app.SCPi.SENSE(<i>Ch</i>).CORRection.OFFSet.COLLect.ACQuire.OPEN = Array(1, 2)</code>
<i>Notes</i>	The command starts the measurement immediately if the trigger source for calibration set to the "Internal" by the command SENS:CORR:TRIG:FREE, otherwise waits for the trigger signal. The command 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>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: <i>Port(0)</i> Measurement port number; <i>Port(1)</i> Frequency port number.
<i>Type</i>	Variant: array of long (write only)
<i>Range</i>	Port number is 1 to 2 (4 for S4VNA). 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>Target</i>	Channel <i>Ch</i> from 1 to 16
<i>Syntax</i>	<i>app</i> .SCPI.SENSE(<i>Ch</i>).CORRection.OFFSet.COLLect.ACQuire.SHORt = Array(1, 2)
<i>Notes</i>	The command starts the measurement immediately if the trigger source for calibration set to the "Internal" by the command SENS:CORR:TRIG:FREE, otherwise waits for the trigger signal. The command 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>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: <i>Port(0)</i> Response port number; <i>Port(1)</i> Stimulus port number.
<i>Type</i>	Variant: array of long (write only)
<i>Range</i>	Port number is 1 to 2 (4 for S4VNA). 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>Target</i>	Channel <i>Ch</i> from 1 to 16
<i>Syntax</i>	<i>app</i> .SCPI.SENSE(<i>Ch</i>).CORRection.OFFSet.COLLect.ACQuire.THRU = Array(1, 2)
<i>Notes</i>	The command starts the measurement immediately if the trigger source for calibration set to the "Internal" by the command SENS:CORR:TRIG:FREE, otherwise waits for the trigger signal. The command 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>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>Type</i>	Variant: array of long (write only)
<i>Range</i>	Port number is 1 to 2 (4 for S4VNA). 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>Target</i>	Channel <i>Ch</i> from 1 to 16
<i>Syntax</i>	<i>app</i> .SCPI.SENSE(<i>Ch</i>).CORRection.OFFSet.COLLect.ACQuire.PMETer = Array(1, 2, 0)
<i>Notes</i>	The command starts the measurement for the channel independently of the trigger initiation and trigger source settings. The command 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>Description</i>	Clears the calibration measurement data when the frequency offset feature is ON (scalar mixer calibration).
<i>Type</i>	Method
<i>Target</i>	Channel <i>Ch</i> from 1 to 16
<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>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</i>(0) Port number 1; <i>Port</i>(1) Port number 2.
<i>Type</i>	Variant: array of long (write only)
<i>Range</i>	Port number is 1 to 2 (4 for S4VNA). 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>Target</i>	Channel <i>Ch</i> from 1 to 16
<i>Syntax</i>	<i>app</i> .SCPI.SENSE(<i>Ch</i>).CORRection.OFFSet.COLLect.METHod.SMIX2 = Array(2, 1)
<i>Equivalent Softkeys</i>	Calibration > Mixer/Converter Calibration > Scalar Mixer Calibration

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

<i>Description</i>	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. If this command is executed before all necessary calibration data for calculating the calibration coefficient is measured, an error occurs when executed.
<i>Type</i>	Method
<i>Target</i>	Channel <i>Ch</i> from 1 to 16
<i>Syntax</i>	<i>app</i> .SCPI.SENSE(<i>Ch</i>).CORRection.OFFSet.COLlect.SAVE
<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.THURU 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>Description</i>	The port number <i>Src</i> writing executes the receiver calibration (T – channel) for the specified port <i>Pt</i> , using the specified source port <i>Src</i> to the right of the equal sign. Then executes the receiver calibration (R – channel) for the specified port <i>Pt</i> , using port <i>Pt</i> as source (the specified source port <i>Src</i> is ignored).
<i>Type</i>	Long (write only)
<i>Range</i>	The number of the source port from 1 to 2 (4 for S4VNA).
<i>Out of Range</i>	Error occurs. Error code: 222.
<i>Target</i>	Port <i>Pt</i> of channel <i>Ch</i> , <i>Ch</i> : channel number 1–16 <i>Pt</i> : port number 1–2 (4 for S4VNA)
<i>Syntax</i>	<i>app</i> .SCPI.SENSE(<i>Ch</i>).CORRection.RECeiver(<i>Pt</i>).COLLect.ACQuire = <i>Src</i>
<i>Notes</i>	The command starts the measurement for the channel independently of the trigger initiation and trigger source settings. The command waits for the completion of the measurement.
<i>Equivalent Softkeys</i>	Calibration > Receiver Calibration > Calibrate Both

SCPI.SENSE(*Ch*).CORRection.RECeiver(*Pt*).COLLect.RCHannel.ACQuire

<i>Description</i>	Method executes the receiver calibration (R – channel) for the specified port <i>Pt</i> , using port <i>Pt</i> as source.
<i>Type</i>	Method
<i>Target</i>	Port <i>Pt</i> of channel <i>Ch</i> , <i>Ch</i> : channel number 1–16 <i>Pt</i> : port number 1–2 (4 for S4VNA)
<i>Syntax</i>	<i>app</i> .SCPI.SENSE(<i>Ch</i>).CORRection.RECeiver(<i>Pt</i>).COLLect.ACQuire
<i>Notes</i>	The command starts the measurement for the channel independently of the trigger initiation and trigger source settings. The command waits for the completion of the measurement.
<i>Equivalent Softkeys</i>	Calibration > Receiver Calibration > Calibrate Reference Receiver

SCPI.SENSE(*Ch*).CORRection.RECeiver(*Pt*).COLLect.TCHannel.ACQuire

<i>Description</i>	The port number <i>Src</i> writing executes the receiver calibration (T – channel) for the specified port <i>Pt</i> , using the specified source port <i>Src</i> to the right of the equal sign.
<i>Type</i>	Long (write only)
<i>Range</i>	The number of the source port from 1 to 4.
<i>Out of Range</i>	Error occurs. Error code: 222.
<i>Target</i>	Port <i>Pt</i> of channel <i>Ch</i> , <i>Ch</i> : channel number 1–16 <i>Pt</i> : port number 1–2 (4 for S4VNA)
<i>Syntax</i>	<i>app</i> .SCPI.SENSE(<i>Ch</i>).CORRection.RECeiver(<i>Pt</i>).COLLect.ACQuire = <i>Src</i>
<i>Notes</i>	The command starts the measurement for the channel independently of the trigger initiation and trigger source settings. The command waits for the completion of the measurement.
<i>Equivalent Softkeys</i>	Calibration > Receiver Calibration > Calibrate Test Receiver

SCPI.SENSE(*Ch*).CORRection.RECeiver(*Pt*).OFFSet.AMPLitude

<i>Description</i>	Sets/gets the power offset value for Receiver Calibration. Receiver calibration is done at the condition of <source power> + <Specified offset>, for selected channel (<i>Ch</i>).
<i>Type</i>	Double (read/write)
<i>Range</i>	-200 to 200.
<i>Unit</i>	dB (decibel)
<i>Preset Value</i>	0
<i>Target</i>	Port <i>Pt</i> of channel <i>Ch</i> , <i>Ch</i> : channel number 1–16 <i>Pt</i> : port number 1–2 (4 for S4VNA)
<i>Syntax</i>	<i>Offset</i> = app.SCPi.SENSE(<i>Ch</i>).CORRection.RECeiver(<i>Pt</i>).OFFSet.AMPLitude app.SCPi.SENSE(<i>Ch</i>).CORRection.RECeiver(<i>Pt</i>).OFFSet.AMPLitude = -10
<i>Equivalent Softkeys</i>	Calibration > Receiver Calibration > Power Offset

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

<i>Description</i>	Turns ON/OFF the state of the receiver correction for the port <i>Pt</i> .
<i>Type</i>	Boolean (read/write)
<i>Parameter</i>	True: Receiver correction ON False: Receiver correction OFF
<i>Preset Value</i>	False
<i>Target</i>	Port <i>Pt</i> of channel <i>Ch</i> , <i>Ch</i> : channel number 1–16 <i>Pt</i> : port number 1–2 (4 for S4VNA)
<i>Syntax</i>	<i>Status</i> = app.SCPi.SENSE(<i>Ch</i>).CORRection.RECeiver(<i>Pt</i>).STATe app.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>Description</i>	Turns ON/OFF the state of the error correction (user calibration of S-parameters).
<i>Type</i>	Boolean (read/write)
<i>Parameter</i>	True: Error correction ON False: Error correction OFF
<i>Preset Value</i>	False
<i>Target</i>	Channel <i>Ch</i> from 1 to 16
<i>Syntax</i>	<i>Status</i> = app.SCPi.SENSE(<i>Ch</i>).CORRection.STATe app.SCPi.SENSE(<i>Ch</i>).CORRection.STATe = True
<i>Equivalent Softkeys</i>	Calibration > Correction

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

<i>Description</i>	Reads out the information about the applied calibration type and the port numbers for the specified trace. The array contains 5 elements: <i>Data</i>(0) calibration type (see below); <i>Data</i>(1) the port number to which the calibration is applied; <i>Data</i>(2) the port number to which the calibration is applied. <i>Data</i>(3) the port number to which the calibration is applied. <i>Data</i>(4) the port number to which the calibration is applied.
<i>Type</i>	Variant: array of Variants (read only)
<i>Parameter</i>	Calibration type in the element <i>Data</i>(0): "RESPO" : Response (Open) "RESPS" : Response (Short) "RESPT" : Response (Thru) "SOLT1" : Full 1–port calibration "SOLT2" : Full 2–port calibration "SOLT3" : Full 3–port calibration "SOLT4" : Full 4–port calibration "1PATH" : One path 2–port calibration "NONE" : Not defined
<i>Target</i>	Trace <i>Tr</i> of channel <i>Ch</i>, <i>Ch</i>: channel number 1–16 <i>Tr</i>: trace number 1–16
<i>Syntax</i>	<i>CalInfo</i> = app.SCPi.SENSE(<i>Ch</i>).CORRection.TYPE(<i>Tr</i>)
<i>Equivalent Softkeys</i>	None

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

<i>Description</i>	Sets/gets the frequency at which the cable loss specified for the cable correction function, when the time domain transformation function is turned ON.
<i>Type</i>	Double (read/write)
<i>Preset Value</i>	1 GHz.
<i>Unit</i>	Hz (Hertz)
<i>Target</i>	Channel <i>Ch</i> from 1 to 16
<i>Syntax</i>	<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>Description</i>	Sets/gets the cable loss value for the cable correction function, when the time domain transformation function is turned ON.
<i>Type</i>	Double (read/write)
<i>Preset Value</i>	0
<i>Unit</i>	dB/m (decibell / meter)
<i>Target</i>	Channel <i>Ch</i> from 1 to 16
<i>Syntax</i>	<i>Value</i> = <i>app</i>.SCPI.SENSE(<i>Ch</i>).CORRection.TRANSform.TIME.LOSS <i>app</i>.SCPI.SENSE(<i>Ch</i>).CORRection.TRANSform.TIME.LOSS = 1.4
<i>Equivalent Softkeys</i>	Analysis > Time Domain > Cable Correction > Cable Loss

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

<i>Description</i>	Sets/gets the cable relative wave speed velocity for the cable correction function, when the time domain transformation function is turned ON.
<i>Type</i>	Double (read/write)
<i>Preset Value</i>	1
<i>Target</i>	Channel <i>Ch</i> from 1 to 16
<i>Syntax</i>	<i>Value</i> = <i>app</i> .SCPI.SENSE(<i>Ch</i>).CORRection.TRANSform.TIME.RVELOCITY <i>app</i> .SCPI.SENSE(<i>Ch</i>).CORRection.TRANSform.TIME.RVELOCITY = 0.66
<i>Equivalent Softkeys</i>	Analysis > Time Domain > Cable Correction > Velocity Factor

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

<i>Description</i>	Turns ON/OFF the state of the cable correction when the time domain transformation function is turned ON.
<i>Type</i>	Boolean (read/write)
<i>Parameter</i>	True: cable correction ON False: cable correction OFF
<i>Preset Value</i>	False
<i>Target</i>	Channel <i>Ch</i> from 1 to 16
<i>Syntax</i>	<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*).CORRection.TRIGger.FREE.STATe

<i>Description</i>	Enables/disables the <i>internal</i> trigger source for calibration. If the <i>internal</i> trigger source for calibration is enabled then a command of the calibration standard measurement starts the measurement immediately. If the internal trigger source for calibration is disabled then the <i>system</i> trigger source is used (which is set for regular measurement with the command SCPI.TRIGger.SEQuence.SOURce) to start the calibration standard measurement. The <i>system</i> trigger source also enables the averaging trigger function (SCPI.TRIGger.SEQuence.AVERage) and the point trigger function (SCPI.TRIGger.SEQuence.POINt) for calibration. Note: When the <i>system</i> trigger source is selected you should avoid the program trigger source (BUS), otherwise the program deadlock is possible. Note: The command does not apply to the electronic calibration, the power calibration and the receiver calibration. The internal trigger always used in these cases.
<i>Type</i>	Boolean (read/write)
<i>Parameter</i>	True: Internal False: System
<i>Preset Value</i>	True
<i>Target</i>	Channel <i>Ch</i> from 1 to 16
<i>Syntax</i>	<i>Status</i> = app.SCPI.SENSE(<i>Ch</i>).CORRection.TRIGger.FREE.STATe app.SCPI.SENSE(<i>Ch</i>).CORRection.TRIGger.FREE.STATe = True
<i>Equivalent Softkeys</i>	Calibration > Cal Trig Source { Internal System }

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

<i>Description</i>	Reads out the corrected S-parameters array. The corrected measurements are complex numbers. Also, see section 14. The array size is $2N$, where N is the number of measurement points. For the n-th point, where n from 1 to N: $Data(2n-2)$ the real part of corrected measurement; $Data(2n-1)$ the imaginary part of corrected measurement. The index of the array starts from 0.
<i>Type</i>	Variant: array of double (read only)
<i>Parameter</i>	<i>Param</i> of String type selects S-parameter: "S11", "S12", "S13", "S14", "S21", "S22", "S23", "S24", "S31", "S32", "S33", "S34", "S41", "S42", "S43", "S44"
<i>Target</i>	Channel <i>Ch</i> from 1 to 16
<i>Syntax</i>	<i>Data</i> = app.SCPi.SENSE(<i>ch</i>).DATA.CORRdata("S11")
<i>Equivalent Softkeys</i>	None

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

<i>Description</i>	Reads out the raw S-parameters array (without correction). The raw measurements are complex numbers. Also, see section 14. The array size is $2N$, where N is the number of measurement points. For the n-th point, where n from 1 to N: $Data(2n-2)$ the real part of corrected measurement; $Data(2n-1)$ the imaginary part of corrected measurement. The index of the array starts from 0.
<i>Type</i>	Variant: array of double (read only)
<i>Parameter</i>	<i>Param</i> of String type selects S-parameter: "S11", "S12", "S13", "S14", "S21", "S22", "S23", "S24", "S31", "S32", "S33", "S34", "S41", "S42", "S43", "S44"
<i>Target</i>	Channel <i>Ch</i> from 1 to 16
<i>Syntax</i>	<i>Data</i> = app.SCPi.SENSE(<i>ch</i>).DATA.RAWData("S11")
<i>Equivalent Softkeys</i>	None

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

<i>Description</i>	Sets/gets the stimulus center value of the sweep range for linear or logarithmic sweep types.
<i>Type</i>	Double (read/write)
<i>Range</i>	from minimum to maximum frequency depends on model.
<i>Out of Range</i>	Sets the value of the limit, which is closer to the specified value.
<i>Preset Value</i>	Center of frequency range depends on model.
<i>Unit</i>	Hz (Hertz)
<i>Target</i>	Channel <i>Ch</i> from 1 to 16
<i>Syntax</i>	<i>Value</i> = app.SCPI.SENSE(<i>Ch</i>).FREQuency.CENTer app.SCPI.SENSE(<i>Ch</i>).FREQuency.CENTer = 1E9
<i>Equivalent Softkeys</i>	Stimulus > Center

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

<i>Description</i>	Sets/gets the value of the fixed frequency for the power sweep.
<i>Type</i>	Double (read/write)
<i>Range</i>	From minimum to maximum frequency depends on model.
<i>Out of Range</i>	Sets the value of the limit, which is closer to the specified value.
<i>Preset Value</i>	Frequency minimum.
<i>Unit</i>	Hz (Hertz)
<i>Target</i>	Channel <i>Ch</i> from 1 to 16
<i>Syntax</i>	$Value = app.SCPi.SENSE(Ch).FREQUENCY.CW$ $app.SCPi.SENSE(Ch).FREQUENCY.CW = 1E9$
<i>Equivalent Softkeys</i>	Stimulus > Power > CW Freq

SCPI.SENSE(*Ch*).FREQUENCY.DATA

<i>Description</i>	Reads out the frequency array of the measurement points for linear, logarithmic or segment sweep type. 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>Type</i>	Variant: array of double (read only)
<i>Unit</i>	Hz (Hertz)
<i>Target</i>	Channel <i>Ch</i> from 1 to 16
<i>Syntax</i>	$Data = app.SCPi.SENSE(Ch).FREQUENCY.DATA$
<i>Equivalent Softkeys</i>	None

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

<i>Description</i>	Sets/gets the stimulus span value of the sweep range for linear or logarithmic sweep types.
<i>Type</i>	Double (read/write)
<i>Range</i>	from minimum to maximum frequency depends on model.
<i>Out of Range</i>	Sets the value of the limit, which is closer to the specified value.
<i>Preset Value</i>	Full frequency span depending on model.
<i>Unit</i>	Hz (Hertz)
<i>Target</i>	Channel <i>Ch</i> from 1 to 16
<i>Syntax</i>	<i>Value</i> = app.SCPI.SENSE(<i>Ch</i>).FREQuency.SPAN app.SCPI.SENSE(<i>Ch</i>).FREQuency.SPAN = 2E9
<i>Equivalent Softkeys</i>	Stimulus > Span

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

<i>Description</i>	Sets/gets the stimulus start value of the sweep range for linear or logarithmic sweep types.
<i>Type</i>	Double (read/write)
<i>Range</i>	from minimum to maximum frequency depends on model.
<i>Out of Range</i>	Sets the value of the limit, which is closer to the specified value.
<i>Preset Value</i>	Frequency minimum.
<i>Unit</i>	Hz (Hertz)
<i>Target</i>	Channel <i>Ch</i> from 1 to 16
<i>Syntax</i>	<i>Value</i> = app.SCPI.SENSE(<i>Ch</i>).FREQuency.STARt app.SCPI.SENSE(<i>Ch</i>).FREQuency.STARt = 1E6
<i>Equivalent Softkeys</i>	Stimulus > Start

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

<i>Description</i>	Sets/gets the stimulus stop value of the sweep range for linear or logarithmic sweep types.
<i>Type</i>	Double (read/write)
<i>Range</i>	from minimum to maximum frequency depends on model.
<i>Out of Range</i>	Sets the value of the limit, which is closer to the specified value.
<i>Preset Value</i>	Frequency maximum.
<i>Unit</i>	Hz (Hertz)
<i>Target</i>	Channel <i>Ch</i> from 1 to 16
<i>Syntax</i>	<i>Value</i> = app.SCPI.SENSE(<i>Ch</i>).FREQuency.STOP app.SCPI.SENSE(<i>Ch</i>).FREQuency.STOP = 1E6
<i>Equivalent Softkeys</i>	Stimulus > Stop

SCPI.SENSE(*Ch*).OFFSet.ADJust.CONTInuous.PERiod

<i>Description</i>	Selects the period for continuous offset adjust feature: 1 – 3 sec; 2 – 10 sec; 3 – 30 sec.
<i>Type</i>	Long (read/write)
<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>Target</i>	Channel <i>Ch</i> from 1 to 16
<i>Syntax</i>	<i>Value</i> = app.SCPI.SENSE(<i>Ch</i>).OFFSet.ADJust.PERiod app.SCPI.SENSE(<i>Ch</i>).OFFSet.ADJust.PERiod = 1
<i>Equivalent Softkeys</i>	Stimulus > Frequency Offset > Offset Adjust > Adjust Period

SCPI.SENSE(*Ch*).OFFSet.ADJust.CONTInuous.STATe

<i>Description</i>	Turns ON/OFF the state of the continuous frequency offset adjust feature.
<i>Type</i>	Boolean (read/write)
<i>Parameter</i>	True: continuous offset adjust feature is ON False: continuous offset adjust feature is OFF
<i>Preset Value</i>	False
<i>Target</i>	Channel <i>Ch</i> from 1 to 16
<i>Syntax</i>	<i>Status</i> = app.SCPI.SENSE(<i>Ch</i>).OFFSet.ADJust.CONTInuous.STATe app.SCPI.SENSE(<i>Ch</i>).OFFSet.ADJust.CONTInuous.STATe = True
<i>Equivalent Softkeys</i>	Stimulus > Frequency Offset > Offset Adjust > Continuous Adjust

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

<i>Description</i>	Executes the offset adjust when the frequency offset feature is ON.
<i>Type</i>	Method
<i>Target</i>	Channel <i>Ch</i> from 1 to 16
<i>Syntax</i>	app.SCPI.SENSE(<i>Ch</i>).OFFset.ADJust.EXECute
<i>Equivalent Softkeys</i>	Stimulus > Frequency Offset > Offset Adjust > Adjust Immediate

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

<i>Description</i>	Selects the port number for frequency offset adjust feature.
<i>Type</i>	Long (read/write)
<i>Range</i>	Port number from 1 to 2 (4 for S4VNA)
<i>Out of Range</i>	Sets the value of the limit, which is closer to the specified value.
<i>Preset Value</i>	1
<i>Target</i>	Channel <i>Ch</i> from 1 to 16
<i>Syntax</i>	<code>Value = app.SCPi.SENSE(<i>Ch</i>).OFFSet.ADJust.PORT</code> <code>app.SCPi.SENSE(<i>Ch</i>).OFFSet.ADJust.PORT = 2</code>
<i>Equivalent Softkeys</i>	Stimulus > Frequency Offset > Offset Adjust > Select Port

SCPI.SENSE(*Ch*).OFFSet.ADJust.STATe

<i>Description</i>	Turns ON/OFF the state of the frequency offset adjust feature.
<i>Type</i>	Boolean (read/write)
<i>Parameter</i>	True: offset adjust feature is ON False: offset adjust feature is OFF
<i>Preset Value</i>	False
<i>Target</i>	Channel <i>Ch</i> from 1 to 16
<i>Syntax</i>	<code>Status = app.SCPi.SENSE(<i>Ch</i>).OFFSet.ADJust.STATe</code> <code>app.SCPi.SENSE(<i>Ch</i>).OFFSet.ADJust.STATe = True</code>
<i>Equivalent Softkeys</i>	Stimulus > Frequency Offset > Offset Adjust > Offset Adjust

SCPI.SENSE(*Ch*).OFFSet.ADJust.VAlue

<i>Description</i>	Sets/gets the frequency offset adjust value of the offset adjust feature.
<i>Type</i>	Double (read/write)
<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>Target</i>	Channel <i>Ch</i> from 1 to 16
<i>Syntax</i>	$Value = app.SCPi.SENSE(Ch).OFFSet.ADJust.VAlue$ $app.SCPi.SENSE(Ch).OFFSet.ADJust.VAlue = 1000$
<i>Equivalent Softkeys</i>	Stimulus > Frequency Offset > Offset Adjust > Adjust Value

SCPI.SENSE(*Ch*).OFFSET.PORT(*Pt*).FREQUENCY.DATA

<i>Description</i>	Reads out 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>Type</i>	Variant: array of double (read only)
<i>Target</i>	Port <i>Pt</i> of channel <i>Ch</i>, <i>Ch</i>: channel number 1–16 <i>Pt</i>: port number 1–2 (4 for S4VNA)
<i>Syntax</i>	<i>Data</i> = app.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>Description</i>	Sets/gets the basic frequency range divisor when the frequency offset feature is ON.
<i>Type</i>	Double (read/write)
<i>Range</i>	from 1 to 100
<i>Out of Range</i>	Sets the value of the limit, which is closer to the specified value.
<i>Preset Value</i>	1
<i>Target</i>	Port <i>Pt</i> of channel <i>Ch</i> , <i>Ch</i> : channel number 1–16 <i>Pt</i> : port number 1–2 (4 for S4VNA)
<i>Syntax</i>	<i>Value</i> = app.SCPi.SENSE(<i>Ch</i>).OFFset.PORT(<i>Pt</i>).FREQuency.DIVisor app.SCPi.SENSE(<i>Ch</i>).OFFset.PORT(<i>Pt</i>).FREQuency.DIVisor = 2
<i>Equivalent Softkeys</i>	Stimulus > Frequency Offset > Port n > Divider

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

<i>Description</i>	Sets/gets the basic range frequency multiplier when the frequency offset feature is ON and offset type is "PORT".
<i>Type</i>	Double (read/write)
<i>Range</i>	from –100 to 100
<i>Out of Range</i>	Sets the value of the limit, which is closer to the specified value.
<i>Preset Value</i>	1
<i>Target</i>	Port <i>Pt</i> of channel <i>Ch</i> , <i>Ch</i> : channel number 1–16 <i>Pt</i> : port number 1–2 (4 for S4VNA)
<i>Syntax</i>	Value = app.SCPi.SENSE(<i>Ch</i>).OFFset.PORT(<i>Pt</i>).FREQuency.MULTiplier app.SCPi.SENSE(<i>Ch</i>).OFFset.PORT(<i>Pt</i>).FREQuency.MULTiplier = 2
<i>Equivalent Softkeys</i>	Stimulus > Frequency Offset > Port n > Multiplier

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

<i>Description</i>	Sets/gets the basic frequency range offset when the frequency offset feature is ON and offset type is "PORT"..
<i>Type</i>	Double (read/write)
<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>Target</i>	Port <i>Pt</i> of channel <i>Ch</i> , <i>Ch</i> : channel number 1–16 <i>Pt</i> : port number 1–2 (4 for S4VNA)
<i>Syntax</i>	<i>Value</i> = app.SCPi.SENSE(<i>Ch</i>).OFFset.PORT(<i>Pt</i>).FREQuency.OFFSet app.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>Description</i>	Sets/gets the frequency sweep start when the frequency offset feature is ON and offset type is "PORT".
<i>Type</i>	Double (read/write)
<i>Unit</i>	Hz
<i>Target</i>	Port <i>Pt</i> of channel <i>Ch</i> , <i>Ch</i> : channel number 1–16 <i>Pt</i> : port number 1–2 (4 for S4VNA)
<i>Syntax</i>	<i>Value</i> = app.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>Description</i>	Sets/gets the frequency sweep stop when the frequency offset feature is ON. and offset type is "PORT".
<i>Type</i>	Double (read/write)
<i>Unit</i>	Hz
<i>Target</i>	Port <i>Pt</i> of channel <i>Ch</i> , <i>Ch</i> : channel number 1–16 <i>Pt</i> : port number 1–2 (4 for S4VNA)
<i>Syntax</i>	<i>Value</i> = app.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>Description</i>	Reads out the receiver 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: <i>Data</i>(<i>n</i>–1) the frequency value at the n-th measurement point.
<i>Type</i>	Variant: array of double (read only)
<i>Target</i>	Channel <i>Ch</i> from 1 to 16
<i>Syntax</i>	<i>Data</i> = <i>app</i> .SCPI.SENSE(<i>Ch</i>).OFFSet.RECeiver.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.RECeiver.FREQuency.DIVisor

<i>Description</i>	Sets/gets the basic frequency range divisor when the frequency offset feature is ON and offset type is "SRCRCv".
<i>Type</i>	Double (read/write)
<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>Target</i>	Channel <i>Ch</i> from 1 to 16
<i>Syntax</i>	<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>Description</i>	Sets/gets the basic range frequency multiplier when the frequency offset feature is ON and offset type is "SRCRCv".
<i>Type</i>	Double (read/write)
<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>Target</i>	Channel <i>Ch</i> from 1 to 16
<i>Syntax</i>	<i>Value</i> = <i>app</i>.SCPI.SENSE(<i>Ch</i>).OFFset.RECeiver.FREQuency.MULTiplier <i>app</i>.SCPI.SENSE(<i>Ch</i>).OFFset.RECeiver.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 > Receivers > Multiplier

SCPI.SENSE(*Ch*).OFFset.RECeiver.FREQuency.OFFSet

<i>Description</i>	Sets/gets the basic frequency range offset when the frequency offset feature is ON and offset type is "SRCRCv".
<i>Type</i>	Double (read/write)
<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>Target</i>	Channel <i>Ch</i> from 1 to 16
<i>Syntax</i>	<i>Value</i> = <i>app</i>.SCPI.SENSE(<i>Ch</i>).OFFset.RECeiver.FREQuency.OFFSet <i>app</i>.SCPI.SENSE(<i>Ch</i>).OFFset.RECeiver.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 > Receivers > Offset

SCPI.SENSE(*Ch*).OFFset.RECeiver.FREQuency.STARt

<i>Description</i>	Sets/gets the frequency sweep start when the frequency offset feature is ON and offset type is "SRCRCv".
<i>Type</i>	Double (read/write)
<i>Unit</i>	Hz
<i>Target</i>	Channel <i>Ch</i> from 1 to 16
<i>Syntax</i>	<i>Value</i> = <i>app</i>.SCPI.SENSE(<i>Ch</i>).OFFset.RECeiver.FREQuency.STARt <i>app</i>.SCPI.SENSE(<i>Ch</i>).OFFset.RECeiver.FREQuency.STARt = 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 > Receivers > Start

SCPI.SENSE(*Ch*).OFFset.RECeiver.FREQuency.STOP

<i>Description</i>	Sets/gets the frequency sweep stop when the frequency offset feature is ON and offset type is "SRCRCv".
<i>Type</i>	Double (read/write)
<i>Unit</i>	Hz
<i>Target</i>	Channel <i>Ch</i> from 1 to 16
<i>Syntax</i>	<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
<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>Description</i>	Reads out 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 <i>n</i>-th point, where <i>n</i> from 1 to N: <i>Data(n-1)</i> the frequency value at the <i>n</i>-th measurement point.
<i>Type</i>	Variant: array of double (read only)
<i>Target</i>	Channel <i>Ch</i> from 1 to 16
<i>Syntax</i>	<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>Description</i>	Sets/gets the basic frequency range divisor when the frequency offset feature is ON and offset type is "SRCRCv".
<i>Type</i>	Double (read/write)
<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>Target</i>	Channel <i>Ch</i> from 1 to 16
<i>Syntax</i>	<i>Value</i> = <i>app</i>.SCPI.SENSE(<i>Ch</i>).OFFset.SOURce.FREQuency.DIVisor <i>app</i>.SCPI.SENSE(<i>Ch</i>).OFFset.SOURce.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 > Source > Divider

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

<i>Description</i>	Sets/gets the basic range frequency multiplier when the frequency offset feature is ON and offset type is "SRCRCv".
<i>Type</i>	Double (read/write)
<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>Target</i>	Channel <i>Ch</i> from 1 to 16
<i>Syntax</i>	<i>Value</i> = <i>app</i>.SCPI.SENSE(<i>Ch</i>).OFFset.SOURce.FREQuency.MULTiplier <i>app</i>.SCPI.SENSE(<i>Ch</i>).OFFset.SOURce.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 > Source > Multiplier

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

<i>Description</i>	Sets/gets the basic frequency range offset when the frequency offset feature is ON and offset type is "SRCRCv".
<i>Type</i>	Double (read/write)
<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>Target</i>	Channel <i>Ch</i> from 1 to 16
<i>Syntax</i>	<i>Value</i> = <i>app</i>.SCPI.SENSE(<i>Ch</i>).OFFset.SOURce.FREQuency.OFFSet <i>app</i>.SCPI.SENSE(<i>Ch</i>).OFFset.SOURce.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 > Source > Offset

SCPI.SENSE(*Ch*).OFFset.SOURce.FREQuency.START

<i>Description</i>	Sets/gets the frequency sweep start when the frequency offset feature is ON and offset type is "SRCRCv".
<i>Type</i>	Double (read/write)
<i>Unit</i>	Hz
<i>Target</i>	Channel <i>Ch</i> from 1 to 16
<i>Syntax</i>	<i>Value</i> = <i>app</i>.SCPI.SENSE(<i>Ch</i>).OFFset.SOURce.FREQuency.START <i>app</i>.SCPI.SENSE(<i>Ch</i>).OFFset.SOURce.FREQuency.START = 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 > Source > Start

SCPI.SENSE(*Ch*).OFFset.SOURce.FREQuency.STOP

<i>Description</i>	Sets/gets the frequency sweep stop when the frequency offset feature is ON and offset type is "SRCRCv".
<i>Type</i>	Double (read/write)
<i>Unit</i>	Hz
<i>Target</i>	Channel <i>Ch</i> from 1 to 16
<i>Syntax</i>	<i>Value</i> = <i>app</i>.SCPI.SENSE(<i>Ch</i>).OFFset.SOURce.FREQuency.STOP <i>app</i>.SCPI.SENSE(<i>Ch</i>).OFFset.SOURce.FREQuency.STOP = 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 > Source > Stop

SCPI.SENSE(*Ch*).OFFSet.STATe

<i>Description</i>	Turns ON/OFF the state of the frequency offset feature.
<i>Type</i>	Boolean (read/write)
<i>Parameter</i>	True: frequency offset feature is ON False: frequency offset feature is OFF
<i>Preset Value</i>	False
<i>Target</i>	Channel <i>Ch</i> from 1 to 16
<i>Syntax</i>	<i>Status</i> = app.SCPi.SENSE(<i>Ch</i>).OFFSet.STATe app.SCPi.SENSE(<i>Ch</i>).OFFSet.STATe = True
<i>Equivalent Softkeys</i>	Stimulus > Frequency Offset > Frequency Offset

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

<i>Description</i>	Sets/gets 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>Type</i>	String (read/write)
<i>Parameter</i>	"PORT" : Port1/Port2 "SRCRcv" : Source/Receivers
<i>Preset Value</i>	"PORT"
<i>Target</i>	Channel <i>Ch</i> from 1 to 16
<i>Syntax</i>	<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>Description</i>	Selects the internal or external source of the reference frequency of 10 MHz.
<i>Type</i>	String (read/write)
<i>Parameter</i>	"INTernal" : Internal source of the reference frequency "EXTernal" : External source of the reference frequency
<i>Preset Value</i>	"INT"
<i>Target</i>	Channel <i>Ch</i> from 1 to 16
<i>Syntax</i>	<i>Param</i> = app.SCPI.SENSE(<i>Ch</i>).ROSCillator.SOURce app.SCPI.SENSE(<i>Ch</i>).ROSCillator.SOURce = "EXT"
<i>Equivalent Softkeys</i>	System > Misc Setup > Ref Source

SCPI.SENSE(*Ch*).SEGMENT.DATA

Description	The array of the segment sweep table. The array has the following format: <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> <Buf> : Always 5, <Flag1> : Stimulus start setting (0 – start/stop, 1 – center/span), <Flag2> : Setting of the <IFBW> field (0 – disabled, 1 – enabled), <Flag3> : Setting of the <Pow> field (0 – disabled, 1 – enabled), <Flag4> : Setting of the field (0 – disabled, 1 – enabled), <Flag5> : Setting of the <Time> field (0 – disabled, 1 – enabled), <N> : Number of segments, <Start <i>n</i>> : Start value of the <i>n</i>-th segment, <Stop <i>n</i>> : Stop value of the <i>n</i>-th segment, <NOP <i>n</i>> : Number of points of the <i>n</i>-th segment, <IFBW <i>n</i>> : IF bandwidth of the <i>n</i>-th segment (if enabled), <Pow <i>n</i>> : Power of the <i>n</i>-th segment (if enabled), <Del <i>n</i>> : Measurement delay of the <i>n</i>-th segment (if enabled), <Time <i>n</i>> : Reserved for future use (if enabled).
Type	Variant: array of double (read/write)
Target	Channel <i>Ch</i> from 1 to 16
Syntax	<i>Data</i> = app.SCPi.SENSE(<i>Ch</i>).SEGMENT.DATA app.SCPi.SENSE(<i>Ch</i>).SEGMENT.DATA = <i>Data</i>
Equivalent Softkeys	Stimulus / Segment Table

SCPI.SENSE(*Ch*).SWEp.POINT.TIME

<i>Description</i>	Sets/gets the value of the delay before measurement in each measurement point.
<i>Type</i>	Double (read/write)
<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>Target</i>	Channel <i>Ch</i> from 1 to 16
<i>Syntax</i>	<i>Value</i> = app.SCPI.SENSE(<i>Ch</i>).SWEp.POINT.TIME app.SCPI.SENSE(<i>Ch</i>).SWEp.POINT.TIME = 5E-6
<i>Equivalent Softkeys</i>	Stimulus > Meas Delay

SCPI.SENSE(*Ch*).SWEep.POINTs

<i>Description</i>	Sets/gets the number of measurement points.
<i>Type</i>	Long (read/write)
<i>Range</i>	2 to 500001 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>Target</i>	Channel <i>Ch</i> from 1 to 16
<i>Syntax</i>	<i>Value</i> = app.SCPI.SENSE(<i>Ch</i>).SWEep.POINTs app.SCPI.SENSE(<i>Ch</i>).SWEep.POINTs = 1001
<i>Equivalent Softkeys</i>	Stimulus > Points

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

<i>Description</i>	Sets/gets the the sweep type.
<i>Type</i>	String (read/write)
<i>Parameter</i>	"LINear" : Linear frequency sweep "LOGarithmic" : Logarithmic frequency sweep "SEGMENT" : Segment frequency sweep "POWER " : Power sweep
<i>Preset Value</i>	"LIN"
<i>Target</i>	Channel <i>Ch</i> from 1 to 16
<i>Syntax</i>	<i>Param</i> = app.SCPI.SENSE(<i>Ch</i>).SWEep.TYPE app.SCPI.SENSE(<i>Ch</i>).SWEep.TYPE = "LOG"
<i>Equivalent Softkeys</i>	Stimulus > Sweep Type

SCPI.SERVICE.CHANnel(1).ACTIVE

<i>Description</i>	Gets the number of the active channel.
<i>Type</i>	Long (read only)
<i>Syntax</i>	<i>Value</i> = app.SCPI.SERVICE.CHANnel.ACTive
<i>Warning</i>	Object <i>CHANnel</i> has an index of 1, which can be omitted in Visual Basic but it cannot be omitted in other programming languages.
<i>Equivalent Softkeys</i>	None

SCPI.SERVICE.CHANnel(1).COUNT

<i>Description</i>	Gets the maximum number of the channels.
<i>Type</i>	Long (read only)
<i>Syntax</i>	<i>Value</i> = app.SCPI.SERVICE.CHANnel.COUNT
<i>Warning</i>	Object <i>CHANnel</i> has an index of 1, which can be omitted in Visual Basic but it cannot be omitted in other programming languages.
<i>Equivalent Softkeys</i>	None

SCPI.SERVICE.CHANnel(*Ch*).TRACe(1).ACTIVE

<i>Description</i>	Gets the active trace number of the specified channel.
<i>Type</i>	Long (read only)
<i>Target</i>	Channel <i>Ch</i> from 1 to 16
<i>Syntax</i>	<i>Value</i> = app.SCPI.SERVICE.CHANnel(<i>Ch</i>).TRACe.ACTive
<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>Description</i>	Gets the maximum number of the traces in the channel.
<i>Type</i>	Long (read only)
<i>Syntax</i>	<i>Value</i> = app.SCPI.SERVICE.CHANnel.TRACe.COUNT
<i>Warning</i>	Object <i>CHANnel</i> has an index of 1, which can be omitted in Visual Basic but it cannot be omitted in other programming languages.
<i>Equivalent Softkeys</i>	None

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

<i>Description</i>	Gets the active marker number of the specified trace of the specified channel.
<i>Type</i>	Long (read only)
<i>Target</i>	Trace <i>Tr</i> of channel <i>Ch</i> , <i>Ch</i> : channel number 1–16 <i>Tr</i> : trace number 1–16
<i>Syntax</i>	<i>Value</i> = app.SCPI.SERVICE.CHANnel(<i>Ch</i>).TRACe.ACTive
<i>Equivalent Softkeys</i>	None

SCPI.SERVICE.PORT.COUNT

<i>Description</i>	Gets the number of the ports.
<i>Type</i>	Long (read only)
<i>Syntax</i>	<i>Value</i> = app.SCPI.SERVICE.PORT.COUNT
<i>Equivalent Softkeys</i>	None

SCPI.SERVICE.SWEep.FREQuency.MAXimum

<i>Description</i>	Gets the upper limit of the analyzer frequency range.
<i>Type</i>	Double (read only)
<i>Syntax</i>	<i>Value</i> = app.SCPI.SERVICE.SWEep.FREQuency.MAXimum
<i>Equivalent Softkeys</i>	None

SCPI.SERVICE.SWEep.FREQuency.MINimum

<i>Description</i>	Gets the lower limit of the analyzer frequency range.
<i>Type</i>	Double (read only)
<i>Syntax</i>	<i>Value</i> = app.SCPI.SERVICE.SWEep.FREQuency.MINimum
<i>Equivalent Softkeys</i>	None

SCPI.SERVICE.SWEep.POINTs

<i>Description</i>	Gets the maximum number of the measurement points.
<i>Type</i>	Double (read only)
<i>Syntax</i>	<i>Value</i> = app.SCPI.SERVICE.SWEep.POINTs
<i>Equivalent Softkeys</i>	None

SCPI.SOURce(Ch).POWER.CENTer

<i>Description</i>	Sets/gets the center value of the power sweep range when sweep type is Power.
<i>Type</i>	Double (read/write)
<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>Target</i>	Channel <i>Ch</i> from 1 to 16
<i>Syntax</i>	<i>Value</i> = app.SCPI.SOURce(<i>Ch</i>).POWER.CENTer app.SCPI.SOURce(<i>Ch</i>).POWER.CENTer = 5
<i>Equivalent Softkeys</i>	Stimulus > Center

SCPI.SOURce(Ch).POWer.LEVeL.IMMeDiate.AMPLitude

<i>Description</i>	Sets/gets the power level for the frequency sweep when port couple feature is set to ON by the command SCPI.SOURce(Ch).POWer.PORT(Pt).COUPLe..
<i>Type</i>	Double (read/write)
<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>Target</i>	Channel <i>Ch</i> from 1 to 16
<i>Syntax</i>	<i>Value</i> = app.SCPI.SOURce(Ch).POWer.LEVeL.IMMeDiate.AMPLitude app.SCPI.SOURce(Ch).POWer.LEVeL.IMMeDiate.AMPLitude = 10
<i>Related commands</i>	SCPI.SOURce(Ch).POWer.PORT(Pt).COUPLe
<i>Equivalent Softkeys</i>	Stimulus > Power > Power

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

<i>Description</i>	Sets/gets the power slope value for the frequency sweep.
<i>Type</i>	Double (read/write)
<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>Target</i>	Channel <i>Ch</i> from 1 to 16
<i>Syntax</i>	<i>Value</i> = app.SCPi.SOURce(<i>Ch</i>).POWer.LEVel.SLOPe.DATA app.SCPi.SOURce(<i>Ch</i>).POWer.LEVel.SLOPe.DATA = 0.2
<i>Equivalent Softkeys</i>	Stimulus > Power > Slope [dB/GHz]

SCPI.SOURce(Ch).POWer.LEVel.SLOPe.STATe

<i>Description</i>	Turns ON/OFF the state of the power slope for the frequency sweep.
<i>Type</i>	Boolean (read/write)
<i>Parameter</i>	True: Power slope ON False: Power slope OFF
<i>Preset Value</i>	False
<i>Target</i>	Channel <i>Ch</i> from 1 to 16
<i>Syntax</i>	<i>Status</i> = app.SCPi.SOURce(<i>Ch</i>).POWer.LEVel.SLOPe.STATe app.SCPi.SOURce(<i>Ch</i>).POWer.LEVel.SLOPe.STATe = True
<i>Equivalent Softkeys</i>	Stimulus > Power > Slope [ON/OFF]

SCPI.SOURce(*Ch*).POWer.PORT(*Pt*).CORRection.COLLection.ACQuire

<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>Type</i>	Method
<i>Target</i>	Port <i>Pt</i> of channel <i>Ch</i> , <i>Ch</i> : channel number 1–16 <i>Pt</i> : port number 1–2 (4 for S4VNA)
<i>Syntax</i>	<i>app</i> .SCPI.SOURce(<i>Ch</i>).POWer.PORT(<i>Pt</i>).CORRection.COLLection.ACQuire
<i>Equivalent Softkeys</i>	Calibration > Power Calibration > Calibrate

SCPI.SOURce(*Ch*).POWer.PORT(*Pt*).CORRection.COLLeT.TABLe.LOSS.DATA

<i>Description</i>	Sets/gets 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</i>(0) the number of the table rows N integer from 0 to 100; <i>Data</i>(2n–1) the frequency of the n–th row of the table from 300 kHz to 8.0 GHz; <i>Data</i>(2n) the loss value of the n-th table row in dB.
<i>Type</i>	Variant: array of double (read/write)
<i>Target</i>	Port <i>Pt</i> of channel <i>Ch</i>, <i>Ch</i>: channel number 1–16 <i>Pt</i>: port number 1–2 (4 for S4VNA)
<i>Syntax</i>	<i>Data</i> = <i>app</i>.SCPI.SOURce(<i>Ch</i>).POWer.PORT(<i>Pt</i>).CORRection.COLLeT.TABLe.LOSS.DATA <i>app</i>.SCPI.SOURce(<i>Ch</i>).POWer.PORT(<i>Pt</i>).CORRection.COLLeT.TABLe.LOSS.DATA = <i>Data</i>
<i>Related Commands</i>	SCPI.SOURce(<i>Ch</i>).POWer.PORT(<i>Pt</i>).CORRection.COLLeT.ACQuire
<i>Equivalent Softkeys</i>	Calibration > Power Calibration > Loss Comp

SCPI.SOURce(*Ch*).POWER.PORT(*Pt*).CORRection.COLLeT.TABLe.LOSS.STATe

<i>Description</i>	Turns ON/OFF the state of the loss compensation function used during the power calibration.
<i>Type</i>	Boolean (read/write)
<i>Parameter</i>	True: Loss compensation ON False: Loss compensation OFF
<i>Preset Value</i>	False
<i>Target</i>	Port <i>Pt</i> of channel <i>Ch</i> , <i>Ch</i> : channel number 1–16 <i>Pt</i> : port number 1–2 (4 for S4VNA)
<i>Syntax</i>	<i>Status</i> = app.SCPi.SOURce(<i>Ch</i>).POWER.PORT(<i>Pt</i>).CORRection.COLLeT.TABLe.LOSS.STATe app.SCPi.SOURce(<i>Ch</i>).POWER.PORT(<i>Pt</i>).CORRection.COLLeT.TABLe.LOSS.STATe = True
<i>Related Commands</i>	SCPI.SOURce(<i>Ch</i>).POWER.PORT(<i>Pt</i>).CORRection.COLLeT.TABLe.LOSS.DATA
<i>Equivalent Softkeys</i>	Calibration > Power Calibration > Loss Compens > Compensation

SCPI.SOURce(Ch).POWer.PORT(Pt).CORRection.DATA

<i>Description</i>	Sets/gets 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; <i>Data(2n)</i> power correction value of the n–th row of the table from –10 to +10 dB.
<i>Type</i>	Variant: array of double (read/write)
<i>Target</i>	Port <i>Pt</i> of channel <i>Ch</i>, <i>Ch</i>: channel number 1–16 <i>Pt</i>: port number 1–2 (4 for S4VNA)
<i>Syntax</i>	<i>Data</i> = app.SCPi.SOURce(<i>Ch</i>).POWer.PORT(<i>Pt</i>).CORRection.DATA app.SCPi.SOURce(<i>Ch</i>).POWer.PORT(<i>Pt</i>).CORRection.DATA = <i>Data</i>
<i>Related Commands</i>	SCPI.SOURce(<i>Ch</i>).POWer.PORT(<i>Pt</i>).CORRection.COLLect.ACQuire
<i>Equivalent Softkeys</i>	None

SCPI.SOURce(Ch).POWer.PORT(Pt).CORRection.STATe

<i>Description</i>	Turns ON/OFF the state of the power correction function.
<i>Type</i>	Boolean (read/write)
<i>Parameter</i>	True: Power correction ON False: Power correction OFF
<i>Preset Value</i>	False
<i>Target</i>	Port <i>Pt</i> of channel <i>Ch</i> , <i>Ch</i> : channel number 1–16 <i>Pt</i> : port number 1–2 (4 for S4VNA)
<i>Syntax</i>	<i>Status</i> = app.SCPi.SOURce(<i>Ch</i>).POWer.PORT(<i>Pt</i>).CORRection.STATe app.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(Pt).LEVel.IMMediate.AMPLitude

<i>Description</i>	Sets/gets the power level of each port for the frequency sweep when port couple feature is set to OFF by the command SCPI.SOURce(Ch).POWer.PORT(Pt).COUPLe.
<i>Type</i>	Double (read/write)
<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>Target</i>	Port <i>Pt</i> of channel <i>Ch</i> , <i>Ch</i> : channel number 1–16 <i>Pt</i> : port number 1–2 (4 for S4VNA)
<i>Syntax</i>	<i>Value</i> = app.SCPI.SOURce(Ch).POWer.PORT(Pt).LEVel.IMMediate.AMPLitude app.SCPI.SOURce(Ch).POWer.PORT(Pt).LEVel.IMMediate.AMPLitude = 10
<i>Related Commands</i>	SCPI.SOURce(Ch).POWer.PORT(Pt).COUPLe
<i>Equivalent Softkeys</i>	Stimulus > Power > Port Power > Port x

SCPI.SOURce(Ch).POWer.PORT(1).COUPle

<i>Description</i>	Turns ON/OFF the state of the port power couple function. Setting the port power couple feature to OFF allows independent power level setting for each port.
<i>Type</i>	Boolean (read/write)
<i>Parameter</i>	True: port power couple ON False: port power couple OFF
<i>Preset Value</i>	True
<i>Target</i>	Channel <i>Ch</i> from 1 to 16
<i>Syntax</i>	<i>Status</i> = app.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 of 1, which can be omitted in Visual Basic but it cannot be omitted in other programming languages.
<i>Related Commands</i>	SCPI.SOURce(<i>Ch</i>).POWer.PORT(<i>Pt</i>).LEVeL.IMMeDiate.AMPLitude
<i>Equivalent Softkeys</i>	Stimulus > Power > Port Couple

SCPI.SOURce(Ch).POWer.SPAN

<i>Description</i>	Sets/gets the power span for the power sweep.
<i>Type</i>	Double (read/write)
<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>Target</i>	Channel <i>Ch</i> from 1 to 16
<i>Syntax</i>	<i>Value</i> = app.SCPI.SOURce(<i>Ch</i>).POWer.SPAN app.SCPI.SOURce(<i>Ch</i>).POWer.SPAN = 50
<i>Equivalent Softkeys</i>	Stimulus > Span

SCPI.SOURce(Ch).POWer.STARt

<i>Description</i>	Sets/gets the power sweep start for the power sweep.
<i>Type</i>	Double (read/write)
<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>Target</i>	Channel <i>Ch</i> from 1 to 16
<i>Syntax</i>	<i>Value</i> = app.SCPI.SOURce(<i>Ch</i>).POWer.STARt app.SCPI.SOURce(<i>Ch</i>).POWer.STARt = 5
<i>Equivalent Softkeys</i>	Stimulus > Start

SCPI.SOURce(Ch).POWer.STOP

<i>Description</i>	Sets/gets the power sweep stop for the power sweep.
<i>Type</i>	Double (read/write)
<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>Target</i>	Channel <i>Ch</i> from 1 to 16
<i>Syntax</i>	<i>Value</i> = app.SCPI.SOURce(<i>Ch</i>).POWer.STOP app.SCPI.SOURce(<i>Ch</i>).POWer.STOP = 5
<i>Equivalent Softkeys</i>	Stimulus > Stop

SCPI.STATus.OPERation.CONDition

<i>Description</i>	Reads out the Operation Status Condition Register.
<i>Type</i>	Long (read only)
<i>Target</i>	IEEE488.2 Status Reporting System (Appendix 1)
<i>Syntax</i>	<i>Value</i> = app.SCPI.STATus.OPERation.CONDition
<i>Equivalent Softkeys</i>	None

SCPI.STATus.OPERation.ENABLE

<i>Description</i>	Sets/gets the Operation Status Enable Register.
<i>Type</i>	Long (read/write)
<i>Range</i>	from 0 to 65535
<i>Preset Value</i>	0
<i>Target</i>	IEE488.2 Status Reporting System (Appendix 1)
<i>Syntax</i>	<i>Value</i> = app.SCPI.STATus.OPERation.ENABLE app.SCPI.STATus.OPERation.ENABLE = <i>Value</i>
<i>Equivalent Softkeys</i>	None

SCPI.STATus.OPERation.EVENT

<i>Description</i>	Reads out the Operation Status Event Register.
<i>Type</i>	Long (read only)
<i>Target</i>	IEE488.2 Status Reporting System (Appendix 1)
<i>Syntax</i>	<i>Value</i> = app.SCPI.STATus.OPERation.EVENT
<i>Equivalent Softkeys</i>	None

SCPI.STATus.OPERation.NTRansition

<i>Description</i>	Sets/gets the Negative Transition Filter of the Operation Status Register.
<i>Type</i>	Long (read/write)
<i>Range</i>	from 0 to 65535
<i>Preset Value</i>	0
<i>Target</i>	IEE488.2 Status Reporting System (Appendix 1)
<i>Syntax</i>	<i>Value</i> = app.SCPI.STATus.OPERation.NTRansition app.SCPI.STATus.OPERation.NTRansition = <i>Value</i>
<i>Equivalent Softkeys</i>	None

SCPI.STATus.OPERation.PTRansition

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

SCPI.STATus.PRESet

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

SCPI.STATus.QUEStionable.CONDiTion

<i>Description</i>	Reads out the Questionable Status Condition Register.
<i>Type</i>	Long (read only)
<i>Target</i>	IEE488.2 Status Reporting System (Appendix 1)
<i>Syntax</i>	<i>Value</i> = <i>app</i> .SCPI.STATus.QUEStionable.CONDiTion
<i>Equivalent Softkeys</i>	None

SCPI.STATus.QUEStionable.ENABLE

<i>Description</i>	Sets/gets the Questionable Status Enable Register.
<i>Type</i>	Long (read/write)
<i>Range</i>	from 0 to 65535
<i>Preset Value</i>	0
<i>Target</i>	IEE488.2 Status Reporting System (Appendix 1)
<i>Syntax</i>	<i>Value</i> = app.SCPI.STATus.QUEStionable.ENABLE app.SCPI.STATus.QUEStionable.ENABLE = <i>Value</i>
<i>Equivalent Softkeys</i>	None

SCPI.STATus.QUEStionable.EVENT

<i>Description</i>	Reads out the Questionable Status Event Register.
<i>Type</i>	Long (read only)
<i>Target</i>	IEE488.2 Status Reporting System (Appendix 1)
<i>Syntax</i>	<i>Value</i> = app.SCPI.STATus.QUEStionable.EVENT
<i>Equivalent Softkeys</i>	None

SCPI.STATus.QUEStionable.LIMit.CHANnel(*Ch*).CONDition

<i>Description</i>	Reads out the Questionable Limit Channel Status Condition Register for channel <i>Ch</i> , <i>Ch</i> : channel number 1–16
<i>Type</i>	Long (read only)
<i>Target</i>	IEE488.2 Status Reporting System (Appendix 1)
<i>Syntax</i>	<i>Value</i> = app.SCPI.STATus.QUEStionable.LIMit.CHANnel(<i>Ch</i>).CONDition
<i>Equivalent Softkeys</i>	None

SCPI.STATus.QUEStionable.LIMit.CHANnel(*Ch*).ENABle

<i>Description</i>	Sets/gets the Questionable Limit Channel Status Enable Register for channel <i>Ch</i> , <i>Ch</i> : channel number 1–16
<i>Type</i>	Long (read/write)
<i>Range</i>	from 0 to 65535
<i>Preset Value</i>	0
<i>Target</i>	IEE488.2 Status Reporting System (Appendix 1)
<i>Syntax</i>	<i>Value</i> = app.SCPI.STATus.QUEStionable.LIMit.CHANnel(<i>Ch</i>).ENABle app.SCPI.STATus.QUEStionable.LIMit.CHANnel(<i>Ch</i>).ENABle = <i>Value</i>
<i>Equivalent Softkeys</i>	None

SCPI.STATus.QUEStionable.LIMit.CHANnel(*Ch*).EVENT

<i>Description</i>	Reads out the Questionable Limit Channel Status Event Register for channel <i>Ch</i> , <i>Ch</i> : channel number 1–16
<i>Type</i>	Long (read only)
<i>Target</i>	IEE488.2 Status Reporting System (Appendix 1)
<i>Syntax</i>	<i>Value</i> = app. SCPI.STATus.QUEStionable.LIMit.CHANnel(<i>Ch</i>).EVENT
<i>Equivalent Softkeys</i>	None

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

<i>Description</i>	Sets/gets the Negative Transition Filter of the Questionable Limit Channel Status Register for channel <i>Ch</i> , <i>Ch</i> : channel number 1–16
<i>Type</i>	Long (read/write)
<i>Range</i>	from 0 to 65535
<i>Preset Value</i>	0
<i>Target</i>	IEE488.2 Status Reporting System (Appendix 1)
<i>Syntax</i>	<i>Value</i> = app.SCPI.STATus.QUEStionable.LIMit.CHANnel(<i>Ch</i>).NTRansition app.SCPI.STATus.QUEStionable.LIMit.CHANnel(<i>Ch</i>).NTRansition = <i>Value</i>
<i>Equivalent Softkeys</i>	None

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

<i>Description</i>	Sets/gets the Positive Transition Filter of the Questionable Limit Channel Status Register for channel <i>Ch</i> , <i>Ch</i> : channel number 1–16
<i>Type</i>	Long (read/write)
<i>Range</i>	from 0 to 65535
<i>Preset Value</i>	65535
<i>Target</i>	IEE488.2 Status Reporting System (Appendix 1)
<i>Syntax</i>	<i>Value</i> = app.SCPi.STATus.QUEStionable.LIMit.CHANnel(<i>Ch</i>).PTRansition app.SCPi.STATus.QUEStionable.LIMit.CHANnel(<i>Ch</i>).PTRansition = <i>Value</i>
<i>Equivalent Softkeys</i>	None

SCPI.STATus.QUEStionable.LIMit.CONDiTion

<i>Description</i>	Reads out the Questionable Limit Status Condition Register.
<i>Type</i>	Long (read only)
<i>Target</i>	IEE488.2 Status Reporting System (Appendix 1).
<i>Syntax</i>	<i>Value</i> = app.SCPi.STATus.QUEStionable.LIMit.CONDiTion
<i>Equivalent Softkeys</i>	None

SCPI.STATus.QUEStionable.LIMit.ENABLE

<i>Description</i>	Sets/gets the Questionable Limit Status Enable Register.
<i>Type</i>	Long (read/write)
<i>Range</i>	from 0 to 65535
<i>Preset Value</i>	0
<i>Target</i>	IEE488.2 Status Reporting System (Appendix 1)
<i>Syntax</i>	<i>Value</i> = app.SCPI.STATus.QUEStionable.LIMit.ENABLE app.SCPI.STATus.QUEStionable.LIMit.ENABLE = <i>Value</i>
<i>Equivalent Softkeys</i>	None

SCPI.STATus.QUEStionable.LIMit.EVENT

<i>Description</i>	Reads out the Questionable Limit Status Event Register.
<i>Type</i>	Long (read only)
<i>Target</i>	IEE488.2 Status Reporting System (Appendix 1)
<i>Syntax</i>	<i>Value</i> = app.SCPI.STATus.QUEStionable.LIMit.EVENT
<i>Equivalent Softkeys</i>	None

SCPI.STATus.QUEStionable.LIMit.NTRansition

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

SCPI.STATus.QUEStionable.LIMit.PTRansition

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

SCPI.STATus.QUEStionable.NTRansition

<i>Description</i>	Sets/gets the Negative Transition Filter of the Questionable Status Register.
<i>Type</i>	Long (read/write)
<i>Range</i>	from 0 to 65535
<i>Preset Value</i>	0
<i>Target</i>	IEE488.2 Status Reporting System (Appendix 1)
<i>Syntax</i>	<i>Value</i> = app.SCPI.STATus.QUEStionable.NTRansition <i>app.SCPI.STATus.QUEStionable.NTRansition</i> = <i>Value</i>
<i>Equivalent Softkeys</i>	None

SCPI.STATus.QUEStionable.PTRansition

<i>Description</i>	Sets/gets the Positive Transition Filter of the Questionable Status Register.
<i>Type</i>	Long (read/write)
<i>Range</i>	from 0 to 65535
<i>Preset Value</i>	65535
<i>Target</i>	IEE488.2 Status Reporting System (Appendix 1)
<i>Syntax</i>	<i>Value</i> = app.SCPI.STATus.QUEStionable.PTRansition <i>app.SCPI.STATus.QUEStionable.PTRansition</i> = <i>Value</i>
<i>Equivalent Softkeys</i>	None

SCPI.STATus.QUEStionable.RLIMit.CHANnel(*Ch*).CONDition

<i>Description</i>	Reads out the Questionable Ripple Limit Channel Status Condition Register for channel <i>Ch</i> , <i>Ch</i> : channel number 1–16
<i>Type</i>	Long (read only)
<i>Target</i>	IEE488.2 Status Reporting System (Appendix 1)
<i>Syntax</i>	<i>Value</i> = app.SCPI.STATus.QUEStionable.RLIMit.CHANnel(<i>Ch</i>).CONDition
<i>Equivalent Softkeys</i>	None

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

<i>Description</i>	Sets/gets the Questionable Ripple Limit Channel Status Enable Register for channel <i>Ch</i> , <i>Ch</i> : channel number 1–16
<i>Type</i>	Long (read/write)
<i>Range</i>	from 0 to 65535
<i>Preset Value</i>	0
<i>Target</i>	IEE488.2 Status Reporting System (Appendix 1)
<i>Syntax</i>	<i>Value</i> = app.SCPI.STATus.QUEStionable.RLIMit.CHANnel(<i>Ch</i>).ENABLE app.SCPI.STATus.QUEStionable.RLIMit.CHANnel(<i>Ch</i>).ENABLE = <i>Value</i>
<i>Equivalent Softkeys</i>	None

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

<i>Description</i>	Reads out the Questionable Ripple Limit Channel Status Event Register for channel <i>Ch</i> , <i>Ch</i> : channel number 1–16
<i>Type</i>	Long (read only)
<i>Target</i>	IEE488.2 Status Reporting System (Appendix 1)
<i>Syntax</i>	<i>Value</i> = app.SCPI.STATus.QUEStionable.RLIMit.CHANnel(<i>Ch</i>).EVENT
<i>Equivalent Softkeys</i>	None

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

<i>Description</i>	Sets/gets the Negative Transition Filter of the Questionable Ripple Limit Channel Status Register for channel <i>Ch</i> , <i>Ch</i> : channel number 1–16
<i>Type</i>	Long (read/write)
<i>Range</i>	from 0 to 65535
<i>Preset Value</i>	0
<i>Target</i>	IEE488.2 Status Reporting System (Appendix 1)
<i>Syntax</i>	<i>Value</i> = app.SCPI.STATus.QUEStionable.RLIMit.CHANnel(<i>Ch</i>). NTRansition <i>app.SCPI.STATus.QUEStionable.RLIMit.CHANnel(<i>Ch</i>). NTRansition</i> = <i>Value</i>
<i>Equivalent Softkeys</i>	None

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

<i>Description</i>	Sets/gets the Positive Transition Filter of the Questionable Ripple Limit Channel Status Register for channel <i>Ch</i> , <i>Ch</i> : channel number 1–16
<i>Type</i>	Long (read/write)
<i>Range</i>	from 0 to 65535
<i>Preset Value</i>	65535
<i>Target</i>	IEE488.2 Status Reporting System (Appendix 1)
<i>Syntax</i>	<i>Value</i> = app.SCPi.STATus.QUEStionable.RLIMit.CHANnel(<i>Ch</i>). PTRansition app.SCPi.STATus.QUEStionable.RLIMit.CHANnel(<i>Ch</i>). PTRansition = <i>Value</i>
<i>Equivalent Softkeys</i>	None

SCPI.STATus.QUEStionable.RLIMit.CONDiTion

<i>Description</i>	Reads out the Questionable Ripple Limit Status Condition Register.
<i>Type</i>	Long (read only)
<i>Target</i>	IEE488.2 Status Reporting System (Appendix 1)
<i>Syntax</i>	<i>Value</i> = app.SCPi.STATus.QUEStionable.RLIMit.CONDiTion
<i>Equivalent Softkeys</i>	None

SCPI.STATus.QUEStionable.RLIMit.ENABLE

<i>Description</i>	Sets/gets the Questionable Ripple Limit Status Enable Register.
<i>Type</i>	Long (read/write)
<i>Range</i>	from 0 to 65535
<i>Preset Value</i>	0
<i>Target</i>	IEE488.2 Status Reporting System (Appendix 1)
<i>Syntax</i>	<i>Value</i> = app.SCPI.STATus.QUEStionable.RLIMit.ENABLE app.SCPI.STATus.QUEStionable.RLIMit.ENABLE = <i>Value</i>
<i>Equivalent Softkeys</i>	None

SCPI.STATus.QUEStionable.RLIMit.EVENT

<i>Description</i>	Reads out the Questionable Ripple Limit Status Event Register.
<i>Type</i>	Long (read only)
<i>Target</i>	IEE488.2 Status Reporting System (Appendix 1)
<i>Syntax</i>	<i>Value</i> = app.SCPI.STATus.QUEStionable.RLIMit.EVENT
<i>Equivalent Softkeys</i>	None

SCPI.STATus.QUEStionable.RLiMit.NTRansition

<i>Description</i>	Sets/gets the Negative Transition Filter of the Questionable Ripple Limit Status Register.
<i>Type</i>	Long (read/write)
<i>Range</i>	from 0 to 65535
<i>Preset Value</i>	0
<i>Target</i>	IEE488.2 Status Reporting System (Appendix 1)
<i>Syntax</i>	<i>Value</i> = app.SCPI.STATus.QUEStionable.RLiMit.NTRansition app.SCPI.STATus.QUEStionable.RLiMit.NTRansition = <i>Value</i>
<i>Equivalent Softkeys</i>	None

SCPI.STATus.QUEStionable.RLiMit.PTRansition

<i>Description</i>	Sets/gets the Positive Transition Filter of the Questionable Ripple Limit Status Register.
<i>Type</i>	Long (read/write)
<i>Range</i>	from 0 to 65535
<i>Preset Value</i>	65535
<i>Target</i>	IEE488.2 Status Reporting System (Appendix 1)
<i>Syntax</i>	<i>Value</i> = app.SCPI.STATus.QUEStionable.RLiMit.PTRansition app.SCPI.STATus.QUEStionable.RLiMit.PTRansition = <i>Value</i>
<i>Equivalent Softkeys</i>	None

SCPI.SYSTem.BEEPer.COMPLete.IMMediate

<i>Description</i>	Generates a beep to notify of the completion of the operation.
<i>Type</i>	Method
<i>Target</i>	Analyzer
<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>Description</i>	Turns ON/OFF the state of the beeper notifying of the completion of the operation.
<i>Type</i>	Boolean (read/write)
<i>Parameter</i>	True: Completion beeper ON False: Completion beeper OFF
<i>Preset Value</i>	True
<i>Target</i>	Analyzer
<i>Syntax</i>	<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.WARNIing.IMMediate

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

SCPI.SYSTem.BEEPer.WARNIing.STATe

<i>Description</i>	Turns ON/OFF the state of the beeper notifying of warning.
<i>Type</i>	Boolean (read/write)
<i>Parameter</i>	True: Warning beeper ON False: Warning beeper OFF
<i>Preset Value</i>	True
<i>Target</i>	Analyzer
<i>Syntax</i>	<code>Status = app.SCPi.SYSTem.BEEPer.WARNIing.STATe</code> <code>app.SCPi.SYSTem.BEEPer.WARNIing.STATe = False</code>
<i>Equivalent Softkeys</i>	System > Misc Setup > Beeper > Beep Warning

SCPI.SYSTem.COMMunicate.ECAL.TEMPerature.SENSor

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

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

<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>Type</i>	String (read/write)
<i>Parameter</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>Target</i>	AutoCal module
<i>Syntax</i>	<i>Param</i> = app.SCPI.SYSTem.COMMunicate.ECAL.IMPedance(<i>Pt</i>) app.SCPI.SYSTem.COMMunicate.ECAL.IMPedance(<i>Pt</i>) = "OPEN"
<i>Equivalent Softkeys</i>	None

SCPI.SYSTem.COMMunicate.ECAL.CHECK

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

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

<i>Description</i>	Sets the thru state between the specified 2 ports (<i>Pt1</i> , <i>Pt2</i>) of AutoCal module connected to the analyzer.
<i>Type</i>	Method
<i>Target</i>	AutoCal module
<i>Syntax</i>	<i>app</i> .SCPI.SYSTem.COMMunicate.ECAL.THRU(1, 2)
<i>Equivalent Softkeys</i>	None

SCPI.SYSTem.CORRection.STATe

<i>Description</i>	Turns ON/OFF the state of the system error correction (factory calibration at the RF port connector).
<i>Type</i>	Boolean (read/write)
<i>Parameter</i>	True: System error correction ON False: System error correction OFF
<i>Preset Value</i>	True
<i>Target</i>	Analyzer
<i>Syntax</i>	<i>Status</i> = app.SCPI.SYSTem.CORRection.STATe <i>app.SCPI.SYSTem.CORRection.STATe</i> = False
<i>Equivalent Softkeys</i>	System > Misc Setup > System Correction

SCPI.SYSTem.CYCLe.TIME.MEASurement

<i>Description</i>	Reads out the measured cycle time. The cycle time is the interval between the start of two adjacent sweeps. The cycle time is averaged by an exponential window with a time constant of about 0.5 sec. If the cycle time is changed more than 100 usec in comparison with the averaged time, the averaging starts anew.
<i>Type</i>	Double (read only)
<i>Unit</i>	second
<i>Target</i>	Analyzer
<i>Syntax</i>	<i>Value</i> = app.SCPI.SYSTem.CYCLe.TIME.MEASurement
<i>Equivalent Softkeys</i>	Display > Properies > Cycle Time

SCPI.SYSTem.DATE

<i>Description</i>	Sets/gets the current date. The array consists of three elements: <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>Type</i>	Variant: array of long (read/write)
<i>Syntax</i>	<i>Data</i> = app.SCPI.SYSTem.DATE <i>app</i>.SCPI.SYSTem.DATE = Array(2009, 9, 9)
<i>Equivalent Softkeys</i>	None

SCPI.SYSTem.HIDE

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

SCPI.SYSTem.LOCal

<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>Type</i>	Method
<i>Target</i>	Analyzer GUI
<i>Syntax</i>	<i>app</i> .SCPI.SYSTem.LOCal
<i>Related Commands</i>	SCPI.SYSTem.REMote SCPI.SYSTem.RWLock
<i>Equivalent Softkeys</i>	None

SCPI.SYSTem.PRESet

<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>Type</i>	Method
<i>Target</i>	Analyzer GUI
<i>Syntax</i>	<i>app</i> .SCPI.SYSTem.PRESet
<i>Equivalent Softkeys</i>	System > Preset > OK

SCPI.SYSTem.PORT.SWITchover.Delay.STATe

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

SCPI.SYSTem.REMote

<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>Type</i>	Method
<i>Target</i>	Analyzer GUI
<i>Syntax</i>	<i>app</i> .SCPI.SYSTem.REMote
<i>Related Commands</i>	SCPI.SYSTem.LOCal SCPI.SYSTem.RWLock
<i>Equivalent Softkeys</i>	None

SCPI.SYSTem.RWLock

<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>Type</i>	Method
<i>Target</i>	Analyzer GUI
<i>Syntax</i>	<i>app</i> .SCPI.SYSTem.RWLock
<i>Related Commands</i>	SCPI.SYSTem.LOCal SCPI.SYSTem.REMote
<i>Equivalent Softkeys</i>	None

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

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

SCPI.SYSTem.SHOW

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

SCPI.SYSTem.TERMinate

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

SCPI.SYSTem.TIME

<i>Description</i>	Sets/gets the current time. The array consists of three elements: <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>Type</i>	Variant: array of long (read/write)
<i>Syntax</i>	<i>Data</i> = app.SCPI.SYSTem.TIME <i>app. app.SCPI.SYSTem.TIME</i> = Array(15, 20, 30)
<i>Equivalent Softkeys</i>	None

SCPI.TRIGger.SEQuence.AVERage

<i>Description</i>	Turns ON/OFF the averaging trigger function. The function executes a sweep the number of times specified by the averaging factor with a single trigger for the channels with the averaging enabled. The averaging process begins again with each trigger. Note: The point trigger function has priority against this command. When the point trigger is enabled the number of pulses equal to (number of points) x (averaging factor) is needed to complete the averaging.
<i>Type</i>	Boolean (read/write)
<i>Parameter</i>	True: averaging trigger function ON False: averaging trigger function OFF
<i>Preset Value</i>	False
<i>Syntax</i>	<i>Status</i> = app.SCPI.TRIGger.SEQuence.AVERage app.SCPI.TRIGger.SEQuence.AVERage = True
<i>Related Commands</i>	SCPI.SENSE(Ch).AVERage.STATe
<i>Equivalent Softkeys</i>	Average > Avg Trigger { On Off }

SCPI.TRIGger.SEQuence.EXTernal.Delay

<i>Description</i>	Sets/gets the response delay with respect to the external trigger signal.
<i>Type</i>	Double (read/write)
<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>Target</i>	Trigger input
<i>Syntax</i>	<i>Param</i> = <i>app</i> .SCPI.TRIGger.EXTernal.Delay <i>app</i> .SCPI.TRIGger.INPut.EXTernal.Delay = 0
<i>Related Commands</i>	SCPI.TRIGger.SEQuence.SOURce
<i>Equivalent Softkeys</i>	Stimulus > Trigger > Ext Trigger > Delay

SCPI.TRIGger.SEQuence.EXTeRnal.POSition

<i>Description</i>	Selects the position of the external trigger. The Analyzer waits for external trigger: • Before sampling, when the frequency of the stimulus port have been set. • Before the frequency setup and subsequent measurement. The frequency change of the stimulus port begins when the external trigger arrives. 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.
<i>Type</i>	String (read/write)
<i>Parameter</i>	"BSAM" : Before sampling "BSET" : Before frequency setup
<i>Preset Value</i>	"BSAM"
<i>Target</i>	Trigger input
<i>Syntax</i>	<i>Param</i> = <i>app</i>.SCPI.TRIGger.EXTeRnal.POSition <i>app</i>.SCPI.TRIGger.INPut.EXTeRnal.POSition = "BSAM"
<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>Description</i>	Selects the edge polarity of the external trigger.
<i>Type</i>	String (read/write)
<i>Parameter</i>	"POSitive" : Positive edge "NEGative" : Negative edge
<i>Preset Value</i>	"NEG"
<i>Target</i>	Trigger input
<i>Syntax</i>	<i>Param</i> = <i>app</i> .SCPI.TRIGger.EXTeRnal.SLOPe <i>app</i> .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>Description</i>	Generates a trigger signal and initiates a sweep under the following conditions. • Trigger source is set to the <i>BUS</i> (set by the command SCPI.TRIGger.SEQuence.SOURce = "BUS"), otherwise an error occurs and the command is ignored. • Analyzer must be in the <i>trigger waiting</i> state, otherwise (the analyzer is in the <i>measurement</i> state or in the <i>hold</i> state) an error occurs and the command is ignored. The command is completed immediately after the generation of the trigger signal (does not wait the end of a sweep).
<i>Type</i>	Method
<i>Target</i>	Analyzer
<i>Syntax</i>	<i>app</i> .SCPI.TRIGger.SEQuence.IMMediate
<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.TRIGger.SEQuence.POINt

<i>Description</i>	Turns ON/OFF the point trigger feature. When the point trigger is turned ON, the external trigger response is the single point. When the point trigger feature is turned OFF, the external trigger response is the entire sweep.
<i>Type</i>	Boolean (read/write)
<i>Parameter</i>	True: point trigger feature ON False: point trigger feature OFF
<i>Preset Value</i>	False
<i>Target</i>	Trigger input
<i>Syntax</i>	<i>Status</i> = app.SCPI.TRIGger.SEQuence.POINt app.SCPI.TRIGger.SEQuence.POINt = True
<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>Description</i>	Generates a trigger signal and initiates a sweep under the following conditions. • Trigger source is set to the <i>BUS</i> (set by the command SCPI.TRIGger.SEQuence.SOURce = "BUS"), otherwise an error occurs and the command is ignored. • Analyzer must be in the <i>trigger waiting</i> state, otherwise (the analyzer is in the <i>measurement</i> state or in the <i>hold</i> state) an error occurs and the command is ignored. As opposed to the SCPI.TRIGger.SEQuence.IMMediate command this command waits the end of the sweep. The method returns control after the end of the sweep caused by the command in one or more channels.
<i>Type</i>	Method
<i>Target</i>	Analyzer
<i>Syntax</i>	<i>app</i> .SCPI.TRIGger.SEQuence.SINGle
<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.TRIGger.SEQuence.SCOPE

<i>Description</i>	Sets or reads out the trigger scope. The trigger scope determines the response on the trigger signal arrival: either starts a sweep of all waiting channels in turn or starts a sweep in the active channel only.
<i>Type</i>	String (read/write)
<i>Parameter</i>	"ALL" : All Channels "ACTive" : Active Channel
<i>Preset Value</i>	"ALL"
<i>Target</i>	Analyzer
<i>Syntax</i>	<i>Param</i> = <i>app</i> .SCPI.TRIGger.SEQuence.SCOPE <i>app</i> .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

Description	Selects the trigger source (see options below). If the the <i>continuous trigger initiation</i> mode is enabled with the command SCPI.INITiate(Ch).CONTInuous = true, the INTernal choice leads to continuous sweep. The choice of another option switches the analyzer to the <i>trigger waiting state</i> from the corresponding source. If the the <i>continuous trigger initiation</i> mode is disabled with the command SCPI.INITiate(Ch).CONTInuous = false, the reaction to SCPI.INITiate(Ch).IMMediate command is different. Selecting INTernal leads to a single sweep in response to the command SCPI.INITiate(Ch).IMMediate, selection another option puts the analyzer in a <i>single trigger waiting</i> state in response to the SCPI.INITiate(Ch).IMMediate command.
Type	String (read/write)
Parameter	"INTernal" : Internal "EXTernal" : External "MANual" : Manual "BUS" : Bus
Preset Value	"INT"
Target	Analyzer
Syntax	Param = app.SCPI.TRIGger.SEQuence.SOURce app.SCPI.TRIGger.SEQuence.SOURce = "BUS"
Related Commands	SCPI.TRIGger.SEQuence.IMMediate SCPI.TRIGger.SEQuence.SINGLE SCPI.IEEE4882.TRG
Equivalent Softkeys	Stimulus > Trigger > Trigger Source > Internal External Manual Bus

SCPI.TRIGger.SEQunce.STATUs

<i>Description</i>	Reads out the the current state of the analyzer.
<i>Type</i>	String (read only)
<i>Return value</i>	"HOLD" : Hold "WAIT" : Waiting for trigger "MEAS" : Measure (sweep in progress)
<i>Syntax</i>	<i>Param</i> = <i>app</i> .SCPI.TRIGger.SEQunce.STATUs
<i>Equivalent Softkeys</i>	None

SCPI.TRIGger.SEQunce.WAIT(STATUs)

<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 (see options below). The command can be used to wait for the end of the sweep initiated by the commands SCPI.TRIGger.SEQunce.IMMEDIATE, SCPI.IEEE4882.TRG or initiated by the external trigger signal. If the <i>continuous initiation mode</i> is turned ON by the command SCPI.INITiate(Ch).CONTInuous = true, then the parameter of the command must be WAIT, otherwise HOLD.
<i>Type</i>	Method
<i>Parameter</i>	Status of String type: "HOLD" : Hold "WAIT" : Wait for Trigger "MEAS" : Measurement
<i>Target</i>	Analyzer
<i>Syntax</i>	<i>app</i> .SCPI.TRIGger.SEQunce.WAIT("HOLD")
<i>Equivalent Softkeys</i>	None

SCPI.TRIGger.OUTPut.FUNcTion

<i>Description</i>	Selects the trigger output function. The trigger output outputs various waveforms depending on the setting of the Output Trigger Function (see the operating manual).
<i>Type</i>	String (read/write)
<i>Parameter</i>	"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
<i>Preset Value</i>	"RTRG"
<i>Target</i>	Trigger output
<i>Syntax</i>	<code>Param = app.SCPi.TRIGger.OUTPut.FUNcTion</code> <code>app.SCPi.TRIGger.INPut.OUTPut.FUNcTion = "ESWP"</code>
<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 }

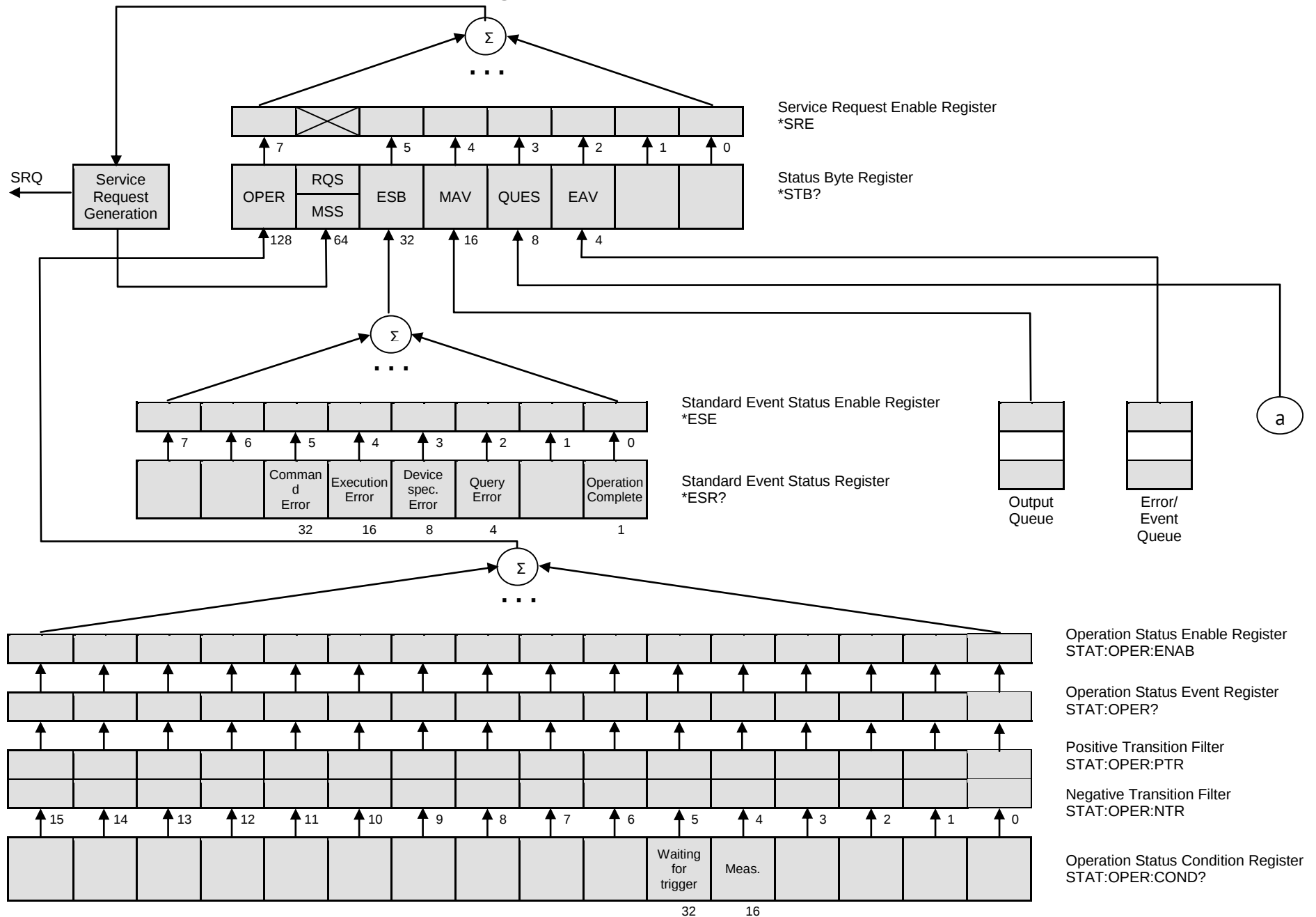
SCPI.TRIGger.OUTPUT.POLarity

<i>Description</i>	Selects the polarity of the trigger output set by the SCPI.TRIGger.OUTPUT.FUNCTION command.
<i>Type</i>	String (read/write)
<i>Parameter</i>	"POSitive" : Positive polarity "NEGative" : Negative polarity
<i>Preset Value</i>	"NEG"
<i>Target</i>	Trigger output
<i>Syntax</i>	<i>Param</i> = app.SCPI.TRIGger.OUTPUT.POLarity app.SCPI.TRIGger.OUTPUT.POLarity = "NEG"
<i>Related Commands</i>	SCPI.TRIGger.OUTPUT.FUNCTION SCPI.TRIGger.OUTPUT.STATe
<i>Equivalent Softkeys</i>	Stimulus > Trigger > Trigger Output > Polarity

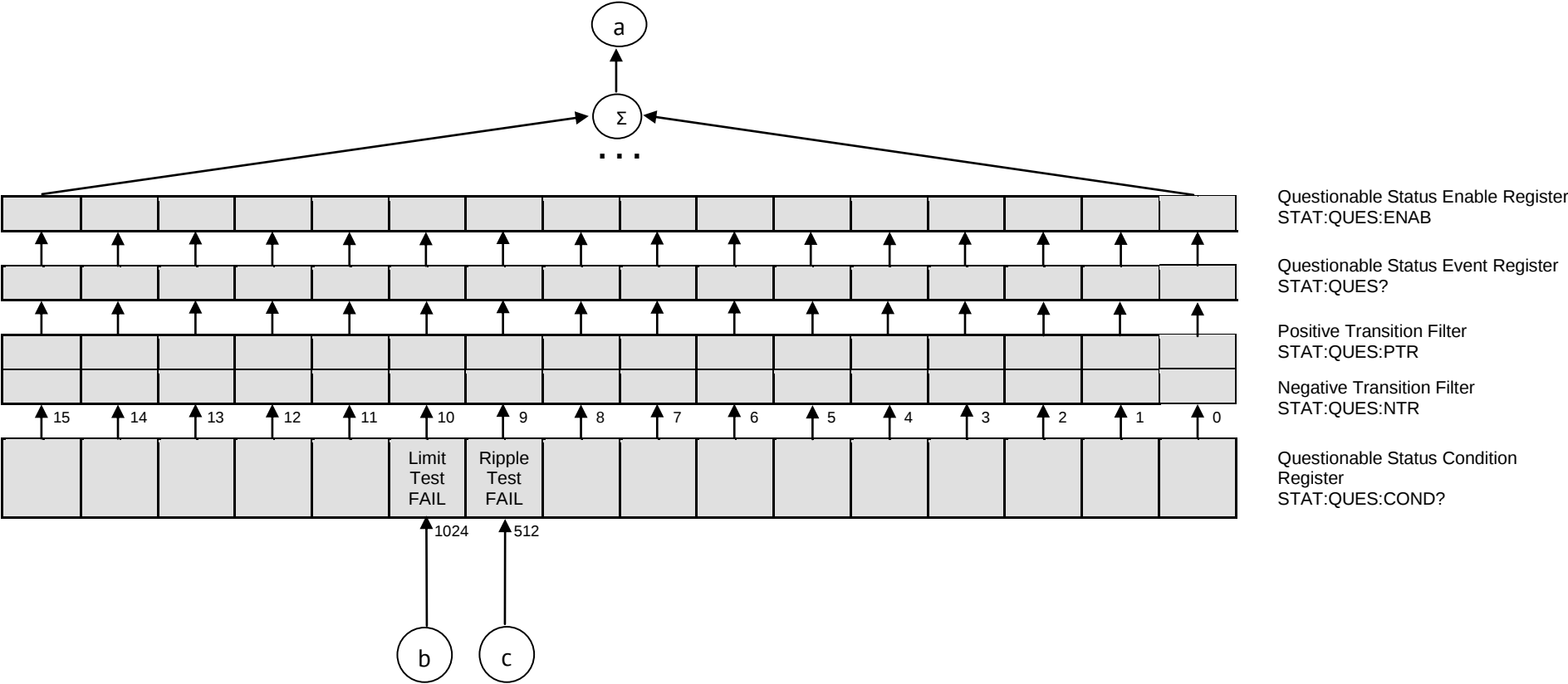
SCPI.TRIGger.OUTPUT.STATe

<i>Description</i>	Turns ON/OFF the state of the trigger output.
<i>Type</i>	Boolean (read/write)
<i>Parameter</i>	True: trigger output ON False: trigger output OFF
<i>Preset Value</i>	False
<i>Target</i>	Trigger output
<i>Syntax</i>	<i>Status</i> = app.SCPi.TRIGger.OUTPUT.STATe app.SCPi.TRIGger.OUTPUT.STATe = True
<i>Related Commands</i>	SCPI.TRIGger.OUTPUT.FUNCTION SCPI.TRIGger.OUTPUT.POLarity
<i>Equivalent Softkeys</i>	Stimulus > Trigger > Trigger Output > Trigger Output

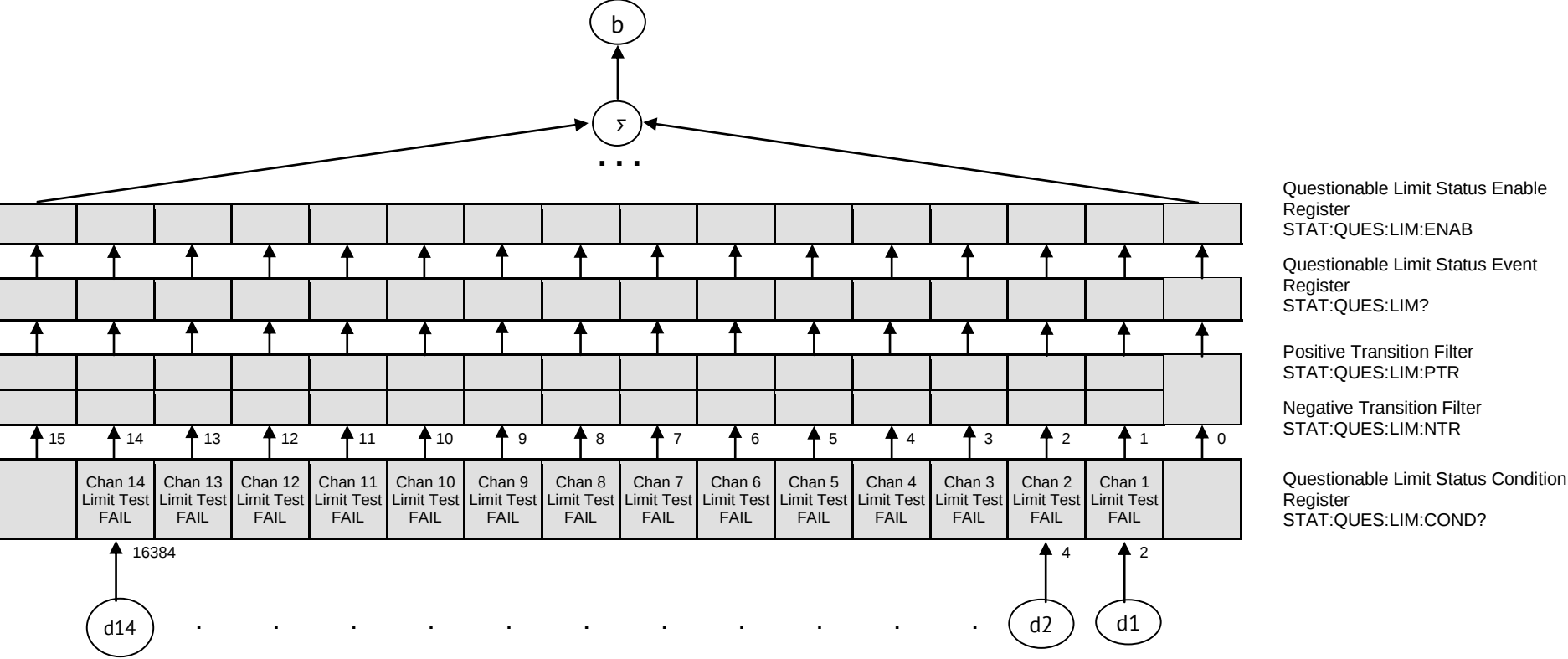
16 Appendix 1. IEE488.2 Status Reporting System



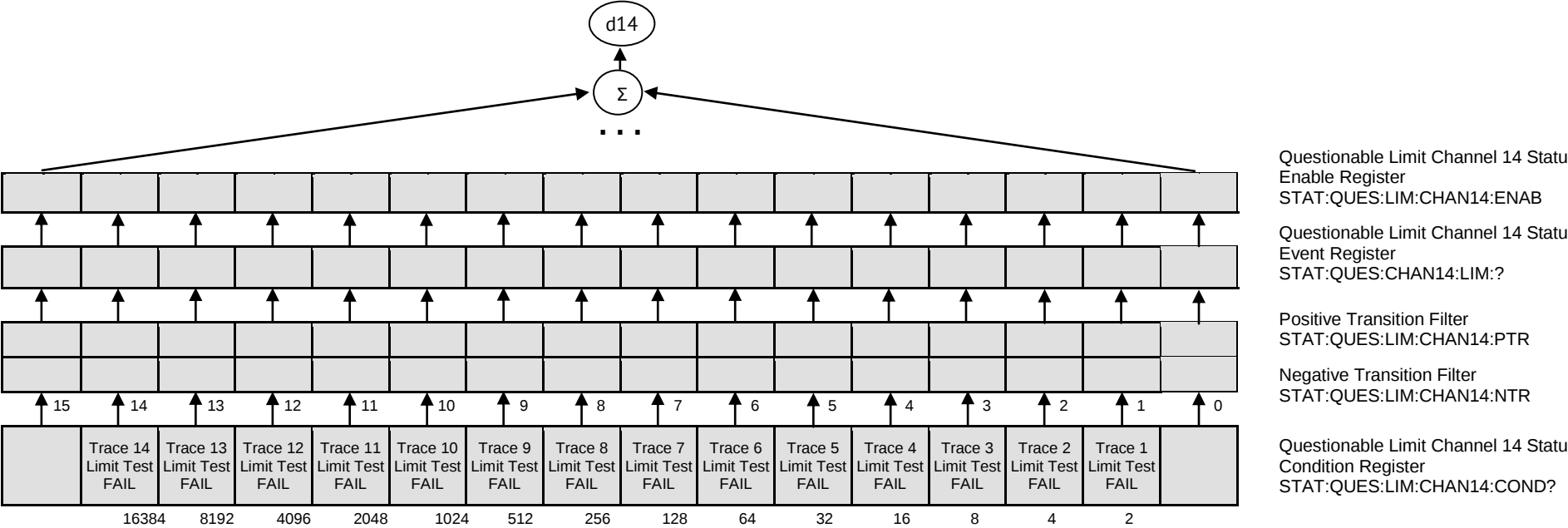
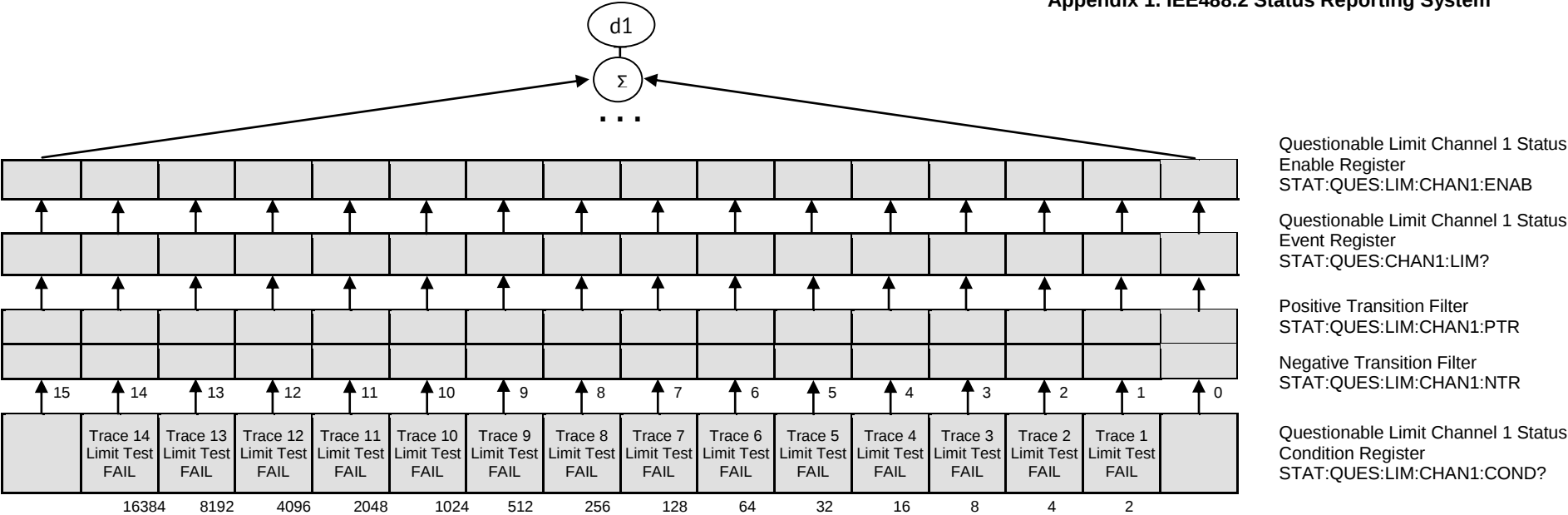
Appendix 1. IEE488.2 Status Reporting System



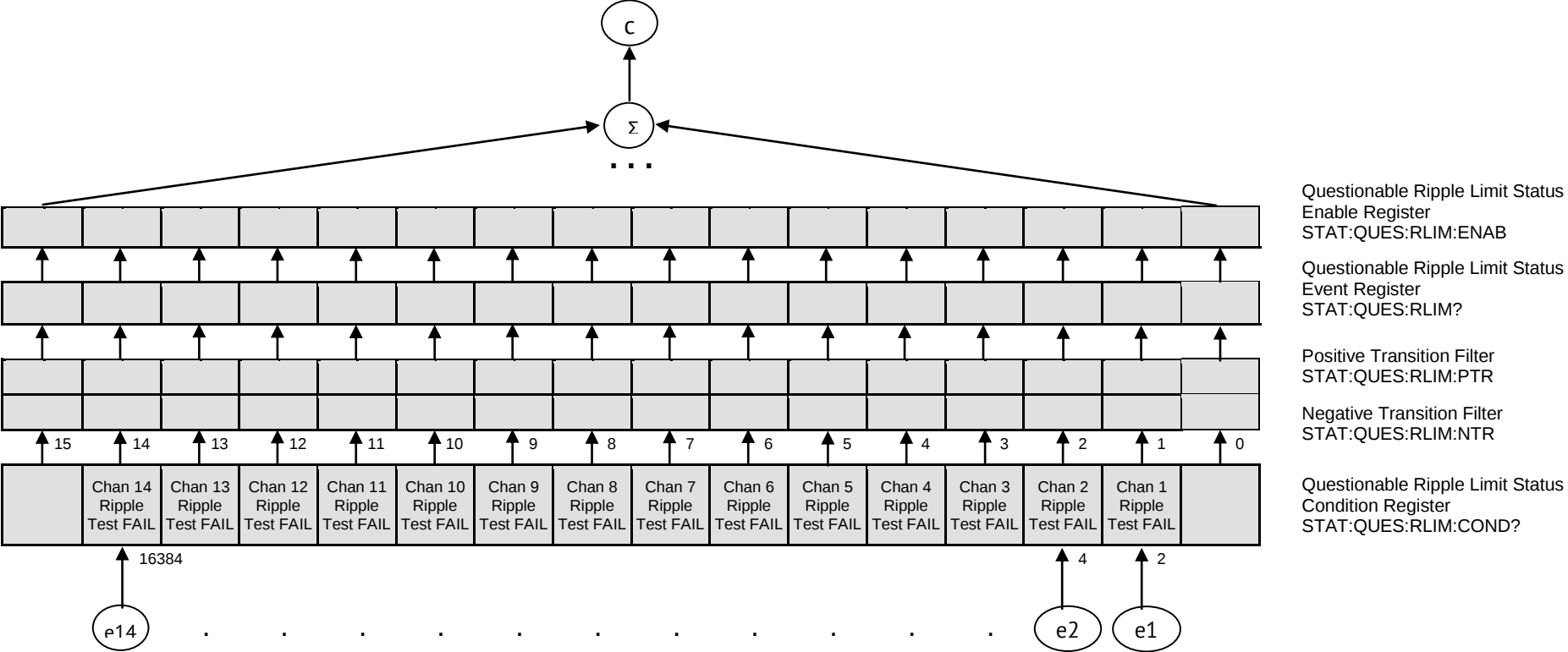
Appendix 1. IEE488.2 Status Reporting System



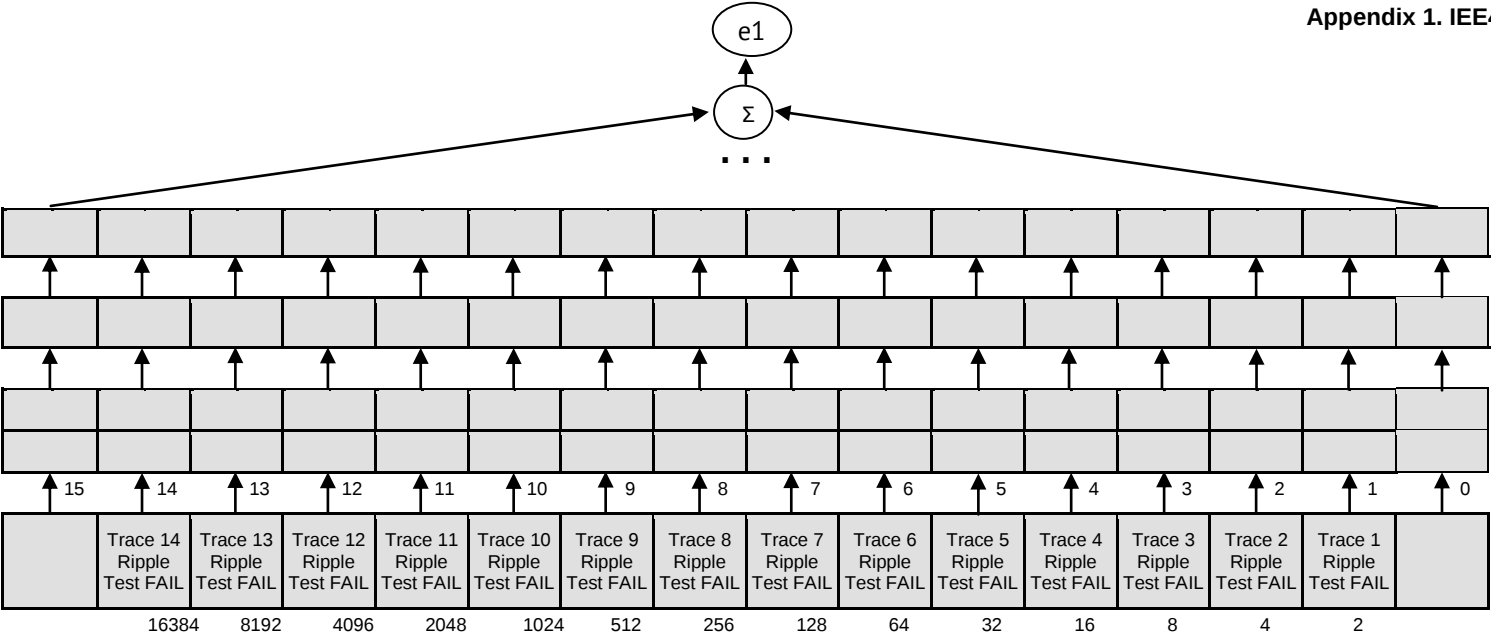
Appendix 1. IEE488.2 Status Reporting System



Appendix 1. IEE488.2 Status Reporting System



Appendix 1. IEE488.2 Status Reporting System



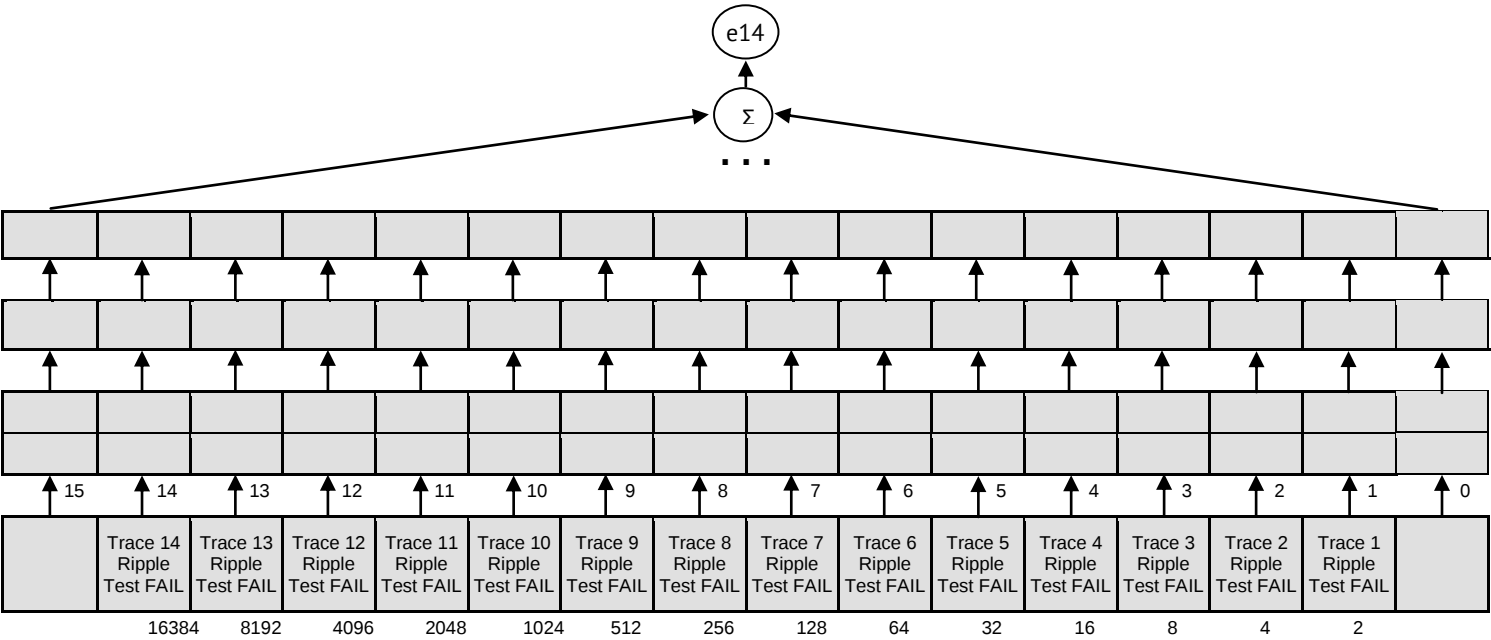
Questionable Ripple Limit Channel 1
Status Enable Register
STAT:QUES:RLIM:CHAN1:ENAB

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"