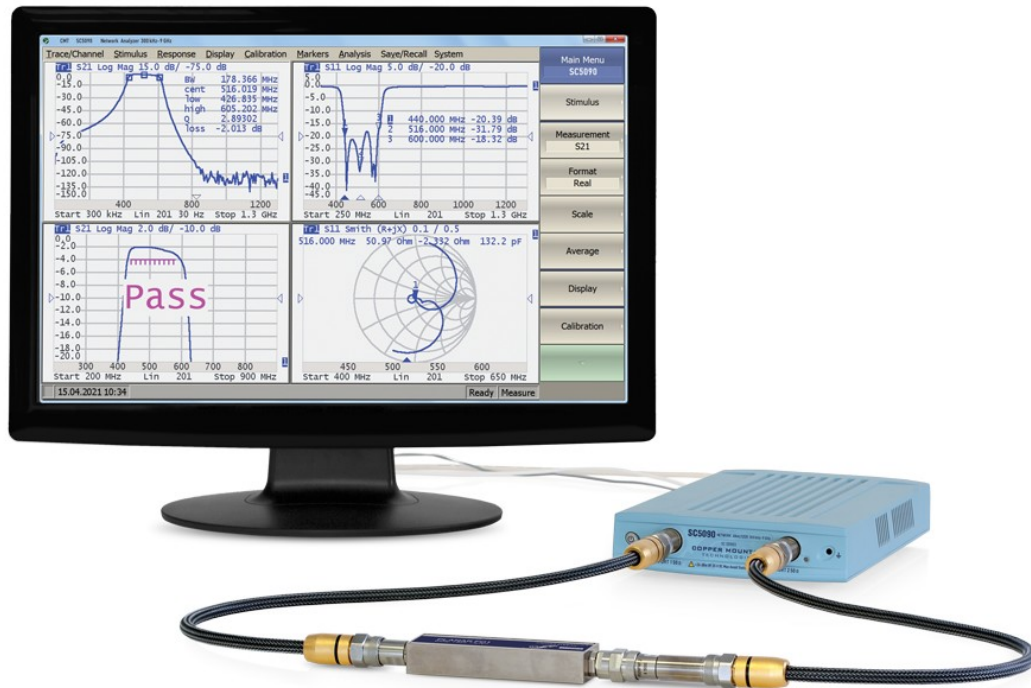




# Network Analyzers using S2VNA software

## Operating and Programming manuals



Revision 21.00 03.11.2021

U.S.: +1.317.222.5400  
Latin America: +1.9154.706.5920

Singapore: +65.63.23.6546  
EMAE: +44 75 03 69 21 13

## Contents

<b>Introduction .....</b>	<b>34</b>
<b>Scope of Manual .....</b>	<b>35</b>
<b>Safety Instructions .....</b>	<b>36</b>
<b>General Overview .....</b>	<b>37</b>
Specifications .....	39
Measurement Capabilities .....	39
Principles of Operation .....	48
The Principle of Measuring S-parameters .....	52
Summarized description of hierarchy .....	53
Internal Data Processing .....	56
<b>Instrument Series .....</b>	<b>59</b>
Compact Series .....	60
S Models, Front Panel .....	61
Rear Panel .....	65
M Models, Front Panel .....	72
Rear Panel .....	75
SC Models, Front Panel .....	78
Rear Panel .....	80
Cobalt Series, Front Panel .....	83
Rear Panel .....	88
Planar Series, Front Panel .....	94
Rear Panel .....	98
<b>Preparation for Use .....</b>	<b>101</b>
Software Installation .....	102
Registering COM Server .....	105
Command Line Parameters .....	106
<b>Getting Started .....</b>	<b>107</b>
Analyzer Preparation for Reflection Measurement .....	109

## Contents

Analyzer Presetting .....	110
Stimulus Setting .....	111
IF Bandwidth Setting .....	113
Number of Traces, Measured Parameter and Display Format Setting .....	115
Trace Scale Setting .....	118
Analyzer Calibration for Reflection Coefficient Measurement .....	119
SWR and Reflection Coefficient Phase Analysis Using Markers .....	121
<b>Program Interface .....</b>	<b>123</b>
Softkey Menu Bar .....	124
Menu Bar .....	126
Channel Window Layout and Functions .....	127
Channel Title Bar .....	129
Trace Status Field .....	130
Diagram .....	134
Trace Layout in the Channel Window .....	136
Markers .....	137
Channel Status Bar .....	138
Instrument Status Bar .....	142
<b>Setting Measurement Conditions .....</b>	<b>145</b>
Channel and Trace Setting .....	146
Channel Allocation .....	148
Number of Traces .....	150
Trace Allocation .....	152
Selection of Active Trace/Channel .....	157
Trace/Channel Window Maximizing .....	159
Stimulus Settings .....	161
Sweep Type .....	162
Sweep Range .....	163

## Contents

Number of Points .....	165
Stimulus Power .....	166
Power Slope Feature .....	168
CW Frequency .....	169
RF Out Function .....	170
Segment Table Editing .....	171
Measurement Delay .....	174
Reverse Sweep Mode .....	175
CW Time Sweep Mode .....	176
Trigger Settings .....	179
Trigger State Diagram .....	180
Trigger Source .....	185
Channel Initiation Mode .....	187
Trigger Scope .....	189
Averaging Trigger .....	190
External Trigger Settings .....	192
External Trigger Event .....	193
External Trigger Polarity .....	194
External Trigger Position .....	195
External Trigger Delay .....	197
Trigger Output .....	198
Enabling Trigger Output .....	199
Trigger Output Polarity .....	200
Trigger Output Function .....	201
Measurement Parameters Settings .....	204
S-Parameters .....	205
Absolute Measurements .....	207
Format Setting .....	209



## Contents

Rectangular Formats .....	210
Polar Format .....	213
Smith Chart Format .....	218
Scale Settings .....	227
Rectangular Scale .....	228
Circular Scale .....	230
Automatic Scaling .....	231
Reference Level Automatic Selection .....	232
Automatic Reference Level Tracking .....	233
Electrical Delay Setting .....	234
Phase Offset Setting .....	237
Measurement Optimization .....	238
IF Bandwidth Setting .....	239
Averaging Setting .....	240
Averaging Trigger .....	242
Smoothing Setting .....	244
Quick Settings Using a Mouse .....	245
Active Channel Selection .....	247
Active Trace Selection .....	248
Measured Parameter Setting .....	249
Display Format Setting .....	250
Trace Scale Setting .....	251
Reference Level Setting .....	252
Reference Level Position .....	253
Sweep Start Setting .....	254
Sweep Stop Setting .....	255
Sweep Center Setting .....	256
Sweep Span Setting .....	257

## Contents

Switching Between Start/Center and Stop/Span Modes .....	258
Start/Center Value Setting .....	259
Stop/Span Value Setting .....	260
Number of Points Setting .....	261
Sweep Type Setting .....	262
IF Bandwidth Setting .....	263
Power Level/CW Frequency Setting .....	264
Marker Stimulus Value Setting .....	265
<b>Calibration and Calibration Kit .....</b>	<b>266</b>
General Information .....	267
Basic Calibration Guidelines .....	268
Measurement Errors .....	271
Systematic Errors .....	272
Error Model .....	278
One-Port Error Model .....	279
Two-Port Error Model .....	280
Analyzer Test Port Definition .....	282
Calibration Steps .....	284
Calibration Standards and Calibration Kits .....	285
Types of Calibration Standards .....	286
Gender of Calibration Standard .....	287
Calibration Kit Management .....	288
Calibration Kit Selection .....	289
Operations on Table of Calibration Kits .....	290
Calibration Standard Definition .....	294
Calibration Standard Model .....	302
Data-Based Calibration Standards .....	306
Scope of Calibration Standard Definition .....	309

## Contents

Classes of Calibration Standards .....	310
Strict Class Assignment Function .....	314
Group Assignment of Port Number Function .....	316
Subclasses of Calibration Standards .....	317
Calibration Methods and Procedures .....	318
Reflection Normalization .....	320
Transmission Normalization .....	323
Full One-Port Calibration .....	326
One-Path Two-Port Calibration .....	329
Full Two-Port Calibration .....	333
Sliding Load Calibration .....	338
Non-Insertable Device Measuring .....	341
Unknown Thru Requirements .....	345
Unknown Thru Calibration .....	347
Unknown Thru Addition .....	349
Adapter Removal/Insertion .....	352
Two-Port TRL Calibration .....	358
Multiline TRL Calibration .....	364
Bandsplit Calibration Using Subclasses .....	366
TRL Calibration Example Using Subclasses .....	367
Sliding Load Calibration Example Using Subclasses .....	368
Waveguide Calibration .....	369
Power Calibration .....	370
Scalar Mixer Calibration .....	375
Receiver Calibration .....	380
Vector Mixer Calibration .....	385
Automatic Calibration Module .....	389
Automatic Calibration Module Features .....	390

## Contents

Automatic Calibration Procedure .....	392
Settings Before Calibrating .....	392
One/Two-Port Calibration .....	394
User Characterization Procedure .....	396
Confidence Check Procedure .....	399
Erasing the User Characterization .....	400
Manual Switch Control .....	401
Error Correction Status .....	402
Error Correction Disabling .....	404
System Impedance Z0 .....	405
Calibration Trigger Source .....	407
<b>Measurement Data Analysis .....</b>	<b>408</b>
Markers .....	409
Reference Marker Feature .....	414
Marker Properties .....	416
Marker Coupling Feature .....	417
Marker Table .....	418
Marker Value Indication Capacity .....	420
Multi Marker Data Display .....	421
Marker Data Arrangement .....	422
Marker Data Alignment .....	423
Memory Trace Value Display .....	424
Marker Discrete Mode .....	425
Marker Position Search Functions .....	426
Maximum and Minimum Search Functions .....	427
Search for Peak .....	429
Search for Target Level .....	432
Search Tracking .....	435

## Contents

Search Range .....	436
Marker Math Functions .....	438
Trace Statistics .....	439
Bandwidth Search .....	441
Flatness .....	444
RF Filter Statistics .....	446
Marker Functions .....	448
Memory Trace Function .....	450
Memory FIFO .....	453
Mathematical Operations .....	456
Trace Hold .....	458
Fixture Simulation .....	459
Port Extension .....	463
Automatic Port Extension .....	466
Port Reference Impedance (Z) Conversion .....	469
De-embedding .....	472
Embedding .....	474
Time Domain Transformation .....	476
Cable Correction Function .....	485
Time Domain Gating .....	489
S-Parameter Conversion .....	494
General S-Parameter Conversion .....	497
Limit Test .....	499
Ripple Limit Test .....	505
Peak Limits Test .....	510
<b>Special Measurement Modes .....</b>	<b>516</b>
Mixer Measurements .....	517
Frequency Offset Mode .....	519

## Contents

Automatic Adjustment of Offset Frequency .....	524
Frequency Extension System .....	529
FET1854 Frequency Extension Module .....	532
Frequency Extension Module FEV .....	534
Selection of Modules in Software .....	537
Configuring Modules Parameters .....	540
Custom Frequency Extender Setup .....	543
DC Measurement .....	547
Direct Receiver Access .....	552
<b>State Saving and Data Output .....</b>	<b>559</b>
Analyzer State .....	560
Channel State .....	564
Calibration Saving/Recalling .....	567
Session Resuming .....	568
Trace Data CSV File .....	569
Trace Data Touchstone File .....	573
<b>System Settings .....</b>	<b>579</b>
Analyzer Presetting .....	579
Graph Printing .....	580
Reference Frequency Oscillator Selection .....	583
System Correction Setting .....	584
Power Trip Function .....	585
Network Settings .....	586
Power Meter Settings .....	589
Beeper Settings .....	593
Analyzer Model .....	594
Analyzer Serial Number .....	594
Security Level .....	595

## Contents

Language .....	597
Create Localize Language File .....	598
Screen Update Setting .....	599
User Interface Setting .....	600
Full Screen .....	601
Font Size .....	602
Trace and Grid Styles .....	604
Color .....	606
Invert Color of Diagram .....	608
Hide/Show Menu Bar .....	610
Hide/Show Horizontal Graticule Label .....	611
Set Vertical Graticule Label .....	612
Hide/Show Sweep Mark .....	614
Hide/Show Date and Time .....	615
Hide/Show Cycle Time .....	616
Interface Presetting .....	617
Save/Load Display Setting .....	618
Demo Mode .....	619
Plugins .....	620
About .....	621
<b>Programming .....</b>	<b>622</b>
Connection Setup .....	623
Analyzer Setting .....	624
Client Setting .....	625
VISA Library .....	626
Network and Local Configuration .....	627
Connecting Multiple Analyzers to Single Computer .....	629
Differences in Use of HiSLIP and Socket Protocols .....	631

## Contents

Terminal Character in Messages to Analyzer .....	632
Terminal Character in Analyzer Responses .....	633
Interrupted Error .....	634
IEEE488.2 Status Reporting System .....	635
Transfer of Binary Data .....	636
SCPI Overview .....	637
Messages .....	638
Command Tree .....	639
Subsystems .....	641
Optional Subsystems .....	642
Long and Short Formats .....	643
Case Sensitivity .....	644
Parameters .....	645
Numeric Values .....	646
Multiplier Prefixes .....	647
Notations .....	648
Booleans .....	649
Character Data .....	650
String Parameters .....	651
Numeric Lists .....	652
Query Commands .....	653
Numeric Suffixes .....	654
Compound Commands .....	655
IEEE488.2 Common Commands Overview .....	656
COM/DOM Overview .....	657
Automation Server .....	658
Registering COM Server .....	659
Automation Controllers .....	660



## Contents

Local and Remote Server .....	661
DCOM Setup .....	663
Instrument Setup .....	663
Remote Computer Setup .....	664
Structure of COM Objects .....	665
Accessing the Application Object .....	667
Object Methods .....	670
Object Properties .....	671
Error Handling .....	672
COM Automation Data Types .....	674
Measurement Data Arrays .....	675
Internal Data Arrays .....	676
Command Reference .....	683
SCPI Command Tree .....	685
IEEE488.2 Common Commands .....	686
*CLS .....	687
*ESE .....	689
*ESR? .....	690
*IDN? .....	691
*OPC .....	692
*OPC? .....	693
*RST .....	695
*SRE .....	696
*STB? .....	697
*TRG .....	698
*TST? .....	700
*WAI .....	701
ABOR .....	703

## Contents

CALCulate .....	704
CALC:CONV .....	714
CALC:CONV:FUNC .....	716
CALC:CORR:EDEL:DIST .....	718
CALC:CORR:EDEL:DIST:UNIT .....	720
CALC:CORR:EDEL:MED .....	722
CALC:CORR:EDEL:RVEL .....	724
CALC:CORR:EDEL:TIME .....	726
CALC:CORR:EDEL:WAV:CUT .....	728
CALC:CORR:OFFS:PHAS .....	730
CALC:CORR:STAT? .....	732
CALC:DATA:FDAT .....	734
CALC:DATA:FMEM .....	737
CALC:DATA:SDAT .....	740
CALC:DATA:SMEM .....	742
CALC:DATA:XAX? .....	744
CALC:FILT:TIME .....	746
CALC:FILT:TIME:CENT .....	748
CALC:FILT:TIME:SHAP .....	750
CALC:FILT:TIME:SPAN .....	752
CALC:FILT:TIME:STAR .....	754
CALC:FILT:TIME:STAT .....	756
CALC:FILT:TIME:STOP .....	758
CALC:FORM .....	760
CALC:FSIM:SEND:DEEM:STAT .....	763
CALC:FSIM:SEND:DEEM:PORT:STAT .....	765
CALC:FSIM:SEND:DEEM:PORT:USER:FIL .....	767
CALC:FSIM:SEND:PMC:STAT .....	769

## Contents

CALC:FSIM:SEND:PMC:PORT:STAT .....	771
CALC:FSIM:SEND:PMC:PORT:USER:FIL .....	773
CALC:FSIM:SEND:ZCON:PORT:Z0 .....	775
CALC:FSIM:SEND:ZCON:PORT:Z0:REAL .....	777
CALC:FSIM:SEND:ZCON:PORT:Z0:IMAG .....	779
CALC:FSIM:SEND:ZCON:STAT .....	781
CALC:FSIM:SEND:ZCON:THE .....	783
CALC:FSIM:STAT .....	785
CALC:FUNC:DATA? .....	787
CALC:FUNC:DOM .....	789
CALC:FUNC:DOM:COUP .....	792
CALC:FUNC:DOM:STAR .....	794
CALC:FUNC:DOM:STOP .....	796
CALC:FUNC:EXEC .....	798
CALC:FUNC:PEXC .....	800
CALC:FUNC:POIN? .....	802
CALC:FUNC:PPOL .....	804
CALC:FUNC:TARG .....	806
CALC:FUNC:TTR .....	808
CALC:FUNC:TYPE .....	810
CALC:HOLD:TYPE .....	813
CALC:HOLD:CLE .....	815
CALC:LIM .....	816
CALC:LIM:DATA .....	818
CALC:LIM:DISP .....	821
CALC:LIM:FAIL? .....	823
CALC:LIM:OFFS:AMPL .....	825
CALC:LIM:OFFS:MARK .....	827

## Contents

CALC:LIM:OFFS:STIM .....	829
CALC:LIM:REP:ALL? .....	831
CALC:LIM:REP:POIN? .....	833
CALC:LIM:REP? .....	835
CALC:MARK .....	837
CALC:MARK:ACT .....	839
CALC:MARK:BWID .....	841
CALC:MARK:BWID:DATA? .....	843
CALC:MARK:BWID:REF .....	845
CALC:MARK:BWID:THR .....	847
CALC:MARK:BWID:TYPE .....	849
CALC:MARK:COUN .....	851
CALC:MARK:COUP .....	853
CALC:MARK:DATA? .....	855
CALC:MARK:DISC .....	857
CALC:MARK:FUNC:DOM .....	859
CALC:MARK:FUNC:DOM:COUP .....	862
CALC:MARK:FUNC:DOM:STAR .....	864
CALC:MARK:FUNC:DOM:STOP .....	866
CALC:MARK:FUNC:EXEC .....	868
CALC:MARK:FUNC:PEXC .....	870
CALC:MARK:FUNC:PPOL .....	872
CALC:MARK:FUNC:TARG .....	875
CALC:MARK:FUNC:TRAC .....	877
CALC:MARK:FUNC:TTR .....	879
CALC:MARK:FUNC:TYPE .....	881
CALC:MARK:MATH:FLAT:DATA? .....	884
CALC:MARK:MATH:FLAT:STAT .....	886

## Contents

CALC:MARK:MATH:FLAT:DOM:STAR .....	888
CALC:MARK:MATH:FLAT:DOM:STOP .....	890
CALC:MARK:REF .....	892
CALC:MARK:SET .....	894
CALC:MARK:X .....	896
CALC:MARK:Y? .....	898
CALC:MATH:FUNC .....	900
CALC:MATH:MEM .....	902
CALC:MST .....	904
CALC:MST:DATA? .....	906
CALC:MST:DOM .....	908
CALC:MST:DOM:STAR .....	910
CALC:MST:DOM:STOP .....	912
CALC:PAR:COUN .....	914
CALC:PAR:DEF .....	916
CALC:PAR:SEL .....	918
CALC:PAR:SPOR .....	920
CALC:RLIM .....	922
CALC:RLIM:DATA .....	924
CALC:RLIM:DISP:LINE .....	926
CALC:RLIM:DISP:SEL .....	928
CALC:RLIM:DISP:VAL .....	930
CALC:RLIM:FAIL? .....	932
CALC:RLIM:REP? .....	934
CALC:SMO .....	936
CALC:SMO:APER .....	938
CALC:TRAN:TIME .....	940
CALC:TRAN:TIME:CENT .....	942

## Contents

CALC:TRAN:TIME:DC:VAL .....	944
CALC:TRAN:TIME:EXTR:DC .....	946
CALC:TRAN:TIME:IMP:WIDT .....	948
CALC:TRAN:TIME:KBES .....	950
CALC:TRAN:TIME:LPFR .....	952
CALC:TRAN:TIME:REFL:TYPE .....	954
CALC:TRAN:TIME:SPAN .....	956
CALC:TRAN:TIME:STAR .....	958
CALC:TRAN:TIME:STOP .....	960
CALC:TRAN:TIME:STAT .....	962
CALC:TRAN:TIME:STEP:RTIM .....	964
CALC:TRAN:TIME:STIM .....	966
CALC:TRAN:TIME:UNIT .....	968
DISPlay .....	970
DISP:COL:BACK .....	974
DISP:COL:GRAT .....	976
DISP:COL:RES .....	978
DISP:COL:TRAC:DATA .....	979
DISP:COL:TRAC:MEM .....	981
DISP:ENAB .....	983
DISP:FONT:SIZE .....	985
DISP:FSIG .....	986
DISP:GLAB .....	988
DISP:IMAG .....	990
DISP:HIDE .....	992
DISP:MARK:TABL .....	993
DISP:MAX .....	994
DISP:PART:FONT:SIZE .....	996

## Contents

DISP:PART:FONT:SIZE:STAT .....	998
DISP:PART:VIS .....	999
DISP:POS .....	1001
DISP:SHOW .....	1003
DISP:SPL .....	1004
DISP:UPD .....	1006
DISP:WIND:ACT .....	1007
DISP:WIND:ANN:MARK:ALIG .....	1009
DISP:WIND:ANN:MARK:SING .....	1011
DISP:WIND:MAX .....	1013
DISP:WIND:SPL .....	1015
DISP:WIND:TITL .....	1017
DISP:WIND:TITL:DATA .....	1019
DISP:WIND:TRAC:ANN:MARK:POS:X .....	1021
DISP:WIND:TRAC:ANN:MARK:POS:Y .....	1023
DISP:WIND:TRAC:MEM .....	1025
DISP:WIND:TRAC:STAT .....	1028
DISP:WIND:TRAC:Y:AUTO .....	1030
DISP:WIND:TRAC:Y:PDIV .....	1032
DISP:WIND:TRAC:Y:RLEV .....	1034
DISP:WIND:TRAC:Y:RLEV:AUTO .....	1036
DISP:WIND:TRAC:Y:RPOS .....	1037
DISP:WIND:X:SPAC .....	1039
DISP:WIND:Y:DIV .....	1041
FORMat .....	1043
FORM:BORD .....	1044
FORM:DATA .....	1046
FORM:PUSH .....	1048

## Contents

FORM:POP .....	1050
HCOPy .....	1051
HCOP .....	1052
HCOP:ABOR .....	1053
HCOP:DATE:STAM .....	1054
HCOP:IMAG .....	1056
HCOP:PAIN .....	1058
INTiate .....	1060
INIT .....	1061
INIT:CONT .....	1063
INIT:CONT:ALL .....	1065
MMEMory .....	1068
MMEM:CAT? .....	1072
MMEM:COPY .....	1074
MMEM:DEL .....	1075
MMEM:LOAD .....	1076
MMEM:LOAD:CHAN .....	1077
MMEM:LOAD:CHAN:CAL .....	1079
MMEM:LOAD:CKIT .....	1080
MMEM:LOAD:LIM .....	1082
MMEM:LOAD:PLOS .....	1083
MMEM:LOAD:RLIM .....	1085
MMEM:LOAD:SEGM .....	1086
MMEM:LOAD:SNP .....	1087
MMEM:LOAD:SNP:FREQ .....	1088
MMEM:LOAD:SNP:TRAC:MEM .....	1089
MMEM:MDIR .....	1091
MMEM:STOR .....	1092



## Contents

MMEM:STOR:CHAN .....	1093
MMEM:STOR:CHAN:CAL .....	1095
MMEM:STOR:CHAN:CLE .....	1096
MMEM:STOR:CKIT .....	1097
MMEM:STOR:FDAT .....	1098
MMEM:STOR:FDAT:SCOP .....	1099
MMEM:STOR:FDAT:FORM .....	1100
MMEM:STOR:FDAT:COMM .....	1102
MMEM:STOR:FDAT:STIM .....	1103
MMEM:STOR:FDAT:SEP .....	1104
MMEM:STOR:IMAG .....	1105
MMEM:STOR:LIM .....	1106
MMEM:STOR:PLOS .....	1107
MMEM:STOR:RLIM .....	1108
MMEM:STOR:SEGM .....	1109
MMEM:STOR:SNP .....	1110
MMEM:STOR:SNP:FORM .....	1111
MMEM:STOR:SNP:SEP .....	1113
MMEM:STOR:SNP:TRAC:TRAN .....	1115
MMEM:STOR:SNP:TYPE? .....	1116
MMEM:STOR:SNP:TYPE:S1P .....	1117
MMEM:STOR:SNP:TYPE:S2P .....	1119
MMEM:STOR:STYP .....	1121
MMEM:TRAN? .....	1123
OUTP .....	1124
SENSe .....	1126
SENS:AVER .....	1142
SENS:AVER:CLE .....	1144

## Contents

SENS:AVER:COUN .....	1145
SENS:BAND .....	1147
SENS:BWID .....	1149
SENS:CORR:CLE .....	1151
SENS:CORR:COEF .....	1152
SENS:CORR:COEF:METH:ERES .....	1154
SENS:CORR:COEF:METH:OPEN .....	1156
SENS:CORR:COEF:METH:SHOR .....	1158
SENS:CORR:COEF:METH:SOLT1 .....	1160
SENS:CORR:COEF:METH:SOLT2 .....	1162
SENS:CORR:COEF:METH:THRU .....	1164
SENS:CORR:COEF:SAVE .....	1166
SENS:CORR:COLL:ADAP:DEL .....	1168
SENS:CORR:COLL:ADAP:LENG .....	1170
SENS:CORR:COLL:ADAP:UNIT .....	1172
SENS:CORR:COLL:ADAP:MED .....	1174
SENS:CORR:COLL:ADAP:PERM .....	1176
SENS:CORR:COLL:ADAP:WAV:CUT .....	1178
SENS:CORR:COLL:METH:ADAP:REM .....	1179
SENS:CORR:COLL:CKIT .....	1180
SENS:CORR:COLL:CKIT:DESC .....	1181
SENS:CORR:COLL:CKIT:LAB .....	1182
SENS:CORR:COLL:CKIT:ORD:LOAD .....	1184
SENS:CORR:COLL:CKIT:ORD:OPEN .....	1186
SENS:CORR:COLL:CKIT:ORD:SEL .....	1188
SENS:CORR:COLL:CKIT:ORD:SHOR .....	1190
SENS:CORR:COLL:CKIT:ORD:THRU .....	1192
SENS:CORR:COLL:CKIT:ORD:TRLL .....	1194

## Contents

SENS:CORR:COLL:CKIT:ORD:TRLT .....	1196
SENS:CORR:COLL:CKIT:ORD:TRLR .....	1198
SENS:CORR:COLL:CKIT:RES .....	1200
SENS:CORR:COLL:CKIT:STAN:ARB .....	1201
SENS:CORR:COLL:CKIT:STAN:C0 .....	1203
SENS:CORR:COLL:CKIT:STAN:C1 .....	1205
SENS:CORR:COLL:CKIT:STAN:C2 .....	1207
SENS:CORR:COLL:CKIT:STAN:C3 .....	1209
SENS:CORR:COLL:CKIT:STAN:COUN? .....	1211
SENS:CORR:COLL:CKIT:STAN:DATA .....	1213
SENS:CORR:COLL:CKIT:STAN:DEL .....	1215
SENS:CORR:COLL:CKIT:STAN:FMAX .....	1217
SENS:CORR:COLL:CKIT:STAN:FMIN .....	1219
SENS:CORR:COLL:CKIT:STAN:INS .....	1221
SENS:CORR:COLL:CKIT:STAN:L0 .....	1222
SENS:CORR:COLL:CKIT:STAN:L1 .....	1224
SENS:CORR:COLL:CKIT:STAN:L2 .....	1226
SENS:CORR:COLL:CKIT:STAN:L3 .....	1228
SENS:CORR:COLL:CKIT:STAN:LAB .....	1230
SENS:CORR:COLL:CKIT:STAN:LOSS .....	1232
SENS:CORR:COLL:CKIT:STAN:REM .....	1235
SENS:CORR:COLL:CKIT:STAN:TYPE .....	1236
SENS:CORR:COLL:CKIT:STAN:Z0 .....	1238
SENS:CORR:COLL:CLE .....	1240
SENS:CORR:COLL:DATA:ISOL .....	1242
SENS:CORR:COLL:DATA:LOAD .....	1244
SENS:CORR:COLL:DATA:OPEN .....	1246
SENS:CORR:COLL:DATA:SHOR .....	1249

## Contents

SENS:CORR:COLL:DATA:THRU:MATC .....	1251
SENS:CORR:COLL:DATA:THRU:TRAN .....	1253
SENS:CORR:COLL:ECAL:CCH .....	1255
SENS:CORR:COLL:ECAL:ERES .....	1256
SENS:CORR:COLL:ECAL:INF? .....	1258
SENS:CORR:COLL:ECAL:ORI:EXEC .....	1260
SENS:CORR:COLL:ECAL:ORI:STAT .....	1261
SENS:CORR:COLL:ECAL:PATH .....	1263
SENS:CORR:COLL:ECAL:SOLT1 .....	1265
SENS:CORR:COLL:ECAL:SOLT2 .....	1266
SENS:CORR:COLL:ECAL:THER:COMP .....	1268
SENS:CORR:COLL:ECAL:UCH .....	1270
SENS:CORR:COLL:ECAL:UTHR:STAT .....	1272
SENS:CORR:COLL:ISOL .....	1274
SENS:CORR:COLL:LOAD .....	1276
SENS:CORR:COLL:OPEN .....	1278
SENS:CORR:COLL:SHOR .....	1280
SENS:CORR:COLL:THRU .....	1282
SENS:CORR:COLL:TRLL .....	1284
SENS:CORR:COLL:TRLT .....	1286
SENS:CORR:COLL:TRLR .....	1288
SENS:CORR:COLL:SUBC .....	1290
SENS:CORR:COLL:METH:ERES .....	1292
SENS:CORR:COLL:METH:OPEN .....	1293
SENS:CORR:COLL:METH:SHOR .....	1296
SENS:CORR:COLL:METH:SOLT1 .....	1298
SENS:CORR:COLL:METH:SOLT2 .....	1300
SENS:CORR:COLL:METH:THRU .....	1302

## Contents

SENS:CORR:COLL:METH:TRL:MULT .....	1304
SENS:CORR:COLL:METH:TRL2 .....	1306
SENS:CORR:COLL:METH:TYPE? .....	1308
SENS:CORR:COLL:SAVE .....	1310
SENS:CORR:COLL:THRU:ADD:DEL .....	1312
SENS:CORR:COLL:THRU:ADD:LENG .....	1314
SENS:CORR:COLL:THRU:ADD:UNIT .....	1316
SENS:CORR:COLL:THRU:ADD:MED .....	1318
SENS:CORR:COLL:THRU:ADD:PERM .....	1320
SENS:CORR:COLL:THRU:ADD:WAV:CUT .....	1322
SENS:CORR:COLL:THRU:ADD:FULL2:COMP .....	1323
SENS:CORR:EXT .....	1325
SENS:CORR:EXT:AUTO:CONF .....	1327
SENS:CORR:EXT:AUTO:DCOF .....	1329
SENS:CORR:EXT:AUTO:LOSS .....	1331
SENS:CORR:EXT:AUTO:MEAS .....	1333
SENS:CORR:EXT:AUTO:PORT .....	1334
SENS:CORR:EXT:AUTO:RES .....	1336
SENS:CORR:EXT:AUTO:STAR .....	1337
SENS:CORR:EXT:AUTO:STOP .....	1339
SENS:CORR:EXT:PORT:FREQ .....	1341
SENS:CORR:EXT:PORT:INCL .....	1343
SENS:CORR:EXT:PORT:LDC .....	1345
SENS:CORR:EXT:PORT:LOSS .....	1347
SENS:CORR:EXT:PORT:TIME .....	1349
SENS:CORR:INF? .....	1351
SENS:CORR:IMP .....	1353
SENS:CORR:IMP:SEL:AUTO .....	1355

## Contents

SENS:CORR:OFFS:CLE .....	1356
SENS:CORR:OFFS:COLL:CLE .....	1357
SENS:CORR:OFFS:COLL:DIR .....	1358
SENS:CORR:OFFS:COLL:ECAL .....	1360
SENS:CORR:OFFS:COLL:LOAD .....	1362
SENS:CORR:OFFS:COLL:METH:SMIX2 .....	1364
SENS:CORR:OFFS:COLL:OPEN .....	1366
SENS:CORR:OFFS:COLL:PMET .....	1368
SENS:CORR:OFFS:COLL:SHOR .....	1370
SENS:CORR:OFFS:COLL:THRU .....	1372
SENS:CORR:OFFS:COLL:SAVE .....	1374
SENS:CORR:PORT:IMP .....	1376
SENS:CORR:REC .....	1378
SENS:CORR:REC:COLL:ACQ .....	1380
SENS:CORR:REC:COLL:RCH:ACQ .....	1382
SENS:CORR:REC:COLL:TCH:ACQ .....	1383
SENS:CORR:REC:OFFS:AMPL .....	1385
SENS:CORR:STAT .....	1387
SENS:CORR:TRAN:TIME:FREQ .....	1389
SENS:CORR:TRAN:TIME:LOSS .....	1391
SENS:CORR:TRAN:TIME:RVEL .....	1393
SENS:CORR:TRAN:TIME:STAT .....	1395
SENS:CORR:TRIG:FREE .....	1397
SENS:CORR:TYPE? .....	1399
SENS:CORR:VMC:COLL:ECAL:SAVE .....	1401
SENS:CORR:VMC:COLL:PORT .....	1402
SENS:CORR:VMC:COLL:LO:FREQ .....	1403
SENS:CORR:VMC:COLL:IF:SEL .....	1405

## Contents

SENS:CORR:VMC:COLL:LOAD .....	1407
SENS:CORR:VMC:COLL:OPEN .....	1408
SENS:CORR:VMC:COLL:SHOR .....	1409
SENS:CORR:VMC:COLL:OPT .....	1410
SENS:CORR:VMC:COLL:SAVE .....	1412
SENS:DATA:CORR? .....	1413
SENS:DATA:RAWD? .....	1415
SENS:FREQ .....	1417
SENS:FREQ:DATA? .....	1419
SENS:FREQ:CENT .....	1421
SENS:FREQ:SPAN .....	1423
SENS:FREQ:STAR .....	1425
SENS:FREQ:STOP .....	1427
SENS:OFFS .....	1429
SENS:OFFS:ADJ .....	1431
SENS:OFFS:ADJ:CONT:PER .....	1433
SENS:OFFS:ADJ:EXEC .....	1435
SENS:OFFS:ADJ:PATH .....	1436
SENS:OFFS:ADJ:PORT .....	1437
SENS:OFFS:ADJ:VAL .....	1438
SENS:OFFS:PORT:DATA? .....	1440
SENS:OFFS:PORT:DIV .....	1442
SENS:OFFS:PORT:MULT .....	1444
SENS:OFFS:PORT:OFFS .....	1446
SENS:OFFS:PORT:STAR .....	1448
SENS:OFFS:PORT:STOP .....	1450
SENS:OFFS:REC:DATA? .....	1452
SENS:OFFS:REC:DIV .....	1454

## Contents

SENS:OFFS:REC:MULT .....	1456
SENS:OFFS:REC:OFFS .....	1458
SENS:OFFS:REC:STAR .....	1460
SENS:OFFS:REC:STOP .....	1462
SENS:OFFS:SOUR:DATA? .....	1464
SENS:OFFS:SOUR:DIV .....	1466
SENS:OFFS:SOUR:MULT .....	1468
SENS:OFFS:SOUR:OFFS .....	1470
SENS:OFFS:SOUR:STAR .....	1472
SENS:OFFS:SOUR:STOP .....	1474
SENS:OFFS:TYPE .....	1476
SENS:ROSC:SOUR .....	1478
SENS:SEGM:DATA .....	1480
SENS:SWE:CW:TIME .....	1482
SENS:SWE:POIN .....	1484
SENS:SWE:POIN:TIME .....	1486
SENS:SWE:REV .....	1488
SENS:SWE:TYPE .....	1490
SENS:VOLT:DC:RANG:UPP .....	1492
SERVice .....	1494
SERV:CHAN:ACT? .....	1495
SERV:CHAN:COUN? .....	1496
SERV:CHAN:TRAC:ACT? .....	1497
SERV:CHAN:TRAC:COUN? .....	1499
SERV:CHAN:TRAC:MARK:ACT? .....	1500
SERV:PORT:COUN? .....	1502
SERV:SWE:FREQ:MAX? .....	1503
SERV:SWE:FREQ:MIN? .....	1504



## Contents

SERV:SWE:POIN? .....	1505
SERV:SWE:POW:MAX? .....	1506
SERV:SWE:POW:MIN? .....	1507
SOURce .....	1508
SOUR:POW .....	1510
SOUR:POW:CENT .....	1512
SOUR:POW:PORT .....	1514
SOUR:POW:PORT:CORR .....	1516
SOUR:POW:PORT:CORR:INT? .....	1518
SOUR:POW:PORT:CORR:COLL .....	1520
SOUR:POW:PORT:CORR:COLL:TABL:LOSS:DATA .....	1522
SOUR:POW:PORT:CORR:COLL:TABL:LOSS .....	1524
SOUR:POW:PORT:CORR:DATA .....	1526
SOUR:POW:PORT:COUP .....	1528
SOUR:POW:SLOP .....	1530
SOUR:POW:SLOP:STAT .....	1532
SOUR:POW:SPAN .....	1534
SOUR:POW:STAR .....	1536
SOUR:POW:STOP .....	1538
STATus .....	1540
STAT:OPER? .....	1543
STAT:OPER:COND? .....	1544
STAT:OPER:ENAB .....	1545
STAT:OPER:NTR .....	1547
STAT:OPER:PTR .....	1549
STAT:PRES .....	1551
STAT:QUES:COND? .....	1552
STAT:QUES:ENAB .....	1553

## Contents

STAT:QUES:LIM:CHAN:COND? .....	1555
STAT:QUES:LIM:CHAN:ENAB .....	1556
STAT:QUES:LIM:CHAN:NTR .....	1558
STAT:QUES:LIM:CHAN:PTR .....	1560
STAT:QUES:LIM:CHAN? .....	1562
STAT:QUES:LIM:COND? .....	1563
STAT:QUES:LIM:ENAB .....	1564
STAT:QUES:LIM:NTR .....	1566
STAT:QUES:LIM:PTR .....	1568
STAT:QUES:LIM? .....	1570
STAT:QUES:NTR .....	1571
STAT:QUES:PTR .....	1573
STAT:QUES:RLIM:CHAN:COND? .....	1575
STAT:QUES:RLIM:CHAN:ENAB .....	1576
STAT:QUES:RLIM:CHAN:NTR .....	1578
STAT:QUES:RLIM:CHAN:PTR .....	1580
STAT:QUES:RLIM:CHAN? .....	1582
STAT:QUES:RLIM:COND? .....	1583
STAT:QUES:RLIM:ENAB .....	1584
STAT:QUES:RLIM:NTR .....	1586
STAT:QUES:RLIM:PTR .....	1588
STAT:QUES:RLIM? .....	1590
STAT:QUES? .....	1591
SYSTem .....	1592
SYST:BEEP:COMP:IMM .....	1597
SYST:BEEP:COMP:STAT .....	1598
SYST:BEEP:WARN:IMM .....	1600
SYST:BEEP:WARN:STAT .....	1601

## Contents

SYST:CAP:IFBW:MAX? .....	1603
SYST:CAP:IFBW:MIN? .....	1604
SYST:CAP:CURR:CONS? .....	1605
SYST:CURR:CONS? .....	1606
SYST:COMM:ECAL:CHEC .....	1607
SYST:COMM:ECAL:DATA? .....	1608
SYST:COMM:ECAL:FREQ:DATA? .....	1609
SYST:COMM:ECAL:POIN? .....	1610
SYST:COMM:ECAL:IMP .....	1611
SYST:COMM:ECAL:READY? .....	1614
SYST:COMM:ECAL:TEMP:SENS? .....	1615
SYST:COMM:ECAL:THRU .....	1616
SYST:COMM:PSEN:NI568x:RES:NAME .....	1617
SYST:COMM:PSEN:READ? .....	1618
SYST:COMM:PSEN:TYPE .....	1619
SYST:COMM:PSEN:ZERO .....	1622
SYST:CONN:SER:NUMB .....	1623
SYST:CORR .....	1624
SYST:CYCL:TIME:MEAS? .....	1626
SYST:CYCL:TIME:METH .....	1628
SYST:CYCL:TIME:REST .....	1630
SYST:DATE .....	1631
SYST:ERR? .....	1633
SYST:FREQ:EXT:RFR:POW .....	1634
SYST:FREQ:EXT:RFP:PSL .....	1636
SYST:FREQ:EXT:LOP:POW .....	1638
SYST:FREQ:EXT:LOP:PSL .....	1640
SYST:FREQ:EXT:TYPE .....	1642

## Contents

SYST:FREQ:EXT:PORT:CONN? .....	1644
SYST:FREQ:EXT:PORT:SER? .....	1645
SYST:FREQ:EXT:PORT:TEMP:SENS? .....	1646
SYST:HIDE .....	1647
SYST:LOC .....	1648
SYST:PRES .....	1649
SYST:REC:DIR:ACC .....	1650
SYST:REC:OVER:POW .....	1651
SYST:READ? .....	1652
SYST:REM .....	1653
SYST:RWL .....	1654
SYST:SERV:PVER:INT .....	1655
SYST:SERV:PVER:LAST .....	1656
SYST:SERV:PVER:NEXT .....	1657
SYST:SHOW .....	1658
SYST:TEMP:SENS? .....	1659
SYST:TEST? .....	1661
SYST:TERM .....	1662
SYST:TIME .....	1663
TRIGger .....	1665
TRIG .....	1667
TRIG:AVER .....	1669
TRIG:EXT:DEL .....	1671
TRIG:EXT:SLOP .....	1673
TRIG:EXT:POS .....	1675
TRIG:OUTP:FUNC .....	1677
TRIG:OUTP:POL .....	1679
TRIG:OUTP:STAT .....	1681

## Contents

TRIG:POIN .....	1683
TRIG:SING .....	1685
TRIG:SCOP .....	1687
TRIG:SOUR .....	1689
TRIG:STAT? .....	1691
TRIG:WAIT .....	1692
Programming Tips .....	1694
Program Sweep Initiation and Waiting .....	1695
Using External Trigger .....	1697
Waiting for Calibration Commands .....	1698
VISA Timeout Considerations .....	1699
Receiving Data Arrays in Text Format .....	1700
Receiving Data Arrays Binary Format .....	1701
IEEE488.2 Status Reporting System .....	1703
Error Codes .....	1710
SCPI and COM Error Codes .....	1711
SCPI Error Codes .....	1714
Programming Examples .....	1715
<b>Maintenance and Storage .....</b>	<b>1722</b>
Maintenance Procedures .....	1723
Storage Instructions .....	1725
<b>Annex 1 — Default Settings Table .....</b>	<b>1726</b>
<b>Glossary .....</b>	<b>1732</b>
<b>Copyright .....</b>	<b>1734</b>

## **Introduction**

This manual contains design, specifications, functional overview, and detailed operation procedures for the Vector Network Analyzer, to ensure effective and safe use of its technical capabilities.

Maintenance and operation of the Analyzer should be performed by qualified engineers with basic experience in the operation of microwave circuits.

This Operating Manual corresponds to S2VNA software version 21.4.1

[Glossary](#) — The abbreviations which are used in this document.

### **Web Sites**

[Copper Mountain Technologies](#)

## Scope of Manual

This manual covers the two-port models of the Copper Mountain Technologies Network Analyzers controlled by the S2VNA software. The Analyzer models are listed below:

- M5045
- M5065
- M5090
- M5180
- S5045
- S5048
- S5065
- S5085
- S7530
- S5180
- S5243
- SC5065
- SC5090
- C1209
- C1220
- C2209
- C2220
- C4209
- C4220
- Planar 304/1
- Planar 804/1
- Planar 814/1

## Safety Instructions

It is highly recommended to follow all safety warnings and precautions provided in this document for operating, servicing, and repairing the Analyzer.

The Analyzer should be used only by skilled and thoroughly trained personnel with the required skills and knowledge of safety precautions.

The Analyzer complies with INSTALLATION CATEGORY II as well as POLLUTION DEGREE 2 as defined in IEC61010–1.

The Analyzer is a MEASUREMENT CATEGORY I (CAT I) device. Do not use the Analyzer as a CAT II, III, or IV device.

The Analyzer is for INDOOR USE only.

The Analyzer has been tested as a stand-alone device and in combination with the accessories supplied by Copper Mountain Technologies, in accordance with the requirements of the standards described in the Declaration of Conformity. If the Analyzer is integrated with another system, compliance with related regulations and safety requirements are to be confirmed by the builder of the system.


Never operate the Analyzer in an environment containing flammable gasses or fumes.

Operators must not remove the cover or any other part of the housing. The Analyzer must not be repaired by the operator. Component replacement or internal adjustment must be performed by qualified maintenance personnel only.

Never operate the Analyzer if the power cable is damaged.

Never connect the test ports to A/C power mains.

Electrostatic discharge can damage the Analyzer whether connected to or disconnected from the DUT. Static charge can build up on your body and damage sensitive internal components of both the Analyzer and the DUT. To avoid damage from electric discharge, observe the following:

- Always use a desktop anti-static mat under the DUT.
- Always wear a grounding wrist strap connected to the desktop anti-static mat via daisy-chained 1 MΩ resistor.
- Connect the post marked  on the body of the Analyzer to the common ground of the test station.

All general safety precautions related to operation of electrically energized equipment must be observed.



Definitions of safety symbols used on the instrument and in the manual are listed below.



Refers to the Manual if the instrument is marked with this symbol.



Alternating current.



Direct current.



On (Supply).



Off (Supply).



A chassis terminal; a connection to the instrument's chassis, which includes all exposed metal surfaces.

---

### **WARNING**

---

This sign denotes a hazard. It calls attention to a procedure, practice, or condition that, if not correctly performed or adhered to, could result in injury or death to personnel.

---

### **CAUTION**

---

This sign denotes a hazard. It calls attention to a procedure, practice, or condition that, if not correctly performed or adhered to, could result in damage to or destruction of part or all of the instrument.

---

### **NOTE**

---

This sign denotes important information. It calls attention to a procedure, practice, or condition that is essential for the user to understand.

## **General Overview**

The Vector Network Analyzer is designed for use in the process of development, adjustment, and testing of various electronic devices in industrial and laboratory facilities, including operation as a component of an automated measurement system. The Analyzer is designed for operation with an external PC, which is not supplied with the Analyzer.

The overview of measurement capabilities of the Analyzer is represented in [Measurement capabilities](#).

The block diagram of the Analyzer is represented in [Principle of operation](#).

## Specifications

The specifications of each Analyzer model can be found in its corresponding [datasheet](#).

## Measurement Capabilities

Measured parameters	S11, S21, S12, S22  Absolute power of the incident, reflected or transmitted DUT signals.  DC voltage at each point of the frequency sweep (optional for Cobalt series).
Number of measurement channels	Up to 16 logical channels. Each logical channel is represented on the screen as an individual channel window. Each logical channel has its own stimulus signal settings such as frequency range, number of test points, power level, etc.
Data traces	Up to 16 data traces can be displayed in each channel window. A data trace represents S-parameter of the DUT or absolute power of the incident, reflected or transmitted DUT signals.
Memory traces	Each of the 16 data traces can be saved into memory for further comparison with the current values.
Data display formats	Logarithmic magnitude, linear magnitude, phase, expanded phase, group delay, SWR, real part, imaginary part, Smith chart format, and polar format.

## Sweep setup features

Sweep type	Linear, logarithmic, and segment frequency sweep, when the stimulus power is a fixed value.
Power sweep	Linear power sweep when the frequency is a fixed value.
CW time sweep	Linear time sweep when the frequency and power are fixed values.

Measured points per sweep	From 2 to 200,001 or to 500,001 depending on model (See corresponding <a href="#">datasheet</a> ).
Segment sweep	A frequency sweep within several user-defined segments. Frequency range, number of points, source power, and IF bandwidth can be set for each segment.
Power settings	The power level can be set the same for all ports or individually for each port in the frequency sweep mode when the stimulus power is a fixed value. The power slope depending on frequency can be set to compensate for high-frequency attenuation in cables.
Sweep trigger	Trigger modes: continuous, single, hold. Trigger sources: internal, manual, external, bus.
Trigger output	<p>Synchronization of external devices with specified events in the Analyzer measurement cycle. A logic signal appears on the «Trigger Out» connector with the specified polarity and according to the specified events.</p> <p>The availability of this feature depends on the Analyzer model (See corresponding <a href="#">datasheet</a>).</p>

## Trace display functions

Trace display	Data trace, memory trace, or simultaneous data and memory traces.
Trace math	Data trace modification by math operations: addition, subtraction, multiplication or division between the data, and memory traces.
Autoscaling	Automatic selection of the scale division and reference level value to have the trace most effectively displayed.
Reference level automatic selection	Automatic selection of the reference level. After selection, the data trace shifts vertically so that the reference level crosses the trace in the middle.
Automatic reference level tracking	Automatic tracking of the reference level after each scan. The tracking method choice is: maximum, minimum, center, or active marker.
Electrical delay	Linear phase correction according the specified electrical delay.
Phase offset	Phase offset by the specified value in degrees.

## Accuracy enhancement

Calibration	Calibration of a test setup (which includes the Analyzer, cables, and adapters) significantly increases the accuracy of measurements. Calibration allows for correction of errors caused by imperfections in the measurement system: directivity, source, and load match, tracking, and isolation.
Calibration methods	<p>The following calibration methods of various sophistication and accuracy enhancement are available:</p> <ul style="list-style-type: none"><li>• reflection and transmission normalization</li><li>• full one-port calibration (SOL)</li><li>• one-path two-port calibration</li><li>• full two-port calibration (SOLT)</li><li>• TRL calibration</li></ul>
Reflection and transmission normalization	The simplest calibration method. It provides limited accuracy.
Full one-port calibration (SOL)	Method of calibration performed for one-port reflection measurements. It ensures high accuracy.
One-path two-port calibration	Method of calibration performed for reflection and one-way transmission measurements, for example, for measuring S11 and S21 only. It ensures high accuracy for reflection measurements, and reasonable accuracy for transmission measurements.
Full two-port calibration (SOLT)	Method of calibration performed for full S-parameter matrix measurement of a two-port DUT. It ensures high accuracy.
Two-port TRL calibration	Method of calibration performed for full S-parameter matrix measurement of a two-port DUT. LRL and LRM types of this calibration are also supported. It provides higher accuracy than a SOLT calibration.
Mechanical calibration kits	It is possible to select one of the predefined calibration kits of various manufacturers or define additional ones.

Electronic calibration modules	Copper Mountain Technologies' automatic calibration modules (ACM's) make Analyzer calibration faster and easier than traditional mechanical calibration and provides the highest accuracy.
Sliding load calibration standard	The use of sliding load calibration standard allows significant increase in calibration accuracy at high frequencies compared to a fixed load calibration standard.
Unknown thru calibration standard	The use of an arbitrary reciprocal two-port thru device instead of a defined by parameters thru during a full two-port calibration allows calibration if the parameters of an available thru are unknown. This method allows calibration of the test setup for measurements of non-insertable devices.
Defining of calibration standards	Different methods of calibration standard definition are available: <ul style="list-style-type: none"> <li>• standard definition by polynomial model</li> <li>• standard definition by database (S-parameters)</li> </ul>
Error correction interpolation	When such settings as start/stop frequencies and number of points are changed, compared to the settings of calibration, interpolation or extrapolation of the calibration coefficients will be applied (Extrapolation is not recommended).
Port Extension	Calibration plane compensation for delay in the test setup.

### Supplemental calibration methods

Power calibration	Method of the port power calibration which allows to maintain more stable power levels at the DUT input. The calibration requires connection of an external USB power meter.
Receiver calibration	Method of the receiver gain calibration to the accurate absolute power measurement.

## Marker functions

Data markers	Up to 16 markers for each trace. A marker indicates the stimulus value and measurement result at a given point of the trace.
Reference marker	Enables indication of any marker value as relative to the reference marker.
Marker search	Search for max, min, peak, or target values on a trace.
Marker search additional features	User-defined search range. Available as either a tracking marker, or as a one-time search.
Setting parameters by markers	Setting of start, stop, and center frequencies from the marker frequency, and setting of reference level by the measurement result of the marker.
Marker math functions	Statistics, bandwidth, flatness, RF filter.
Statistics	Calculation and display of mean, standard deviation and peak-to-peak values of the trace.
Bandwidth	Determines bandwidth between cutoff frequency points for an active marker or absolute maximum. The bandwidth value, center frequency, lower frequency, higher frequency, Q value, and insertion loss are displayed.
Flatness	Displays gain, slope, and flatness between two markers on a trace.
RF filter	Displays insertion loss and peak-to-peak ripple of the passband, and the maximum signal magnitude in the stopband. The passband and stopband are defined by two pairs of markers.



## Data analysis

Port impedance conversion	This function converts S-parameters measured at the Analyzer's nominal port impedance into values which would be found if measured at arbitrary port impedance.
De-embedding	This function allows mathematical exclusion of the effects of the fixture circuit connected between the calibration plane and the DUT. This circuit should be described by an S-parameter matrix in a Touchstone file.
Embedding	This function allows mathematical simulation of the DUT parameters after virtual insertion of a fixture circuit between the calibration plane and the DUT. This circuit should be described by an S-parameter matrix in a Touchstone file.
S-parameter conversion	This function allows conversion of the measured S-parameters to the following parameters: reflection impedance and admittance, transmission impedance and admittance, and inverse S-parameters.
Time domain transformation	<p>This function performs transformation from frequency domain into response of the DUT to various stimulus types in time domain. Modeled stimulus types: bandpass impulse, lowpass impulse, and lowpass step. Time domain span is set arbitrarily from zero to maximum, which is determined by the frequency steps. Various window shapes allow optimizing the tradeoff between resolution and the level of spurious sidelobes.</p> <p>The availability of this feature depends on the Analyzer model (See corresponding <a href="#">datasheet</a>).</p>
Time domain gating	<p>This function mathematically removes unwanted responses in time domain, allowing for measurement of the frequency response without the influence of selected fixture elements. Gating filter types: bandpass or notch. For better tradeoff between gate resolution and the level of spurious sidelobes the following filter shapes are available: maximum, wide, normal, and minimum.</p> <p>The availability of this feature depends on the Analyzer model (See corresponding <a href="#">datasheet</a>).</p>

## Mixer / converter measurements

Scalar mixer / converter measurements	<p>The scalar method allows measurement of scalar transmission S-parameters of mixers and other devices having different input and output frequencies. No external mixers or other devices are required. The scalar method employs port frequency offset when there is a difference between receiver frequency and source frequency.</p> <p>The availability of this feature depends on the Analyzer model (See corresponding <a href="#">datasheet</a>).</p>
Vector mixer / converter measurements	<p>The vector method allows measuring of the mixer transmission S-parameter magnitude and phase. The method requires an external reference mixer and an LO common to both the external reference mixer and the mixer under test.</p> <p>The availability of this feature depends on the Analyzer model (See corresponding <a href="#">datasheet</a>).</p>
Scalar mixer / converter calibration	<p>The most accurate method of calibration applied for measurements of mixers in frequency offset mode. OPEN, SHORT, and LOAD calibration standards are used. An external power meter is required and should be connected to the USB port directly or via USB/GPIB adapter.</p> <p>The availability of this feature depends on the Analyzer model (See corresponding <a href="#">datasheet</a>).</p>
Vector mixer /converter calibration	<p>Method of calibration applied for vector mixer measurements. OPEN, SHORT, and LOAD calibration standards are used.</p> <p>The availability of this feature depends on the Analyzer model (See corresponding <a href="#">datasheet</a>).</p>
Automatic adjustment of frequency offset	<p>This function performs automatic frequency offset adjustment when scalar mixer / converter measurements are performed to compensate for LO frequency inaccuracies internal to the DUT.</p> <p>The availability of this feature depends on the Analyzer model (See corresponding <a href="#">datasheet</a>).</p>

## Other features

Familiar graphical user interface	Intuitive graphical user interface ensures fast and easy Analyzer operation.
Printout/saving of traces	The traces and data printout function has a preview feature. Previewing, saving, and printing can be performed using MS Word, Image Viewer for Windows, or the Analyzer Print Wizard.
Linux OS support	<p>The Linux version of the analyzer software is designed to run on x86 PCs running Linux.</p> <p><b>NOTE:</b> Tests must be performed to determine if the analyzer software is compatible with a particular version of Linux.</p>

## Remote control

COM/DCOM	Remote control via COM/DCOM. COM automation is used when the user program is running on the local PC. DCOM automation is used when the user program is running on the LAN-networked PC. Automation of the instrument can be achieved in any COM/DCOM-compatible language or environment, including Python, C++, C#, VB.NET, LabVIEW, MATLAB, Octave, VEE, Visual Basic (Excel), and others.
SCPI	Remote control using textual commands SCPI (Standard Commands for Programmable Instruments). Text messages are delivered over PC networks using HiSLIP or TCP/IP Socket network protocols. VISA Library is recommended to support HiSLIP protocol. The TCP/IP Socket protocol can be supported by the VISA library or directly programmed in any language or environment that supports TCP/IP Sockets. The VISA library is free and widely used software in the field of testing and measurement.

## Principles of Operation

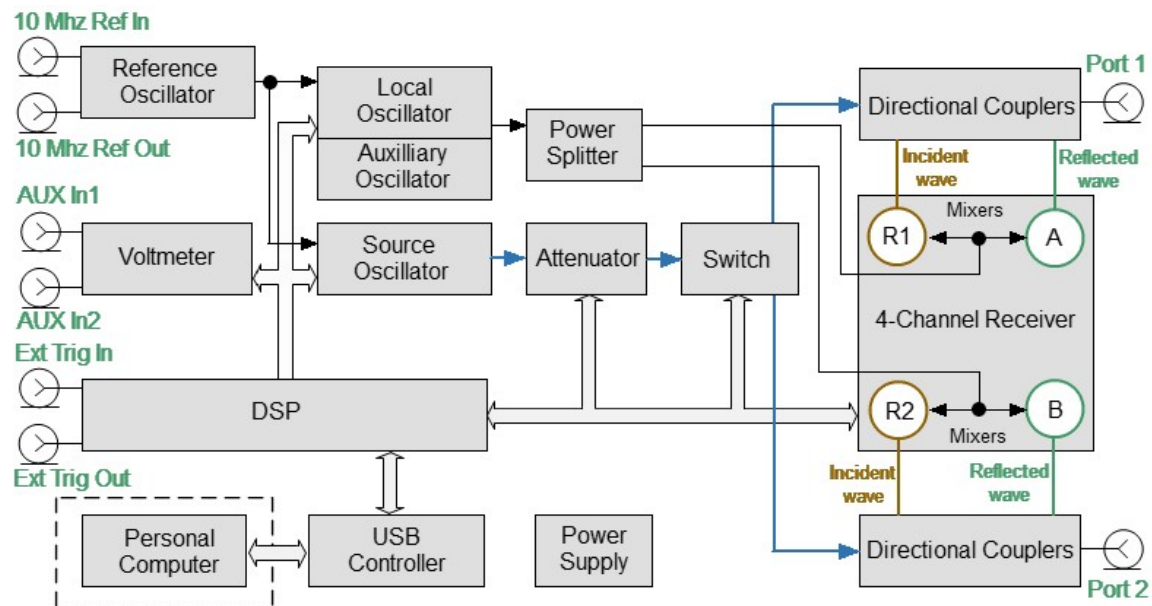
A Vector Network Analyzer (VNA) is a tool for accurate measurement of complex transmission and reflection coefficients (S-parameters) of a Device Under Test (DUT).

The analyzers described in this manual are USB VNAs. These VNAs consists of an RF measurement module (Analyzer) and supplied processing software – an application which runs on a Windows or Linux based PC or laptop, connected to the Analyzer's hardware via a USB interface. This application controls the RF measurement module, receives and post-processes received raw data and presents the calibrated results to the user in a variety of graphical formats.

The Analyzers described in this manual differ in such parameters as frequency range, output power, measurement speed, dynamic range, and measurement accuracy. Direct access to receivers and the possibility of connecting frequency extension modules significantly affects the design and functions of the Analyzer.

For a detailed description of different series and models of analyzers see [Instrument Series](#).

The complete specification and supported features list are given in the Datasheet of the corresponding Analyzer.



The block diagram of the Analyzer

The Analyzer consists of the following functional blocks: a Reference Oscillator, a Source Oscillator, a Local Oscillator, a power control Attenuator, a Switch, a Power Splitter, two Dual Directional Couplers, a four-channel Receiver, a digital signal processor (DSP), and a Power Supply.

A tunable Source Oscillator is the test signal source. This Source Oscillator design is based on digital frequency synthesizers. An internal Reference Oscillator provides the source oscillator with a stable reference signal.

The Local Oscillator (LO) generates signals using digital frequency synthesizers at an offset from the Source Oscillator which is equal to the Intermediate Frequency (IF) which will be digitized by the VNA IF circuit. An Auxiliary Oscillator may be used to provide an offset LO for frequency offset measurements.

The Local Oscillator is the source of the LO signal for the receiver.

The Power Splitter distributes the LO signal between the four Receivers.

A programmable Attenuator controls the power level of the test signal. This Attenuator is an executive unit of the automatic power control system. For example, when a power calibration has been completed, the Power Correction function uses this Attenuator. Also, the Analyzer can sweep over the output power range at a fixed frequency of test signal using this Attenuator. The Attenuator is controlled by setting the signal power level at the output of the measurement port. The range of signal power levels is specified at the output for power sweep mode.

Switch changes the direction of the test signal through the DUT, switching the Source Oscillator signal between the two Directional Couplers. Thus, any port can be the source or receiver of a signal. If port 1 is the source, port 2 will be the receiver and vice versa. The user can measure all the S-parameters by making only one DUT connection and allow the VNA to switch the direction of measurement.

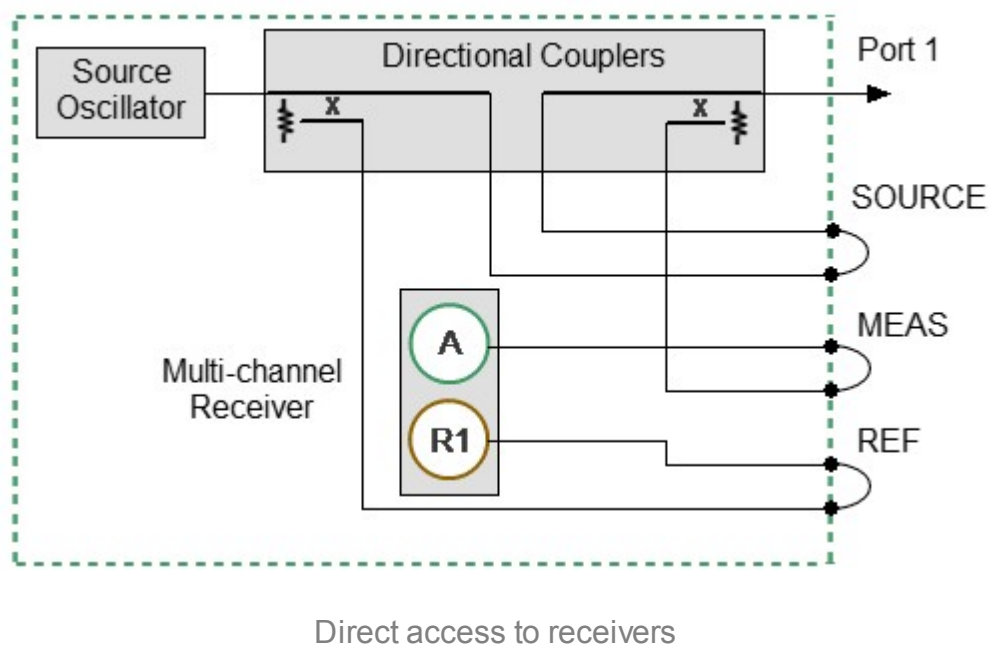
Directional Couplers separate the incident wave and reflected waves of signal transmitted through a DUT.

The incident and reflected signals from the two Directional Couplers are applied to a multi-channel Receiver. The multi-channel Receiver of the two-port Analyzer consists of four identical channels (two channels per port). The reference receiver processes the incident wave, a measuring receiver processes the reflected wave, the signal entering the port. The Reference receiver is indicated as R with the index corresponding to the port number. The measuring receiver is indicated as A or B. Receiver Mixers convert the signal to an IF frequency. For Planar 304/1, 804/1, and 814/1 models, this is a first IF conversion. After filtering, the signal is converted to a second IF. Analog-to-digital converters in a multi-channel receiver convert these IF signals to a sequence of digital samples and supply them to DSP. The DSP performs primary signal processing (filtering, phase difference estimation, magnitude measurement). The user selected IF Bandwidth is applied by the DSP filter.

After the primary signal processing, the DSP transmits the information to the control software (S2VNA) running on an external PC. Communication is provided by a USB controller. This software applies calibration and performs the final calculations and displays the measurement results on the screen of PC. The Software also controls the operation of the hardware of the Analyzer.

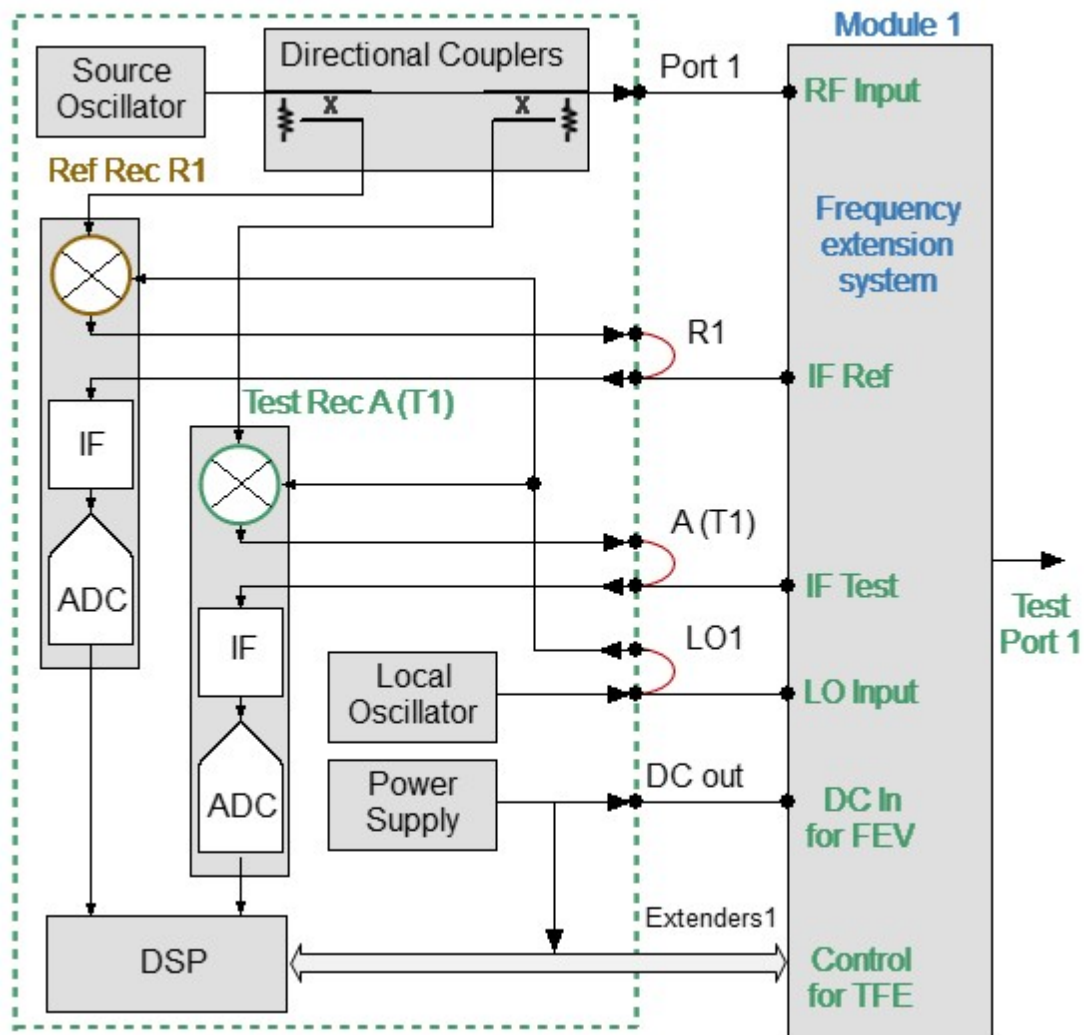
## Adjustable port configuration with direct access to the receivers

Planar 814/1, Cobalt C2209, and C2220 analyzer models allow direct access to the measurement receivers (See Figure below). This feature is intended for a variety of measurement applications requiring a wider dynamic and power range. Direct receiver access enables testing of high power devices. Additional amplifiers, attenuators, various filters, and matching pads for each of the ports can be introduced in the reference oscillator and receiver paths to ensure optimal operation of the receivers.



## Frequency extension system module connection

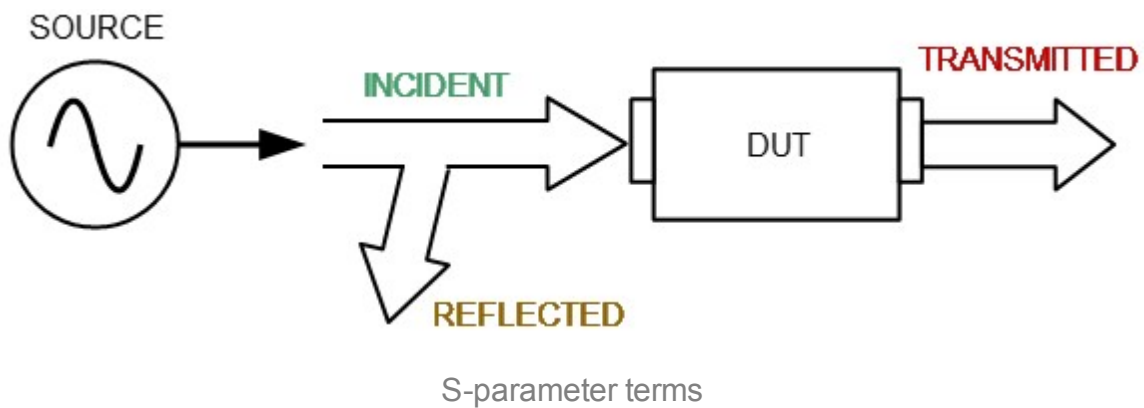
Cobalt C4209 and C4220 analyzer models have additional ports with jumper cable assemblies on the configurable front panel for connecting Frequency Extension Modules. These modules, equipped with coaxial or waveguide connectors, are intended to extend the measurement frequency range from 18 to 110 GHz. The Analyzer provides power and control for Frequency Extension Modules. The power and control connectors for operating with Frequency Extension Modules are located on the rear panel of the Analyzer. The connection diagram is shown below.



Frequency extension system module connection

## The Principle of Measuring S-parameters

The Analyzer emits a test signal (stimulus) out of a port connected to the DUT. The frequency of the test signal changes in the specified range discretely from point to point. At each frequency point, the Analyzer simultaneously measures the magnitude and phase of the signal transmitted through and reflected from the DUT. These are compared with the magnitude and phase of the incident test signal. The Analyzer calculates the S-parameters of the DUT at each frequency point based on this comparison (See figure below).



The S-parameter is a relation between the complex magnitudes of two waves:

$$S_{mn} = \frac{\text{outgoing wave at Port } m}{\text{incoming wave at Port } n}$$

Providing the *incoming wave at Port*, except  $n = 0$ , where  $m, n$  denote the DUT port number.

In accordance with the measurement plan, the Analyzer performs a frequency sweep in one direction in a specified frequency range. Then, if needed, the Analyzer performs a frequency sweep in reverse direction in a specified frequency range. The frequency changes discretely, according to the number of measuring points specified. S-parameters of the DUT are measured for each measuring point.

For a two-port DUT the Analyzer measures the full scattering matrix:

$$S = \begin{bmatrix} S_{11} & S_{12} \\ S_{21} & S_{22} \end{bmatrix}$$

For the measurement of  $S_{11}$ ,  $S_{21}$  parameters, test Port 1 will operate as a signal source. The incident and reflected waves will be measured by Port 1. The transmitted wave will be measured by Port 2.

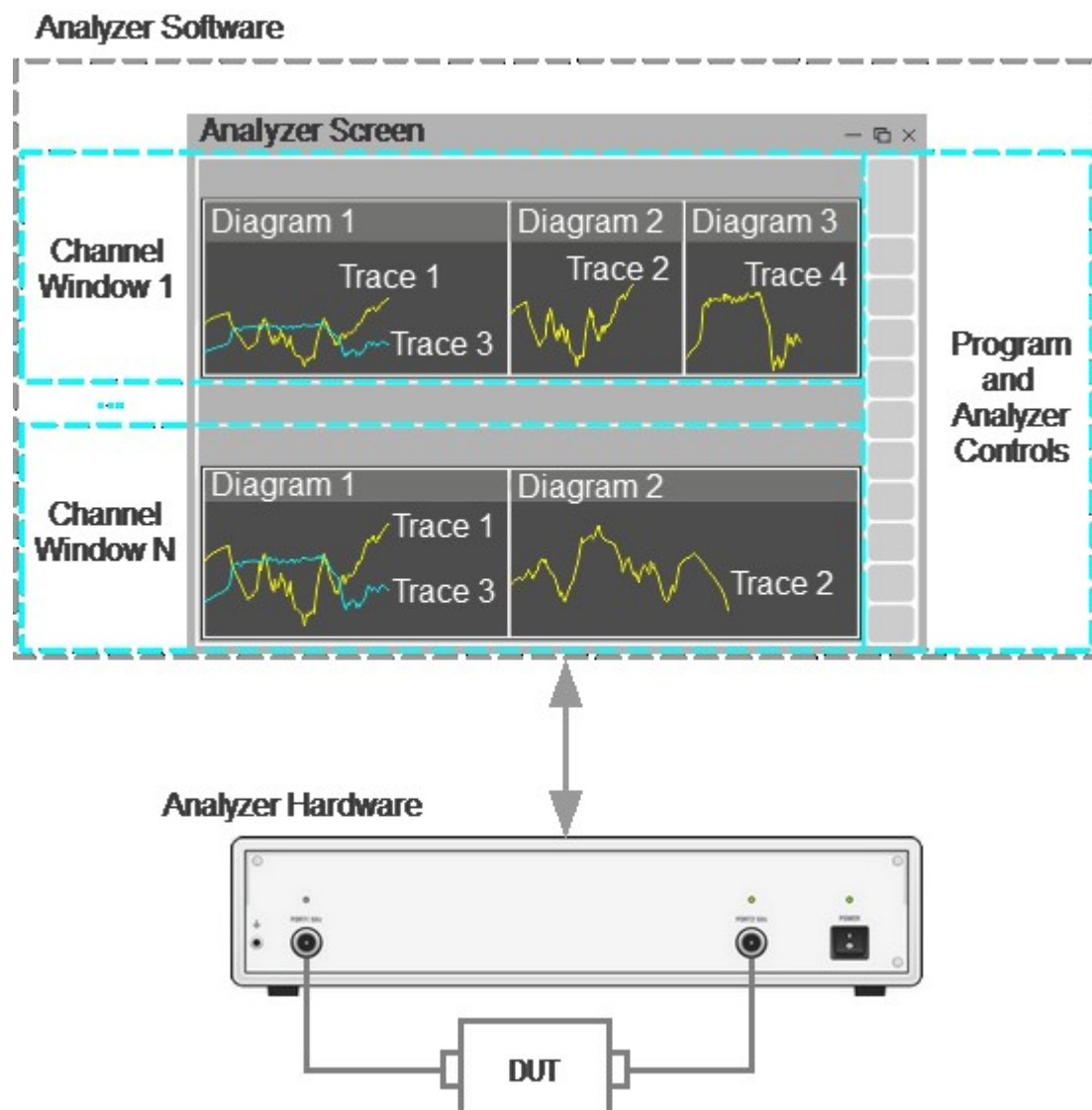
For the measurement of  $S_{12}$ ,  $S_{22}$  parameters, test Port 2 will operate as a signal source. The incident and reflected waves will be measured by Port 2. The transmitted wave will be measured by Port 1.



## Summarized description of hierarchy

The following hierarchy of measurement, processing, and display tools is used during operation of the Analyzer (See figure below):

- **Analyzer Hardware** makes radio frequency measurements of the DUT parameters and performs primary processing of measurement results.
- **Analyzer Software** (supplied with the analyzer) controls the operation of the analyzer components and performs the final mathematical processing and display of the measurement results.



Hierarchy of measuring, processing, and displaying tools

**Analyzer Software** is displayed as **Analyzer Screen** on the control PC screen, which contains the following:

- **Channel Windows** – the diagram area in which the **Channel** is displayed. For a detailed description of the controls, see [Channel Window Layout and Functions](#).
- **Program and Analyzer Controls**: menu bar, analyzer status bar, and program button bar. For a detailed description of the controls, see [Screen Layout and Functions](#).

**Channel** – a logical analyzer created by the control program to perform DUT measurements with set parameters. The analyzer control software supports up to 16 channels simultaneously, processing them one at a time. Thus, the same DUT can be sequentially measured by 16 logic analyzers with individual settings.

The channel settings are:

- [Sweep Type](#)
- [Sweep Range](#)
- [Number of Points](#)
- [Stimulus Power](#)
- [Trigger Settings](#)
- [IF Bandwidth Setting](#)
- [Calibration and Cal Kits](#)
- [Average Setting](#)

The measurement results of the DUT in the channel are displayed in traces.

**Trace** – a sequence of data points measured (data trace) or memorized (memory trace) by the analyzer, connected by a line. Each channel contains up to 16 Traces. The trace is characterized by the following parameters:

- [Measurement Parameters](#)
- [Format](#) and [Scale](#)
- [Memory Trace](#)
- [Smoothing](#)

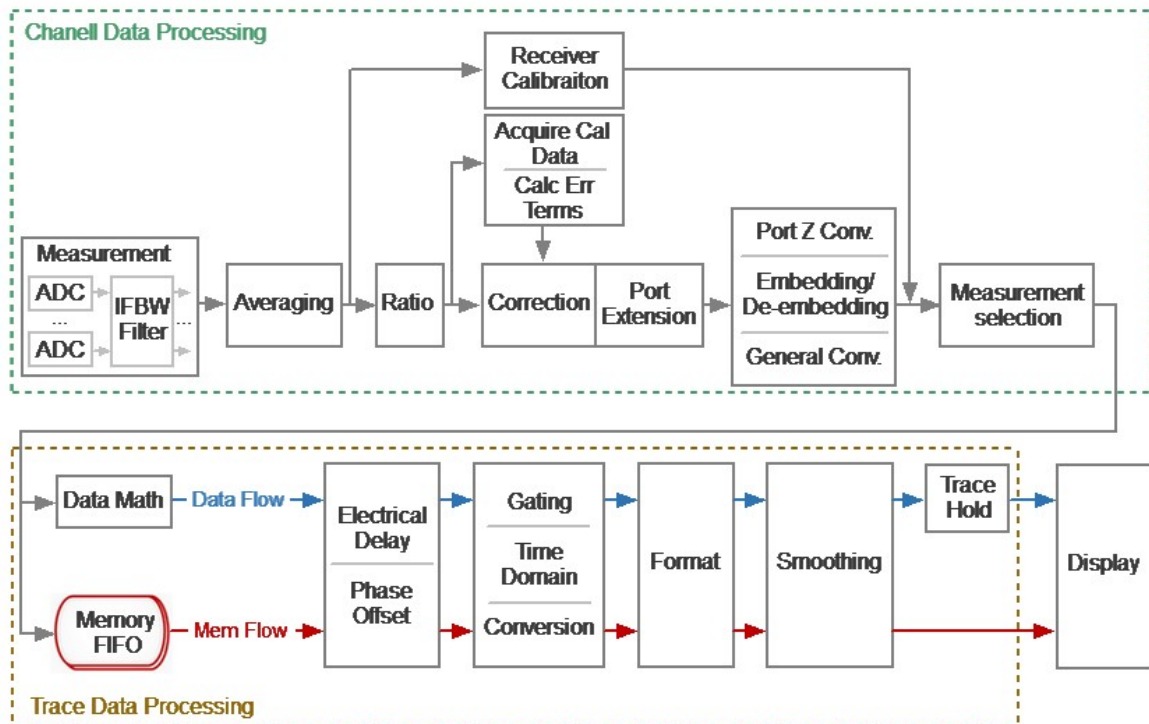
The following functions apply to the trace:

- [Markers](#)
- [Electrical Delay](#)
- [Phase Offset](#)
- [Time Domain Gating](#)
- [S-Parameter Conversion](#)
- [Limit Test](#)

Each channel window can display up to 16 charts simultaneously. Convenient placement of traces in the channel window is designated as **Diagram**. Traces can be placed in a single chart or grouped according to user settings in different charts. For a detailed description of working with diagrams, see [Trace Allocation](#).

## Internal Data Processing

The following figure shows a flowchart of the Analyzer's internal data processing flow. For a detailed description of remote control access to internal data arrays see in [Internal Data Arrays](#).



Data Processing Flowchart

The Analyzer's internal data processing consists of the following stages:

- **Measurement** is a converting analog signals of receivers into digital ones (receivers R1 and R2 receive the signal of the incident wave, receivers A and B receive the signal reflected from the device under test or passed through the DUT. The received analog measurement signals are converted by ADC (analog-to-digital converters) into digital IF signals and transmitted to the digital processor. The digital processor performs a discrete Fourier transform (DFT) of the IF signals. The analyzer IF bandwidth is equivalent to the bandwidth of the DFT filter. The digital output of each receiver is represented as complex numbers). For more details see [Principle of Operation](#).
- **Averaging** is averaging of the measured data of the receivers for a given number of scan cycles. For more details see [Averaging Setting](#).
- **Receiver Calibration** is a gain correction of individual receivers for absolute measurements. For more details see [Receiver Calibration](#).
- **Ratio** is calculating S-parameters by dividing the complex values of two receiver signals. See [The Principle of Measuring S-parameters](#).

- **Acquire Cal Data** is measuring calibration standards. Complex measured data of all standards are stored in memory. For more details see [Calibration Methods and Procedures](#).
- **Calc Error Terms** is calculation of calibration coefficients based on measurement data of calibration standards in accordance with the selected calibration method. Calculated calibration coefficients are stored in memory. After calculating the calibration coefficients, the measurement data of the calibration standards is deleted. For more details see [Systematic Errors](#).
- **Correction** is an application of calibration coefficients to raw S-parameters. At this stage, systematic measurement errors introduced by the analyzer and the measuring setup are eliminated. For more details see [Calibration Methods and Procedures](#).
- **Port extension** is a fixture simulation in which the addition or removal of a transmission line of a given length for each test port is mathematically simulated. This allows to offset the calibration reference plane by the length of the line. For more details see [Port Extension](#).
- **Port Z Conv** is the fixture simulation to convert the reference impedance to an arbitrary impedance value. See [Port Reference Impedance \(Z\) Conversion](#).
- **De-embedding** is the fixture simulation to eliminate the influence of a certain circuit from the measurement results. See [De-embedding](#).
- **Embedding** is the fixture simulation for embedding some virtual circuit in the measured circuit. See [Embedding](#).
- **General Conv** is a conversion of a matrix of S-parameters into a matrix of Z, Y, T, H, ABCD - parameters. See [General S-Parameter Conversion](#).
- **Measurement Selector** is a selection of display of measured S-parameter or absolute (receiver) data. Data for the trace is selected from a matrix of corrected S-parameters or corrected receiver data. See [Measurement Parameters Settings](#).
- **Memory FIFO** is copying current measurements to memory (S-parameter or receiver data). The software contains a set of cells for storing measurements (memory). It is possible to record up to 8 of these saves. In this case, if all 8 saves are occupied, then the next save will delete the save created by the very first in time. Further, the memory data is processed in parallel with the measured data. See [Memory Trace Function](#).
- **Data Math** — mathematical operations between measured data and data in memory. When using FIFO memory, the operation is performed with active memory. Available functions: add measured data to memory data, subtract memory data from measured data, multiply/divide measured data by memory data. The result of the operation replaces the measured data. See [Memory Trace Function](#).

- **Electrical Delay** is adding/removing a transmission line of a given length without loss for the measurement port. Unlike port extension, the method is applied individually for each trace. See [Electrical Delay Setting](#).
- **Phase Offset** is setting a constant phase offset of the trace. See [Phase Offset Setting](#).
- **Time Domain** is conversion of the measured S-parameter in the frequency domain into the response of the circuit under investigation in the time domain. See [Time Domain Transformation](#).
- **Gating** is a removal of unwanted responses in the time domain. See [Time Domain Gating](#).
- **Conversion (S-parameter conversion function)** is conversion of the measured S-parameter into following ones: impedance ( $Z_r$ ) and admittance ( $Y_r$ ) in reflection measurement, impedance ( $Z_t$ ) and admittance ( $Y_t$ ) in transmission measurement, inverse S-parameter ( $1/S$ ), impedance ( $Z_{tsh}$ ) and admittance ( $Y_{tsh}$ ) in transmission shunt measurements, S-parameter complex conjugate (Conj). See [S-Parameter Conversion](#).
- **Format** is selection of the display format of the measured data on the trace. See [Format Setting](#).
- **Smoothing** is an averaging of adjacent points of the trace by a moving window. See [Smoothing Setting](#).
- **Trace Hold** is holding the maximum or minimum values of the trace. See [Trace Hold](#).
- **Display** — data processing for displaying on the screen in the form of a trace of a given format. Scaling is applied to the traces according to the data format, according to selected reference line position and value and scale/grid settings. See [Channel Window Layout and Functions](#).

## Instrument Series

This section describes the different series and models of analyzers.

Series of two-port Analyzers:

- [Compact Series](#), excluding the S, M, SC models
- [Cobalt Series](#)
- [Planar Series](#)

The front and rear panels of each analyzer are shown further in this section, along with the controls located on those panels.

## Compact Series

The Compact Series Analyzers deliver a lab grade performance in a compact package.

The Analyzers in this series are divided by function:

- [S models](#) — these models deliver a lab grade performance, with all the features engineers have come to expect to be included in standard software.
- [M models](#) — these models provide metrology-level characteristics, but do not support the following functions: Vector Mixer Calibration, TRL Calibration, Frequency Offset, Time Domain, Gating.
- [SC models](#) — these models have an excellent dynamic range, high measurement speed and high output power, and support a full set of standard software functions.

---

### NOTE

The S5243 can be a rack-mounted VNA.

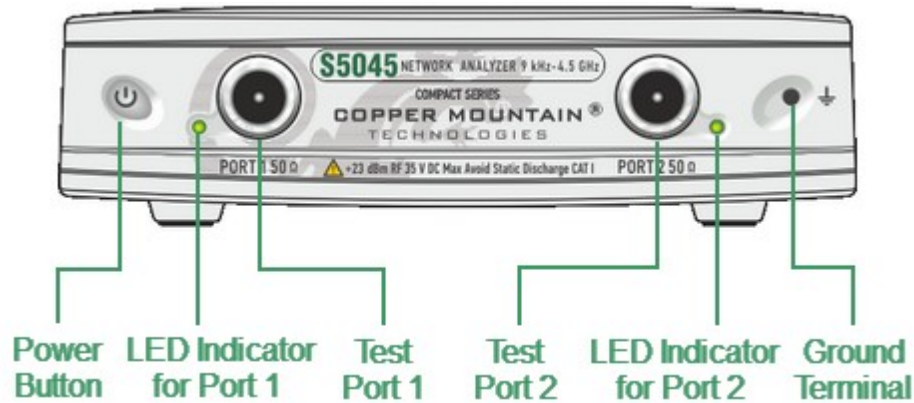
---



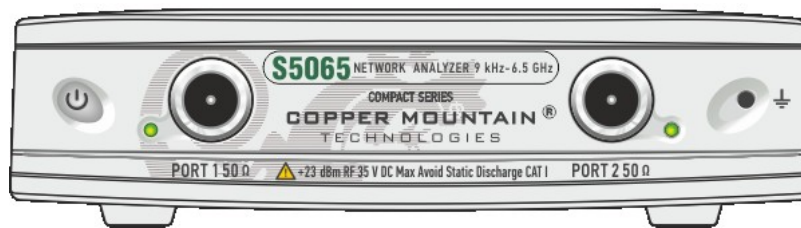
## S Models, Front Panel

A Compact VNA models that deliver lab grade performance in a compact package, with all the features engineers have come to expect included standard in software.

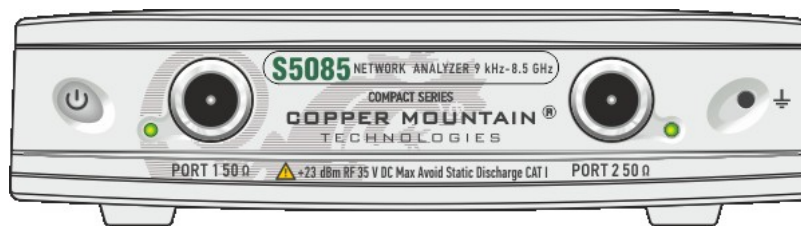
The front view of the Analyzers is represented in the figures below.



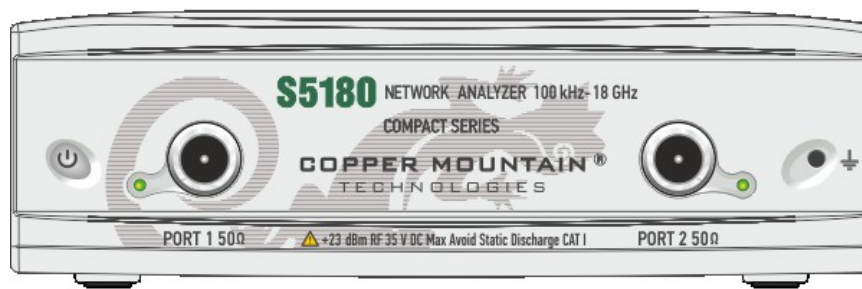
S5045 front panel



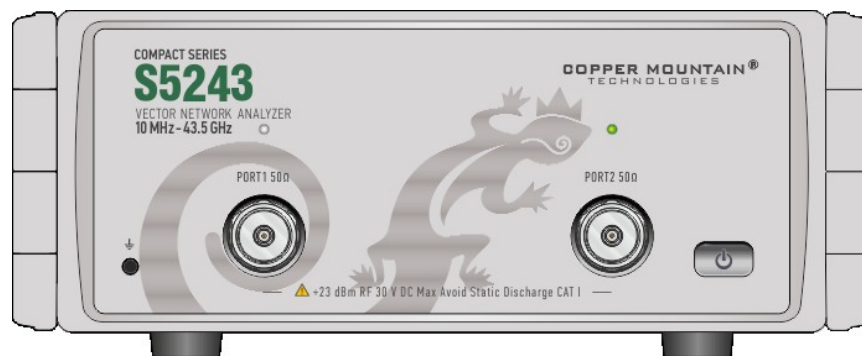
S5065 front panel



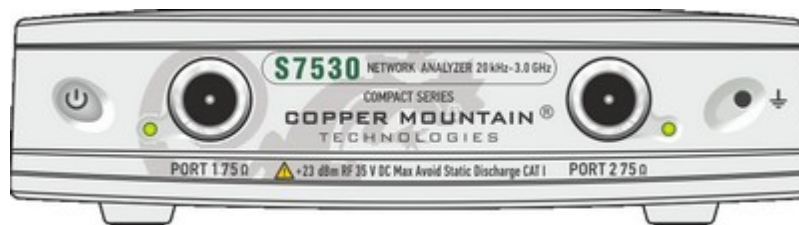
S5085 front panel



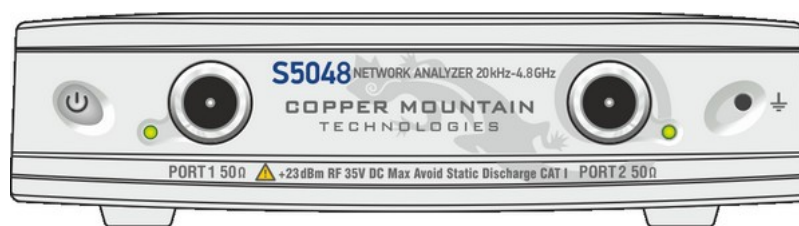
S5180 front panel



S5243 front panel



S7530 rear panel



S5048 front panel

## Part of front panel

### Power Button



Switches the Analyzer ON and OFF.

The analyzer can be turned on/off at any time. The VNA loads its operating firmware from the PC each time upon powering up. The process will take approximately 10 seconds, after which the analyzer will be ready for operation.

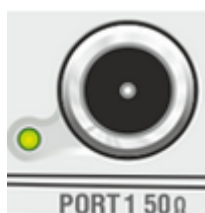
---

#### NOTE

The USB driver will be installed onto the PC when the analyzer is turned on for the first time. The driver installation procedure is described in [Software Installation](#). Some PCs may require re-installation of the driver in case of change of the USB port.

---

### Test Ports



Test Port 1 and 2 are intended for DUT connection. S models have type-N female test ports connector type. Model S5243 has NMD 2.4 mm test ports.



Each test port has an LED indicator. A test port can be used either as a source of the stimulus signal or as a receiver of the response signal from the DUT. The stimulus signal can only appear on the only one port at a time.

Connecting the DUT to only one test port on the Analyzer allows the measurement of reflection parameters (e.g. S11 or S22) of the DUT.

Connecting the DUT to all test ports of the Analyzer allows for the measurement of the full S-parameter matrix of the DUT.

---

#### NOTE

The LED indicator identifies the test port which is operating as a signal source.

---

---

**CAUTION**

Do not exceed the maximum allowed power of the input RF signal (or maximum DC voltage) indicated on the front panel. This may lead to damage of the Analyzer.

---

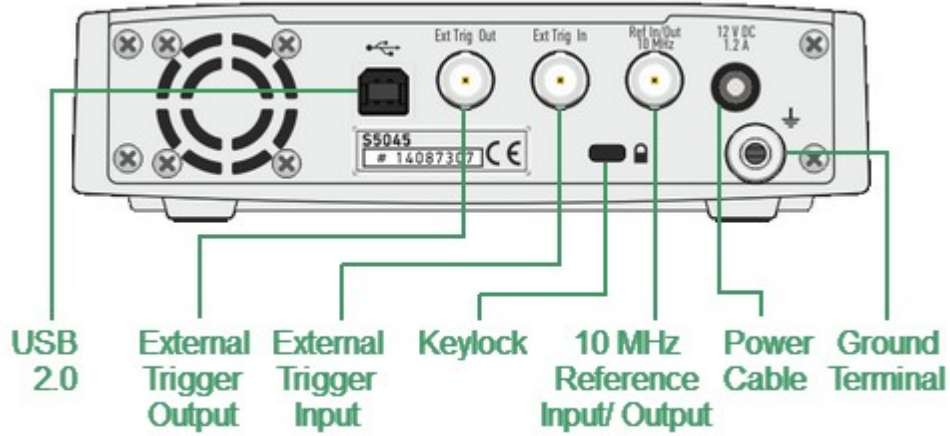
**Ground Terminal**

Use the terminal for grounding.

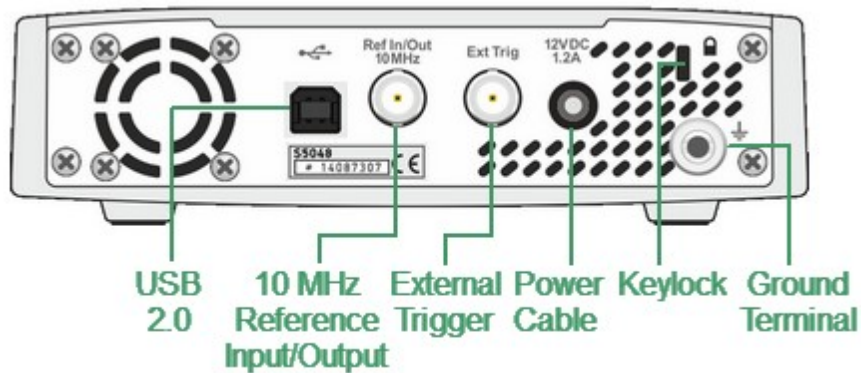
To avoid damage from electric discharge, connect the ground terminal on the body of the Analyzer to a reliable earth ground shared with the DUT in the test environment.

## Rear Panel

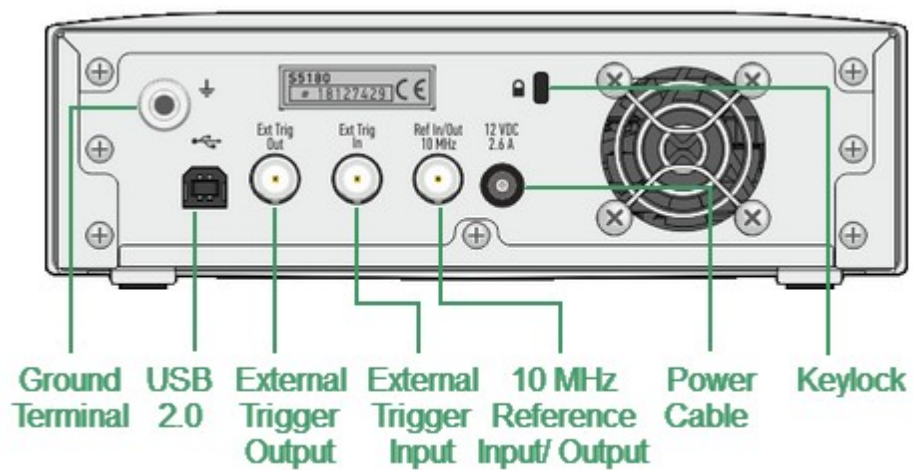
The rear view of the Analyzers is represented in the figures below.



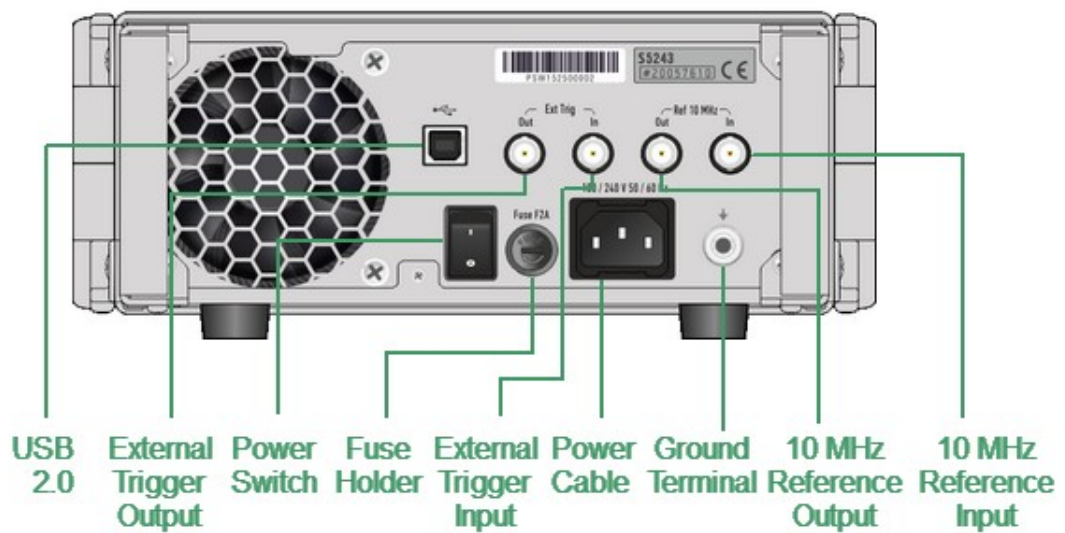
S5045, S5065, S5085 rear panel

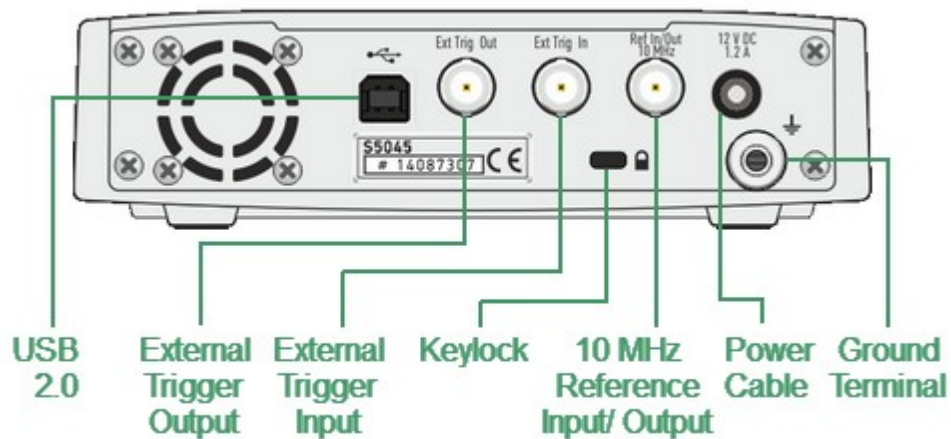


S5075, S5048 rear panel

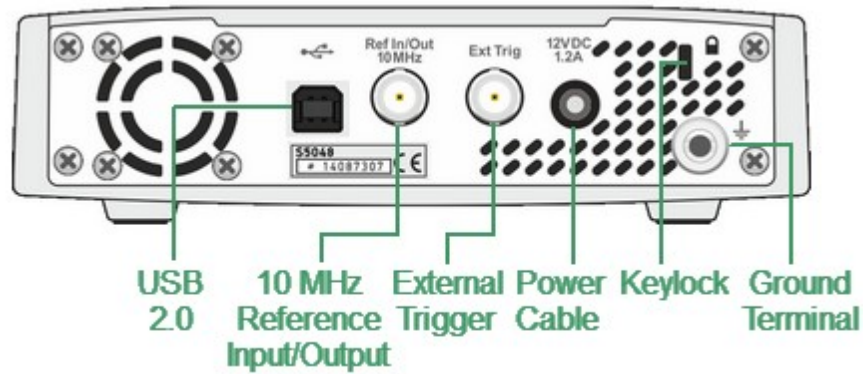


S5180 rear panel

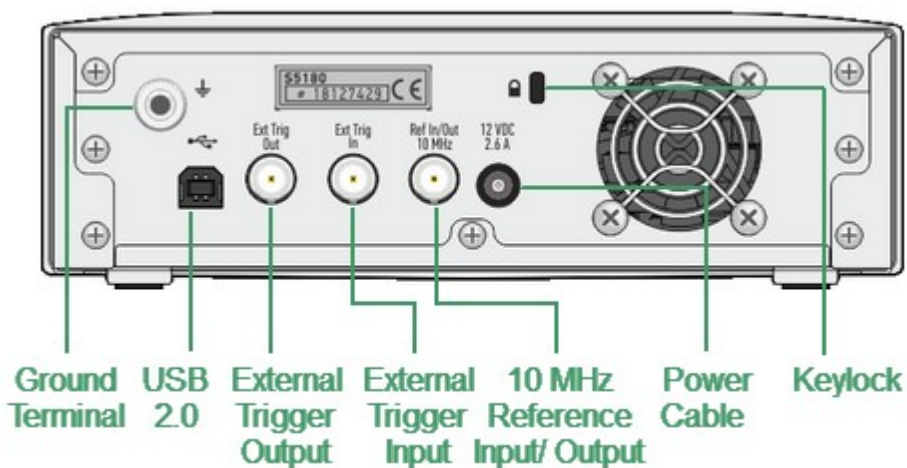




S5045, S5065, S5085 rear panel

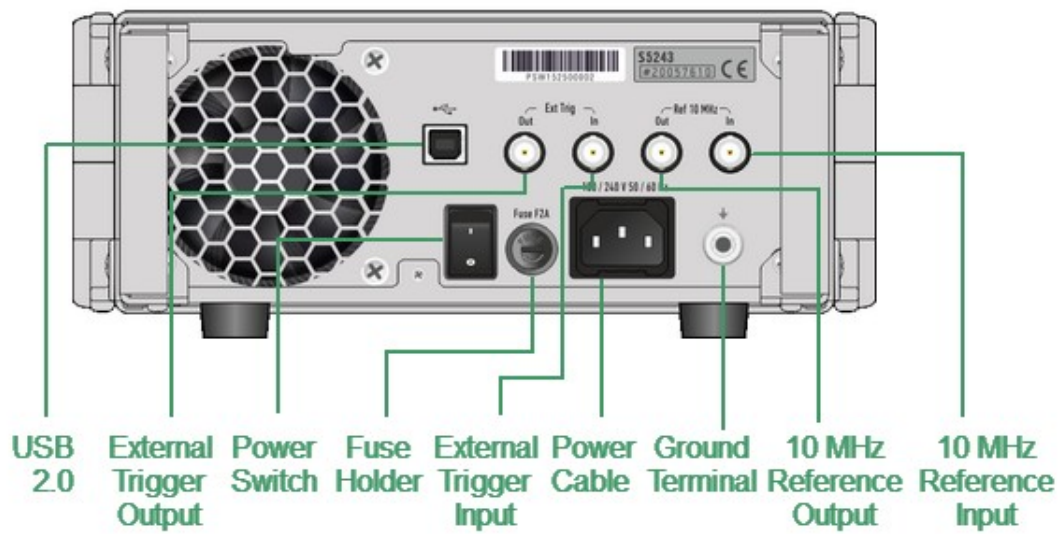


S5048, S5075 rear panel



S5180 rear panel





S5243 rear panel

## Parts of rear panel

### Power Cable Receptacle



The power cable receptacle (S5243 model only) is intended for 100 VAC to 240 VAC 50/60 Hz power cable connection.



The power supply receptacle is intended for an external DC power supply voltage from 9 to 15 V; alternatively, the power supply can be powered by a battery, including a vehicle battery, through an appropriate vehicle power cable. The DC connection requires a 3.5x1.35 mm plug with positive center conductor.

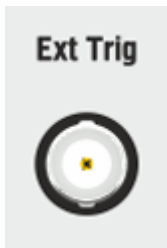
### Power Switch



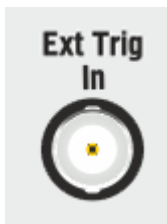
The power switch serves as the disconnecting device (device that cuts off power supply) of the VNA. The power supply must be cut off to avoid such danger is electric shock, during prolonged non-use of the device.



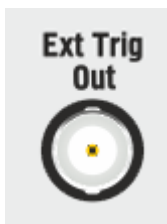
### External Trigger Signal Input Connector



This connector allows to connect an external trigger source. Connector type is BNC female. For input and signal parameters, see instrument specification.

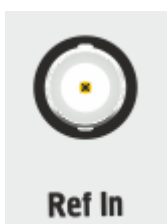


### External Trigger Signal Output Connector (except S5048, S7530 models)



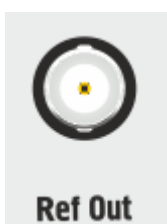
The External Trigger Signal Output port can be used to provide trigger to an external device. The port outputs signal with various waveforms depending on the setting of the Output Trigger Function: before frequency setup pulse, before sampling pulse, after sampling pulse, ready for external trigger, end of sweep pulse, measurement sweep.

### Internal Reference Frequency Output Connector (S5243 model only)



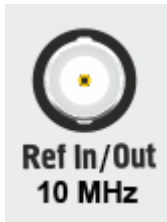
External reference frequency is 10 MHz, input level is 2 dBm  $\pm$  2 dB, input impedance at «Ref In» is 50  $\Omega$ . Connector type is BNC female.

### Reference Frequency Input (S5243 model only)



Output reference signal level is 3 dBm  $\pm$  2 dB at 50  $\Omega$  impedance. Connector type is BNC female.

### Reference Frequency Input/Output Connector (except S5243 model)



External reference frequency is 10 MHz, input level is 2 dBm  $\pm$  3 dB, input impedance 50  $\Omega$ . Output reference signal level is 3 dBm  $\pm$  2 dB into 50  $\Omega$  impedance. Connector type is BNC female.

### USB 2.0 High Speed Port



The USB port is intended for connection to an external PC.

### Fuse Holder (S5243 only)



A Fuse protects the Analyzer from the excessive current.

## Ground Terminal



To avoid electric shock, use the this terminal for grounding.

The Ground terminal allows to connect directly the body of the Analyzer to the test station ground in order to ensure electrical safety.

## Keylock

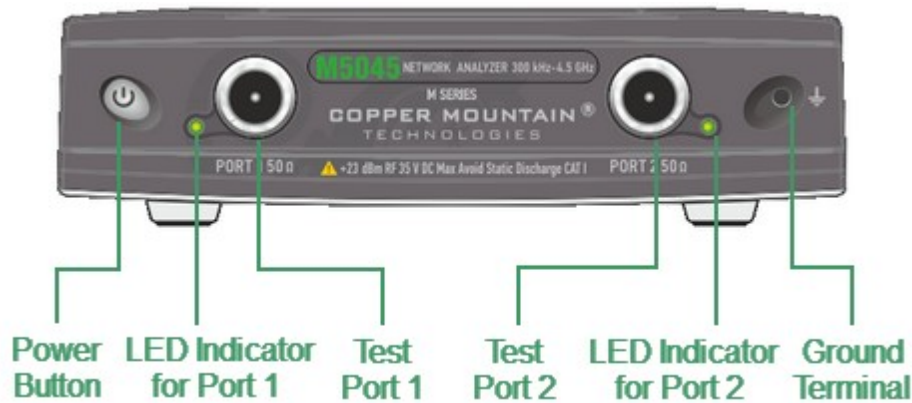


A Kensington lock (K-Slot) is part of an anti-theft system.

## M Models, Front Panel

A Compact VNA models that deliver metrology grade performance in a more economical package that excludes a number of advanced features: Vector Mixer Calibration, TRL Calibration, Frequency Offset, Time Domain, Gating.

The front view of the Analyzers is represented in the figures below.



M5045 front panel



M5065 front panel



M5090 front panel



M5180 front panel

## Part of front panel

### Power Button



Switches the Analyzer ON and OFF.

The analyzer can be turned on/off at any time. The VNA loads its operating firmware from the PC each time upon powering up. The process will take approximately 10 seconds, after which the analyzer will be ready for operation.

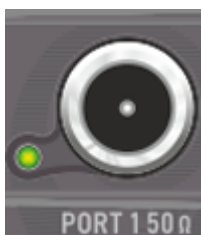
---

#### NOTE

The USB driver will be installed onto the PC when the analyzer is turned on for the first time. The driver installation procedure is described in [Software Installation](#). Some PCs may require re-installation of the driver in case of change of the USB port.

---

## Test Ports



Test Port 1 and 2 are intended for DUT connection. M models have type-N female test ports connector type.

Each test port has an LED indicator. A test port can be used either as a source of the stimulus signal or as a receiver of the response signal from the DUT. The stimulus signal can only appear on the only one port at a time.

Connecting the DUT to only one test port on the Analyzer allows the measurement of reflection parameters (e.g. S11 or S22) of the DUT.

Connecting the DUT to all test ports of the Analyzer allows for the measurement of the full S-parameter matrix of the DUT.

---

**NOTE**

The LED indicator identifies the test port which is operating as a signal source.

---

---

**CAUTION**

Do not exceed the maximum allowed power of the input RF signal (or maximum DC voltage) indicated on the front panel. This may lead to damage of the Analyzer.

---

### Ground Terminal

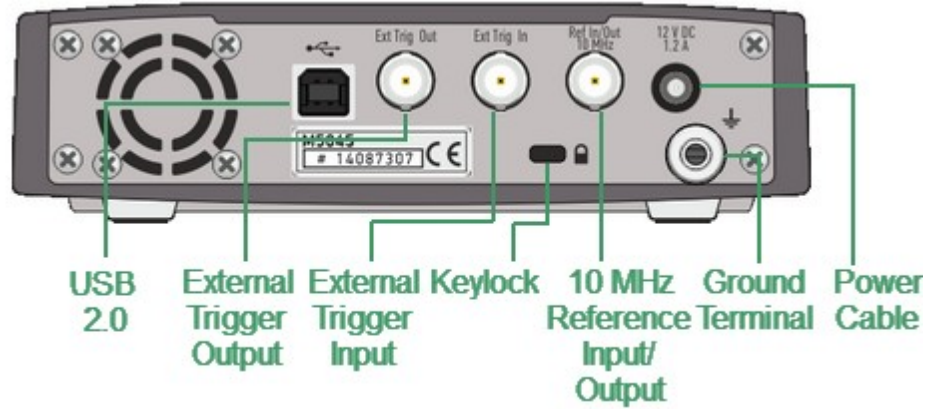


Use the terminal for grounding.

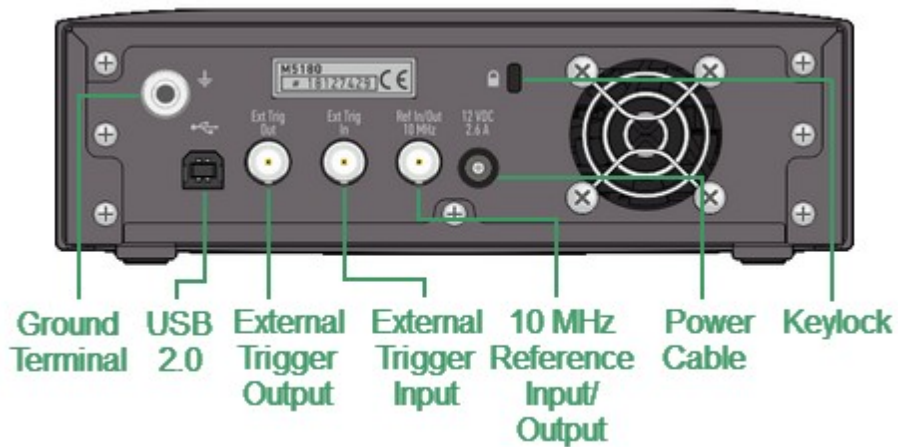
To avoid damage from electric discharge, connect the ground terminal on the body of the Analyzer to a reliable earth ground shared with the DUT in the test environment.

## Rear Panel

The rear view of the Analyzers is represented in the figures below.



M5045, M5065, M5090 rear panel



M5180 rear panel

## Parts of rear panel

### Power Cable Receptacle



The power supply receptacle is intended for an external DC power supply voltage from 9 to 15 V; alternatively, the power supply can be powered by a battery, including a vehicle battery, through an appropriate vehicle power cable. The DC connection requires a 3.5x1.35 mm plug with positive center conductor.

### External Trigger Signal Input Connector



This connector allows to connect an external trigger source. Connector type is BNC female. For input and signal parameters, see instrument specification.

### External Trigger Signal Output Connector



The External Trigger Signal Output port can be used to provide trigger to an external device. The port outputs signal with various waveforms depending on the setting of the Output Trigger Function: before frequency setup pulse, before sampling pulse, after sampling pulse, ready for external trigger, end of sweep pulse, measurement sweep.

### Reference Frequency Input/Output Connector



Output reference signal level is 3 dBm  $\pm$  2 dB at 50  $\Omega$  impedance. Connector type is BNC female.



## USB 2.0 High Speed Port



The USB port is intended for connection to an external PC.

## Ground Terminal



To avoid electric shock, use the this terminal for grounding.

The Ground terminal allows to connect directly the body of the Analyzer to the test station ground in order to ensure electrical safety.

## Keylock

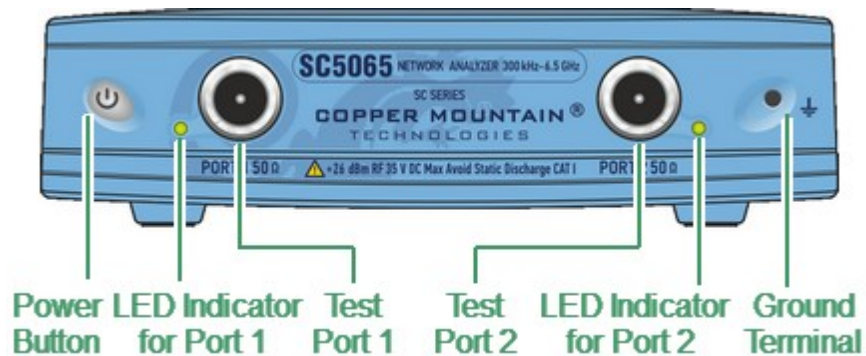


A Kensington lock (K-Slot) is part of an anti-theft system

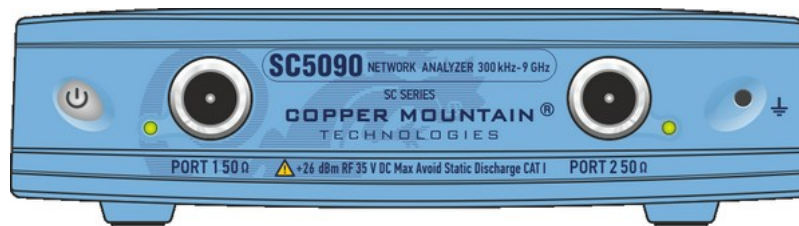
## SC Models, Front Panel

A Compact VNA models that deliver features excellent dynamic range, fast measurement speed, and high output power, in a compact package, with the maximum standard software feature set.

The front view of the Analyzers is represented in the figures below.



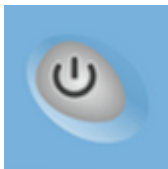
SC5065 front panel



SC5090 front panel

### Part of front panel

#### Power Button



Switches the Analyzer ON and OFF.

The analyzer can be turned on/off at any time. The VNA loads its operating firmware from the PC each time upon powering up. The process will take approximately 10 seconds, after which the analyzer will be ready for operation.

---

**NOTE**

The USB driver will be installed onto the PC when the analyzer is turned on for the first time. The driver installation procedure is described in [Software Installation](#). Some PCs may require re-installation of the driver in case of change of the USB port.

---

**Test Ports**

Test Port 1 and 2 are intended for DUT connection. SC models have type-N female test ports connector type.

Each test port has an LED indicator. A test port can be used either as a source of the stimulus signal or as a receiver of the response signal from the DUT. The stimulus signal can only appear on the only one port at a time.

Connecting the DUT to only one test port on the Analyzer allows the measurement of reflection parameters (e.g. S11 or S22) of the DUT.

Connecting the DUT to all test ports of the Analyzer allows for the measurement of the full S-parameter matrix of the DUT.

---

**NOTE**

The LED indicator identifies the test port which is operating as a signal source.

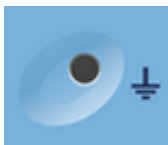
---

---

**CAUTION**

Do not exceed the maximum allowed power of the input RF signal (or maximum DC voltage) indicated on the front panel. This may lead to damage of the Analyzer.

---

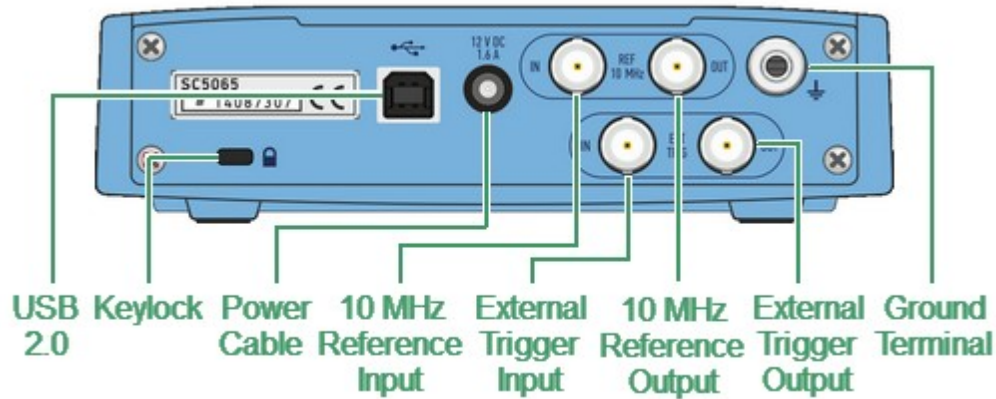
**Ground Terminal**

Use the terminal for grounding.

To avoid damage from electric discharge, connect the ground terminal on the body of the Analyzer to a reliable earth ground shared with the DUT in the test environment.

## Rear Panel

The rear view of the Analyzers is represented in the figures below.



SC5060, SC5090 rear panel

### Parts of rear panel

#### Power Cable Receptacle



The power supply receptacle is intended for an external DC power supply voltage from 9 to 15 V; alternatively, the power supply can be powered by a battery, including a vehicle battery, through an appropriate vehicle power cable. The DC connection requires a 3.5x1.35 mm plug with positive center conductor.

#### External Trigger Signal Input Connector



This connector allows to connect an external trigger source. Connector type is BNC female. For input and signal parameters, see instrument specification.

#### External Trigger Signal Output Connector



The External Trigger Signal Output port can be used to provide trigger to an external device. The port outputs signal with various waveforms depending on the setting of the Output Trigger Function: before frequency setup pulse,

before sampling pulse, after sampling pulse, ready for external trigger, end of sweep pulse, measurement sweep.

### Internal Reference Frequency Input Connector



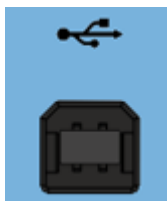
External reference frequency is 10 MHz, input level is 2 dBm  $\pm$  2 dB, input impedance at «Ref In» is 50  $\Omega$ . Connector type is BNC female.

### External Reference Frequency Output Connector



Output reference signal level is 3 dBm  $\pm$  2 dB at 50  $\Omega$  impedance. Connector type is BNC female.

### USB 2.0 High Speed Port



The USB port is intended for connection to an external PC.

### Ground Terminal



To avoid electric shock, use the this terminal for grounding.

The Ground terminal allows to directly connect the body of the Analyzer to the test station ground in order to ensure electrical safety.

## Keylock

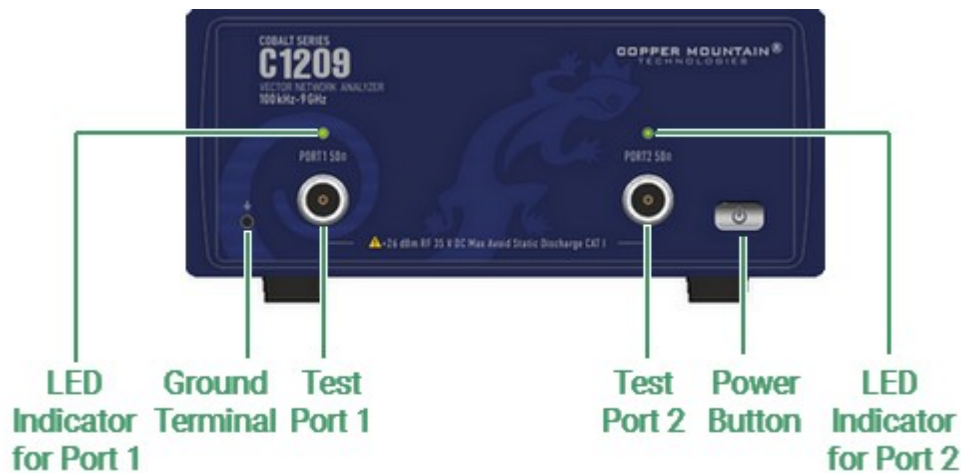


A Kensington lock (K-Slot) is part of an anti-theft system.

## Cobalt Series, Front Panel

This is a series of rack-mounted VNAs - a series that delivers industry-leading dynamic range and sweep speed, with all the features engineers have come to expect included standard in our software. This series also includes analyzers that are compatible with frequency extenders or have direct access to receivers. The Auxiliary Board Option is available on all Cobalt VNAs at the time of order (factory-installed) or as an add-on at a later date.

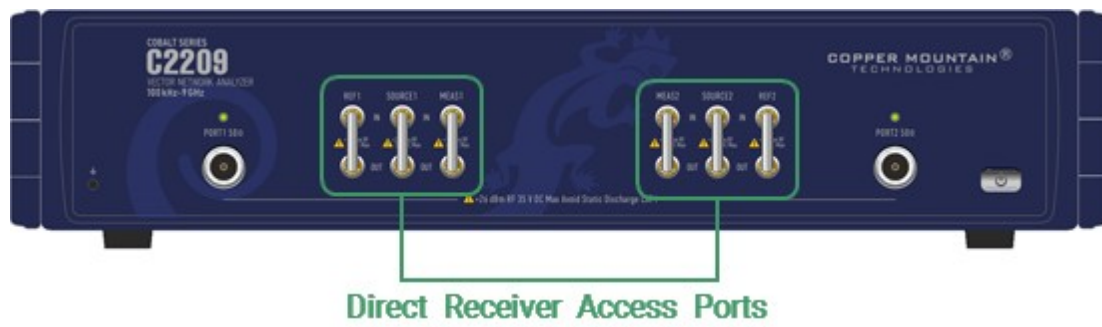
The front view of the Analyzers is represented in the figures below.



C1209 front panel



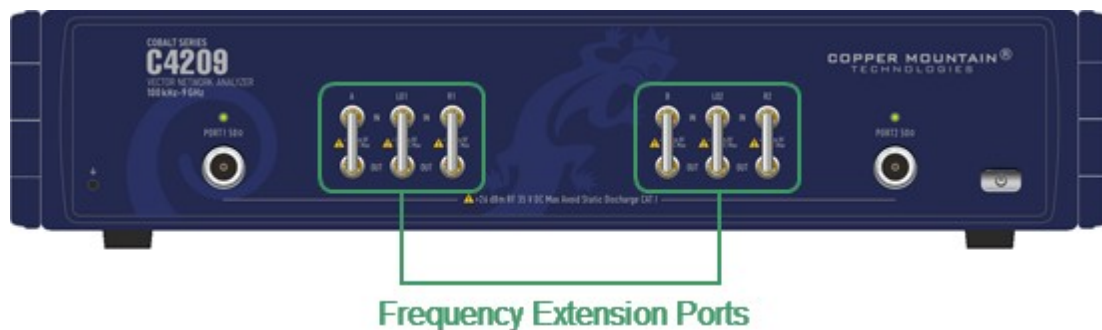
C1220 front panel



C2209 front panel



C2220 front panel



C4209 front panel

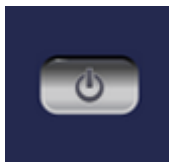




C4220 front panel

## Part of front panel

### Power Button



Switches the Analyzer ON and OFF.

The analyzer can be turned on/off at any time. The VNA loads its operating firmware from the PC each time upon powering up. The process will take approximately 10 seconds, after which the analyzer will be ready for operation.

### NOTE

The USB driver will be installed onto the PC when the analyzer is turned on for the first time. The driver installation procedure is described in [Software Installation](#). Some PCs may require re-installation of the driver in case of change of the USB port.

## Test Ports



Test Port 1 and 2 are intended for DUT connection. C1209, C2209, C4209 models have type-N female test ports connector type. C1220, C2220, C4220 models have NMD 3.5 mm male test ports connector type.

Each test port has an LED indicator. A test port can be used either as a source of the stimulus signal or as a receiver of the response signal from the DUT. The stimulus signal can only appear on the only one port at a time.



Connecting the DUT to only one test port on the Analyzer allows the measurement of reflection parameters (e.g. S11 or S22) of the DUT.

Connecting the DUT to all test ports of the Analyzer allows for the measurement of the full S-parameter matrix of the DUT.

---

#### NOTE

The LED indicator identifies the test port which is operating as a signal source.

---

#### CAUTION

Do not exceed the maximum allowed power of the input RF signal (or maximum DC voltage) indicated on the front panel. This may lead to damage of the Analyzer.

---

### Ground Terminal



Use the terminal for grounding.

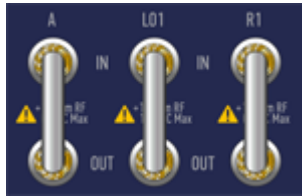
To avoid damage from electric discharge, connect the ground terminal on the body of the Analyzer to a reliable earth ground shared with the DUT in the test environment.

### Adjustable Ports Configurations (C2209, C2220 models only)



Adjustable port configurations with direct access to the receivers of the VNA provide for a variety of test applications requiring wider dynamic and power range. Direct receiver access enables testing of high power devices. Additional amplifiers, attenuators, various filters and matching pads for each of the ports may be introduced in reference oscillator and receiver path to ensure optimal operation of the receivers.

## Additional Ports for Frequency Extension System (C4209, C4220 models only)

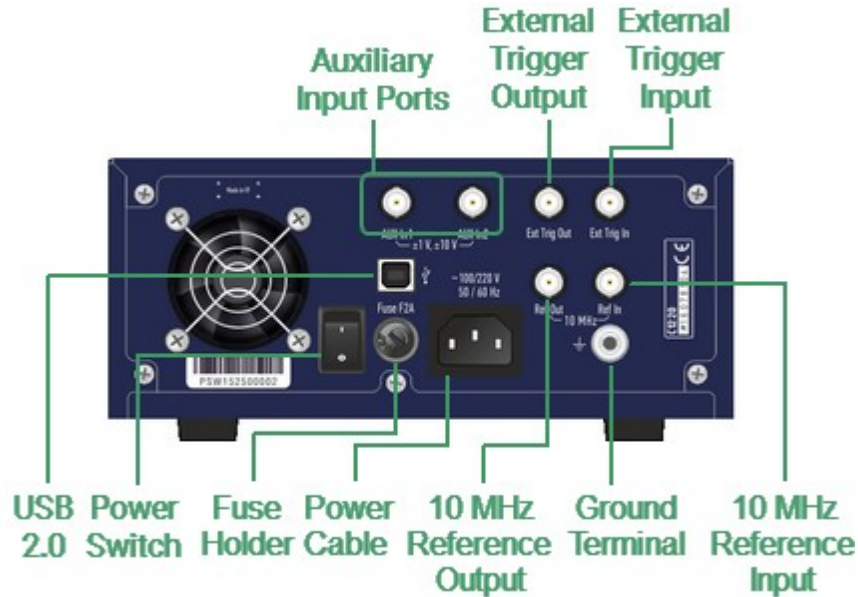


C4209, C4220 analyzers have additional ports on the front panel for connecting frequency extension system.

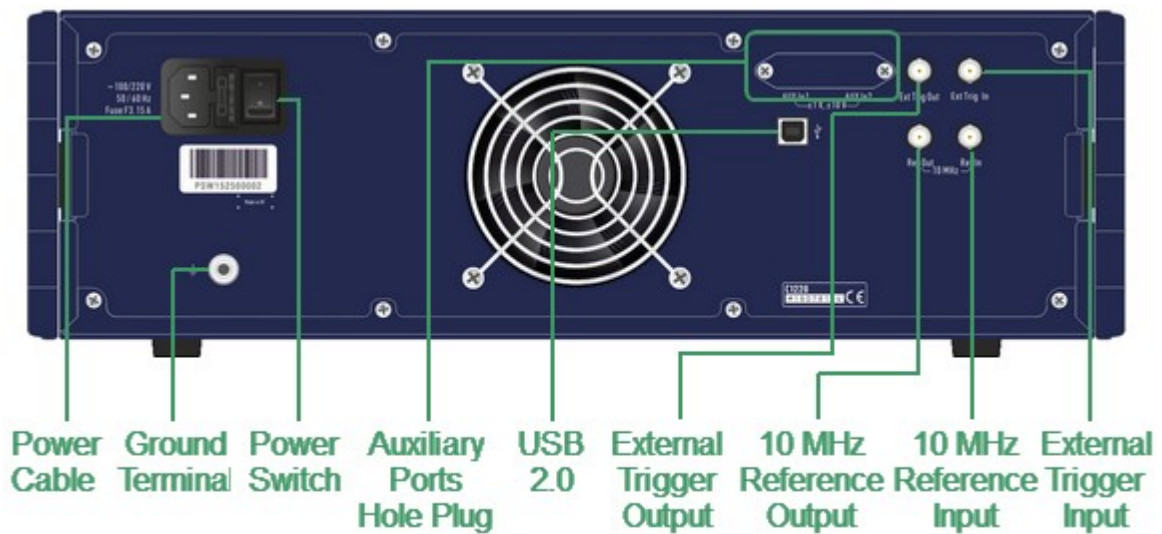
The signal from the analyzer's local oscillator (LO1 output) is supplied to the frequency extension system. Signals from the extension system are fed to the inputs of the test and reference receivers of the analyzer (A in and R1 in). Unless frequency extension system is used, each pair of ports is connected with a jumper cable assembly.

## Rear Panel

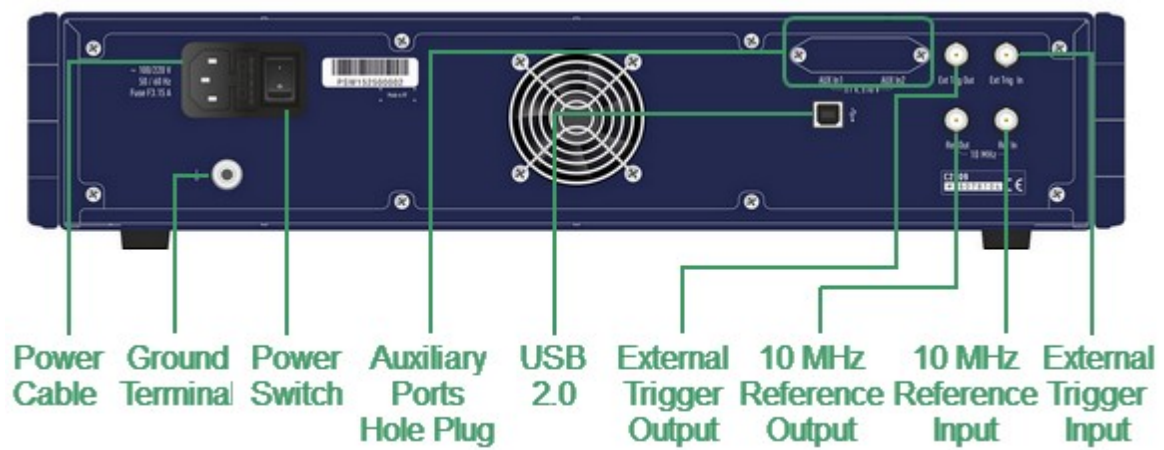
The rear view of the Analyzers is represented in the figures below.



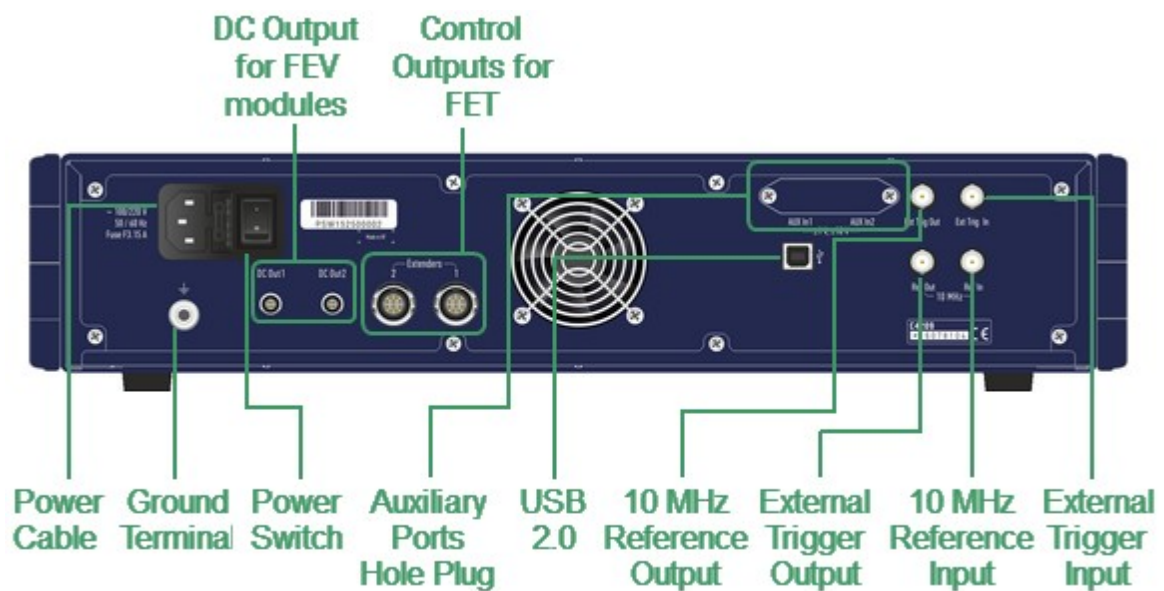
C1209 rear panel



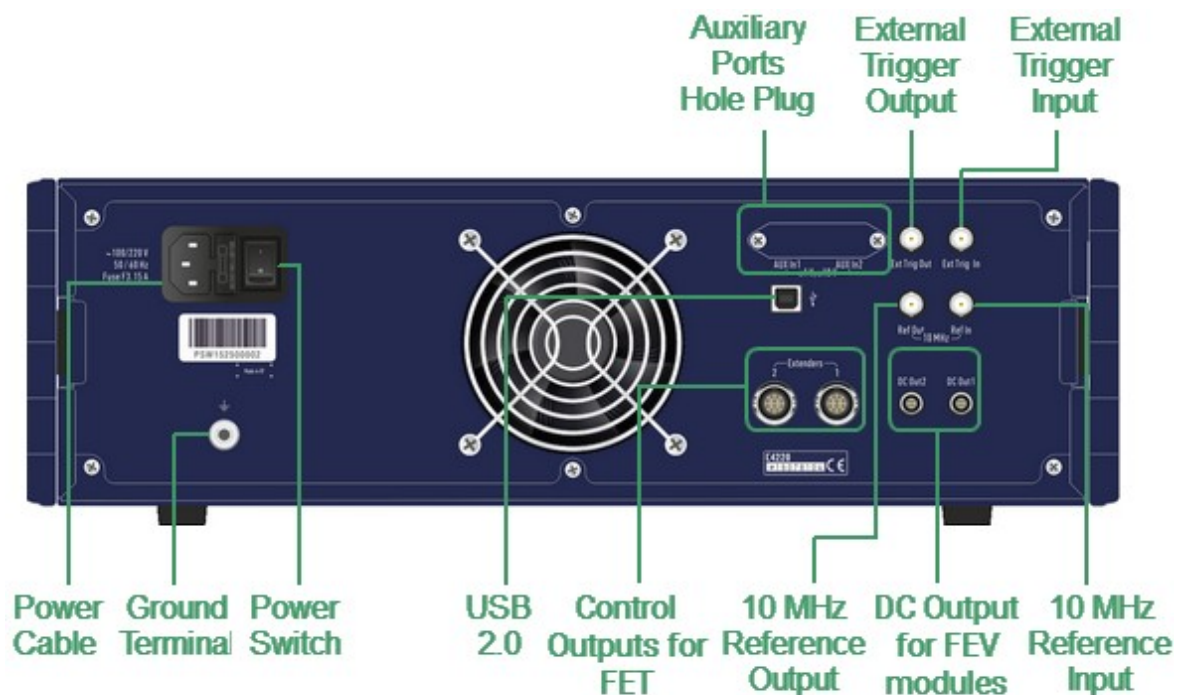
C1220, C2220 rear panel



C2209 rear panel



C4209 rear panel



C4220 rear panel

## Part of rear panel

### Power Cable Receptacle



The power cable receptacle is intended for 100 VAC to 240 VAC 50/60 Hz power cable connection. The cable receptacle has a built-in switch and fuse holder (except C1209 model). The power switch serves as the disconnecting device (device that cuts off power supply) of the VNA. The power supply must be cut off to avoid such danger is electric shock, during prolonged non-use of the device. A Fuse protects the Analyzer from the excessive current.

### Power Switch (C1209 only)



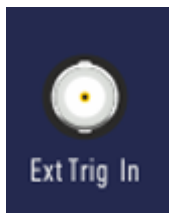
The power switch serves as the disconnecting device (device that cuts off power supply) of the VNA. The power supply must be cut off to avoid such danger is electric shock, during prolonged non-use of the device.

### Fuse Holder (C1209 only)



A Fuse protects the Analyzer from the excessive current.

### External Trigger Signal Input Connector



This connector allows to connect an external trigger source. Connector type is BNC female. For input and signal parameters, see instrument specification.

### External Trigger Signal Output Connector



The External Trigger Signal Output port can be used to provide trigger to an external device. The port outputs signal with various waveforms depending on the setting of the Output Trigger Function: before frequency setup pulse, before sampling pulse, after sampling pulse, ready for external trigger, end of sweep pulse, measurement sweep.

### Internal Reference Frequency Output Connector



External reference frequency is 10 MHz, input level is 2 dBm  $\pm$  2 dB, input impedance at «Ref In» is 50  $\Omega$ . Connector type is BNC female.

### External Reference Frequency Input Connector



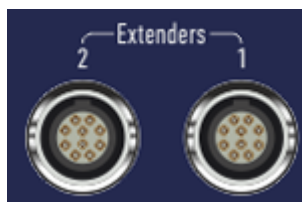
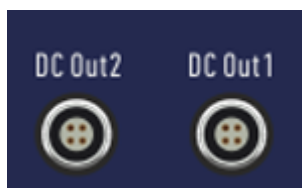
Input reference signal level is  $3 \text{ dBm} \pm 2 \text{ dB}$  at  $50 \Omega$  impedance. Connector type is BNC female.

### USB 2.0 High Speed Port



The USB port is intended for connection to an external PC.

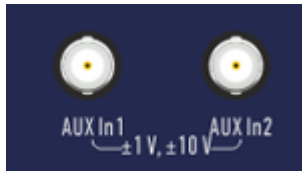
### Power and Control Connectors for Frequency Extension System (C4209, C4220 only)



External frequency extenders are powered by a DC voltage from the analyzer power supply. DC Out connectors are used for powering FEV frequency extension modules. Extender connectors are used for powering and controlling FET frequency extension modules. These connectors include power and control lines.



### Auxiliary Input Ports (HW-C-AUX option for Cobalt series)



A two-channel DC voltmeter board can be optionally included in the Analyzer. A DC voltmeter measures voltage synchronously with the sweeping frequency when measuring S-parameters. Two additional ports AUX in1 and AUX in2 are voltmeter inputs.



The hole for the connectors is closed with a plug if the option is not present.

### Ground Terminal

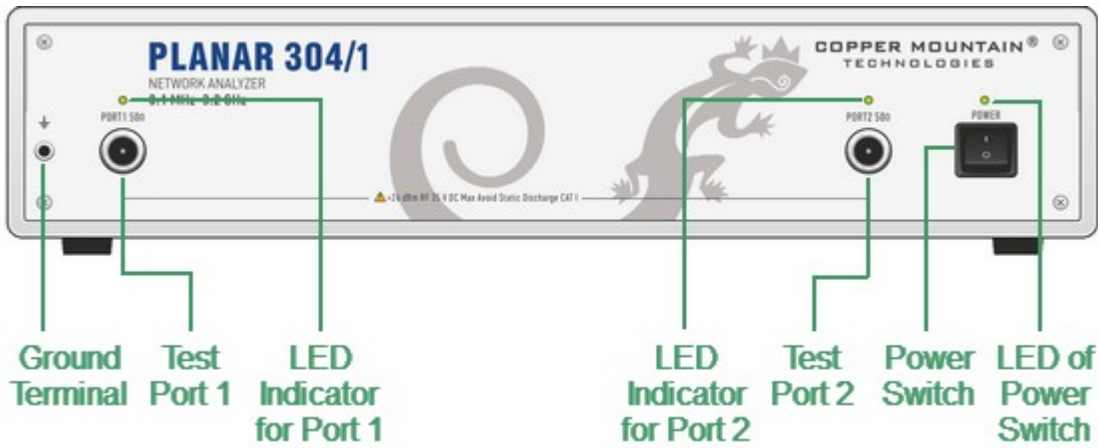


To avoid electric shock, use the this terminal for grounding. The Ground terminal allows to connect directly the body of the Analyzer to the test station ground in order to ensure electrical safety.

# Planar Series, Front Panel

This is a series of full-size rack-mounted VNAs that delivers lab grade performance, with the maximum standard software feature set (except Planar 304/1). This series includes an Analyzer with direct access to receivers.

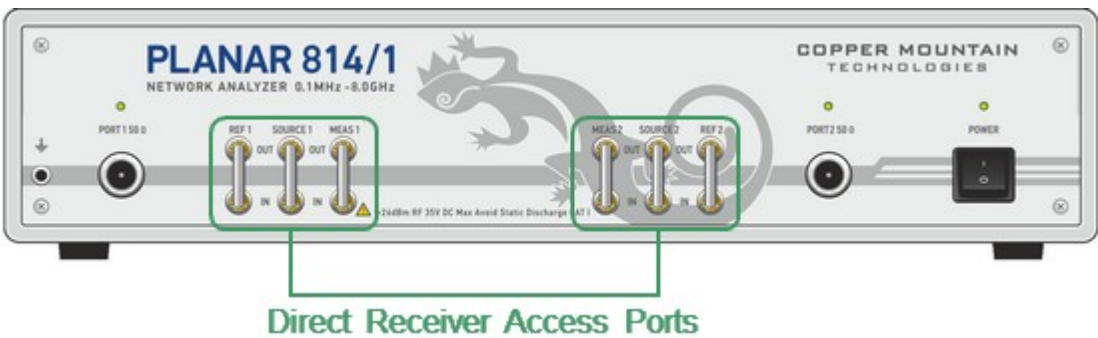
The front view of the Analyzers is represented in the figures below.



Planar 304/1 front panel



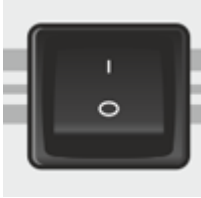
Planar 804/1 front panel



Planar 814/1 front panel

## Part of front panel

### Power Switch



Switches the Analyzer ON and OFF. The switch interrupts the power line of the instrument in this model. The Analyzer can be turned on/off at any time. The VNA loads its operating firmware from the PC each time upon powering up. The process will take approximately 10 seconds, after which the analyzer will be ready for operation.

The LED of power is on when the analyzer is running.

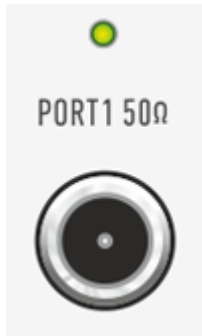
---

#### NOTE

The USB driver will be installed onto the PC when the analyzer is turned on for the first time. The driver installation procedure is described in [Software Installation](#). Some PCs may require re-installation of the driver in case of change of the USB port.

---

## Test Ports



Test Port 1 and 2 are intended for DUT connection. Planar models have type-N female test ports connector type.

Each test port has an LED indicator. A test port can be used either as a source of the stimulus signal or as a receiver of the response signal from the DUT. The stimulus signal can only appear on the only one port at a time.

Connecting the DUT to only one test port on the Analyzer allows the measurement of reflection parameters (e.g. S11 or S22) of the DUT.

Connecting the DUT to all test ports of the Analyzer allows for the measurement of the full S-parameter matrix of the DUT.

---

### NOTE

The LED indicator identifies the test port which is operating as a signal source.

---

### CAUTION

Do not exceed the maximum allowed power of the input RF signal (or maximum DC voltage) indicated on the front panel. This may lead to damage of the Analyzer.

---

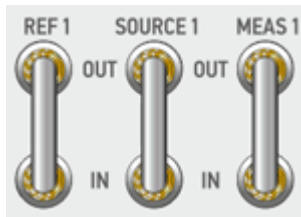
## Ground Terminal



Use the terminal for grounding.

To avoid damage from electric discharge, connect the ground terminal on the body of the Analyzer to a reliable earth ground shared with the DUT in the test environment.

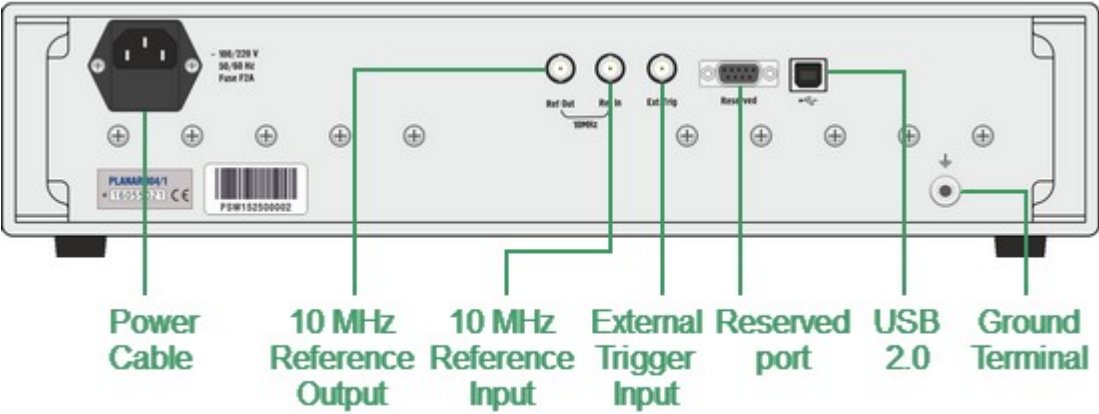
## Adjustable Ports Configurations (Planar 814/1 model only)



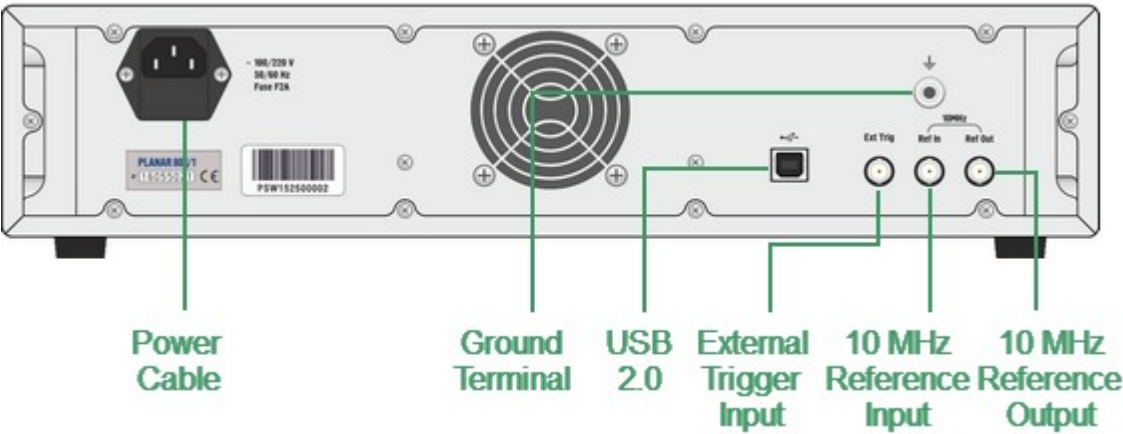
Adjustable port configurations with direct access to the receivers of the VNA provide for a variety of test applications requiring wider dynamic and power range. Direct receiver access enables testing of high power devices. Additional amplifiers, attenuators, various filters and matching pads for each of the ports may be introduced in reference oscillator and receiver path to ensure optimal operation of the receivers.

# Rear Panel

The rear view of the Analyzers is represented in the figures below.



PLANAR 304/1 rear panel



PLANAR 804/1, PLANAR 814/1 rear panel

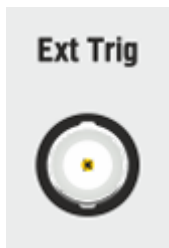
## Parts of rear panel

### Power Cable Receptacle



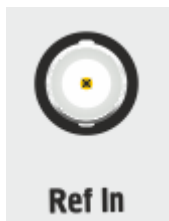
The power cable receptacle is intended for 100 VAC to 240 VAC 50/60 Hz power cable connection.

### External Trigger Signal Input Connector



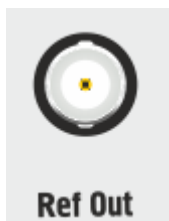
This connector allows to connect an external trigger source. Connector type is BNC female. For input and signal parameters, see instrument specification.

### Internal Reference Frequency Input Connector



External reference frequency is 10 MHz, input level is 2 dBm  $\pm$  2 dB, input impedance at «Ref In» is 50  $\Omega$ . Connector type is BNC female.

### Internal Reference Frequency Output Connector



Output reference signal level is 3 dBm  $\pm$  2 dB at 50  $\Omega$  impedance. Connector type is BNC female.

### USB 2.0 High Speed Port



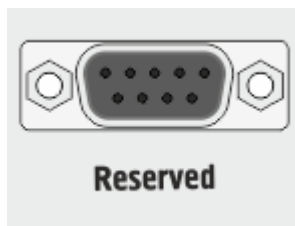
The USB port is intended for connection to an external PC.

### Ground Terminal



To avoid electric shock, use the this terminal for grounding. The Ground terminal allows to connect directly the body of the Analyzer to the test station ground in order to ensure electrical safety.

### Reserved Port (Planar 304/1 model only)



Do not use this port.



## Preparation for Use

Unpack the Analyzer and other accessories.

---

### CAUTION

Please keep packaging to safely ship the instrument for annual calibration!

---

First, install the S2VNA software from the shipped flash-drive or [www.coppermountaintech.com](http://www.coppermountaintech.com) on the PC that will be used to operate of the Analyzer. The software installation procedure is described in [Software Installation](#).

Connect the Analyzer to a 100 to 240 VAC 50/60 Hz power source by means of the external power supply (S models and M models) or power cable (Planar and Cobalt models) supplied with the instrument. Connect the USB-port of the Analyzer to the PC using the USB Cable supplied in the package.

Warm up the Analyzer for the time stated in its [datasheet](#).

Assemble the test setup using cables, connectors, fixtures, etc., which allow DUT connection to the Analyzer.

Perform calibration of the Analyzer. Calibration procedures are described in [Calibration and Calibration Kit](#).

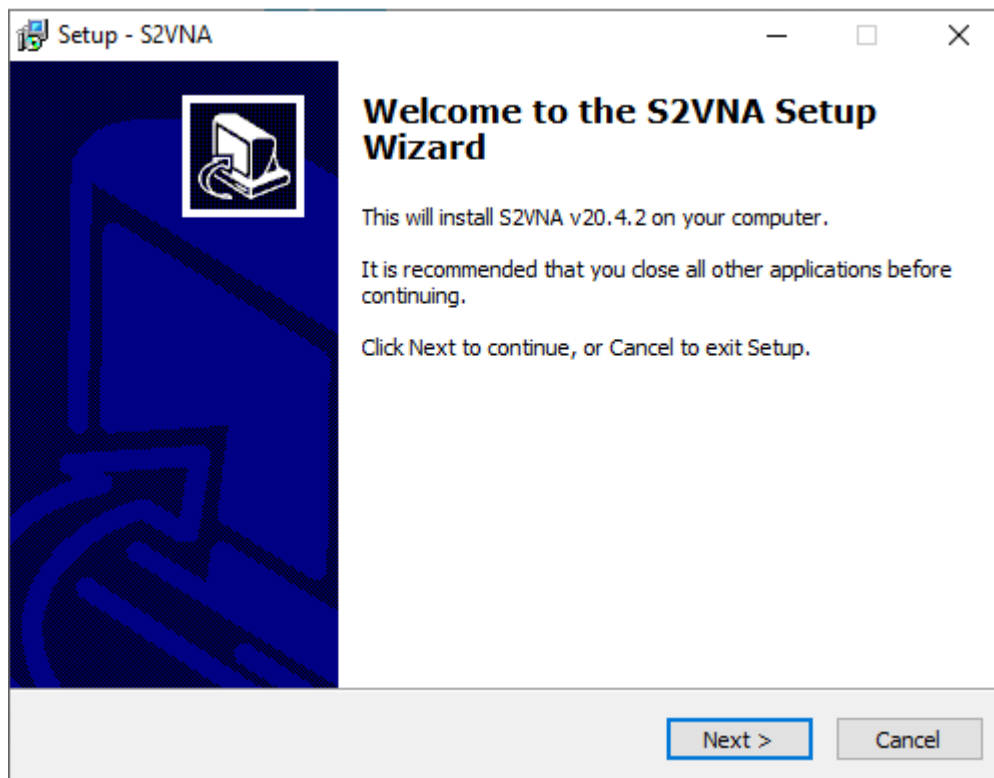
## Software Installation

Minimal system requirements	<p>x86 compatible PC running WINDOWS or LINUX</p> <p>WINDOWS 7 or higher</p> <p>LINUX Ubuntu 14.04, Linux Mint 17, Linux Debian 8.9 or higher</p> <p>1.5 GHz Minimum Clock Speed</p> <p>4 GB RAM Minimum</p> <p>USB 2.0 High Speed</p>
-----------------------------	--

### Windows installation procedure

Find the analyzer software installer file Setup\_S2VNA\_vX.X.X.exe in the shipped flash-drive or download it from [www.coppermountaintech.com](http://www.coppermountaintech.com). Where X.X.X stands for version number.

Run the Setup\_S2VNA\_vX.X.X.exe installer file. Follow the instructions of the installation wizard.



Installation wizard

Default installation paths for files in WINDOWS	Software components	Path
	S2VNA.exe	C:\VNA\S2VNA
	VNA driver	C:\VNA\S2VNA\Driver
	Documentation	C:\VNA\S2VNA\Doc
	Programming Examples	C:\VNA\S2VNA\Programming Examples
	Data Files	C:\VNA\S2VNA

## Linux installation procedure

1. Download the analyzer software file CMT\_S2VNA\_X.X.X\_x86\_64.appimage from [www.coppermountaintech.com](http://www.coppermountaintech.com), where X.X.X is stands for version number.
2. Make it executable

```
chmod a+x S2VNA_X.X.X_x86_64.ApplImage
```

### 3. Run

```
$ ./S2VNA_X.X.X_x86_64.ApplImage
```

First time app will ask to enter root password for adding permissions for working with USB devices. You can do it by yourself by adding file /etc/udev/rules.d/cmt.rules containing:

```
SUBSYSTEM=="usb", ATTRS{idVendor}=="2226", MODE="0666"  
SUBSYSTEM=="usb_device", ATTRS{idVendor}=="2226", MODE="0666"
```

Operating and Programming manual is the same as for the Windows application, except that COM automation does not apply. Download it from [www.coppermountaintech.com](http://www.coppermountaintech.com).

User's data file location:

```
$ ~/.vna-portable/drive_c/users/<user>/Application Data/S2VNA
```

### Running More Than One Device on a PC in Linux

Up to 16 environment configurations can be used. It's allowed to store individual settings for different devices without copy application:

```
$ ./S2VNA_X.X.X_x86_64.ApplImage -Conf=1  
$ ./S2VNA_X.X.X_x86_64.ApplImage -Conf=2
```

All application settings stored in Linux current user's folder:

```
$ ~/.vna-portable
```

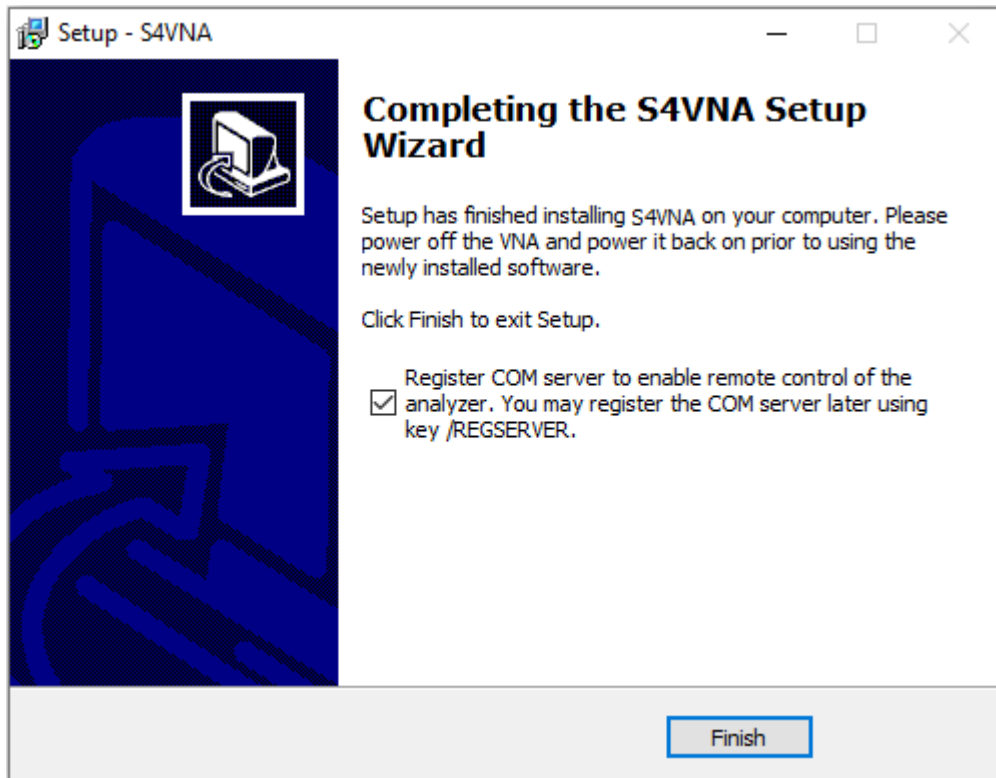
User's settings for all configurations stored inside it:

```
$ ~/.vna-portable/drive_c/users/<user>/Application Data/S2VNA  
$ ~/.vna-portable/drive_c/users/<user>/Application Data/S2VNA.1  
$ ~/.vna-portable/drive_c/users/<user>/Application Data/S2VNA.2
```

## Registering COM Server

Registration of the COM server is required when using COM automation. If using SCPI automation or if automation is not required, registration can be skipped.

Registration of the COM server is performed during the installation of the analyzer software. The COM server registration window is shown in the figure below.



Registering COM Server

Registration can be done after installing the software. To register the COM server, run the executable module SxVNA from the command prompt with the /regserver keyword. To unregister the COM server of the analyzer, run the executable module from the command prompt with the /unregserver keyword. Administrative rights are required to register/unregister COM server.

The following is an example of the COM server registration command:

```
S2VNA.exe /regserver
```

## Command Line Parameters

Below is the full list of supported parameters for the S2VNA command line.

S2VNA [optional parameters]

For example:

S2VNA /visible:off

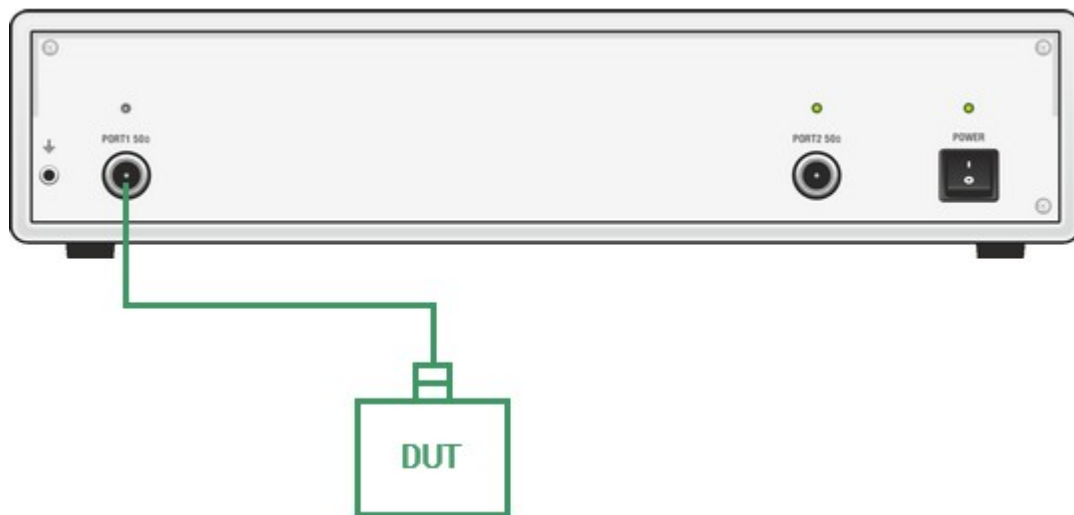
Parameter	Description
/?	Displays the help message
/SocketServer:<on off>	Enables or disables TCP/IP socket server
/SocketPort:<num>	Assigns socket server port [default is 5025]
/HislipServer:<on off>	Enables or disables HiSLIP server
/HislipPort:<num>	Assigns HiSLIP server port [default is 4880]
/SerialNumber:<num>	Connects to the Analyzer with the specified serial number
/simulate:<on off>	Turnes on/off simulation [default is off]
/visible:<on off>	Show or hides GUI [default is on]
/regserver	Registers COM server
/unregserver	Unregisters COM server

## Getting Started

This section is organized as a sample session of the Analyzer. It describes the main techniques for measurement by example measuring of the reflection coefficient parameters of the DUT. SWR and reflection coefficient phase of the DUT will be analyzed.

In this example, only one test port of the analyzer is used for reflection coefficient measurement. The instrument sends the stimulus to the input of the DUT and then receives the reflected wave. If the DUT is a two-port device, its unused port should be terminated with a LOAD standard. The results of these measurements can be represented in various formats.

A typical setup for reflection coefficient measurement is shown below.



Reflection Measurement Circuit

To measure SWR and reflection coefficient phase of the DUT in the given example, go through the following steps:

- Prepare the Analyzer for reflection measurement.
- Set stimulus parameters (frequency range, number of points).
- Set IF bandwidth.
- Set the number of traces to 2, assign measured parameters and display format to the traces.
- Set the scale of the traces.
- Perform a one-port calibration.
- Analyze SWR and reflection coefficient phase using markers.

---

**NOTE**

The analyzer can be controlled via softkey panel located on the right-hand part of the screen. The analyzer also allows to perform quick control by the mouse (See [Quick Setting Using a Mouse](#)).

---



## Analyzer Preparation for Reflection Measurement

Turn on the Analyzer and warm it up for the period of time stated in its [specifications](#) (40 minutes typically).

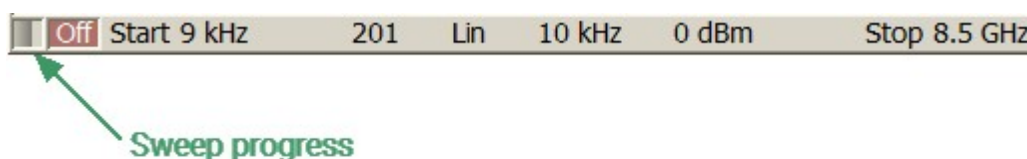
### Ready status indication

The bottom line of the screen displays the instrument status bar. It should read **Ready**.



### Sweep progress indication

The channel status bar is located above this bar. The sweep indicator in the left-hand part of this bar should display sweep progress



Connect the DUT to Port 1 of the Analyzer. The DUT can be connected directly to the port if the type of connectors are the same and the gender is opposite. Otherwise use the appropriate cables and adapters for connection of the DUT input to the Analyzer test port. Use a calibrated torque wrench for tightening the connectors.

## Analyzer Presetting

Before starting the measurement session, reset the analyzer to the initial condition. The initial condition setting is described in [Annex 1](#).

---

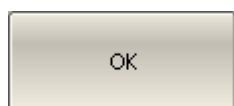
### NOTE

Softkeys controlling the analyzer are located on the vertical panel on the right side of the analyzer screen (See [Softkey Menu Bar](#)).

---



To restore the initial condition of the Analyzer, use the following softkeys:



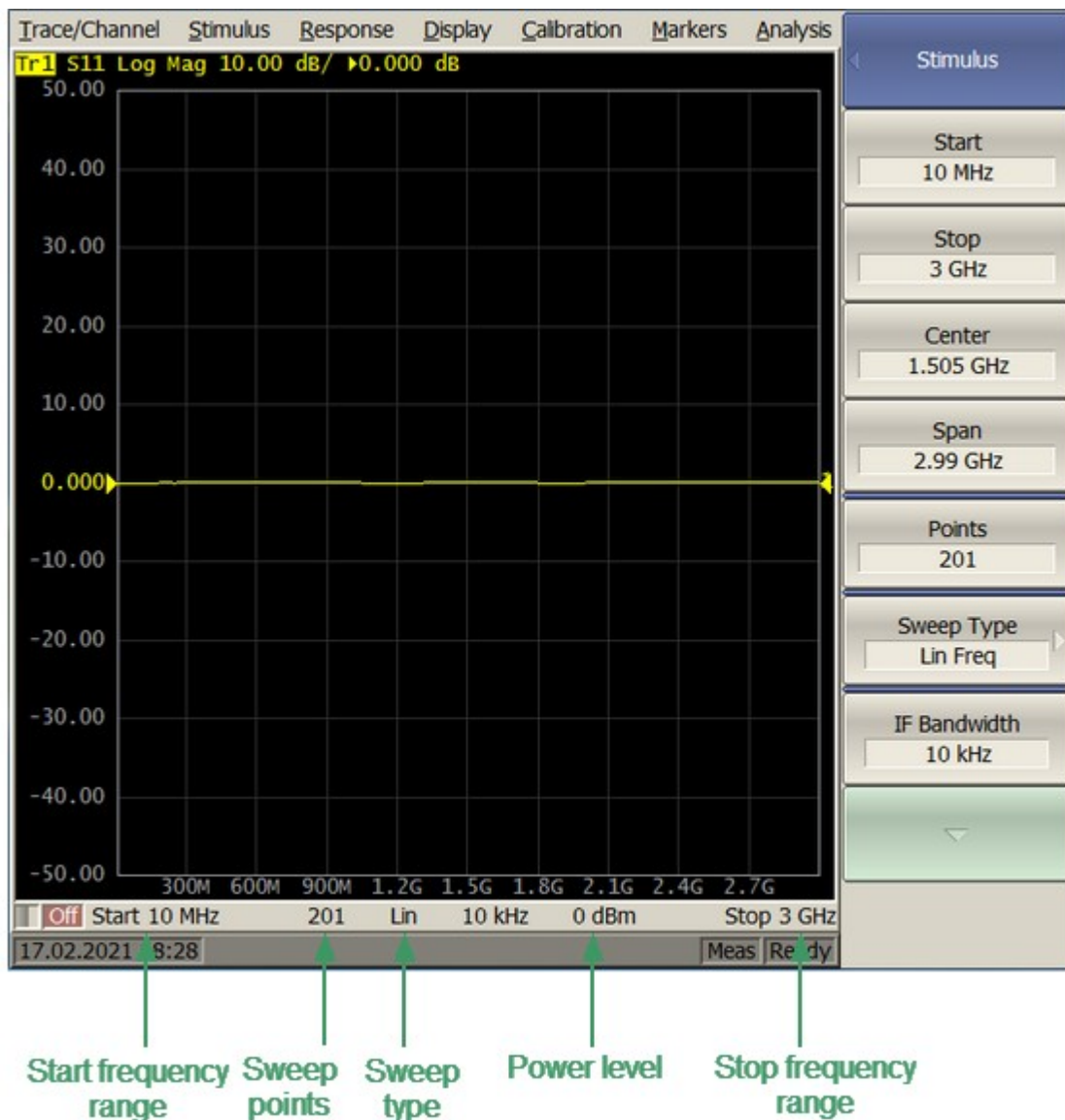
**System > Preset > OK**

---

## Stimulus Setting

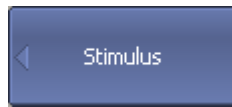
After presetting the analyzer, the stimulus parameters will be as follows:

- Full frequency range of the instrument
- Linear sweep type
- 201 points
- Power level of 0 dBm

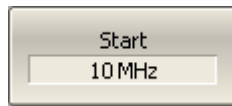


Settable parameters in the channel status bar

For the current example, set the frequency range to from 10 MHz to 3 GHz.

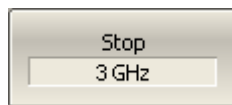


To set the start frequency of the frequency range to 10 MHz, use the following softkeys:



**Stimulus > Start**

Then enter «1», «0» from the keyboard. Complete the setting by pressing the «M» key. (Capital "M").



To set the stop frequency of the frequency range to 3 GHz, use the following softkeys:

**Stimulus > Stop**

Then enter «3» from the keyboard. Complete the setting by pressing «G» key.

---

To return to the main menu, click the top softkey (colored in blue).

---

---

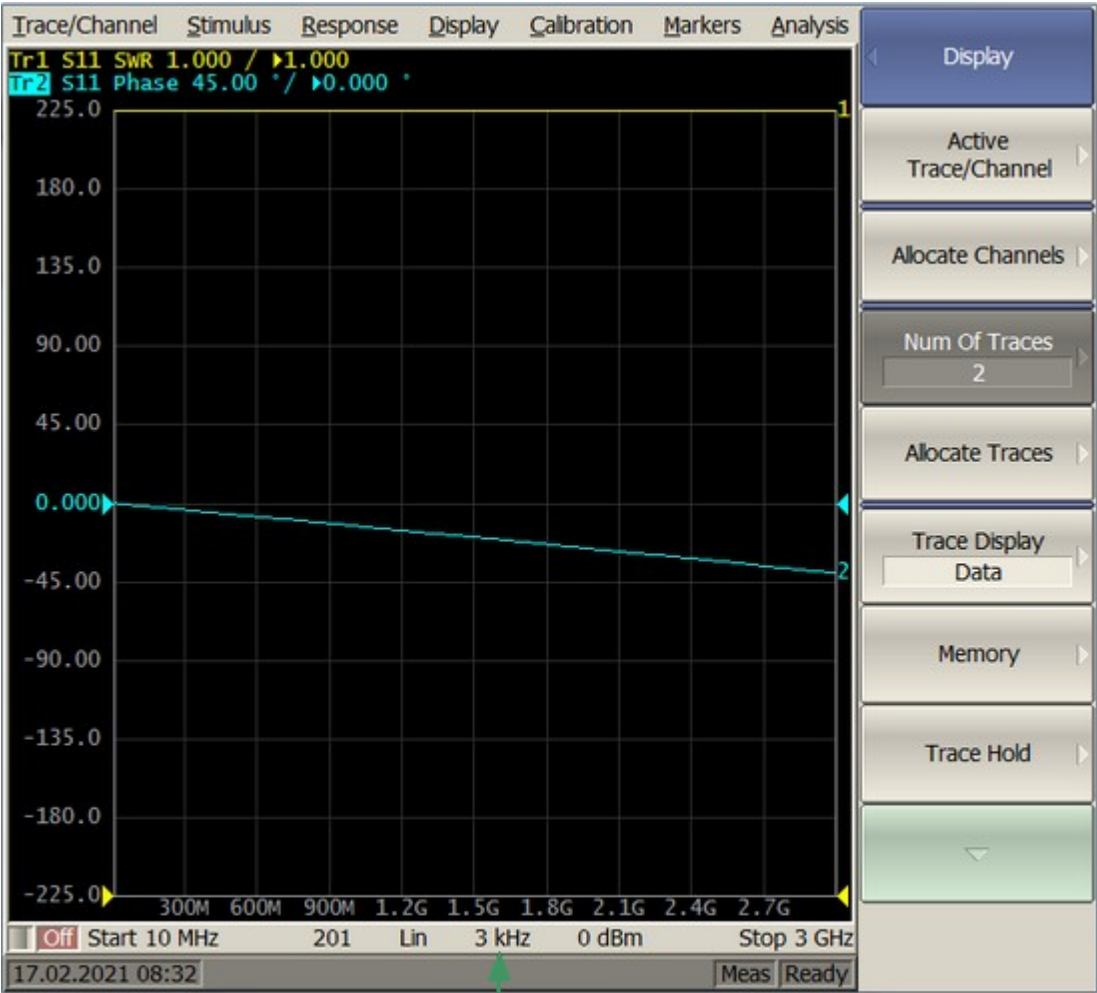
**NOTE**

The **Start** and **Stop** values of the frequency range can be set using the mouse (See [Sweep Start Setting](#)).

---

# IF Bandwidth Setting

For the current example, set the IF bandwidth to 3 kHz.



IF bandwidth

IF bandwidth value in the channel status bar



To set the IF bandwidth to 3 kHz, use the following softkeys:

**Average > IF Bandwidth**



Then enter «3» from the keyboard and complete the setting by pressing the «k» key.

---

To return to the main menu, click the top softkey (colored in blue).

---

---

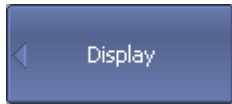
**NOTE**

IF bandwidth can be set using the mouse (See [IF Bandwidth Setting](#)).

---

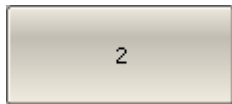
## Number of Traces, Measured Parameter and Display Format Setting

In the current example, two traces are used for simultaneous display of the two parameters (SWR and reflection coefficient phase).



To set the number of traces, use the following softkeys:

**Display > Num of Traces > 2**

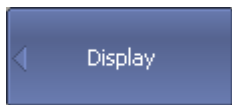


---

To return to the main menu, click the top softkey (colored in blue).

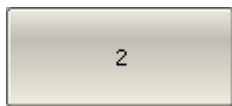
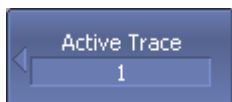
---

Activate the trace before assigning the measurement parameters.



To activate the second trace, use the following softkeys:

**Display > Active Trace/Channel > Active Trace > 2**



---

To return to the main menu, click the top softkey (colored in blue).

---

---

### NOTE

The active trace can be selected using the mouse (See [Active Trace Selection](#)).

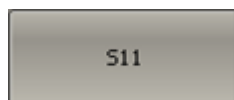
---

Assign the S11-parameter to the second trace. This parameter is already assigned to the first trace by default.



To assign a parameter to the trace, use the following softkeys:

**Measurement > S11**



---

**NOTE**

The measured parameter can be set using the mouse (See [Measured Data Setting](#)).

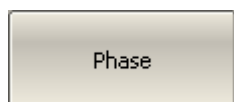
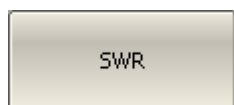
Then, assign SWR display format to the first trace and the reflection coefficient phase display format to the second trace.



To set the active trace display format, use the following softkeys:

**Format > SWR** (for the first trace),

**Format > Phase** (for the second trace).



To return to the main menu, click the top softkey (colored in blue).

---

**NOTE**

The display format can be set using the mouse (See [Display Format Setting](#)).



The set parameters will be displayed in the trace status bar (See figure below).

Trace/Channel	Stimulus	Response	Display	Calibration	Markers
Tr1	S11	SWR	1.000 / ▶1.000		
Tr2	S11	Phase	45.00 ° / ▶0.000 °		
	225.0				

Active trace    Measured parameter    Display format

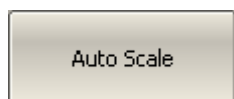
Settable parameters in the trace status field

## Trace Scale Setting

For convenience of operation, change the trace scale using automatic scaling function (See [Automatic Scaling](#)).



To set the scale of the active trace by the auto-scaling function, use the following softkeys:



**Scale > Auto Scale**

---

To return to the main menu, click the top softkey (colored in blue).

---

### NOTE

The trace scale can also be set manually using the softkeys or using the mouse (Setting with softkeys is described in [Rectangular Scale](#), setting by mouse in [Trace Scale Setting](#)).

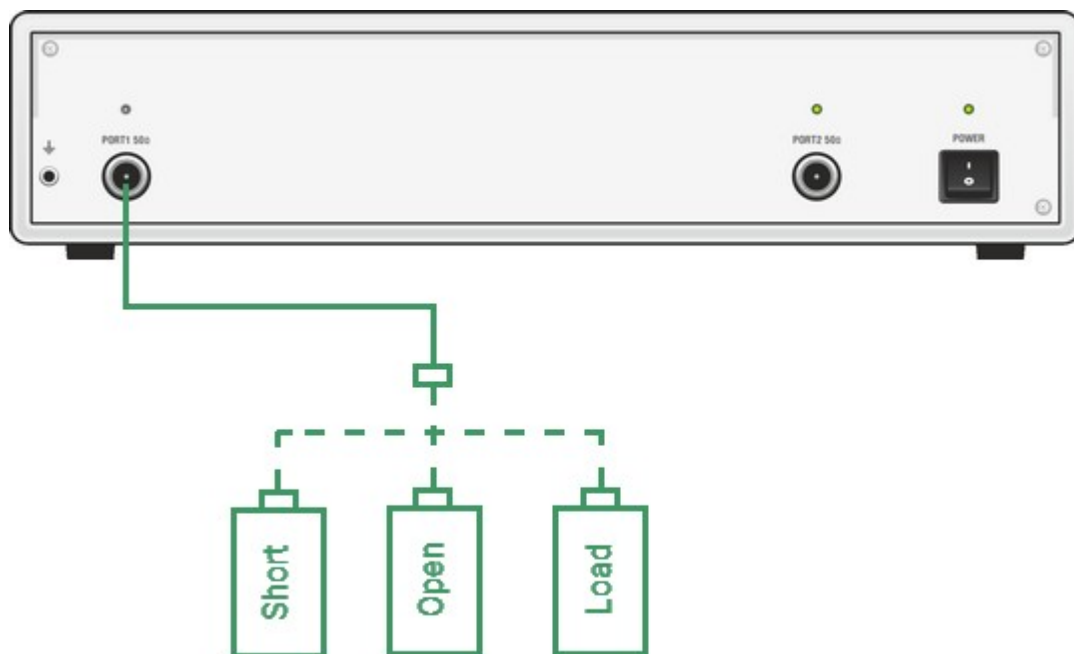
---

## Analyzer Calibration for Reflection Coefficient Measurement

Calibration of the entire measurement setup — which includes the Analyzer, cables and adapters involved for the DUT connection — greatly enhances the accuracy of the measurement.

To perform full one-port calibration, prepare the kit of calibration standards: OPEN, SHORT and LOAD. To perform proper calibration, select the correct kit type in the program. This kit contains a description and specifications of the standards

To perform full one-port calibration, connect calibration standards to the test port one after another and perform measurement, as shown below.



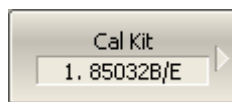
Full one-port calibration circuit

An Keysight 85032E calibration kit is used in this example.



To select the calibration kit, use the following softkeys:

**Calibration > Cal Kit**

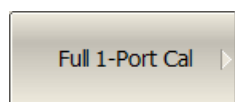


Then select the kit being used from the table at the bottom of the screen (See figure below).

	Label	Description	Select	Predefined	Modified	#STDs	
2	85032B/E	Type-N 50Ω 6GHz Cal Kit (Agilent)	<input type="checkbox"/>	Yes	No	6	
3	85032F	Type-N 50Ω 9GHz Cal Kit (Agilent)	<input type="checkbox"/>	Yes	No	8	
4	85054B	Type-N 50Ω 18GHz Cal Kit with Sliding Load (Agilent)	<input type="checkbox"/>	Yes	No	8	
5	85054D	Type-N 50Ω 18GHz Cal Kit (Agilent)	<input type="checkbox"/>	Yes	No	6	
6	05CK10A-150	Type-N 50Ω 18GHz Cal Kit (Rosenberger)	<input type="checkbox"/>	Yes	No	6	
7	8850Q	Type-N 50Ω 18GHz Cal Kit (Maurly Microwave)	<input checked="" type="checkbox"/>	Yes	No	6	
8	8850C	Type-N 50Ω 18GHz Cal Kit with Sliding Load (Maurly Mic)	<input type="checkbox"/>	Yes	No	8	
9	85033D/E	3.5 mm 6GHz/9GHz Cal Kit (Agilent)	<input type="checkbox"/>	Yes	No	6	
10	85052B	3.5 mm 26.5GHz Cal Kit with Sliding Load (Agilent)	<input type="checkbox"/>	Yes	No	10	

### Calibration kits list

To perform full one-port calibration (SOL), execute measurements of the three standards in turn. After completion, the table of calibration coefficients will be calculated and saved into the memory of the Analyzer. Before starting calibration, disconnect the DUT from the Analyzer.

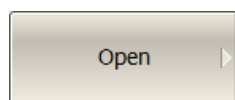


To perform full one-port calibration, use the following softkeys:

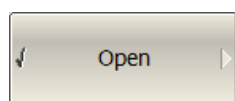
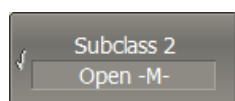


#### Calibration > Calibrate > Full 1-Port Cal

Connect an OPEN standard and click **Open**.



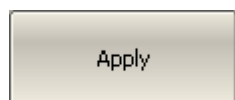
Select a subclass of the OPEN standard according to the gender of its connector (**Male/Female**). The instrument status bar will indicate **Calibration in progress...** while the measurement is in progress. On completion of the measurement a check mark will appear on the left side of the softkey.



Connect a SHORT standard and follow the same procedure as with the OPEN standard.



Connect a LOAD standard and click **Load**.

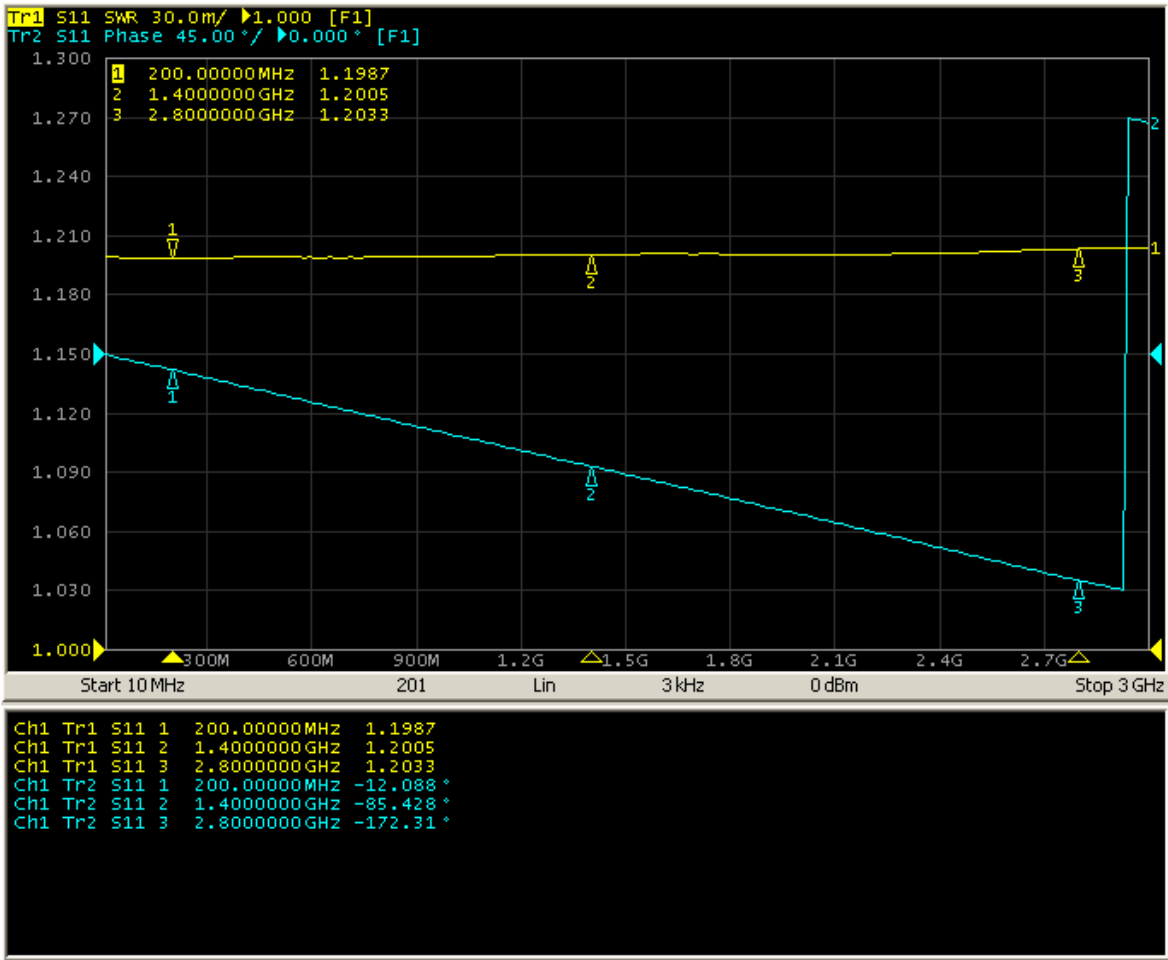


To complete the calibration procedure and calculate the table of calibration coefficients, click the **Apply** softkey.

Connect the DUT to the calibrated analyzer port again after calibration is done.

# SWR and Reflection Coefficient Phase Analysis Using Markers

This section describes how to determine the measurement values at three frequency points using markers. The Analyzer screen view is shown the screenshot below. In the current example, a reflection standard of SWR = 1.2 is used as a DUT.



SWR and reflection coefficient phase measurement example

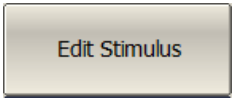


To create a new marker, use the following softkeys:

**Markers > Add Marker**



A new marker is placed in the center of the stimulus axis and assigned to be active. To edit just created marker, use the following soft keys:



**Markers > Edit Stimulus**

---

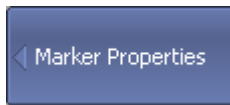
Then enter the frequency value in the input field in the graph, e.g. to enter frequency 200 MHz, press «2», «0», «0» and «M» keys on the keypad.

Repeat the above procedure three times to enable three markers at different frequency points.

---

By default, only active trace markers are displayed on the screen. To enable display markers of two traces simultaneously, activate the marker table.

---



To open the marker table, use the following softkeys:

**Markers > Properties > Marker Table**



---

**NOTE**

For more details on working with markers, see [Markers](#), [Markers](#), [Marker Stimulus Value Setting](#).

---

## Program Interface

The control program on the PC screen is displayed as the Analyzer Screen. The Analyzer screen contains:

- [Channel windows](#) to display measurement results in the form of traces and numerical values.
- [Menu bar](#) and [Softkey menu bar](#) to control the Analyzer.
- [Instrument status bar](#) to display information about the state of the Analyzer.

A detailed description of the program window elements is given further in this section.

The Analyzer Screen, with the main elements highlighted, is shown in the figure below.



Analyzer screen layout

## Softkey Menu Bar

The softkey menu bar along the right side of the screen is the main menu of the program.

---

### NOTE

The top line of the screen contains the menu bar, which provides direct access to certain submenus of the softkey menu. This is a secondary menu which can optionally be hidden.

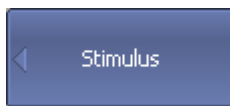
---

The softkey menu bar consists of a series of panels. Each panel represents one of the submenus of the softkey menu. All the panels are integrated to form the complete multilevel menu system, providing access to all the Analyzer functions.

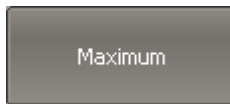
The menu softkeys can be navigated using a mouse.

Alternatively, the menu can be navigated using the «↑», «↓», «←», «→», «Enter», «Esc», and «Home» keys on an external keyboard.

The types of softkeys are described below:



The top softkey is the menu title key. It allows to return to the previous menu. If it is displayed in blue, the keyboard can be used to navigate within the softkey menu.



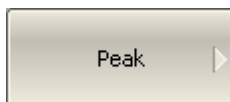
If the softkey is highlighted in dark gray, pressing the «Enter» key on the keyboard will activate this softkey. The highlight can be shifted from key to key using «↑» and «↓» arrows on the keyboard.



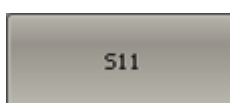
A large dot on the softkey indicates the current selection in a list of alternative settings.



A check mark in the left part of the softkey indicates an active function, which can be switched on/off.



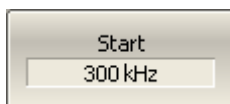
Softkeys with right arrows provide access to a lower level menu.



A softkey with a text field allows for the selected function indication.

---





Softkeys with a value field allow for entering/selection of the numerical settings.



This navigation softkey appears when the softkey menu overflows the menu screen area. The softkey menu can be scrolled through with this softkey.

---

In addition to «↑», «↓», the «←», «→», «Esc», «Home» keys can be used to navigate the softkey menu:

- «←» key brings up the upper level of the menu
- «→» key brings up the lower level of the menu, if there is a highlighted softkey with a right arrow
- «Esc» key functions similarly to the «←» key
- «Home» key brings up the main menu

---

**NOTE**

The above keys of the keyboard allow navigation within the softkey menu only if there is no active entry field. In this case the menu title softkey is highlighted in blue.

---

# Menu Bar

The menu bar is located at the top of the screen (See figure below). This is a secondary menu providing direct access to certain submenus of the main menu. It also contains the most frequently used softkeys' functions.



Menu Bar

The menu bar can be optionally hidden to gain more screen space for the channel window.

---

**NOTE**

To hide the menu bar, use the following softkeys:

**Display > Properties > Menu Bar**

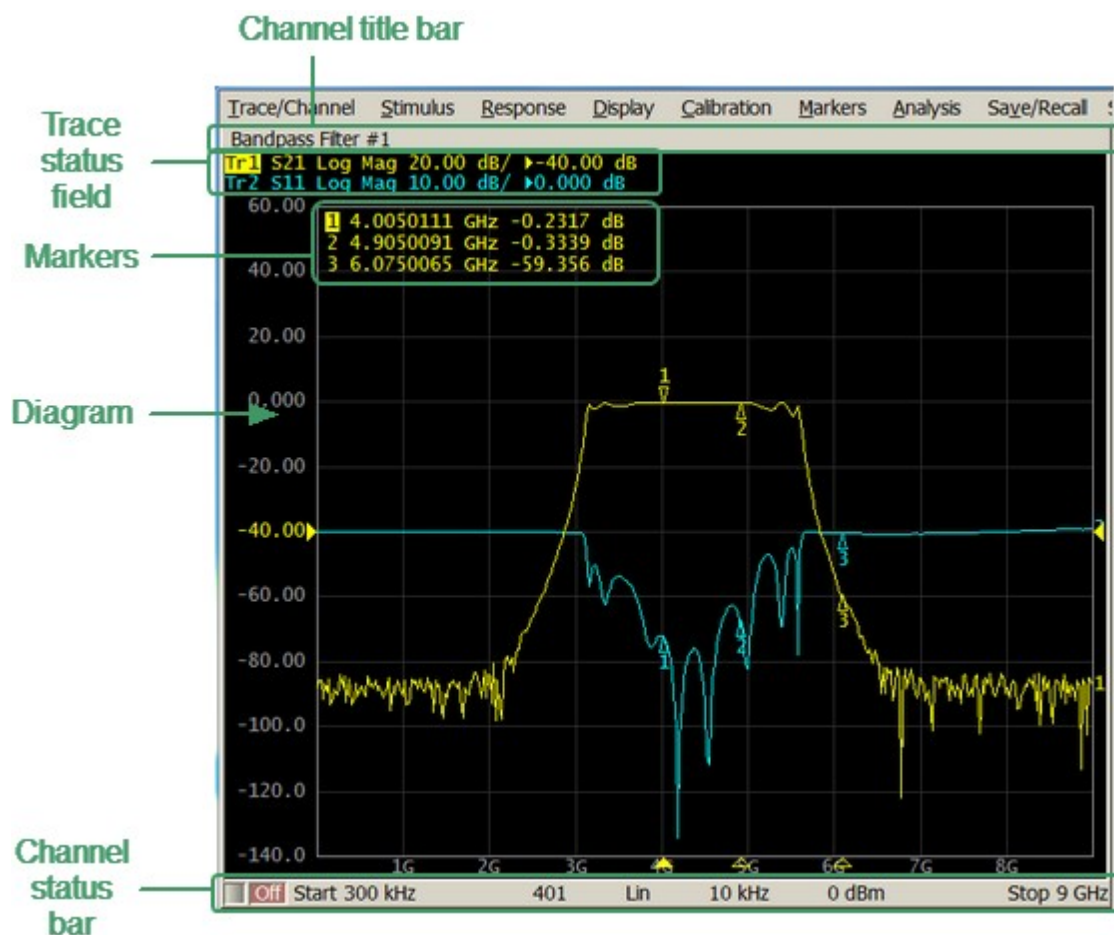
---

## Channel Window Layout and Functions

The channel window displays the measurement results in the form of traces and numerical values. The screen can display up to 16 channel windows simultaneously. Each window corresponds to one channel. The analyzer hardware processes channels sequentially.

In turn, each channel window can display up to 16 traces of measured parameters. If there is more than one trace in a channel window, the way they are displayed can be changed in the diagram (See [Trace Layout in the Channel Window](#)).

The general view of the channel window is represented in the figure below.



Channel window

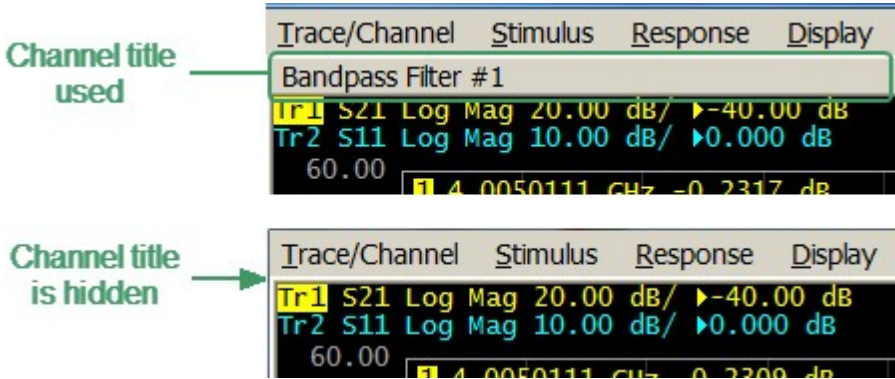
Each channel window contains a [Channel title](#) (hidden by default) to be defined by the user, [Trace status field](#) to display the name and parameters of the traces, [Diagram](#) for displaying traces, as well as information about the channel status in the form of the [Channel Status Bar](#). To display the measurement values at the indicated trace points, use the [Markers](#) feature.

A channel can be considered to be a separate logical analyzer with the following settings:

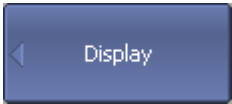
- Stimulus signal settings:
  1. [Frequency range](#)
  2. [Number of Points](#)
  3. [Sweep Type](#)
  4. [Power level](#)
- [IF Bandwidth](#) and [average](#)
- [Calibration](#)

# Channel Title Bar

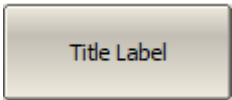
The channel title feature allows for a comment to be entered for each channel window. The channel title bar can be hidden to gain more screen space for the trace diagram.



Channel title bar



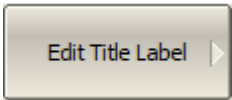
To show/hide the channel title bar, use the following softkeys:



**Display > Title Label**

[DISP:WIND:TITL](#)

Turns the channel title display ON/OFF.



The channel title edit mode can be accessed by using the following softkeys:

**Display > Edit Title Label > Edit Title Label**

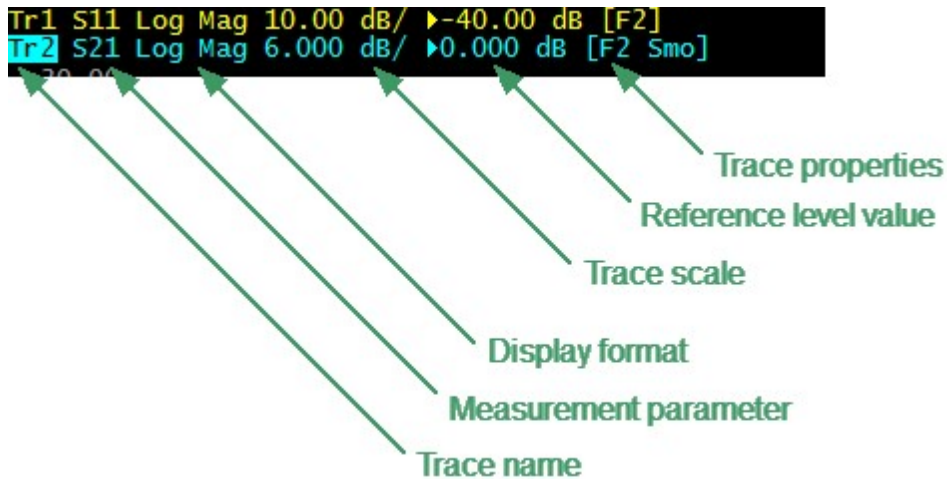
Alternatively, click on the title area in the channel title bar.

[DISP:WIND:TITL:DATA](#)

Sets or reads out the channel title label.

## Trace Status Field

The trace status field displays the name and parameters of a trace. The number of lines in the field depends on the number of active traces in the channel. The trace status field is represented in the figure below.



Trace status field

Each line contains the data of one trace of the channel:

- Trace name from «Tr1» to «Tr16». The active trace name is highlighted in an inverted color.
- Measured parameter: S11, S21, S12, S22, or absolute power value: A(n), B(n), R1(n), R2(n)
- Display format, e.g. «Log Mag».
- Trace scale in measurement units per scale division, e.g. «10.0 dB/».
- Reference level value, e.g. «▶0.00 dB», where «▶» is the symbol of the reference level.
- Trace status is indicated as symbols in square brackets (See table below).

Status	Symbols	Definition
Error Correction	RO	OPEN response calibration
	RS	SHORT response calibration

Status	Symbols	Definition
	<b>RT</b>	THRU response calibration
	<b>OP</b>	One-path two-port calibration
	<b>F1</b>	Full one-port SOL calibration
	<b>F2</b>	Full two-port SOLT and TRL calibration
		Full three-port SOLT and TRL calibration
		Full four-port SOLT and TRL calibration
	<b>SMC</b>	Scalar mixer calibration
Other Calibrations	<b>RC</b>	Receiver calibration
	<b>PC</b>	Power calibration
Data Analysis	<b>Z0</b>	Port impedance conversion
	<b>Dmb</b>	Fixture de-embedding
	<b>Emb</b>	Fixture embedding
	<b>Pxt</b>	Port extension
Trace Display	No indication	Data trace, memory does not exist
	<b>Dat</b>	Data trace only, memory exists
	<b>D&amp;M</b>	Data and memory traces
	<b>M</b>	Memory trace
	<b>Off</b>	Data and memory traces OFF

Status	Symbols	Definition
Trace Hold	No indication	Trace hold OFF
	<b>Max Hold</b>	Hold the maximum value
	<b>Min Hold</b>	Hold the minimum value
Math Operations	<b>D+M</b>	Data + Memory
	<b>D-M</b>	Data – Memory
	<b>D*M</b>	Data * Memory
	<b>D/M</b>	Data / Memory
Electrical Delay	<b>Del</b>	Electrical delay other than zero
Smoothing	<b>Smo</b>	Trace smoothing
Gating	<b>Gat</b>	Time domain gating
Conversion	<b>Zr</b>	Reflection impedance
	<b>Zt</b>	Transmission impedance
	<b>Yr</b>	Reflection admittance
	<b>Yt</b>	Transmission admittance
	<b>1/S</b>	S-parameter inversion
	<b>Ztsh</b>	Transmission-shunt impedance
	<b>Ytsh</b>	Transmission-shunt admittance
	<b>Conj</b>	Conjugation



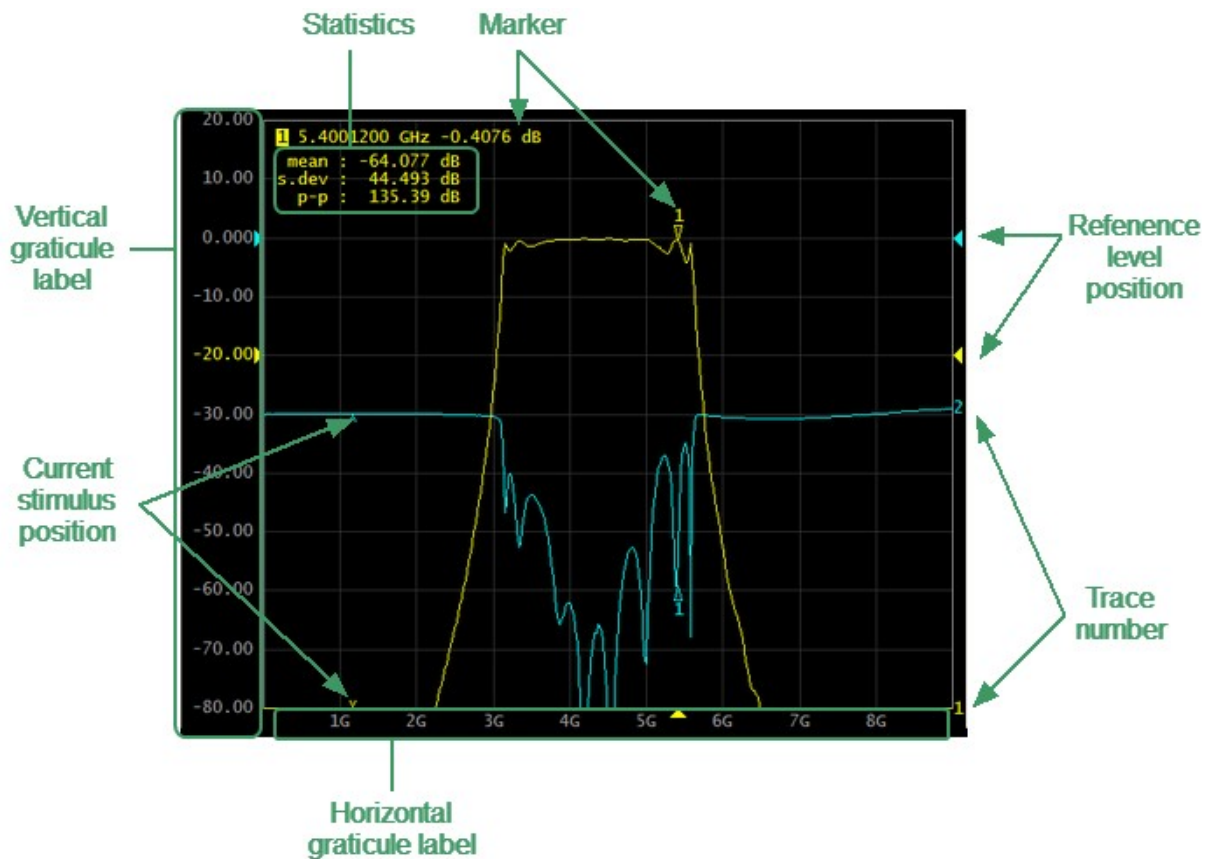
**NOTE**

Using the trace status field, the trace parameters can be easily modified using the mouse (as described in [Quick Setting Using a Mouse](#)).

---

## Diagram

The graph area in the channel window is called a diagram. The diagram displays traces and numeric data.



Diagram

The diagram contains the following elements:

- **Vertical graticule label** displays the vertical axis numeric data for the active trace. The data for all traces can be displayed or hidden to gain more screen space for the trace display.
- **Horizontal graticule label** displays stimulus axis numeric data (frequency, power level or time). The horizontal graticule label can be hidden to gain more screen space for the trace display.
- **Reference level position** indicates the reference level position of the trace.
- **Markers** indicates the measured values at points along the active trace. The markers for all traces can be simultaneously displayed.
- **Marker functions:** statistics, bandwidth, flatness, RF filter.
- **Trace number** allows trace identification when printing in black and white.

- **Current stimulus position** indicator appears when sweep duration exceeds 1.5 sec.
- 

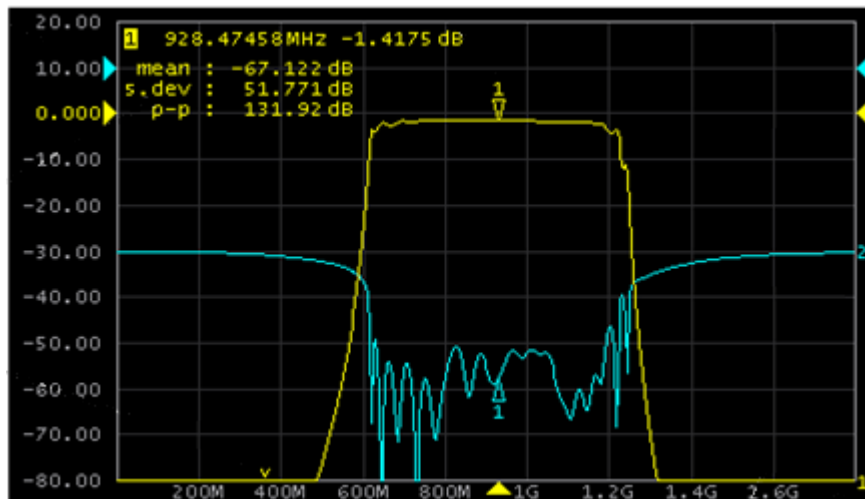
**NOTE**

Using the graticule labels, the trace parameters can be easily modified using the mouse (as described in [Quick Setting Using a Mouse](#)).

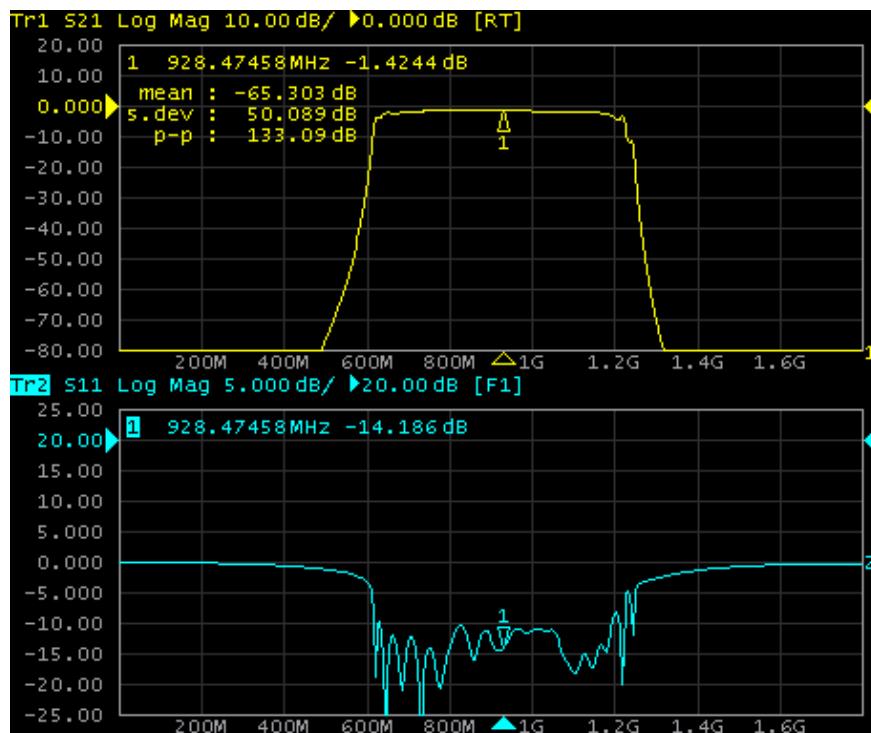
---

## Trace Layout in the Channel Window

If the number of the displayed traces is more than one, the traces can be rearranged. All the traces can be allocated to one diagram or each trace can be displayed on an individual diagram (See figures below).



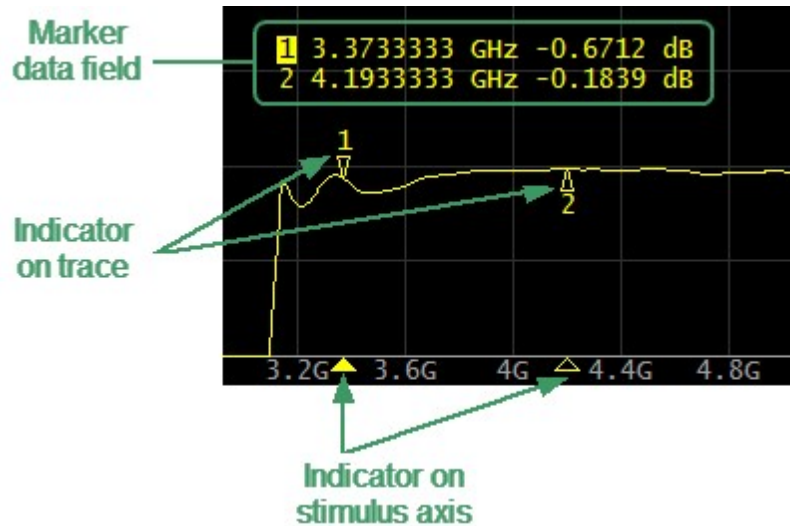
The one trace in one channel window (sample)



Two traces in one channel window (sample)

## Markers

The markers indicate the stimulus values and the measured values at selected points of the trace (See figure below).



Markers

The markers are numbered from 1 to 15. The reference marker is indicated with an R symbol. The active marker is indicated in the following manners:

- Its number is highlighted with inverse color.
- The indicator on the trace is located above the trace.
- The stimulus indicator is fully colored.

---

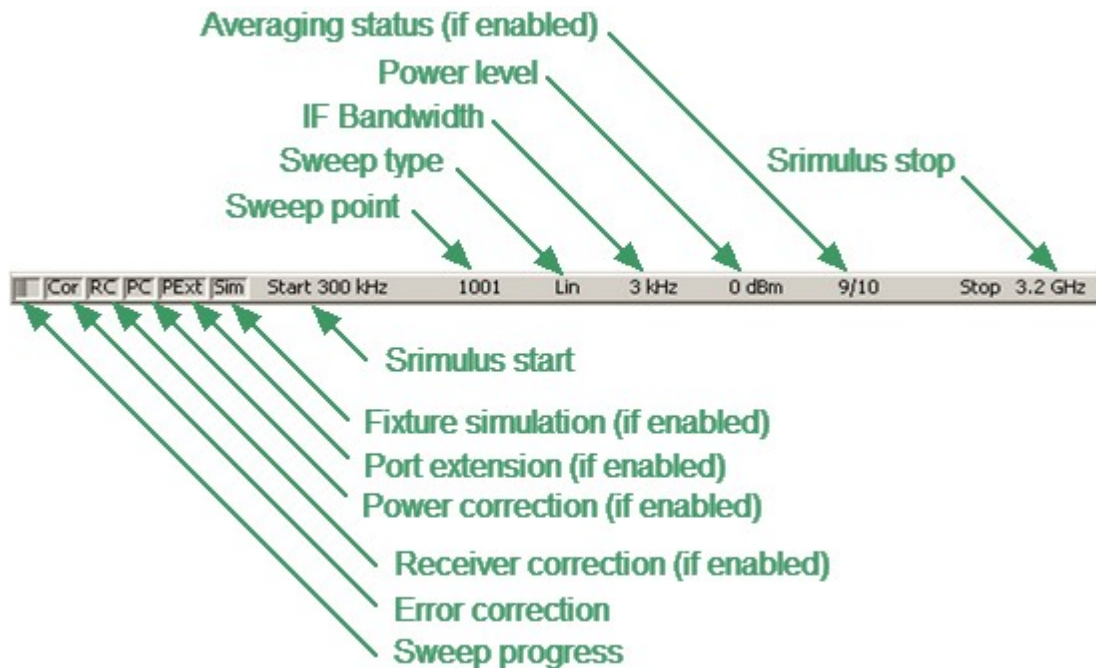
### NOTE

The use of markers is described in the [Markers](#).

---

## Channel Status Bar

The channel status bar is located in the bottom part of the channel window (See figure below).



Channel status bar

It contains the following elements:

- **Sweep progress** field displays a progress bar when the channel data is being updated.
- **Error correction** field displays the integrated status of error correction for S-parameter traces. The values of this field are represented in the table below. For a detailed description see [Error Correction Status](#).

Symbol	Definition	Note
<b>Cor</b>	Error correction is enabled. The stimulus settings are the same for the measurement and the calibration.	If the function is active for all traces – black characters on a gray background.
<b>C?</b>	Error correction is enabled. The stimulus settings are not the same for the measurement and the calibration. Interpolation is applied.	If the function is active only for some of the traces (other traces are not calibrated) – white characters on a red background.

Symbol	Definition	Note
<b>C!</b>	Error correction is enabled. The stimulus settings are not the same for the measurement and the calibration. Extrapolation is applied.	
<b>Off</b>	Error correction is turned off.	For all traces. White characters on a red background.
<b>—</b>	No calibration data. No calibration was performed.	

- **Receiver correction** field displays the integrated status of receiver correction for absolute power measurement traces. The values of this field are represented in the table below. For a detailed description on correcting receivers see [Receiver Calibration](#).

Symbol	Definition	Note
<b>RC</b>	Receiver correction is enabled. The stimulus settings are the same for the measurement and the calibration.	If the function is active for all the traces – black characters on a gray background.  If the function is active only for some of the traces (other traces are not calibrated) – white characters on a red background.
<b>RC?</b>	Receiver correction is enabled. The stimulus settings are not the same for the measurement and the calibration. Interpolation is applied.	
<b>RC!</b>	Receiver correction is enabled. The stimulus settings are not the same for the measurement and the calibration. Extrapolation is applied.	

- **Power correction** field displays the integrated status of power correction for all the traces. The values of this field are represented in the table below. For a detailed description of power correction, see [Power Calibration](#).

Symbol	Definition	Note
<b>PC</b>	Power correction is enabled. The stimulus settings are the same for the measurement and the calibration.	If the function is active for all traces – black characters on a gray background.
<b>PC?</b>	Power correction is enabled. The stimulus settings are not the same for the measurement and the calibration. Interpolation is applied.	If the function is active only for some of the traces (other traces are not calibrated) – white characters on a red background.
<b>PC!</b>	Power correction is enabled. The stimulus settings are not the same for the measurement and the calibration. Extrapolation is applied.	

- **Port extension** field indicates whether this function is active and applied to one or all traces.. If the function is enabled for all traces, black characters will be displayed on a gray background. If the function is enabled just for some of the traces, white characters will be displayed on a red background. For a detailed description see [Port Extension](#).
- **Fixture simulation** field indicates whether this function is active and applied to one or all traces. Fixture simulation includes the following operations: Z0 conversion, embedding, and de-embedding. If the function is enabled for all traces, black characters will be displayed on a gray background. If the function is enabled just for some of the traces, white characters will be displayed on a red background. For a detailed description see [Fixture Simulation](#).
- **Stimulus start** field allows for display and entry of the start frequency or power, depending on the sweep type. This field can be switched to indicate the stimulus center frequency, in this case the word «Start» will change to «Center». For a detailed description of stimulus setting, see [Sweep Range](#).
- **Number of Points** field allows for display and entry of the number of sweep points. The number of points can be set from 2 to the instrument maximum. For a detailed description see [Number of Points](#).
- **Sweep type** field allows for display and selection of the sweep type. The values of this field are represented in the table below. For a detailed description see [Sweep Type](#).



Symbol	Definition
<b>Lin</b>	Linear frequency sweep.
<b>Log</b>	Logarithmic frequency sweep.
<b>Segm</b>	Segment frequency sweep.
<b>Pow</b>	Power sweep.

- **IF bandwidth** field allows for display and setting of the IF bandwidth. The values can be set from the instrument minimum up to 1 MHz for some models. For a detailed description see [IF Bandwidth Setting](#).
- **Power level** field allows for display and entry of the port output power. In power sweep mode, the field switches to indicate the CW frequency of the source. For a detailed description see [Stimulus Power](#).
- **Averaging status** field displays the averaging status, if this function is enabled. The first number is the averaging current counter value, the second one is the averaging factor. For a detailed description see [Averaging Setting](#).
- **Stimulus stop** field allows for display and entry of the stop frequency or power, depending on the sweep type. This field can be switched to indication of stimulus span, in this case the word «Stop» will change to «Span». For a detailed description of stimulus setting, see [Sweep Range](#).

## Instrument Status Bar

The instrument status bar is located at the bottom of the screen.



Instrument status bar

### Messages in the instrument status bar

Field Description	Message	Instrument Status	Note
DSP status	<b>Not Ready</b>	No communication between DSP and PC.	
	<b>Loading</b>	DSP firmware is loading.	
	<b>Ready</b>	DSP is running normally.	
Sweep status	<b>Meas</b>	A sweep is in progress.	For a detailed description see <a href="#">Trigger Settings</a> .
	<b>Hold</b>	A sweep is on hold.	
	<b>Ext</b>	Waiting for <b>External</b> trigger.	
	<b>Man</b>	Waiting for <b>Manual</b> trigger.	
	<b>Bus</b>	Waiting for <b>Bus</b> trigger.	
Calibration	<b>Calibration...</b>	Calibration standard measurement is in progress.	
RF signal	<b>RF Off</b>	Stimulus signal output is turned OFF.	For a detailed description see <a href="#">RF Out Function</a> .

Field Description	Message	Instrument Status	Note
External reference frequency	<b>ExtRef</b>	External reference frequency input (10 MHz) is turned on.	For a detailed description see <a href="#">Reference Frequency Oscillator Selection</a> .
Display update	<b>Update Off</b>	Display update is turned OFF.	For a detailed description see <a href="#">Screen Update Setting</a> .
System correction status	<b>Sys Corr OFF</b>	System correction is turned OFF.	For a detailed description see <a href="#">System Correction Setting</a> .
Factory calibration error	<b>Factory Cal Failure:</b>  (PW, PT, LOP, ED, IF, VM, xPW, xPT)	Calibration data in ROM has an error.  <b>THE ANALYZER IS DAMAGED AND REQUIRES FACTORY REPAIR.</b>	
External power meter status	<b>Power Meter: message</b>	When external power meter is connected to the Analyzer via USB the following messages are displayed: <b>connection, connection error, ready, measurement, zero setting, zero setting error.</b>	For a detailed description see <a href="#">Power Meter Setting</a> .
Power Trip function	<b>Port n Power Trip at Overload</b>	Port n overload, stimulus signal output is turned OFF.	For a detailed description

Field Description	Message	Instrument Status	Note
			see <a href="#">Power Trip Function</a> .

## Setting Measurement Conditions

The section describes how to set the various measurement conditions of the Analyzer.

### Measurement procedure

To perform measurements, do the following according to each measurement task:

- Set the number, parameters, and traces of the logical channels involved in the measurements. For a detailed description see [Channel and Trace Setting](#).
- Set the stimulus signal parameters. For a detailed description see [Stimulus Settings](#).
- Assign the measured parameters display format, and scale to the traces. For a detailed description see [Measurement Parameters Settings](#), [Format Setting](#), [Scale Settings](#).
- If necessary, set the related trigger settings. For a detailed description see [Trigger Settings](#).
- Set filtering parameters to improve the signal to noise ratio. For a detailed description see [Measurement Optimization](#).

This section also describes how to quickly set the parameters of the analyzer using a mouse. For a detailed description see [Quick Setting Using a Mouse](#).

## Channel and Trace Setting

The Analyzer supports 16 channels, each of which allows for measurements with stimulus parameter settings different from the other channels. The parameters related to a logical channel are listed in the table below.

### Channel parameters

N	Parameter Description
1	<a href="#">Sweep Type</a>
2	<a href="#">Sweep Range</a>
3	<a href="#">Number of Points</a>
4	<a href="#">Stimulus Power Level</a>
5	<a href="#">Power Slope Feature</a>
6	<a href="#">CW Frequency</a>
7	<a href="#">Segment Sweep Table</a>
8	<a href="#">Trigger Mode</a>
9	<a href="#">IF Bandwidth</a>
10	<a href="#">Averaging</a>
11	<a href="#">Calibration</a>
12	<a href="#">Fixture Simulator</a>
13	<a href="#">General S-Parameter Conversion</a>

Each channel window can contain up to 16 different traces. Each trace is assigned a measured parameter (S-parameter), display format, and other parameters. The parameters related to a trace are listed in the table below.

### Trace parameters

N	Parameter Description
1	<a href="#">Measured Parameter</a>
2	<a href="#">Display Format</a>
3	<a href="#">Scale Settings</a>
4	<a href="#">Electrical Delay, Phase Offset</a>
5	<a href="#">Memory Trace, Math Operation</a>
6	<a href="#">Smoothing</a>
7	<a href="#">Markers</a>
8	<a href="#">Time Domain</a>
9	<a href="#">Parameter Transformation</a>
10	<a href="#">Limit Test</a>
11	<a href="#">Ripple Limit Test</a>
12	<a href="#">Peak Limit Test</a>

## Channel Allocation

A channel is represented on the screen as an individual channel window. The screen can display up to 16 channel windows simultaneously. By default, one channel window opens. If two or more channel windows need to be opened, one of the layouts shown below can be selected. The available options for number and layout of the channel windows on the screen are as follows:



Channel window layout

In accordance with the layouts, the channel windows do not overlap each other. The channels open sequentially from low to high.

---

### NOTE

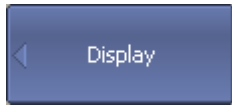
For each open channel window, set the stimulus parameters, adjust other settings, and perform calibration. For a detailed description see [Stimulus Settings](#).

Before changing a channel parameter setting or performing calibration of a channel, ensure that the channel is selected as active. For a detailed description see [Selection of Active Trace/Channel](#).

---

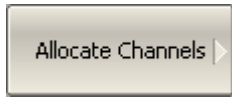
The measurements are executed for open channel windows sequentially. Measurements for any hidden channel windows are not performed.





To set the channel window layout, use the following softkeys:

**Display > Allocate Channels**



Then select the required number and layout of the channel windows in the menu.

---

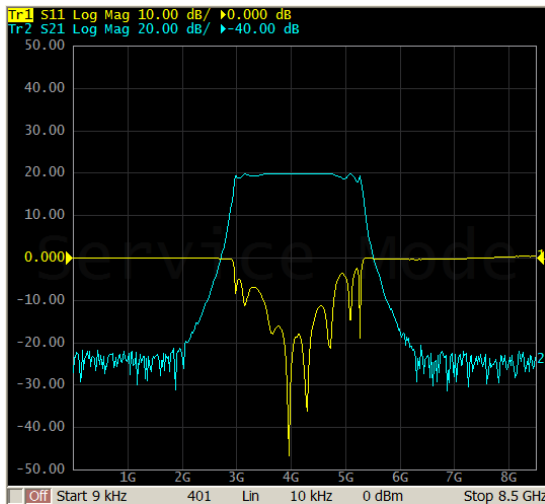
[DISP:SPL](#)

Sets or reads out the number of the channel window layout on the screen.

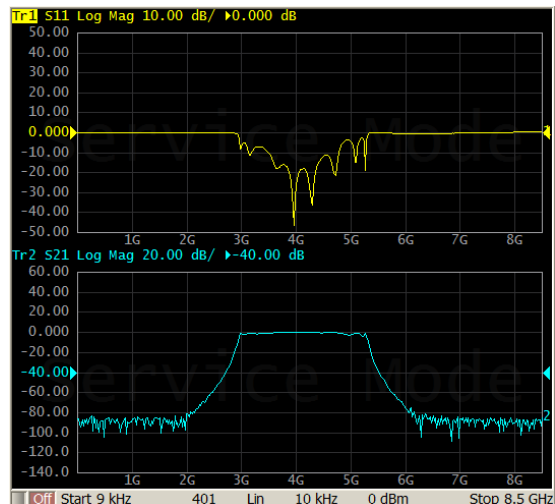
---

## Number of Traces

Traces can be displayed in one diagram, overlapping each other, or in separate diagrams within a channel window (See figures below).



Displaying two traces on the same diagram



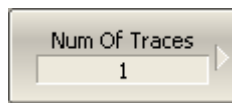
Displaying two traces on two different diagrams

The trace settings are made in two steps: trace number and trace layout within the channel window. By default, the channel window contains one trace. If two or more traces need to be enabled, the number of traces can be set as described below.



To set the number of the traces, use the following softkeys:

### Display > Num Of Traces



Then select the number of traces from the menu.

[CALC:PAR:COUN](#)

Sets or reads out the number of traces in the channel.

All traces are assigned individual names, which cannot be changed. The trace name contains its number. The trace names are as follows: **Tr1**, **Tr2**, ... **Tr16**.

Each trace is assigned the following initial settings: measured parameter, format, scale, and color, which can be modified:

- The measured parameters of the first four traces default to the following values: **S11**, **S21**, **S12**, **S22**. For a detailed description of changing measured parameter see [S Parameter](#).

- By default, the display format for all traces is set to logarithmic magnitude (dB). For a detailed description of changing display format see [Format Setting](#).
  - By default, the scale parameters are set as follows: division is set to 10 dB, reference level value is set to 0 dB, and the reference level position is in the middle of the diagram. For a detailed description of changing scale parameters see [Scale Settings](#).
  - The trace color is determined by its number. This color can be changed. For a detailed description of changing color see [Set Color](#).
- 

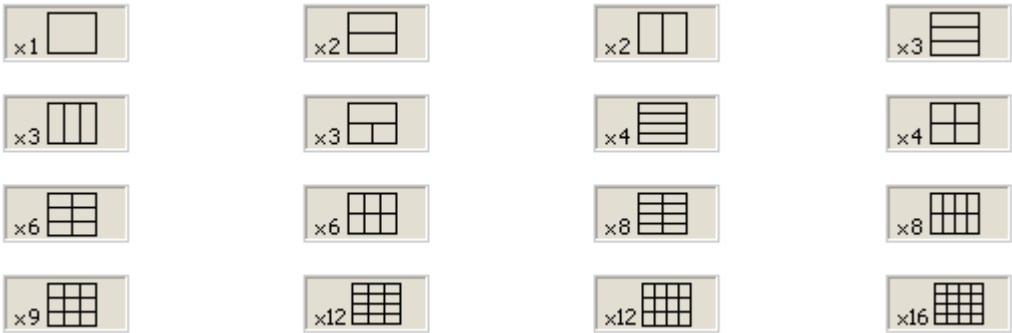
**NOTE**

The full cycle of the trace update depends on the S-parameters measured and the calibration method. For example, the full cycle might consist of a single sweep with either Port 1 or Port 2 as the source, or it might include two successive sweeps of Port 1 then of Port 2. To have two traces (S11 and S22) measured, two successive sweeps need to be performed. Two successive sweeps are also performed when full two-port calibration is employed, independently of the number of the traces and S-parameters measured.

---

# Trace Allocation

By default, traces are displayed overlapping one other in the diagram. If you wish to display the traces in separate diagrams, the number and layout of the diagrams can be set in the channel window as shown below.

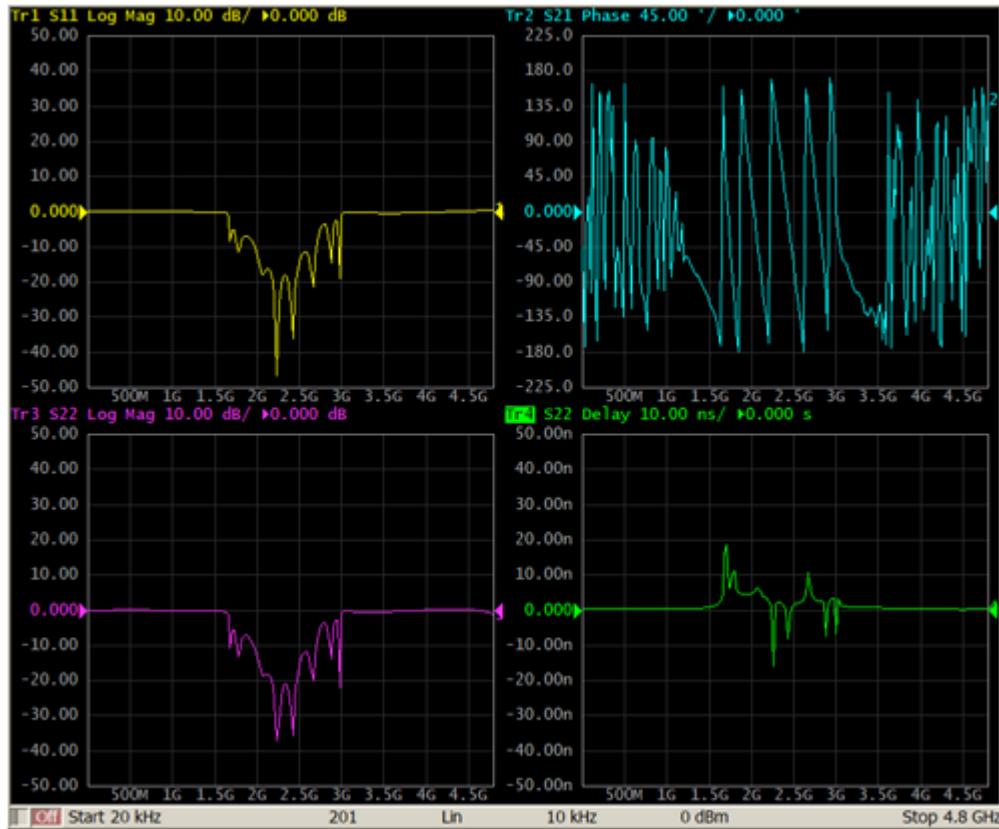


Options for diagram placement in the channel

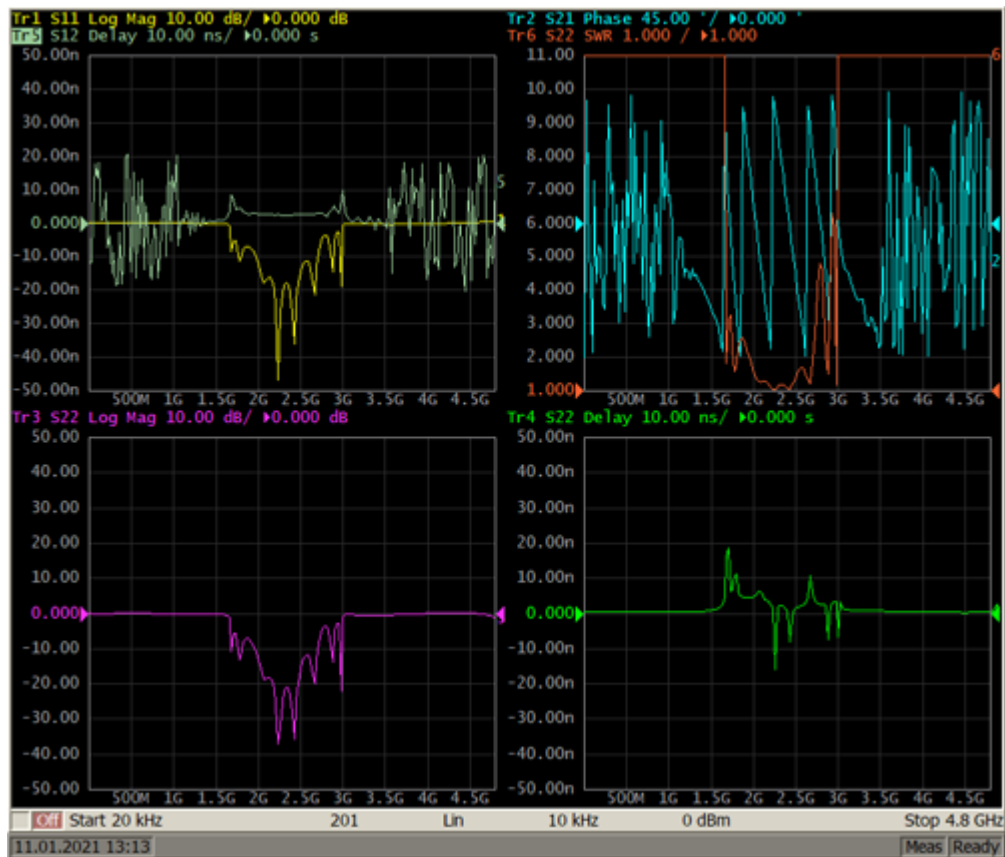
Unlike channel windows, the number of traces and layout of the trace in diagrams are not related. The number of traces and the number of diagrams are set independently.

Placing traces in a diagram:

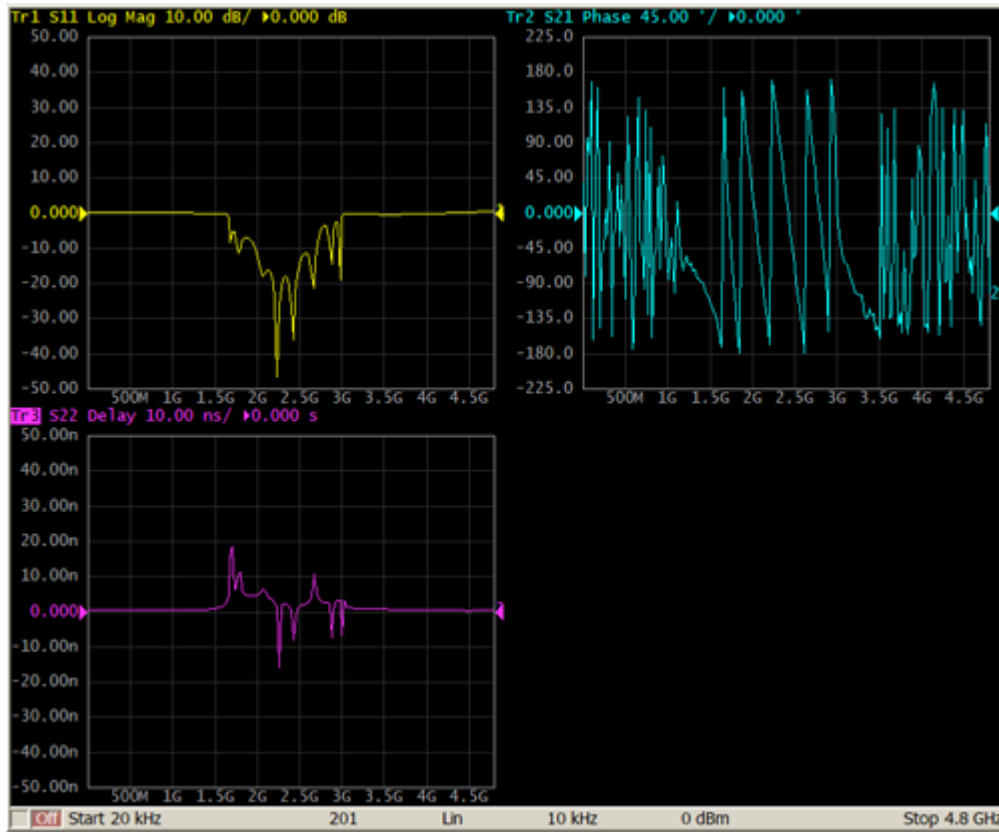
- If the number of traces and the number of diagrams are equal, all the traces will be displayed separately, each in an individual diagram.



- If the number of traces is greater than the number of diagrams, traces will be assigned successively (beginning from the smallest trace number) to the number of available diagrams. When all diagrams are utilized, the process will continue from the first diagram (the following in succession traces will be added in diagrams).



- If the number of traces is smaller than the number of diagrams, empty diagrams will be displayed.



If two or more traces are displayed in one diagram, the vertical scale will be shown for the active trace.

#### NOTE

The Analyzer can optionally show vertical graticule labels for all the traces in the diagram. By default, this feature is disabled. Activation of this function is described in [Set Response Graticule Mode \(Y\)](#).

If two or more traces are displayed in one diagram, markers data will be shown for the active trace.

---

**NOTE**

There are two options for displaying marker data for all the traces simultaneously:

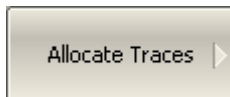
- Using the [Marker Table Feature](#).
  - Deactivating identification of the [active trace marker only](#), which is set by default.
- 

The stimulus axis is the same for all the traces of the channel, except when [Time Domain Transformation](#) is applied to some of the traces. In this case, the displayed stimulus axis will correspond to the active trace.

---



To allocate the traces in diagrams, use the following softkeys:



**Display > Allocate Traces**

Then select the desired number and layout of separate diagrams in the menu.

---

[DISP:WIND:SPL](#)

Sets or reads out the number of the graph layout in the channel window.

---

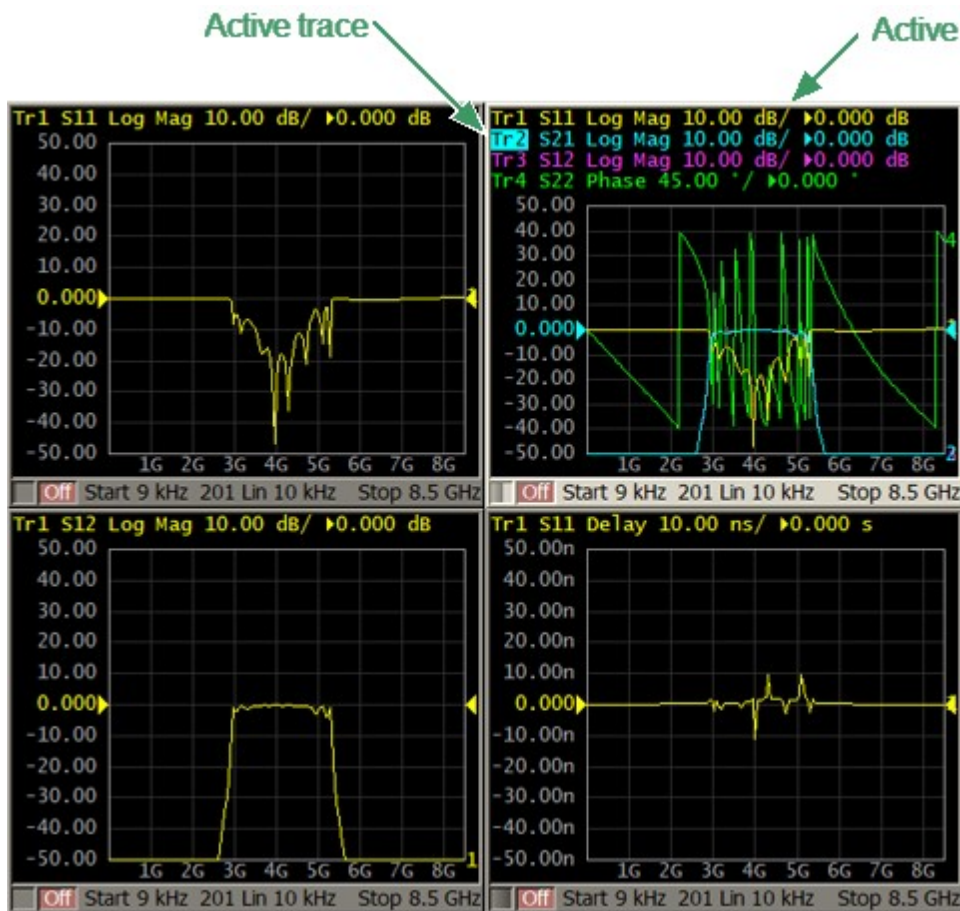


## Selection of Active Trace/Channel

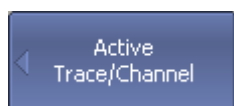
The selected control commands are applied to the active channel or the active trace, respectively.

The boundary line of the active channel window is highlighted in a light color. The active trace belongs to the active channel and its title is highlighted in an inverse color.

Before setting the parameters of a channel or trace, that channel or trace needs to be activated.

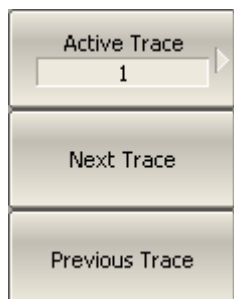


Active Trace/Channel



To activate a trace/channel, use the following softkeys:

### Display > Active Trace/Channel



Then activate the trace by entering the number in the Active Trace softkey or using **Previous Trace** or **Next Trace** softkeys.

The active channel can be selected in a similar way.



---

[DISP:WIND:ACT](#)

Sets the active channel.

---

[CALC:PAR:SEL](#)

Selects the active trace in the channel.

---

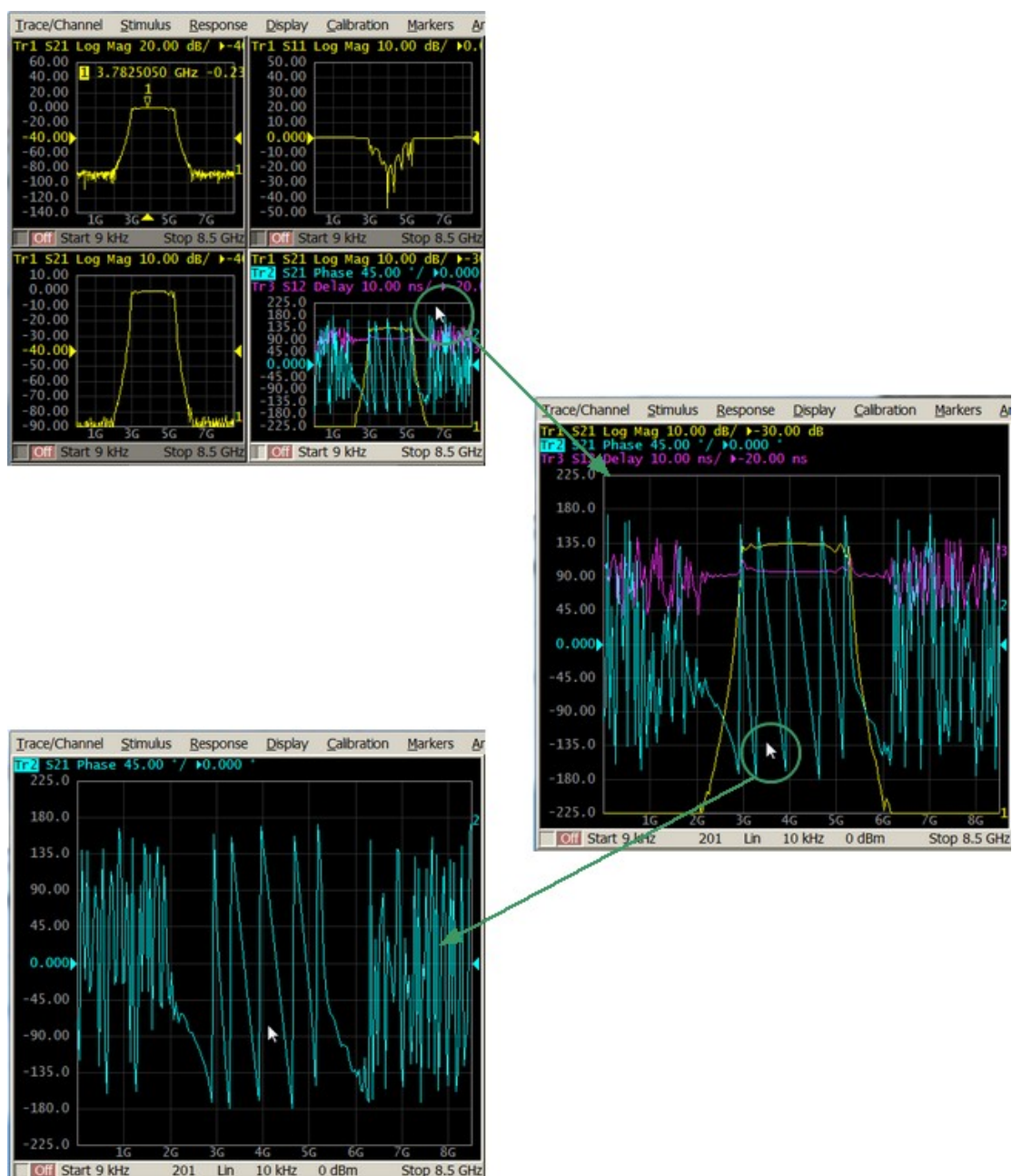
#### NOTE

Active trace/channel can be selected using the mouse (See [Active Trace Selection](#) and [Active Channel Selection](#)).

---

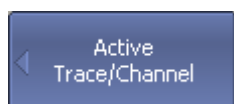
## Trace/Channel Window Maximizing

When there are several channel windows displayed, the active channel window can be temporarily expanded to full screen size. The other channel windows will not be visible, but this will not interrupt measurements in those channels.

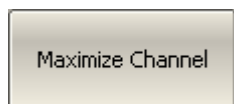


### Active Channel/Trace Window Maximizing

Similarly, when there are several traces displayed in a channel window, the active trace can be temporarily expanded. The other traces will not be visible, but this will not interrupt measurement of those traces.



To enable/disable active channel maximizing function, use the following softkeys:

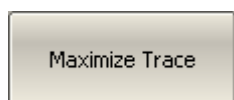


**Display > Active Trace/Channel > Active Channel**

---

[DISP:MAX](#)

Turns the maximization of the active channel window ON/OFF.



To enable/disable active trace maximizing function, use the following softkeys:

**Display > Active Trace/Channel > Active Trace**

---

[DISP:WIND:MAX](#)

Turns the active trace maximization inside the specified channel ON/OFF.

---

#### NOTE

Channel and trace maximization can also be controlled achieved by a double click on the channel/trace. To return to the initial state, double click on channel/trace.

---

## Stimulus Settings

This section describes how to set the stimulus signal parameters.

Stimulus — a signal with a known amplitude and phase, fed by the Analyzer to the device under test.

The stimulus parameter settings apply to each channel. Before setting the stimulus parameters of a channel the channel must be made active (See [Selection of Active Trace/Channel](#)).

---

**NOTE**

To make maximize measurement accuracy, perform measurements with the same stimulus settings as were used for calibration.

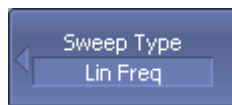
---

## Sweep Type

The sweep type determines how the stimulus range is scanned:

- by frequency (linear frequency sweeps, logarithmic frequency sweeps or segment sweep mode)
- by power (linear power sweep)

The channel to which the function is applied must be preselected as active (See [Selection of Active Trace/Channel](#) ).



To set the sweep type, use the following softkeys:

### Stimulus > Sweep Type

Then select the sweep type:



- **Lin Freq** —Linear frequency sweep;
- **Log Freq** — Logarithmic frequency sweep;
- **Segment** — Segment frequency sweep;
- **Power Sweep** — Power sweep.

---

[SENS:SWE:TYPE](#)

Sets or reads out the sweep type.

---

### NOTE

The **Sweep Type** can be selected using the mouse (See [Sweep Type Setting](#)).

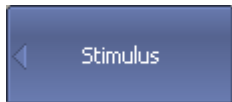
---

## Sweep Range

The sweep range should be set for the linear and logarithmic frequency sweeps (Hz) and for the linear power sweep (dBm).

The sweep range can be set using either Start/Stop or Center/Span values.

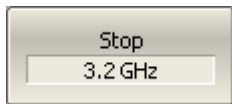
The channel to which the function is applied must be preselected as active (See [Selection of Active Trace/Channel](#)).



To enter the start and stop values of the sweep range, use the following softkeys:



**Stimulus > Start**



**Stimulus > Stop**

---

[SENS:FREQ:STAR](#)

Sets or reads out the stimulus start value of the sweep range for linear or logarithmic sweep type.

---

[SENS:FREQ:STOP](#)

Sets or reads out the stimulus stop value of the sweep range for linear or logarithmic sweep type.

---

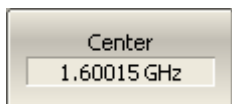
[SOUR:POW:STAR](#)

Sets or reads out the power sweep start value when the power sweep type is active.

---

[SOUR:POW:STOP](#)

Sets or reads out the power sweep stop value when the power sweep type is active.



To enter center and span values of the sweep range, use the following softkeys:



**Stimulus > Center**

**Stimulus > Span**

---

[SENS:FREQ:CENT](#)

Sets or reads out the stimulus center value of the sweep range for linear or logarithmic sweep type.

---

<a href="#">SENS:FREQ:SPAN</a>	Sets or reads out the stimulus span value of the sweep range for linear or logarithmic sweep type.
<a href="#">SOUR:POW:CENT</a>	Sets or reads out the center value of the power sweep type.
<a href="#">SOUR:POW:SPAN</a>	Sets or reads out the power span when the power sweep type is active.
<b>NOTE</b>	If span is set to 0 the <a href="#">CW Time Sweep Mode</a> is automatically turned on.
<b>NOTE</b>	If power sweep is activated, the values on the <b>Start</b> , <b>Stop</b> , <b>Center</b> and <b>Span</b> softkeys will be represented in <b>dBm</b> .
<b>NOTE</b>	<p>The <b>Start</b>, <b>Stop</b>, <b>Center</b> and <b>Span</b> values of the sweep range can be set using the mouse (See <a href="#">Sweep Start Setting</a>, <a href="#">Sweep Stop Setting</a>, <a href="#">Sweep Center Setting</a>, <a href="#">Sweep Range Setting</a>).</p> <p>Switch between <b>Start/Center</b> and <b>Stop/Span</b> modes with the mouse (See <a href="#">Switching Between Start/Center and Stop/Span Modes</a>).</p> <p>The <b>Start/Center</b> and <b>Stop/Span</b> values can be set using the mouse (See <a href="#">Start/Center Value Setting</a>, <a href="#">Stop/Span Value Setting</a>).</p>



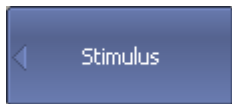
## Number of Points

The number of points is the number of measurements gathered in a sweep cycle in the range of stimulus change.

The number of points should be set for the linear and logarithmic frequency sweeps, and for the linear power sweep.

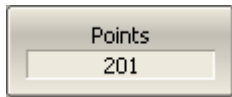
Increase the number of points to get a larger trace resolution. To increase measurement performance, reduce the number of points to values that provide an acceptable trace resolution. To maintain high accuracy, the number of points in the calibration and in the actual measurements must be the same.

The channel to which the function is applied must be preselected as active (See [Selection of Active Trace/Channel](#) )..



To enter the number of points, use the following softkeys:

**Stimulus > Points**



---

[SENS:SWE:POIN](#)

Sets or reads out the number of measurement points.

---

### NOTE

The number of **Points** can be set using the mouse (See [Number of Points Setting](#)).

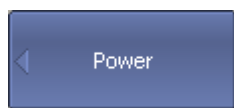
---

## Stimulus Power

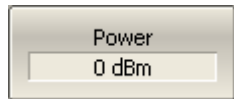
The stimulus power level should be set for the linear and logarithmic frequency sweeps.

For the segment sweep type, the method of power level setting described in this section can be used only if the same power level is set for all the segments of the sweep. For setting of individual power levels for each segment, see [Segment Table Editing](#).

The channel to which the function is applied must be preselected as active (See [Selection of Active Trace/Channel](#) ).



To enter the power level value when port couple feature is ON, use the following softkeys:



**Stimulus > Power > Power**

---

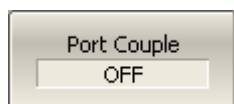
[SOUR:POW](#)

Sets or reads out the power level for the frequency sweep type.

---

## Setting the Power Level for Each Port

By default, the power levels of all test ports are set to equal value. This function is called Port Couple. This function can be optionally disabled and the power level of each port can be set individually.



To set the power level for each port individually, first disable the Power Couple function:

**Stimulus > Power > Port Couple [ON | OFF]**

---

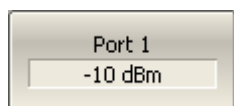
[SOUR:POW:PORT:COUP](#)

Turns the port power couple ON/OFF.

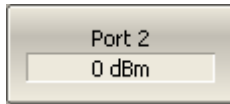
---



Then set the power level for each port:



**Stimulus > Power > Port Power > [Port 1 | Port 2]**



---

#### [SOUR:POW:PORT](#)

Set or reads out the power level of each port for the frequency sweep type when the port couple feature is set to OFF by the [SOUR:POW:PORT:COUP](#) command.

---

---

#### NOTE

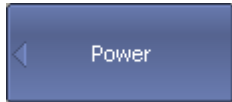
Setting the **Power** level is possible using the mouse (See [Power Level/CW Frequency Setting](#)).

---

## Power Slope Feature

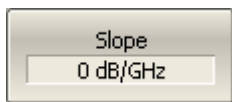
The power slope feature allows for compensation of power loss with increasing frequency in the connecting cables. The power slope can be set for the linear, logarithmic, and segment frequency sweep types.

The channel to which the function is applied must be preselected as active (See [Selection of Active Trace/Channel](#) ).



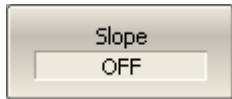
To enter the power slope value, use the following softkeys:

**Stimulus > Power > Slope**



To enable/disable the power slope function, use the following softkeys:

**Stimulus > Power > Slope [ ON|OFF]**



---

[SOUR:POW:SLOP](#)

Sets or reads out the power slope value for the frequency sweep type.

---

[SOUR:POW:SLOP:STAT](#)

Turns the power slope ON/OFF.

---

## CW Frequency

The CW frequency setting determines the fixed frequency for the linear power sweep.

The channel to which the function is applied must be preselected as active (See [Selection of Active Trace/Channel](#) ).



To enter the CW frequency value, use the following softkeys:

**Stimulus > Power > CW Freq**



---

[SENS:FREQ](#)

Sets or reads out the fixed frequency value when the power sweep type is selected.

---

### NOTE

**CW frequency** can be set using the mouse (See [Power Level/CW Frequency Setting](#)).

---

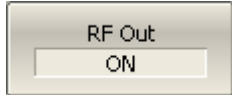
## RF Out Function

The RF Out function allows for temporary disabling of the stimulus signal. While the stimulus is disabled, measurements cannot be performed.



To disable/enable stimulus, use the following softkeys:

**Stimulus > Power > RF Out**



---

[OUTP](#)

Turns the RF signal output ON/OFF.

---

### NOTE

The **RF Out** function is applied to the Analyzer, not to individual channels. Indication of RF Out status appears in the instrument status bar (See [Instrument Status Bar](#))



## Segment Table Editing

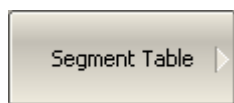
The segment table determines the sweep parameters when segment sweep type is used (See [Sweep Type](#)).

The channel to which the function is applied must be preselected as active (See [Selection of Active Trace/Channel](#)).



To open the segment table, use the following softkeys:

**Stimulus > Segment Table**



When switching to the **Segment Table** submenu, the segment table will open in the lower part of the application. When exiting the **Segment Table** submenu, the segment table will be hidden.

The segment table layout is shown below (See figure below). The table has three mandatory columns: start frequency, stop frequency, and number of points, and three columns which can be optionally enabled/disabled: IF bandwidth, power level, and delay time.

	Start	Stop	Points	IFBW	Power
1	300 kHz	800 MHz	11	100 Hz	10 dBm
2	800 MHz	1.12 GHz	51	3 kHz	0 dBm
3	1.12 GHz	1.99 GHz	101	30 kHz	-10 dBm
4	1.99 GHz	2.28 GHz	51	3 kHz	0 dBm
5	2.28 GHz	3.2 GHz	11	100 Hz	10 dBm

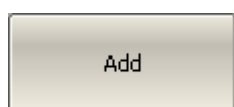
Total Points: 225

The segment table

Each row describes one segment. The table can contain one or more rows. The number of segments is limited only by the instrument's maximum number of points.

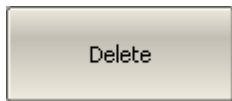


To add a segment to the table, click the **Add** softkey. The new segment row will be entered below the highlighted one.



To delete a segment, click the **Delete** softkey. The highlighted segment will be deleted.

---



---

For any segment it is necessary to set the mandatory parameters: frequency range (start and stop) and number of points. The frequency range can be set either as Start / Stop, or as Center / Span.

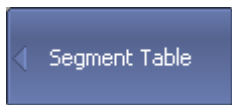


To set the frequency range representation mode, click the **Freq Mode** softkey to select between the **Start/Stop** and **Center/Span** options.



---

For any segment, the following additional parameter columns can be enabled: IF bandwidth, power level, and delay time. If such a column is disabled, the corresponding value set for linear sweep will be used (same for all the segments).



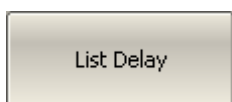
To enable the IF bandwidth column, click the **List IFBW** softkey.



To enable the power level column, click the **List Power** softkey.



To enable the delay time column, click the **List Delay** softkey.



---

[SENS:SEGM:DATA](#)

Sets or reads out the array of the segment sweep table.

---

To set a parameter, click on its value field and enter the value. To navigate in the table, use the keys on the keyboard.

---

**NOTE**

Adjacent segments must not overlap in the frequency domain.

---

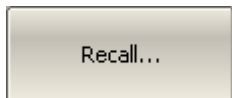
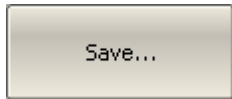


The segment table can be saved into \*.lim file to a hard disk and later recalled.



To save the segment table, click the **Save...** softkey.

Then enter the file name in the appeared dialog.



To recall the segment table, click **Recall...** softkey.

Then select the file name in the appeared dialog.

---

[MMEM:STOR:SEGM](#)

Saves the segment table into a file.

---

[MMEM:LOAD:SEGM](#)

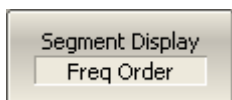
Recalls the segment table file. The file must be saved using the MMEM:STOR:SEGM command.

---

The segment sweep graph has two methods of horizontal axis representation: the frequency-based and order-based. In the first, the axis is displayed according the frequency. In the second, the axis is displayed according to the measuring point number.



To set the frequency axis display mode, click the **Segment Display** softkey and select the **Freq Order** or **Base Order** option.



---

[DISP:WIND:X:SPAC](#)

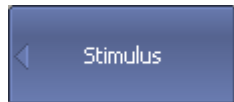
Sets or reads out the display method of the graph horizontal axis for the segment sweep.

---

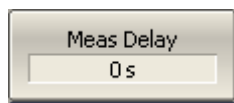
## Measurement Delay

The measurement delay function allows for adding an additional time delay at each measurement point between the moment when the source output frequency becomes stable and the start of the measurement. This capability can be useful for measurements of electrically-long devices.

The channel to which the function is applied must be preselected as active (See [Selection of Active Trace/Channel](#) ).



To set the measurement delay time, use the following softkeys:



**Stimulus > Meas Delay**

---

[SENS:SWE:POIN:TIME](#)

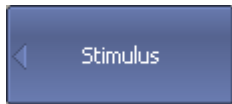
Sets or reads out the delay before measurement in each measurement point.

---

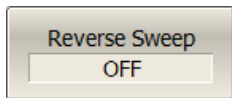
## Reverse Sweep Mode

In the reverse sweep mode, the sweep starts from the stop frequency and stops at the start frequency.

The channel to which the function is applied must be preselected as active (See [Selection of Active Trace/Channel](#) ).



To set the reverse sweep mode, click the **Reverse Sweep** softkey and select the ON or OFF option.



**Stimulus > Reverse Sweep**

---

[SENS:SWE:REV](#)

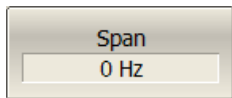
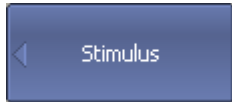
Sets or reads out the ON/OFF status of the reverse sweep function

---

## CW Time Sweep Mode

In the CW time sweep mode, the Analyzer displays measured data as a function of time when the stimulus frequency is fixed. This function is automatically turned on when the Stimulus Span is set to zero.

The channel to which the function is applied must be preselected as active (See [Selection of Active Trace/Channel](#) ).



To enable CW time sweep mode, set the Span value to zero using the following softkeys:

### **Stimulus > Span**

A horizontal scale will then display the time.

Set **Stimulus > Center** to the frequency under test.

Other sweep settings (number of points, power level, IF bandwidth) can be set arbitrarily, depending on the measurement task.

In the CW time sweep mode, the following elements change from frequency representation to temporal representation:

- stimulus axis labels
- marker stimulus value
- SCPI commands response:

[CALC:DAT:XAX?](#)

[CALC:TRAC:DATA:XAX?](#)

[CALC:MARK:X](#)

### **NOTE**

The sweep time is determined by the following formula:

$$T_{st} = N \left( \frac{1.19}{IFBW} + T_{md} + T_{hw} \right),$$

where  $N$  — number of points,

$IFBW$  — IF bandwidth,

$T_{md}$  — measurement delay,

$T_{hw}$  — hardware delay (depends on the Analyzer model and cannot be changed).

---

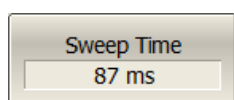
The Analyzer automatically calculates the sweep time value based on the current settings: number of points, IF bandwidth, measurement delay. Arbitrary value can be set for sweep time, in this case the Analyzer corrects the [measurement delay](#) value. To set the minimum possible sweep time, set the measurement delay to zero.

---



To set the sweep time value, use the following softkeys:

**Stimulus > Sweep Time**



To set the measurement delay, use the following softkeys:

**Stimulus > Meas Delay**



---

## **WARNING**

To maintain correct sweep time value when using the C4209 or C4220 Analyzers without a frequency extenders, the digital cables connecting the Analyzer and the frequency extenders must be disconnected.

---

The sweep time should not be confused with the measurement cycle time displayed in the Analyzer status bar (See [Hide/Show Cycle Time](#)). The table below shows the difference between sweep time and cycle time.

	<b>Sweep Time value</b>	<b>Cycle Time value</b>
Method	Theoretically estimated	Actually measured
Scope	One channel	All sweeping channels
Sweep direction	One sweep direction <sup>1</sup>	All sweep directions <sup>2</sup>
Range	From the first sweep point to the last sweep point, excluding the time between sweeps	Between the start points of two consecutive measurement cycles, including the time between sweeps
<sup>1</sup> One port is the stimulus source.  <sup>2</sup> All ports can be used as a stimulus source.		

If one channel is open and a measurement is made in one sweep direction, the sweep time and cycle time are close. The difference is that the sweep time value does not include the delay between the sweeps.

## Trigger Settings

This section describes the trigger settings.

A trigger is a signal or event that starts the analyzer measurement cycle. The measurement cycle, by default, includes measurement of all opened channels. The analyzer measures the channels sequentially one after another in one measurement cycle. At some conditions the channel can be excluded from the measurement cycle (See [Channel Initiation Mode](#) and [Trigger Scope](#)).

For a detailed description of trigger state diagram see [Trigger State Diagram](#).

The trigger settings include:

- Selection the trigger source (See [Trigger Source](#)).
- Selection channel initialization mode (See [Channel Initiation Mode](#)).
- Setting the trigger scope (See [Trigger Scope](#)).
- Setting the averaging trigger function (See [Averaging Trigger](#)).

An external device can be used as a trigger source. For a detailed description of external trigger settings see [External Trigger Settings](#).

The trigger output of the Analyzer can be a trigger source for other devices (See [Trigger Output](#)).

## Trigger State Diagram

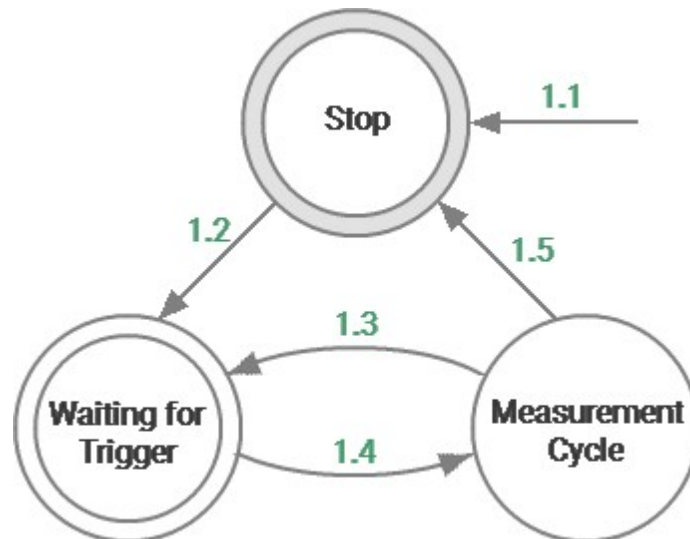
The trigger system operates at two levels: at the analyzer level and at the channel level.

### Analyzer States

The Analyzer can be in one of the following three states:

- **Stop** — the Analyzer waits for any channel to enter the **Initiated** state.
- **Waiting for Trigger** — the Analyzer waits for the trigger signal. If the **Internal** trigger source (see [Trigger Source](#)) is selected, it is automatically generated.
- **Measurement Cycle** — all initiated channels are measured in turn.

The figure below shows the states of the Analyzer, and the transitions between them.



Analyzer states and transitions

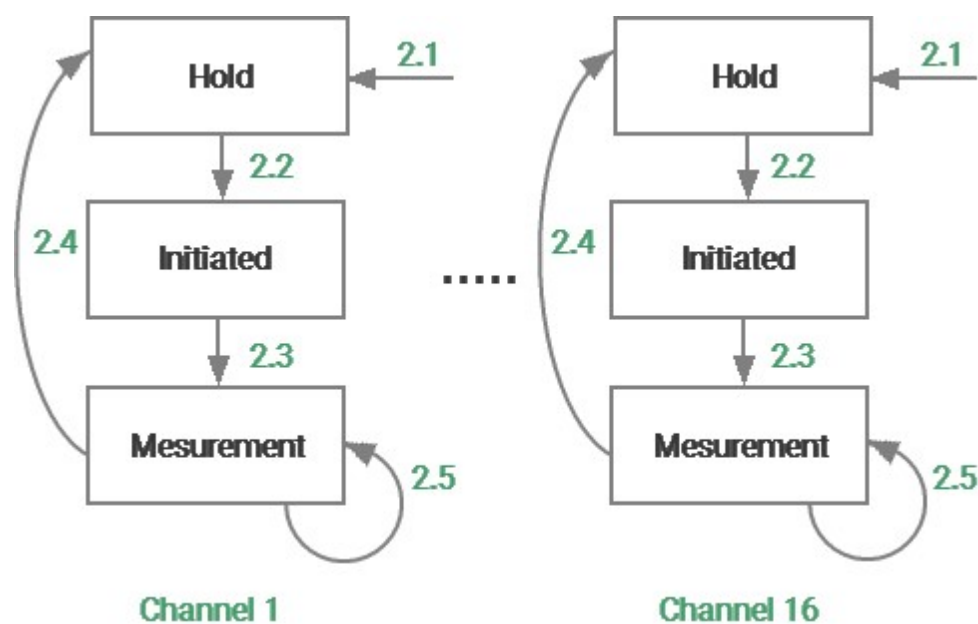
### Channel States

Channels can be in one of the three following states:

- **Hold** — the channel waits for the initiation. If the continuous initiation mode (see [Channel Initiation Mode](#)) is selected, the channel is automatically initiated.
- **Initiated** — the channel waits for the measurement after the trigger signal and measurement of other channels in the queue.
- **Measurement** — the channel is measured.

The figure below shows the channel states, and the transitions between them.





Channel states and transitions

The table below describes the transitions between analyzer and channel states.

Transition	Condition	Button	Command
<b>1.1</b>  To Stop	Power on	—	—
	Reset	<b>Preset</b>	<a href="#">SYST:PRESet</a> , <a href="#">*RST</a>
	Abort of the current measurement cycle.	<b>Trigger Restart</b> >	<a href="#">ABORT</a>
	Changing Analyzer settings by user or by the SCPI command.	For example: <b>Stimulus Start</b> >	For example: <a href="#">SENS:FREQ:START</a>
<b>1.2</b>  Stop Waiting Trigger → for	One or more channels make the <a href="#">transition 2.2</a> to the Initiated state.	—	—

Transition	Condition	Button	Command
<b>1.3</b>  Waiting for Trigger → Measurement Cycle	Automatically, if the trigger source is set to <b>Internal</b> .	Trigger source Internal >	<a href="#">TRIG:SOUR INT</a>
	At a signal arrival at the external trigger input, if the trigger source is set to <b>External</b> .	Trigger source External >	<a href="#">TRIG:SOUR EXT</a>
	At a softkey pressing, if the trigger source is set to <b>Manual</b> .	Trigger source Manual >  Trigger Trigger >	<a href="#">TRIG:SOUR MAN</a>
	Upon receipt of SCPI command, if the trigger Source is set to <b>Bus</b> .	Trigger source Bus >	<a href="#">TRIG:SOUR BUS</a> <a href="#">TRIG:SING</a> , <a href="#">TRIG</a> , <a href="#">*TRG</a>
<b>1.4</b>  Measurement Cycle → Waiting for Trigger	At the end of a measurement cycle, when at least one channel has the <b>Continuous</b> initiation mode.	Trigger Continuous >	<a href="#">INIT:CONT ON</a>
	After measuring a point, when the <b>On Point</b> trigger function is active.	Ext Trigger > Event > On Point	<a href="#">TRIG:POIN ON</a>
<b>1.5</b>  Measurement Cycle → Stop	At the end of a measurement cycle, when the <b>Continuous</b> initiation mode is	Trigger Hold All Channels >	—

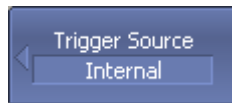
Transition	Condition	Button	Command
	disabled for all channels.		
2.1 To Hold	The same condition as <a href="#">transition 1.1</a>	—	—
	When the Initiation Mode of the channel has been set to <b>Hold</b> .	<b>Trigger Hold</b> >	<a href="#">INIT:CONT OFF</a>
2.2 Hold Initiated →	Every time if the <b>Continuous</b> initialization mode of the channel is turned on.	<b>Trigger Continuous</b> >	<a href="#">INIT:CONT ON</a>
	Once when the <b>Single</b> initiation mode of the channel has been set.	<b>Trigger Single</b> >	<a href="#">INIT</a>
2.3 Initiated → Measurement	Upon the occurrence of one of the conditions <a href="#">transition 1.3</a> and after measurement of other channels in the queue.	—	—
2.4 Measurement → Hold	At the end of channel measurement.	—	—
2.5	if the averaging trigger function is on, the	<b>Average Avg Trigger</b> > <b>&gt; On</b>	<a href="#">TRIG:AVER</a>

Transition	Condition	Button	Command
Repeat measurement	measurement repeats N times, where N is averaging factor.		

## Trigger Source

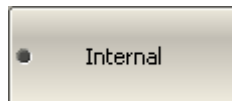
One of four trigger sources can be selected. This setting works at the analyzer level.

Trigger Source	Function
<b>Internal</b> [default]	The analyzer generates a trigger signal automatically when needed.
<b>External</b>	A trigger signal is a logic signal at the external trigger input (See <a href="#">External Trigger Settings</a> ).
<b>Bus</b>	The trigger signal is generated by a command from the program controlling the Analyzer via SCPI or COM.
<b>Manual</b>	The trigger signal is generated by pressing the <b>Trigger</b> softkey in the analyzer Software.



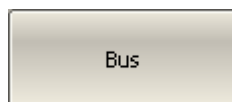
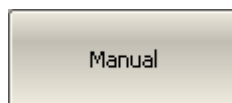
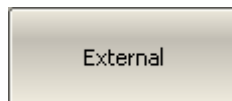
To set the trigger source, use the following softkeys:

**Stimulus > Trigger > Trigger Source**



Then select the required trigger source:

- **Internal**
- **External**
- **Manual**
- **Bus**



---

[TRIG:SOUR](#)

Selects the trigger source.



Trigger softkey generates the trigger in manual trigger mode.

**Stimulus > Trigger > Trigger**

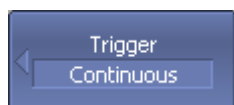
---

## Channel Initiation Mode

The channel initiation mode determines whether or not the channel will be included in the measurement cycle when a trigger signal is detected.

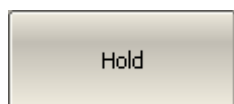
The channel to which the function is applied must be preselected as active (See [Selection of Active Trace/Channel](#) ).

Channel Initiation Mode	Function
<b>Continuous</b> [default]	The channel automatically transits to the <a href="#">Initiated state</a> at the end of each measurement.
<b>Single</b>	The channel is initiated once. At the end of the measurement, the channel goes into the <a href="#">Hold state</a> .
<b>Hold</b>	The channel is idle and not updating.



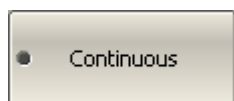
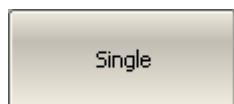
To set the channel initiation mode, use the following softkeys:

### Stimulus > Trigger



Then select the required channel initiation mode:

- **Hold**
- **Single**
- **Continuous**

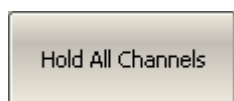


### [INIT:CONT](#)

Turns the continuous initiation mode ON/OFF.

### [INIT](#)

Sets the single initiation mode once.



To set the appropriate mode for all channels, use the following softkeys:

### Stimulus > Trigger > Hold All Channels

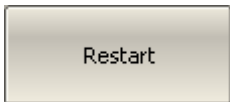


**Stimulus > Trigger > Continuous All Channels**

---

[INIT:CONT:ALL](#)

Turns the continuous initiation mode for all channels ON/OFF.



Restart softkey aborts the sweep and transits the Analyzer to stop state, then if there are channels in the continuous initiation state the Analyzer transits to the waiting for a trigger state (See [Trigger State Diagram](#)).

**Stimulus > Trigger > Restart**

---

[ABOR](#)

Aborts the sweep.

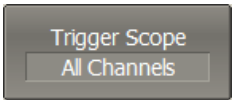
---



# Trigger Scope

The trigger scope function selects whether all initiated channels or an active channel, if initiated, will be measured on a trigger condition.

Trigger Scope	Function
<b>All Channel</b>  [default]	All initiated <sup>1</sup> channels will be measured on a trigger condition.
<b>Active Channel</b>	Only the active channel, if initiated, will be measured on a trigger condition .
<sup>1</sup> For a detailed description of the channel initiation mode settings, see <a href="#">Channel Initiation Mode</a> .	



To set the trigger scope, use the following softkeys:

**Stimulus > Trigger > Trigger Scope [All Channels | Active Channel]**

[TRIG:SCOP](#)

Sets or reads out the trigger scope.

## Averaging Trigger

The averaging trigger function allows for completing the averaging with a single trigger signal. This function affects the channels in which the averaging function is enabled (See [Averaging Setting](#)).

Averaging Trigger	Function
<b>OFF</b>  [default]	One sweep is performed in response to one trigger signal regardless of the state of the channel averaging function. If the channel averaging is turned on, N trigger signals are required to complete the averaging process (where N is the averaging factor). The trigger signal does not reset the result of the previous averaging.
<b>ON</b>	N sweeps are performed in response to one trigger signal for the channel with the averaging on (where N is the averaging factor). One trigger signal is required to complete the averaging process in the channel. The trigger signal starts a new averaging cycle in the channel.

The averaging trigger function is convenient in conjunction with an external, software (BUS), or manual trigger source. When the function is enabled, the averaging result can be obtained on one trigger signal by performing a number of sweeps equal to the averaging factor (See [Averaging Setting](#)). When the internal trigger source is used it is recommended to turn OFF this function.

---

### NOTE

If the trigger event function is set to [On Point](#), then it takes precedence over the averaging trigger function. In this case, to complete averaging, the number of trigger pulses equal to the number of points multiplied by the averaging factor is required.

---

---

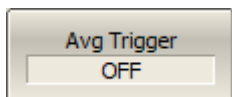
**NOTE**

If multiple channels are open at the same time, one trigger signal starts a measurement cycle the required number of times for the channels with averaging on, and once for channels with averaging off.

---



To enable/disable the averaging trigger function, use the following softkeys:

**Average > Avg Trigger**

The function changes between the values:

- **ON**
- **OFF**

---

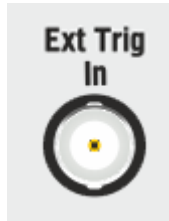
**[TRIG: AVER](#)**

Turns the averaging trigger function ON/OFF.

---

## External Trigger Settings

This section describes settings of the external trigger. The logic signal at the **Ext Trig In** on the rear panel of the analyzer is an external trigger signal (See [Instrument Series](#)).



External Trigger Signal Input Connector

To work with an external trigger:

- Select trigger source **External** (See [Trigger Source](#)).
- Set the external trigger event, polarity, position and delay (See the subsections in this section).

## External Trigger Event

This setting allows to select the external trigger event.

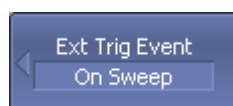
Trigger event	Function
<b>On sweep</b> [default]	One trigger signal starts a full measurement cycle, that is, the measurement of all frequency points of all channels included in the measurement cycle.
<b>On point</b>	One trigger signal starts the measurement of one frequency point of a channel. The next trigger signal starts the measurement of the next frequency point of the channel, and so on.

---

### NOTE

If the **Averaging Trigger** function and the **On point** trigger function are enabled at the same time, the **On point** trigger function has priority. In this case,  $N * P$  trigger signals are required to complete the averaging, where  $N$  is the averaging factor, and  $P$  is the number of points.

---

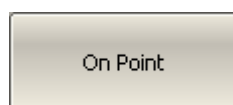
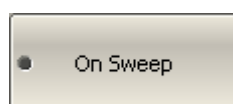


To select an external trigger event, use the following softkeys:

**Stimulus > Trigger > Ext Trigger > Event**

Then select the required external trigger event:

- **On Sweep**
- **On Point**



---

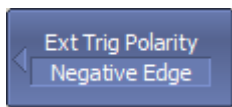
### [TRIG:POIN](#)

Turns the point trigger feature ON/OFF.

---

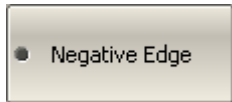
## External Trigger Polarity

Trigger polarity	Function
<b>Negative Edge</b> [default]	The negative edge of the input signal of an external trigger is a trigger signal.
<b>Positive Edge</b>	The positive edge of the input signal of an external trigger is a trigger signal.



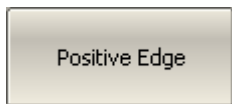
To select external trigger polarity, use the following softkeys:

**Stimulus > Ext Trigger > Polarity**



Then select the required external trigger polarity:

- **Negative Edge**
- **Positive Edge**

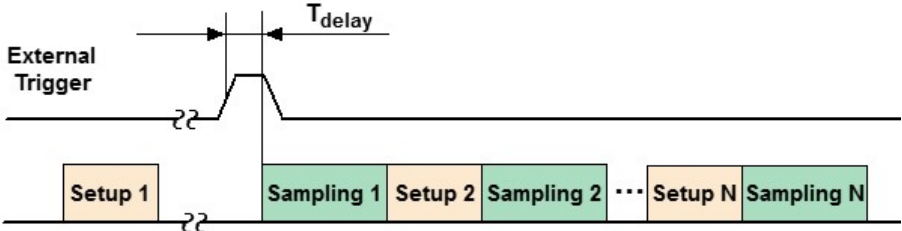
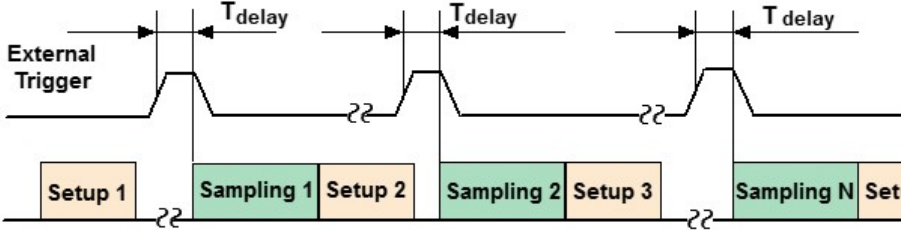
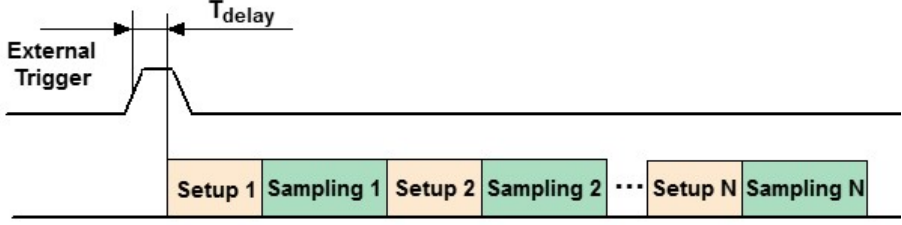


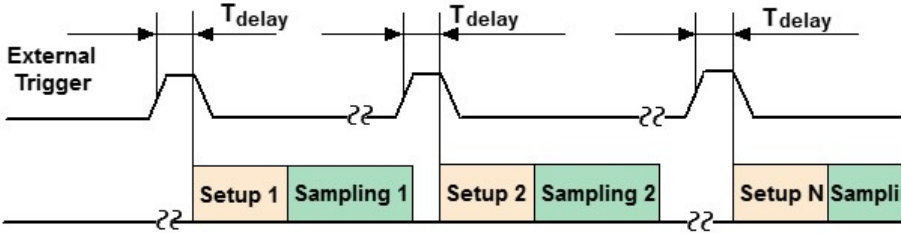
[TRIG:EXT:SLOP](#)

Sets or reads out the polarity of the external trigger.

## External Trigger Position

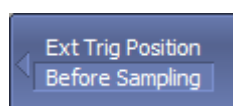
The position of the external trigger determines the moment when the analyzer expects an external trigger signal — before the frequency setup or before measuring (ADC sampling). The frequency setup precedes the measurement for each frequency point.

Trigger Position	Function
<b>Before Sampling</b>  [default]	<p>The trigger signal is expected before the ADC sampling when the frequency is already set. After sampling the Analyzer automatically transits to the next frequency (See figure below).</p>  <p>External Trigger Delay Before Sampling, Point trigger is OFF</p>  <p>External Trigger Delay Before Sampling, Point trigger is ON</p>
<b>Before Setup</b>	<p>The trigger signal is expected before the frequency setup. The frequency setup starts when the trigger signal arrives (See figure below). After the frequency setup is completed, the analyzer begins ADC sampling.</p>  <p>External Trigger Delay Before Setup, Point trigger is OFF</p>

Trigger Position	Function
	 <p>External Trigger Delay Before Setup, Point trigger is ON</p>

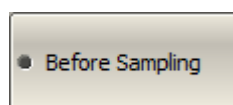
#### NOTE

This function is intended for use in conjunction with the **On Point** trigger function. In case of the **On Sweep** trigger function the trigger position will be performed only for the first sweep point.



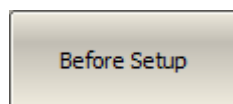
To select external trigger position, use the following softkeys:

**Stimulus > Trigger > Ext Trigger > Position**



Then select the required external trigger position:

- **Before Sampling**
- **Before Setup**



[TRIG:EXT:POS](#)

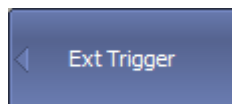
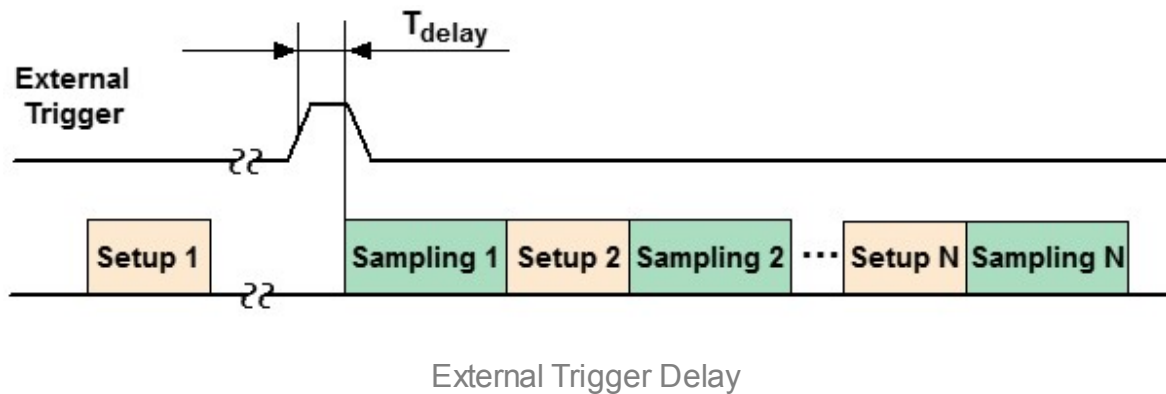
Selects the position of the external trigger.



## External Trigger Delay

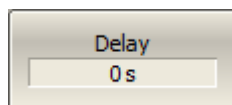
The external trigger delay sets the response delay with respect to the external trigger signal (See figure below).

The delay range and resolution depend on the Analyzer model (See corresponding [datasheet](#)).



To set the external trigger delay, use the following softkeys:

**Stimulus > Trigger > Ext Trigger > Delay**



[TRIG:EXT:DEL](#)

Sets or reads out the response delay with respect to the external trigger signal.

## Trigger Output

This section describes settings of the trigger output. The trigger output is a special analyzer connector used to output a logical signal from the Analyzer.

---

**NOTE**

The availability of the trigger output connector depends on the Analyzer model (See corresponding [datasheet](#)).

---

The trigger output is designed to synchronize external devices with the Analyzer measurement cycle.



External Trigger Output Connector

To work with trigger output:

- Turn on trigger output (See [Enabling Trigger Output](#)).
- Set the polarity of the trigger (See [Trigger Output Polarity](#)).
- Select the trigger signal condition (See [Trigger Output Function](#)).

## Enabling Trigger Output

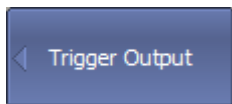
Trigger Output	Function
OFF	The trigger output is disabled.
ON	The trigger output is enabled.

---

### NOTE

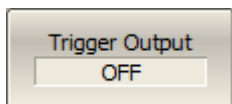
If the **Ready for Trigger** function is selected (See [Trigger Output Function](#)), the trigger source must be set to **External** to enable the trigger output (See [Trigger Source](#)).

---



To enable/disable the trigger output, use the following softkeys:

**Stimulus > Trigger > Trigger Output > Trigger Output**



The function changes between the values:

- ON
- OFF

---

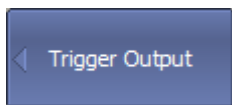
[TRIG:OUTP:STAT](#)

Turns the trigger output ON/OFF.

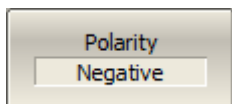
---

## Trigger Output Polarity

Trigger Output Polarity	Function
Negative	The negative edge of the signal at the trigger output corresponds to the event.
Positive	The positive edge of the signal at the trigger output corresponds to the event.



To select the polarity of the trigger output, use the following softkeys:



**Stimulus > Trigger > Trigger Output > Polarity**

The function changes between the values:

- **Negative**
- **Positive**

---

[TRIG:OUTP:POL](#)

Sets or reads out the trigger output.

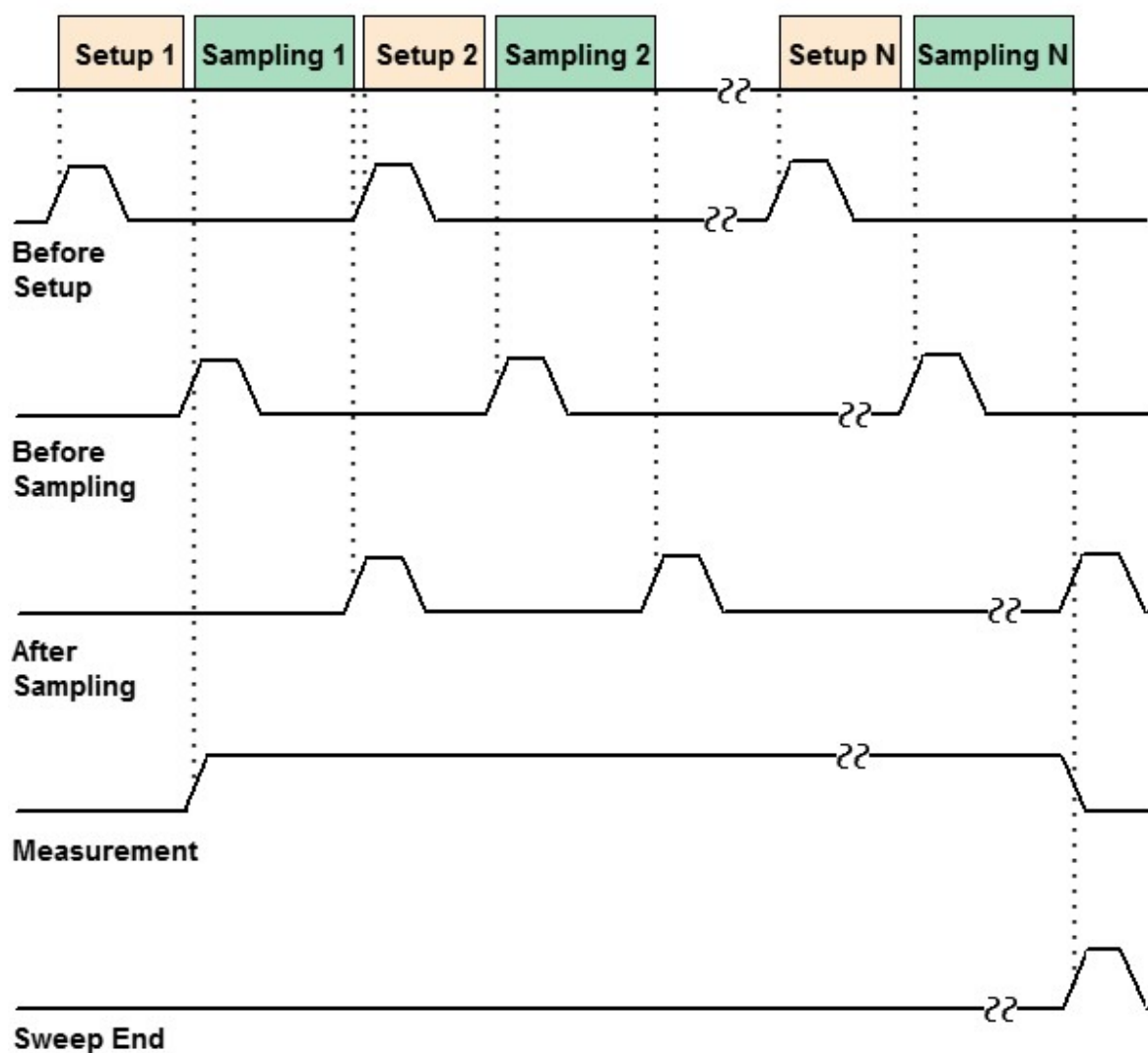
---

## Trigger Output Function

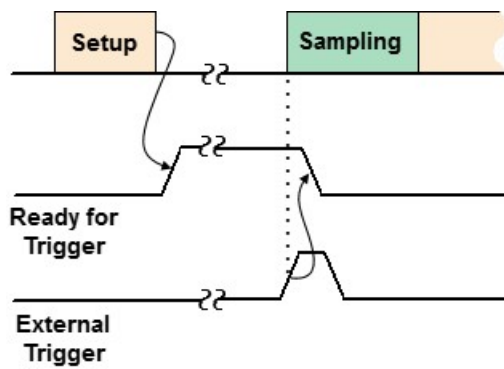
The purpose of the trigger output depends on the selected function.

Trigger Output Function	Function
<b>Before Setup</b>	Single pulse before setup frequency.
<b>Before Sampling</b>	Single pulse before sampling.
<b>After Sampling</b>	Single pulse after sampling.
<b>Ready for Trigger</b>	Indicates the ready for external trigger state. The signal position depends on the external trigger position setting.  After the arrival of the external trigger the ready for trigger signal is deselected and the measurement has begun.
<b>Sweep End</b>	Single pulse at the end of the sweep.
<b>Measurement</b>	The pulse duration is equal to the duration of the measurement from the first to the last point.

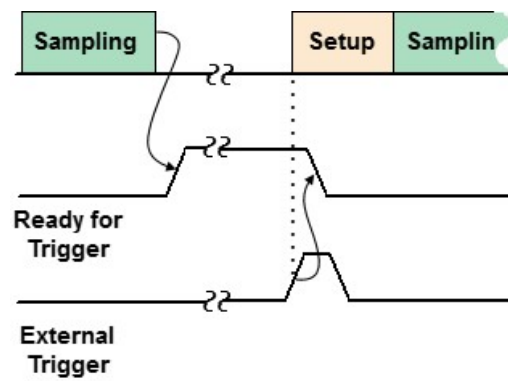
The figures below show the trigger output signal generation, depending on the selected trigger condition.



Trigger Output (except Ready for Trigger)

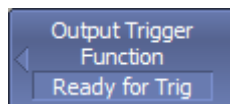


External Trigger set before sampling



External trigger set before setup

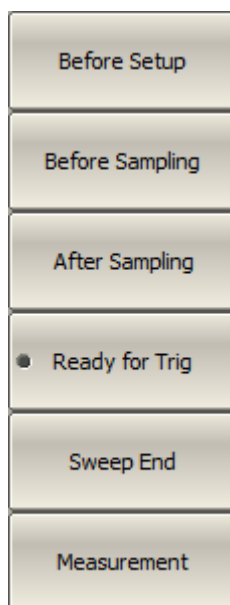
Trigger Output (Ready for Trigger only)



To select the function of the trigger output, use the following softkeys:

**Stimulus > Trigger > Trigger Output > Function**

Then select the required function of the trigger output:



- **Before Setup**
- **Before Sampling**
- **After Sampling**
- **Ready for Trig**
- **Sweep End**
- **Measurement**

[TRIG:OUTP:FUNC](#)

Selects the trigger output function.

## Measurement Parameters Settings

This section describes the settings for the measurement parameter selection. The parameter selection applies to traces within a channel.

The Analyzers allows for:

- S-Parameter measurement (See [S-Parameters](#)).
- Absolute power measurement at the receiver input (See [Absolute Measurements](#)).
- DC measurement (option) (See [DC Measurement](#)).



## S-Parameters

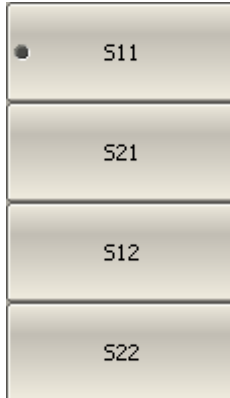
A measured S-parameter (S11, S21, S12, S22) is set for each trace. The trace to which the function is applied must be preselected as active (See [Selection of Active Trace/Channel](#)).

For a detailed description of the principle of measuring S-parameters see in [Principle of measuring S-parameters](#).



To set the measured parameter, use the following softkey:

**Measurement**



Then select the desired parameter by the corresponding softkey.

---

[CALC:PAR:DEF](#)

Selects the measurement parameter of the trace.

---

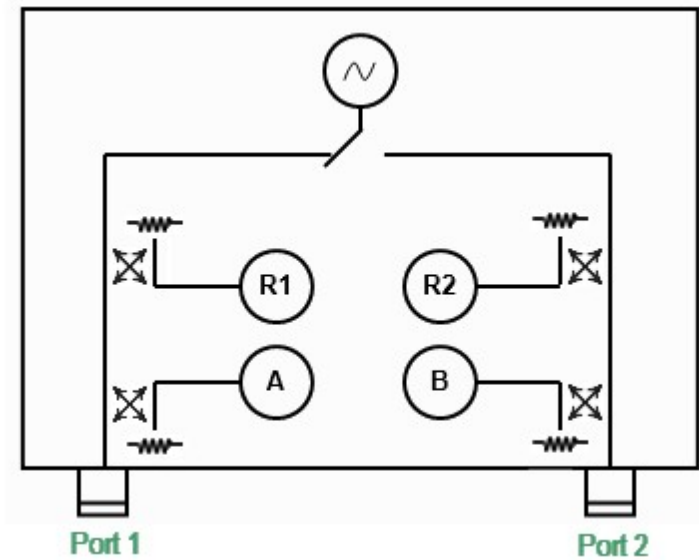
**NOTE**

Measured data can be set using the mouse (See [Measured Data Setting](#)).

---

# Absolute Measurements

Absolute measurements are measurements of the absolute power of a signal at a receiver input. Unlike relative measurements of S-parameters, which represent a relation between the signals at inputs of two receivers, absolute measurements determine the signal power at the input of one receiver. A two-port Analyzer has four independent receivers: **A, B, R1, R2** (See figure below).



Analyzer block diagram

The R1 and R2 are reference signal receivers. The A and B are test signal receivers. The A and R1 receivers are located in Port 1. The B and R2 receivers are located in Port 2. There are six types of absolute measurements depending on the port number (See table below).

Symbols	Definition
A(1)	Test signal receiver A (Source Port 1)
A(2)	Test signal receiver A (Source Port 2)
B(1)	Test signal receiver B (Source Port 1)
B(2)	Test signal receiver B (Source Port 2)
R1(1)	Reference signal receiver R1 (Source Port 1)

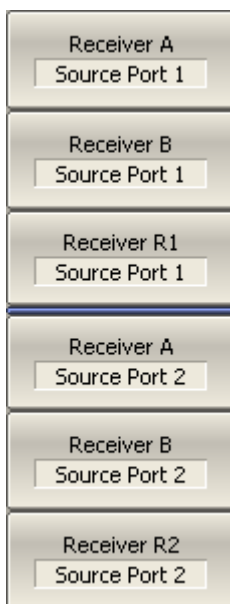
Symbols	Definition
<b>R2(2)</b>	Reference signal receiver R2 (Source Port 2)

A measured absolute parameter is set for each trace. The trace to which the function is applied must be preselected as active (See [Selection of Active Trace/Channel](#) ).



To select absolute measurement, click softkeys:

#### **Measurement > Absolute**



Then select the required parameter:

- **Receiver A, Source Port 1**
- **Receiver B, Source Port 1**
- **Receiver R1, Source Port 1**
- **Receiver A, Source Port 2**
- **Receiver B, Source Port 2**
- **Receiver R2, Source Port 2**

[CALC:PAR:DEF](#)

Selects the measurement parameter of the trace.

[CALC:PAR:SPOR](#)

Sets or reads out the number of the stimulus port when performing absolute measurements.

#### **NOTE**

In absolute measurement mode, **dBm** measurement units are used for logarithmic magnitude format, and **W** measurement units are used in linear magnitude format. Other formats are not applicable to absolute measurements.

## Format Setting

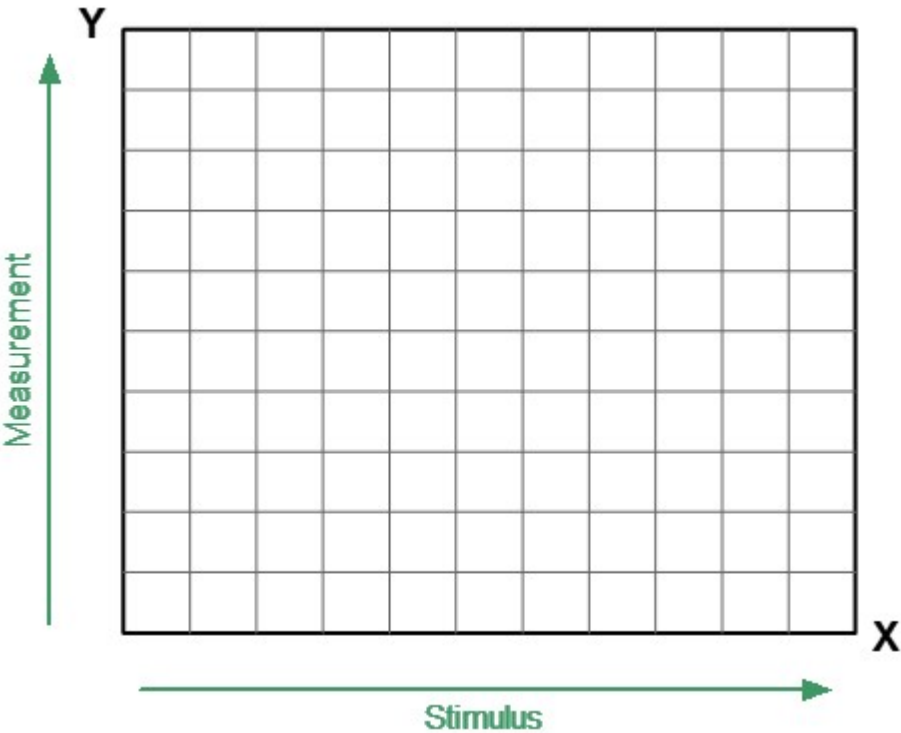
The format setting determines how measured data will be presented on the diagram.

The Analyzer offers three S-parameter measurement display types:

- [Rectangular format](#)
- [Polar format](#)
- [Smith chart format](#)

# Rectangular Formats

In this format, stimulus values are plotted along X-axis and the measured data are plotted along Y-axis (See figure below).



Rectangular format

To display complex-valued S-parameters along the scalar Y-axis, it must be transformed into a real number. Rectangular formats involve various types of transformation of an S-parameter

$$S = a + j \cdot b,$$

where *a* — real part of S-parameter complex value,

*b* — imaginary part of S-parameter complex value.

There are eight types of rectangular formats depending on the measured value plotted along Y-axis (See table below).

## Rectangular Formats

Format Type Description	Label	Data Type (Y-axis)		Measurement Unit (Y-axis)
Logarithmic	Log Mag	S-parameter	logarithmic	Decibel (dB)

Format Type Description	Label	Data Type (Y-axis)	Measurement Unit (Y-axis)
Magnitude		magnitude: $20 \cdot \log S $ , $ S  = \sqrt{a^2 + b^2}$	
Voltage Standing Wave Ratio	<b>SWR</b>	$\frac{1+ S }{1- S }$	Dimensionless value
Phase	<b>Phase</b>	S-parameter phase from $-180^\circ$ to $+180^\circ$ : $\frac{180}{\pi} \cdot \arctg \frac{b}{a}$	Degree (°)
Expanded Phase	<b>Expand Phase</b>	S-parameter phase, range expanded to from below $-180^\circ$ to over $+180^\circ$	Degree (°)
Group Delay	<b>Group Delay</b>	Signal propagation delay within the DUT: $-\frac{d\varphi}{d\omega}$ , $\varphi = \arctg \frac{b}{a}$ , $\omega = 2\pi \cdot f$	Second (sec.)
Linear Magnitude	<b>Lin Mag</b>	S-parameter linear magnitude: $\sqrt{a^2 + b^2}$	Dimensionless value
Real Part	<b>Real</b>	S-parameter real part: $a = \text{re}(S)$	Dimensionless value
Imaginary Part	<b>Imag</b>	S-parameter imaginary part: $b = \text{im}(S)$	Dimensionless value

The format for each trace of the channel can be selected individually. The trace must be activated before setting the format.



To choose a rectangular format, use the following softkey:

### **Format**

Then select the desired format:

- **Logarithmic magnitude**
- **SWR**
- **Phase**
- **Expanded phase**
- **Group delay**
- **Linear magnitude**
- **Real part**
- **Imaginary part**

---

### [CALC:FORM](#)

Sets or reads out the trace format.

---

---

### **NOTE**

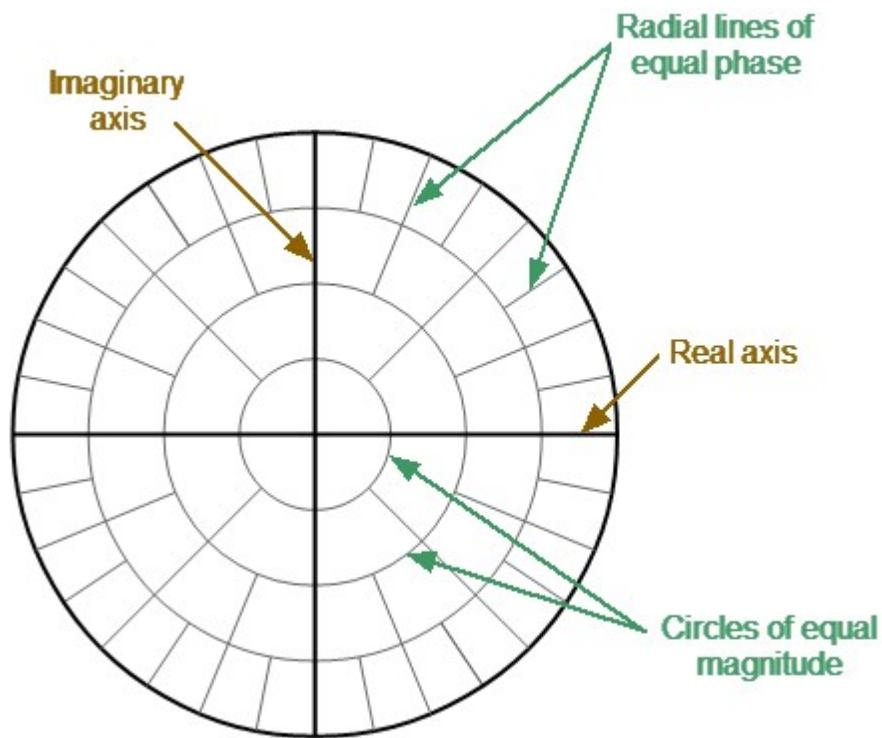
The display format can be set using the mouse (See [Display Format Setting](#)).

---



## Polar Format

The Polar format is used to display the amplitude and phase of the reflection coefficient ( $\Gamma$ ) when measuring S11 or S22. The complex reflection coefficient values are displayed on the polar diagram in the complex plane. The complex plane is formed by the real horizontal and the imaginary vertical axes. The grid lines correspond to points of equal amplitude and phase (See figure below).



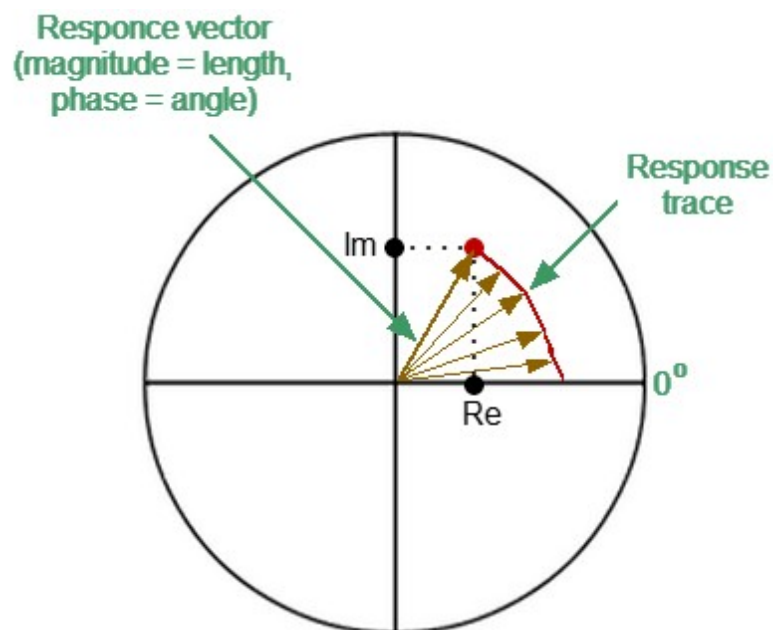
Polar format

---

**NOTE**

On circular diagrams (Polar and Smith chart), any point of the trace can be defined in the following two ways (See figure below):

- Coordinates of the point (Re, Im) on the real and imaginary coordinate axes.
- Parameters of the vector directed to the point from the center of the diagram. The length of this vector is equal to the response amplitude, and the angle between the vector and the positive part of the real coordinate axis is equal to the phase of the response. The angle is calculated counterclockwise.



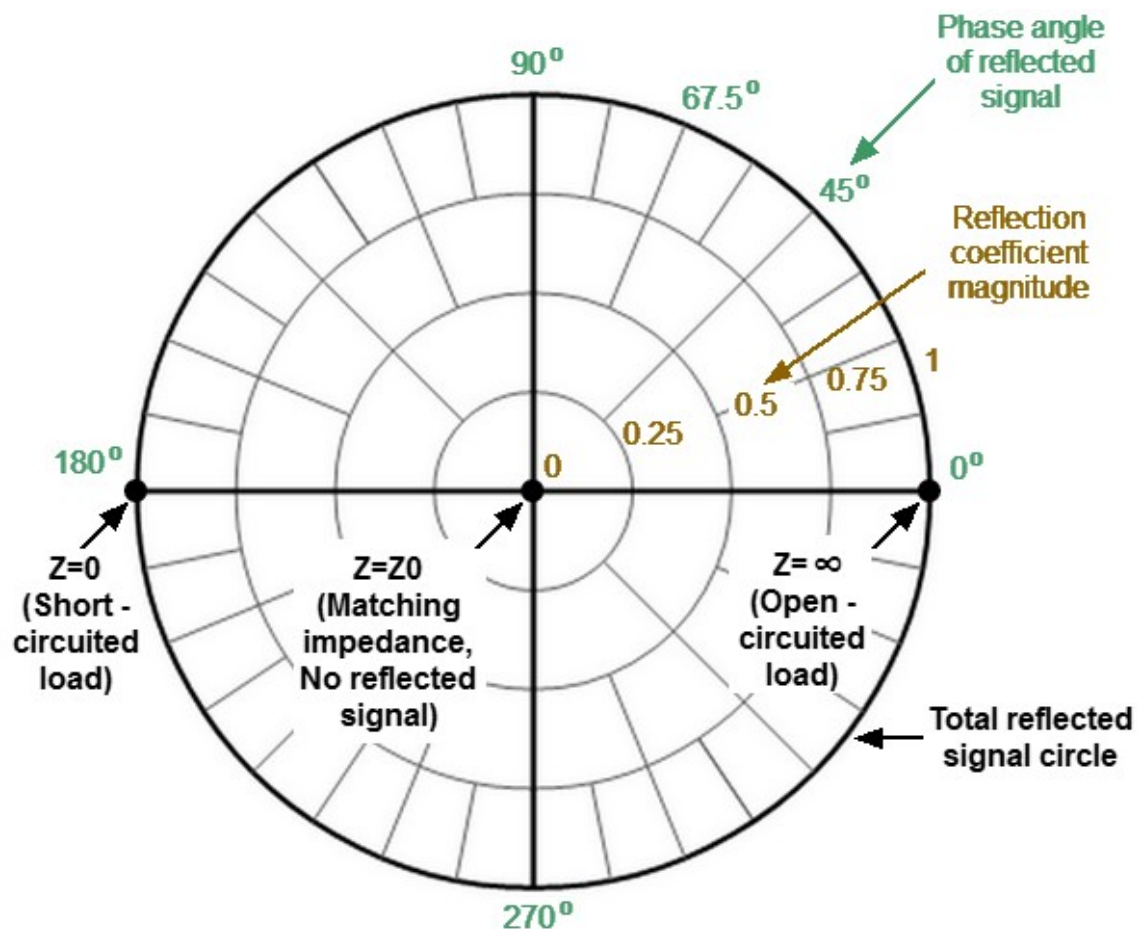
---

**NOTE**

Traces on all types of Smith chart and polar format are the same, the analyzer replaces the base grid and default marker format when switching formats.

---

The Polar format diagram with the characteristic points is shown in the figure below.



Properties of Polar format

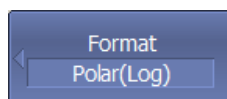
Basic properties of the Polar format:

- The center of the diagram corresponds to the reflection coefficient  $\Gamma = 0$  (reference impedance  $Z_0$  on the input test port of the DUT when measuring  $S_{11}$ ,  $S_{22}$ , matched circuit, no reflection).
- The outer circle of the diagram corresponds to the reflection coefficient  $\Gamma = 1$  ( $|S_{ii}| = 1$ , unmatched circuit, total reflection).
- Points with the same amplitude are located on a circle with the center coinciding with the center of the diagram.
- Points with the same phase are located on a line starting from the center.
- At the rightmost point of the horizontal axis, the impedance has an infinitely large value (Open circuited load).
- At the leftmost point of the horizontal axis, the impedance value is zero (Short circuited load).

The polar graph does not have a frequency axis, so frequency is indicated by markers. There are three types of polar formats corresponding to the data displayed by the marker; the traces remain the same for all the format types (See table below).

Format Type Description	Label	Data Displayed by Marker	Measurement Unit
Linear Magnitude and Phase	<b>Polar (Lin)</b>	S-parameter linear magnitude	Dimensionless value
		S-parameter phase	Degree (°)
Logarithmic Magnitude and Phase	<b>Polar (Log)</b>	S-parameter logarithmic magnitude	Decibel (dB)
		S-parameter phase	Degree (°)
Real and Imaginary Parts	<b>Polar (Re/Im)</b>	S-parameter real part	Dimensionless value
		S-parameter imaginary part	Dimensionless value

The format for each trace of the channel can be selected individually. The trace must be activated before setting the format.

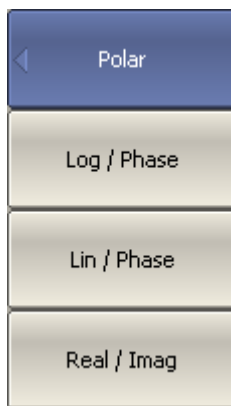


To choose a Polar format, use the following softkeys:

#### **Format > Polar**

Then select the desired format:

- **Logarithmic magnitude and phase**
- **Linear magnitude and phase**
- **Real and imaginary parts**



---

[CALC:FORM](#)

Sets or reads out the trace format.

---

---

**NOTE**

The display format can be set using the mouse (See [Display Format Setting](#)).

---

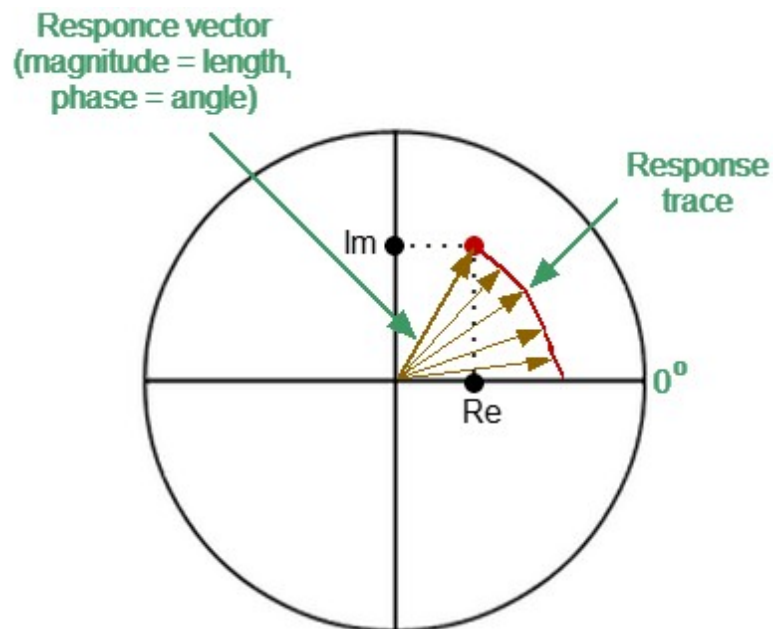
## Smith Chart Format

---

### NOTE

On circular diagrams (Polar and Smith chart), any point of the trace can be defined in the following two ways (See figure below):

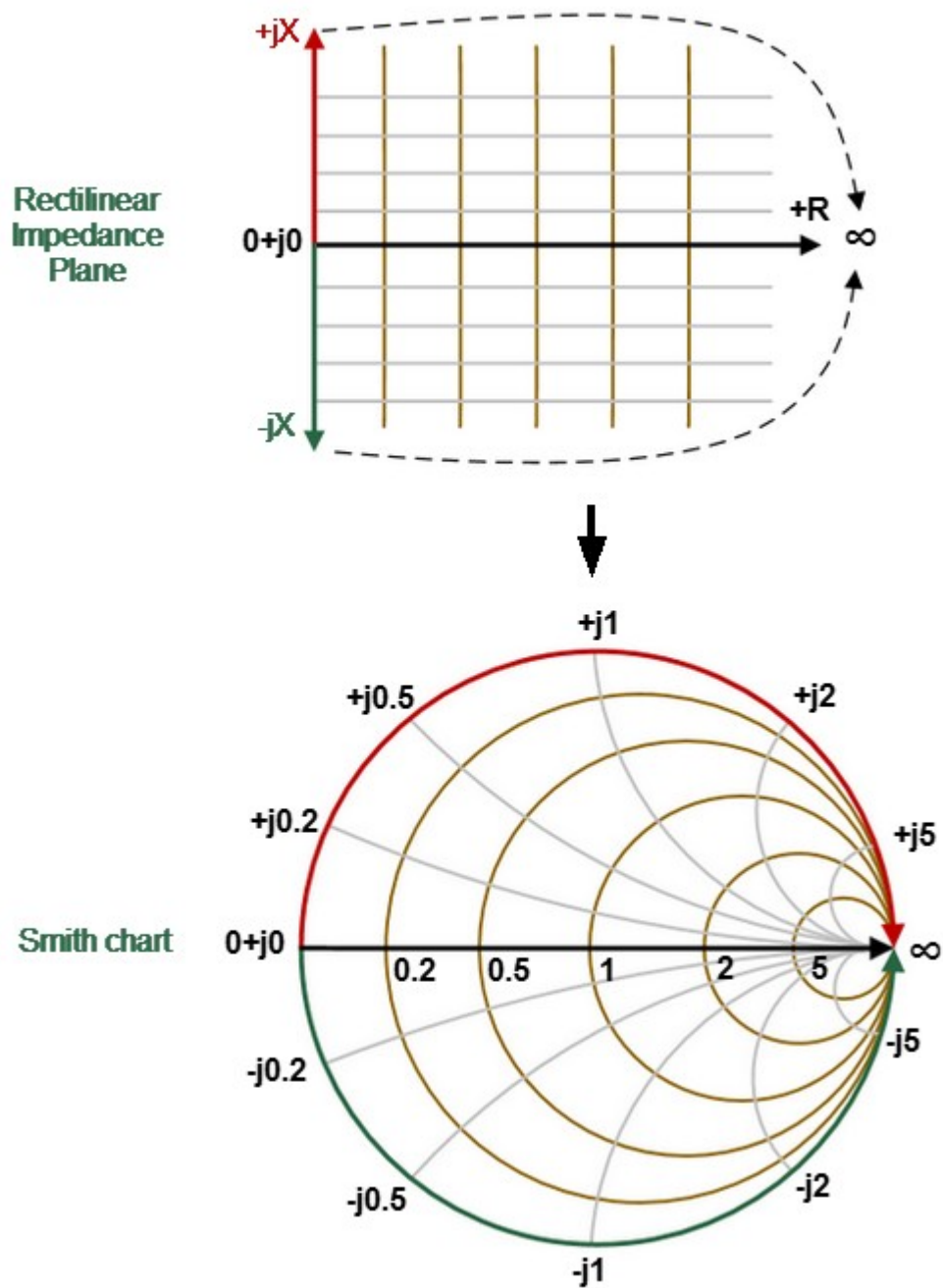
- Coordinates of the point (Re, Im) on the real and imaginary coordinate axes.
- Parameters of the vector directed to the point from the center of the diagram. The length of this vector is equal to the response amplitude, and the angle between the vector and the positive part of the real coordinate axis is equal to the phase of the response. The angle is calculated counterclockwise.



---

The Smith chart is a circular chart on which the measured complex reflection coefficients ( $S_{11}$ ,  $S_{22}$ ) are compared with the normalized impedance of the DUT.

The Smith chart is formed from a rectilinear impedance plane by collapsing the area with positive resistance into a single unit circle (See figure below).



Converting Rectilinear Impedance Plane to Smith Chart

Basic properties of the Smith chart (See figure below):

- Each point on the diagram is equivalent to the complex impedance of the DUT:

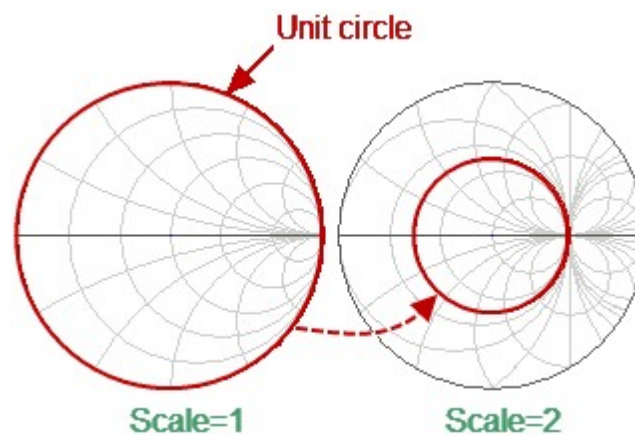
$$Z = R + jX,$$

where  $R$  — real part of the impedance (resistance),  $X$  — imaginary part of the impedance (reactance).

- The horizontal axis is resistance; reactance on this axis is equal to zero.
- Grid lines of the diagram consist of circles of constant resistance and arcs of constant reactance.
- The center of the diagram corresponds to the system reference impedance ( $Z/Z_0 = 1$ ).
- At the rightmost point of the horizontal axis, the impedance has an infinitely large value (Open circuited load).
- At the leftmost point of the horizontal axis, the impedance value is zero (Short circuited load).
- The outer circle of the diagram at scale = 1 (or unit circle) corresponds to a zero resistance value (reactance only). The measured points inside the unit circle correspond to the passive load, the points outside to the active load.

#### NOTE

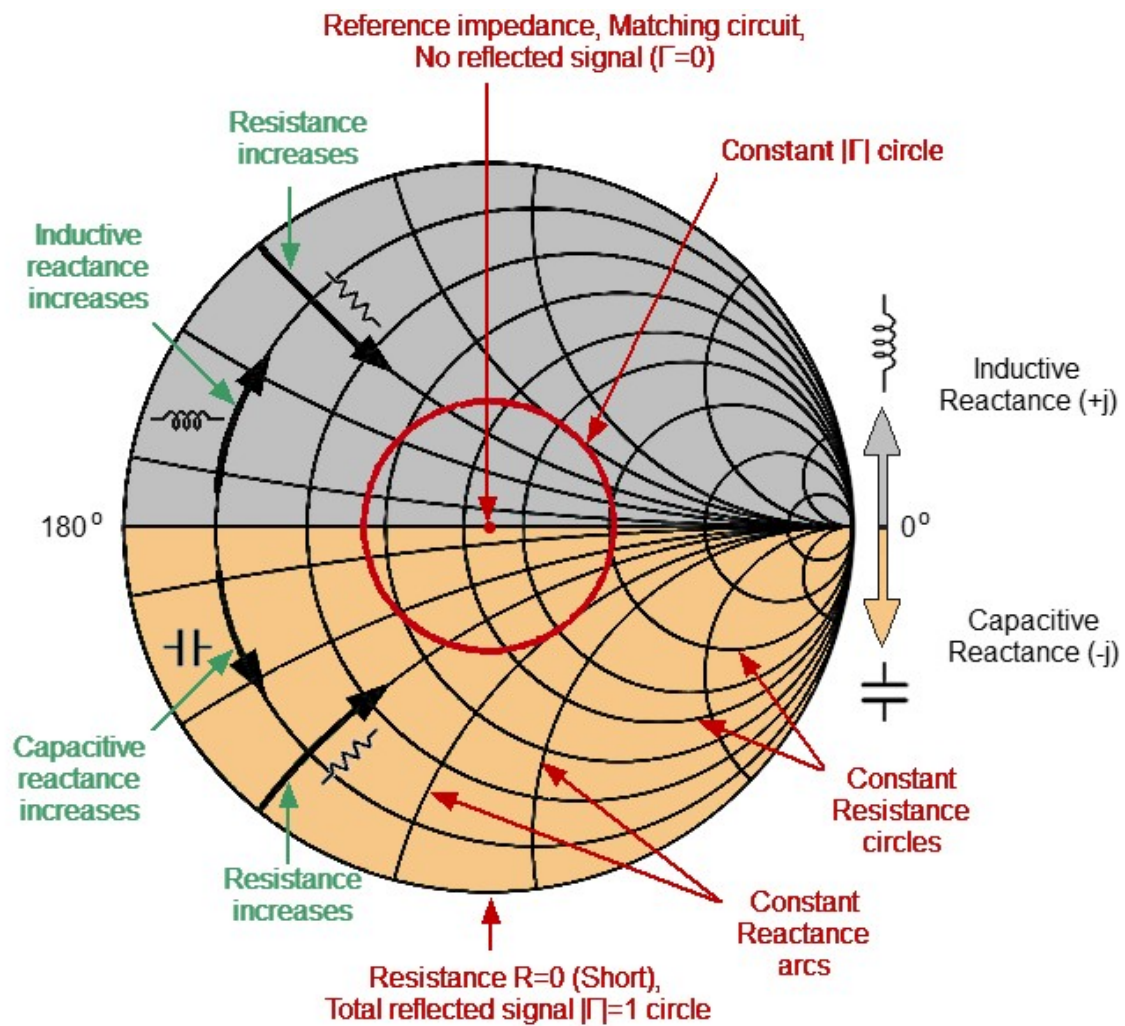
Location of the unit circle at a scale greater than 1



- The upper and lower halves of the diagram correspond to the positive (inductive) and negative (capacitive) reactive components of impedance.
- Reflection coefficient value ( $\Gamma$ ) at any point of the diagram is determined by the distance from it to the center of the diagram. Thus, any circle with the center coinciding with the center of the diagram contains equal values of the modulus of the reflection coefficient. The center of the diagram corresponds to a matched circuit with no reflect signal ( $\Gamma = 0$ ). The unit circle diagram corresponds to an unmatched circuit with total reflection  $|\Gamma| = 1$ .

Use the Smith chart to assess circuit mismatch and determine whether the load is resistive, inductive, capacitive, or complex. The Smith chart format is useful for looking for mismatch introduced by parasitic elements connected in series with the DUT.



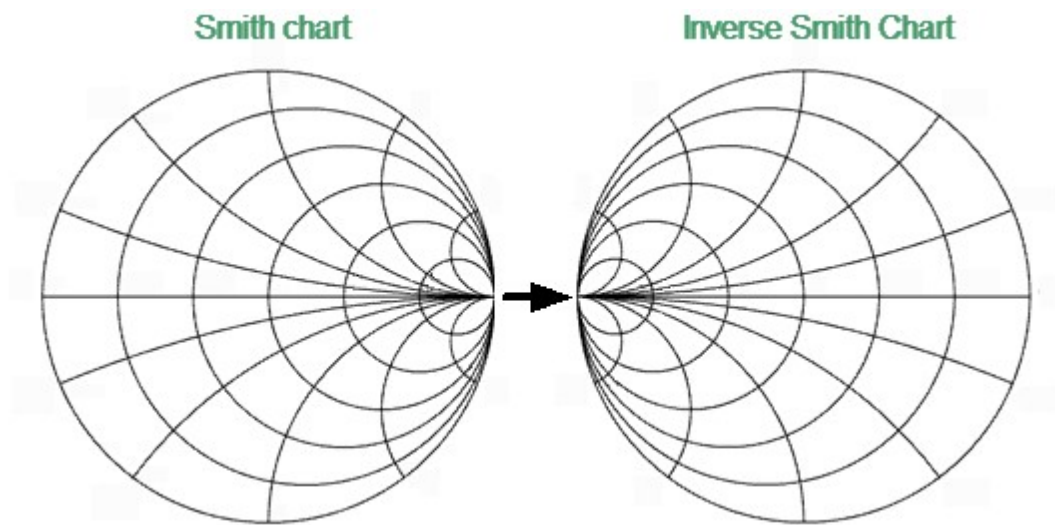


Smith chart properties

### Inverse Smith Chart (Complex Admittance)

The Inverse Smith chart is a circular chart on which the measured complex reflection coefficients ( $S_{11}$ ,  $S_{22}$ ) are compared with the normalized DUT admittance. Complex admittance is the inverse of complex impedance.

To build an Inverse Smith chart, mirror the Smith chart on the horizontal axis (See figure below).



Convert Smith Chart to Inverse Smith Chart

Basic properties of the Inverse Smith chart:

- Each point on the diagram is equivalent to the complex conductance of the DUT:

$$Y = G + jB,$$

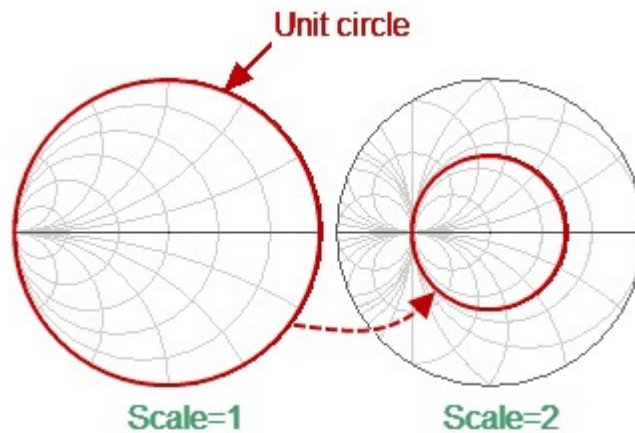
where  $G$  — real part of conductivity (conductance),  $B$  — imaginary part of conductivity (susceptance).

- The horizontal axis is only conductance; susceptance on this axis is equal to zero.
- The grid lines of the diagram consist of circles of constant conductance width and arcs of constant susceptance width.
- The center of the diagram corresponds to the reference conductivity of the system ( $Y/Y_0 = 1$ ).
- At the leftmost point of the horizontal axis, admittance is infinitely large (Short circuited load).
- At the rightmost point of the horizontal axis, admittance is equal to zero (Open circuited load).
- The outer circle at scale = 1 (or unit circle) corresponds to the zero value of conductance (susceptance only). The measured points inside the unit circle correspond to the passive load, the points outside to the active load.

---

**NOTE**

Position of the unit circle at a scale greater than 1



- 
- The upper and lower halves of the diagram correspond to the negative (inductive) and positive (capacitive) reactive components (admittance).
  - The reflection coefficient display ( $\Gamma$ ) on the Inverse Smith chart coincides with its display on the Smith chart. The center of the diagram corresponds to a matched circuit with no reflected signal ( $\Gamma=0$ ). The unit circle diagram corresponds to an unmatched circuit with total reflection  $|\Gamma| = 1$ .

Use the Inverse Smith chart (admittance diagram) to search for a mismatch introduced by the parasitic elements shunting the DUT.

The Smith chart format does not have a frequency axis, so frequency is indicated by markers.

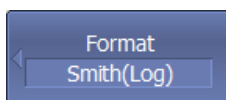
There are five types of Smith chart formats (See table below) corresponding to the data displayed by the marker; the traces remain the same for all the format types.

Format Type Description	Label	Data Displayed by Marker	Measurement Unit
Linear Magnitude and Phase	<b>Smith (Lin)</b>	S-parameter magnitude linear	Dimensionless value
		S-parameter phase	Degree (°)
Logarithmic Magnitude and Phase	<b>Smith (Log)</b>	S-parameter magnitude logarithmic	Decibel (dB)

Format Type Description	Label	Data Displayed by Marker	Measurement Unit
		S-parameter phase	Degree (°)
Real and Imaginary Parts	<b>Smith (Re/Im)</b>	S-parameter real part	Dimensionless value
		S-parameter imaginary part	Dimensionless value
Complex Impedance (at Input)	<b>Smith (R + jX)</b>	Resistance at input: $R = re(Z_{inp})$ $Z_{inp} = Z_0 \frac{1+S}{1-S}$	Ohm (Ω)
		Reactance at input: $X = im(Z_{inp})$	Ohm (Ω)
		Equivalent capacitance or inductance: $C = -\frac{1}{\omega X}, \quad X < 0$ $L = \frac{X}{\omega}, \quad X > 0$	Farad (F) Henry (H)
Complex admittance (at Input)	<b>Smith (G + jB)</b>	Conductance at input: $G = re(Y_{inp})$ $Y_{inp} = \frac{1}{Z_0} \cdot \frac{1-S}{1+S}$	Siemens (S)
		Susceptance at input: $B = imp(Y_{inp})$	Siemens (S)

Format Type Description	Label	Data Displayed by Marker	Measurement Unit
		Equivalent capacitance or inductance:  $C = \frac{B}{\omega}, \quad B > 0$  $L = -\frac{1}{\omega B}, \quad B < 0$	Farad (F)  Henry (H)
Z0 — test port impedance. Z0 setting is described in <a href="#">System Impedance Z0</a> .			

The format for each trace of the channel can be selected individually. The trace must be activated before setting the format.



To choose a Smith chart format, use the following softkeys:

#### Format > Smith



Then select the desired format:

- **Logarithmic magnitude and phase**
- **Linear magnitude and phase**
- **Real and imaginary parts**
- **Complex impedance (at input)**
- **Complex admittance (at input)**

#### [CALC:FORM](#)

Sets or reads out the trace format.

#### NOTE

The display format can be set using the mouse (See [Display Format Setting](#)).



## Scale Settings

The section describes how to set the scale for the different available formats.

The scale setting options depend on the selected data display format: rectangular format or circular format. For a detailed description of the scale settings for the different formats, see [Rectangular Scale](#) and [Circular Scale \(Polar and Smith\)](#).

It is possible to apply the [Automatic Scaling](#) function for both formats.

These functions are also available when using the rectangular format:

- [Reference Level Automatic Selection](#)
- [Automatic Reference Level Tracking](#)

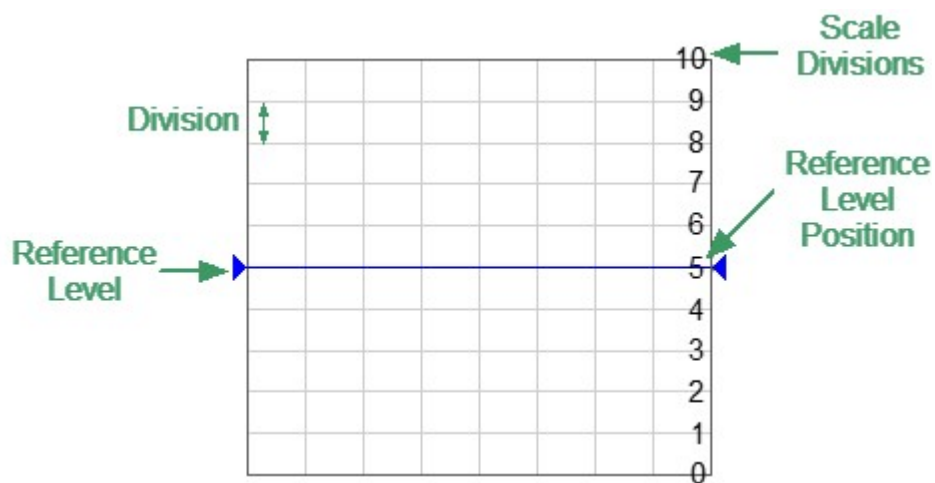
The scaling function is under trace settings.

This section also describes the electric delay setting functions (See [Electrical Delay Setting](#)) and phase offsets (See [Phase Offset Setting](#)).

## Rectangular Scale

For [rectangular format](#), the following parameters can be set (See figure below):

- scale division
- reference level value
- reference level position
- number of scale divisions



Rectangular scale

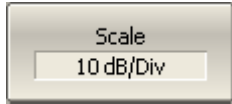
The scale of each trace can be set independently. The trace to which the function is applied must be preselected as active (See [Selection of Active Trace/Channel](#)).





To set the scale of a trace, use the following softkeys:

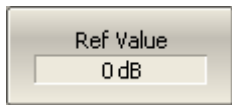
**Scale > Scale**



---

[DISP:WIND:TRAC:Y:PDIV](#)

Sets or reads out the trace scale. Sets the scale per division.



To set the reference level, use the following softkeys:

**Scale > Ref Value**

---

[DISP:WIND:TRAC:Y:RLEV](#)

Sets the value of the reference line (response value on the reference line).



To set the position of the reference level, use the following softkeys:

**Scale > Ref Position**

---

[DISP:WIND:TRAC:Y:RPOS](#)

Sets the position of the reference line.



To set the number of trace scale divisions, use the following softkeys:

**Scale > Divisions**

**NOTE:** The number of scale divisions affects all traces of the channel.

---

[DISP:WIND:Y:DIV](#)

Sets the number of the vertical scale divisions.

---

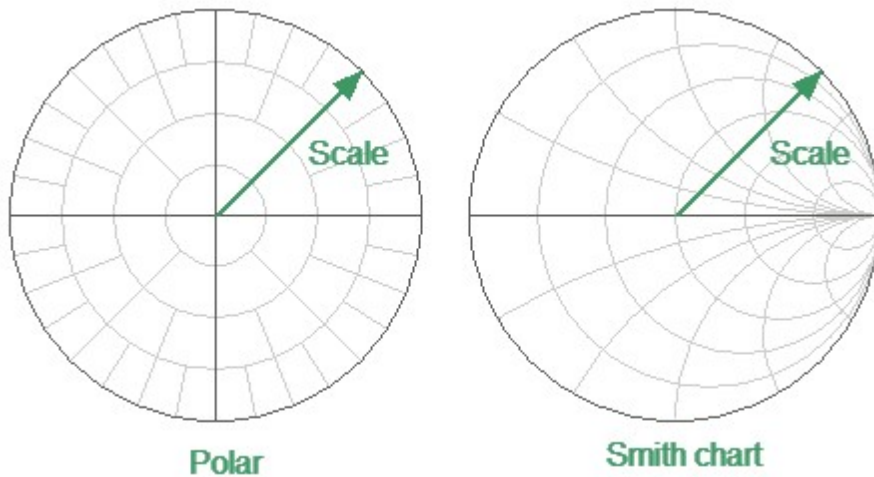
**NOTE**

The trace scale, value of the reference level, and reference level position can be set using the mouse (See [Quick Setting Using a Mouse](#)).

---

## Circular Scale

For [Polar formats](#) and [Smith chart formats](#), the outer circle value can be set (See figure below).

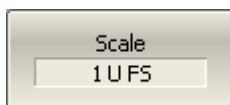


Circular Scale

The scale of each trace can be set independently. The trace to which the function is applied must be preselected as active (See [Selection of Active Trace/Channel](#) ).



To set the scale of the circular graphs, use the following softkeys:



**Scale > Scale**

[DISP:WIND:TRAC:Y:PDIV](#)

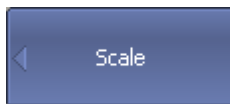
Sets or reads out the trace scale. Sets the full scale value.

## Automatic Scaling

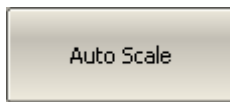
The automatic scaling function automatically adjusts the trace scale so that the trace of the measured value fits into the diagram entirely.

In rectangular format, two parameters are adjustable: scale division and reference level position. In circular format, the outer circle value is adjusted.

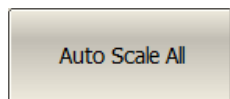
The function can be applied to the active trace or to all traces of the active channel.



To automatically select the scale of the active trace, use the following softkeys:



**Scale > Auto Scale**



To automatically select the reference level of all traces of the active channel, use the following softkeys:

**Scale > Auto Scale All**

---

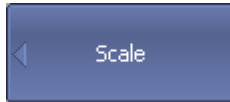
[DISP:WIND:TRAC:Y:AUTO](#)

Executes the auto scale function for the trace.

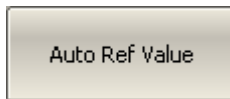
---

## Reference Level Automatic Selection

This function automatically selects the reference level in rectangular coordinates. After selection, the trace of the measured value shifts vertically so that the reference level crosses the trace in the middle. The scale division is unaffected. The function can be applied to the active trace or to all traces of the active channel.



To automatically select the reference level of the active trace, use the following softkeys:

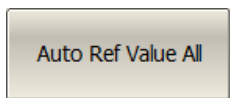


**Scale > Auto Ref Value**

---

[DISP:WIND:TRAC:Y:RLEV:AUTO](#)

Executes the auto reference function for the trace.



To automatically select the reference level of all traces of the active channel, use the following softkeys:

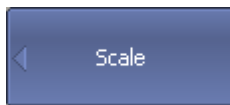
**Scale > Auto Ref Value All**

---

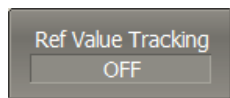
## Automatic Reference Level Tracking

The automatic reference level tracking function tracks the reference level of a trace. When enabled, the trace reference level is updated with each scan according to the selected method: maximum, minimum, center, or according to the value of the active marker. The scale per division value does not change. The function is only applicable to the rectangular format.

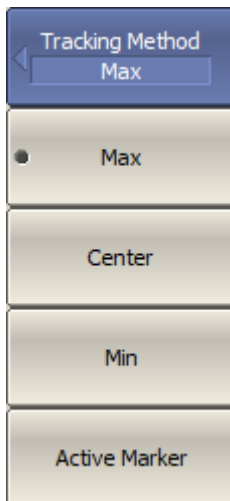
The trace to which the function is applied must be preselected as active (See [Selection of Active Trace/Channel](#)).



To enable / disable reference level tracking, use the following softkeys:



**Scale > Ref Value Tracking [ON | OFF]**



To select a reference level tracking method, use the following softkeys:

**Scale > Tracking Method**

Select method:

- **Max** — maximum trace value.
  - **Center** — average trace value.
  - **Min** — minimum trace value.
  - **Activ Marker** — active marker value.
-

## Electrical Delay Setting

The electrical delay function compensates for the electrical delay of the trace measurement. This function is useful during measurements of phase deviations from linear, for example.

If the electrical delay setting is other than zero, the S-parameter value will be corrected in accordance with the following formula:

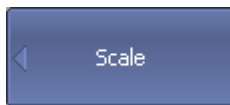
$$S = S_{meas} \cdot e^{j \cdot 2\pi \cdot f \cdot t},$$

where  $f$  — frequency, Hz,

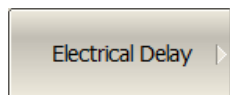
$t$  — electrical delay, sec.

The electrical delay can be specified in seconds or as an equivalent length (meters, feet, inches). When the equivalent length is used, it is also possible to select media (coax or waveguide) and to set the velocity factor and WG Cutoff (for waveguide only).

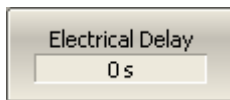
The electrical delay is set for each trace independently. The trace to which the function is applied must be preselected as active (See [Selection of Active Trace/Channel](#) ).



To set the electrical delay in seconds, use the following softkeys:



**Scale > Electrical Delay > Electrical Delay**

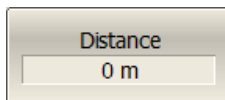


---

[CALC:CORR:EDEL:TIME](#)

Sets or reads out the value of the electrical delay.

---



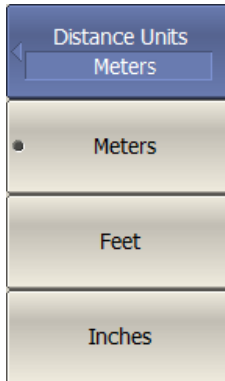
To set the electrical delay to an equivalent length, use the following softkeys:

**Scale > Electrical Delay > Distance**

---

[CALC:CORR:EDEL:DIST](#)

Sets or reads out the value of the equivalent distance in the electrical delay function.



To set the units of equivalent length, use the following softkeys:

**Scale > Electrical Delay > Distance Units**

Select unit:

- **Meters**
- **Feet**
- **Inches**

---

[CALC:CORR:EDEL:DIST:UNIT](#)

Sets or reads out the distance units in the electrical delay function.



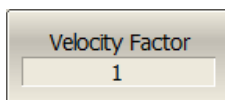
To set the transmission line type, use the following softkeys:

**Scale > Electrical Delay > Media [Coax | Waveguide]**

---

[CALC:CORR:EDEL:MED](#)

Sets or reads out the type of media in the electrical delay function.



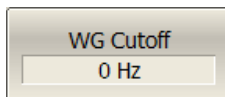
To enter the velocity factor value, use the following softkeys:

**Scale > Electrical Delay > Velocity Factor**

---

[CALC:CORR:EDEL:RVEL](#)

Sets or reads out the value of the velocity factor used to calculate between delay and distance in the electrical delay function.



To enter the WG Cutoff value, use the following softkeys:

**Scale > Electrical Delay > WG Cutoff**

---

[CALC:CORR:EDEL:WAV:CUT](#)

Sets or reads out the value of the waveguide cutoff frequency in the electrical delay function if the type of media set to the

---

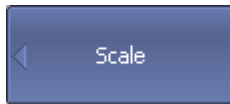
**WAVeguide** by the command  
[CALC:CORR:EDEL:MED.](#)

---



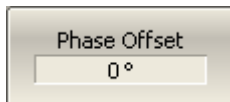
## Phase Offset Setting

The phase offset function adds the constant offset to the phase of a trace. The value of the phase offset is set in degrees for each trace independently. The trace must be activated before setting the phase offset.



To set the phase offset, use the following softkeys:

**Scale > Phase Offset**



---

[CALC:CORR:OFFS:PHAS](#)

Sets or reads out the value of the phase offset.

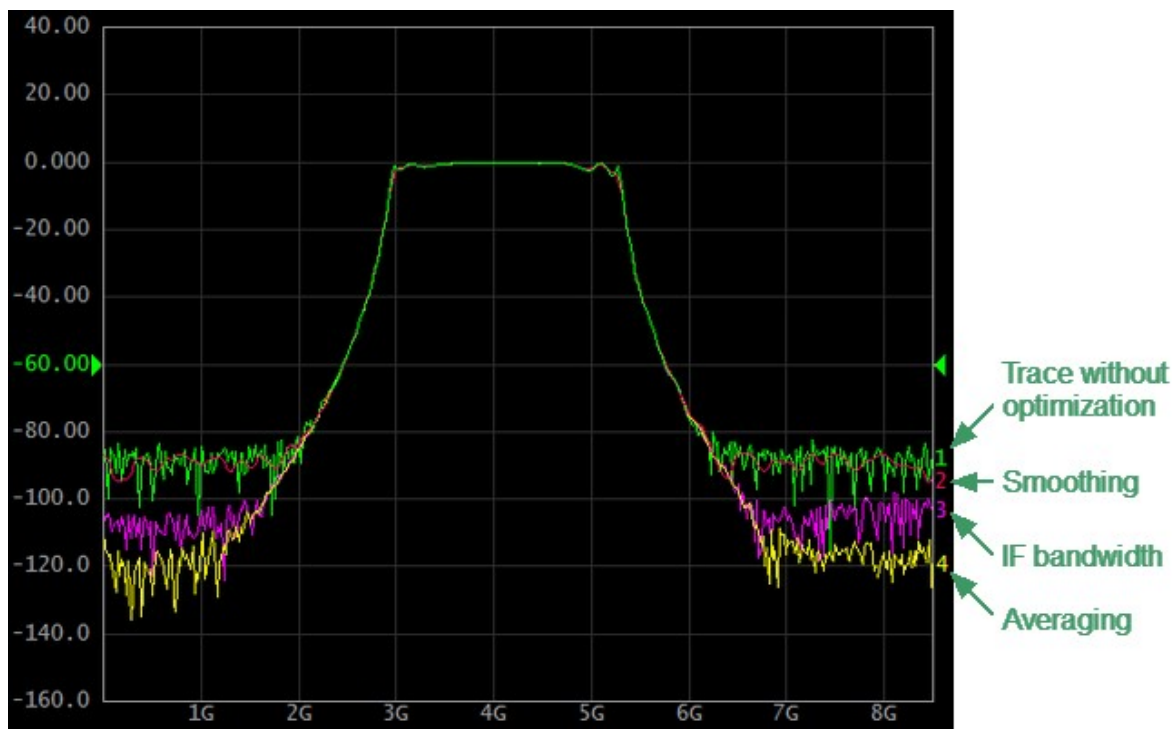
---

## Measurement Optimization

This section describes ways to optimize the measurement:

- Narrowing the IF bandwidth of measurement receivers increases the signal-to-noise ratio and extends the dynamic range of measurements. This increases the value of the sweep time. For a detailed description see [IF bandwidth](#).
- Averaging allows to increase the signal-to-noise ratio and extend the dynamic range of the measurements. Averaging does not increase the value of the sweep time, but the averaging result is complete after N sweeps, where N is an averaging factor. For a detailed description see [Averaging](#).
- Smoothing does not change the dynamic range of the measurements, but reduces the noise emissions of the signal. For a detailed description see [Smoothing](#).

The figure below shows an example of applying different filtering methods to the signal: the IF bandwidth is reduced by a factor of 10, averaging factor is set to 100, and smoothing is applied with an aperture of 2%.



Example of the application of different measurement optimization

## IF Bandwidth Setting

The IF bandwidth setting selects the bandwidth of the receivers. The IF bandwidth value takes value from the following series: 1 Hz, 1.5 Hz, 2 Hz, 3 Hz, 5 Hz, 7 Hz, 10 Hz, 15 Hz, 20 Hz ... 1 MHz, 2 MHz. The maximum IF bandwidth value depends on Analyzer model (See corresponding [datasheet](#)).

Narrowing the IF bandwidth increases the signal-to-noise ratio and extends the dynamic range of measurements. Narrowing the IF bandwidth by 10 will nominally extend the dynamic range by 10 dB. Narrowing the IF bandwidth increases the measurement time.

The IF bandwidth is set for each channel independently. The channel to which the function is applied must be preselected as active (See [Selection of Active Trace/Channel](#)).



To set the IF bandwidth, use the following softkeys:

**Average > IF Bandwidth**



---

[SENS:BAND](#)  
([SENS:BWID](#))

Sets or reads out the IF bandwidth.

---

### NOTE

IF bandwidth can be set using the mouse (See [IF Bandwidth Setting](#)).

---

## Averaging Setting

Averaging of each measurement point is performed over several sweeps. The benefits of the averaging function are similar to those of IF bandwidth narrowing. It increases the signal-to-noise ratio and extends the dynamic range of measurements.

Averaging of each measurement point is made across multiple sweeps in accordance with the following formula:

$$\begin{cases} M_i = S_i, & i = 0 \\ M_i = \left(1 - \frac{1}{n}\right) \cdot M_{i-1} + \frac{S_i}{n}, & i > 0, n = \min(i + 1, N) \end{cases}$$

where  $M_i$  —  $i$ -th sweep averaging result,

$S_i$  —  $i$ -th sweep measurement parameter (S-parameter) value,

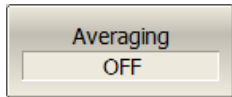
$N$  — averaging factor from 1 to 999; the higher the factor value, the stronger the averaging effect.

When the averaging function is enabled, the current number of iterations and the averaging factor, e.g. «9/10», will appear in the channel status bar. The averaging process is considered stable when the two numbers are equal.

The averaging should be set for each channel individually. The channel to which the function is applied must be preselected as active (See [Selection of Active Trace/Channel](#) ).



To toggle the averaging function on/off, use the following softkeys:



**Average > Averaging [ON | OFF]**

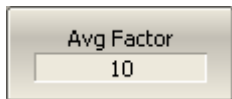
---

[SENS:AVER](#)

Turns the measurement averaging function ON/OFF.



To set the averaging factor, use the following softkeys:



**Averaging > Avg Factor**

---

[SENS:AVER:COUN](#)

Sets or reads out the averaging factor when the averaging function is turned on.

---

## Averaging Trigger

The averaging trigger function allows for completing the averaging with a single trigger signal. This function affects the channels in which the averaging function is enabled (See [Averaging Setting](#)).

Averaging Trigger	Function
<b>OFF</b>  [default]	One sweep is performed in response to one trigger signal regardless of the state of the channel averaging function. If the channel averaging is turned on, N trigger signals are required to complete the averaging process (where N is the averaging factor). The trigger signal does not reset the result of the previous averaging.
<b>ON</b>	N sweeps are performed in response to one trigger signal for the channel with the averaging on (where N is the averaging factor). One trigger signal is required to complete the averaging process in the channel. The trigger signal starts a new averaging cycle in the channel.

The averaging trigger function is convenient in conjunction with an external, software (BUS), or manual trigger source. When the function is enabled, the averaging result can be obtained on one trigger signal by performing a number of sweeps equal to the averaging factor (See [Averaging Setting](#)). When the internal trigger source is used it is recommended to turn OFF this function.

---

### NOTE

If the trigger event function is set to [On Point](#), then it takes precedence over the averaging trigger function. In this case, to complete averaging, the number of trigger pulses equal to the number of points multiplied by the averaging factor is required.

---

---

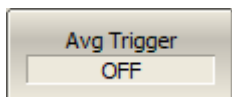
**NOTE**

If multiple channels are open at the same time, one trigger signal starts a measurement cycle the required number of times for the channels with averaging on, and once for channels with averaging off.

---



To enable/disable the averaging trigger function, use the following softkeys:

**Average > Avg Trigger**

The function changes between the values:

- **ON**
- **OFF**

---

**[TRIG: AVER](#)**

Turns the averaging trigger function ON/OFF.

---

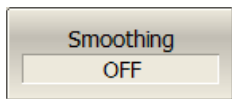
## Smoothing Setting

Smoothing averages the adjacent points of the trace by the moving window. The window aperture is set as a percent of the total number of trace points.

Smoothing does not increase the dynamic range of the Analyzer, nor does it increase measurement time. Smoothing helps to reduce noise bursts. Smoothing is set for each trace independently. The trace to which the function is applied must be preselected as active (See [Selection of Active Trace/Channel](#)).



To toggle the smoothing function ON/OFF, use the following softkeys:

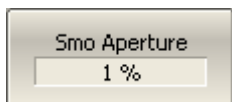


**Averaging > Smoothing [ON | OFF]**

---

[CALC:SMO](#)

Turns the trace smoothing ON/OFF.



To set the smoothing aperture, use the following softkeys:

**Averaging > Smo Aperture**

---

[CALC:SMO:APER](#)

Sets or reads out the smoothing aperture when performing the smoothing function.

---

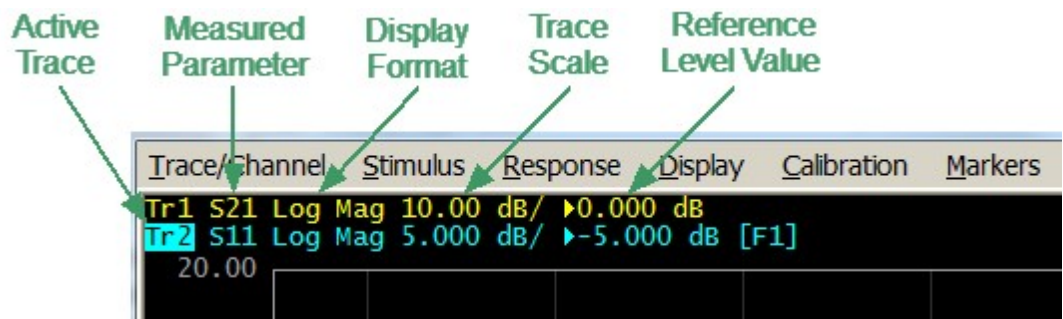


# Quick Settings Using a Mouse

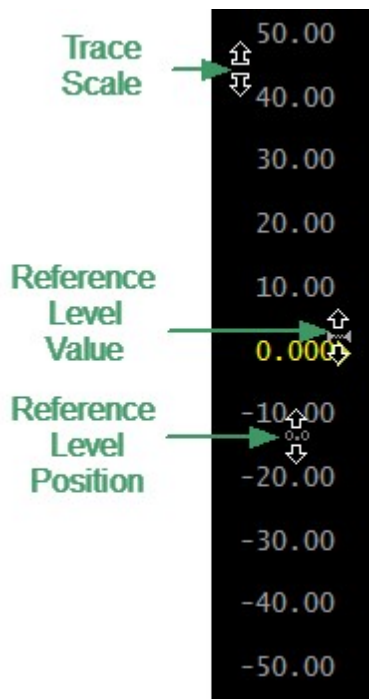
This section describes mouse operations, which allows to set the channel parameters quickly and easily. In a channel window, when hovering over the field where a channel parameter can be modified, the mouse pointer will change its icon to indicate edit mode. In text and numerical fields, edit mode will be indicated by underlined symbols.

**NOTE** The mouse operations described in this section help to adjust the most frequently used settings. The complete set of channel functions can be accessed via the softkey menu.

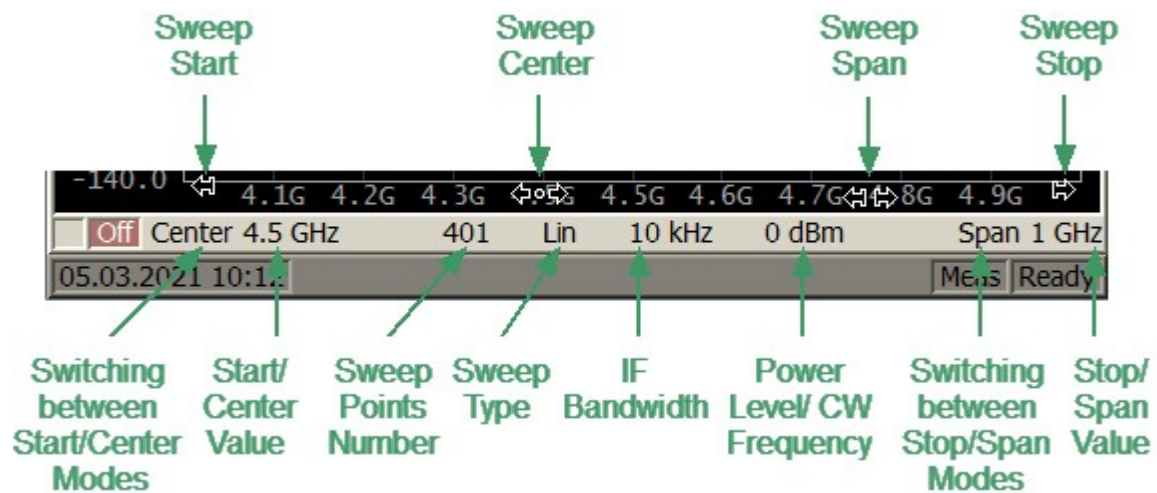
The figures below show areas and labels for quick parameter setting.



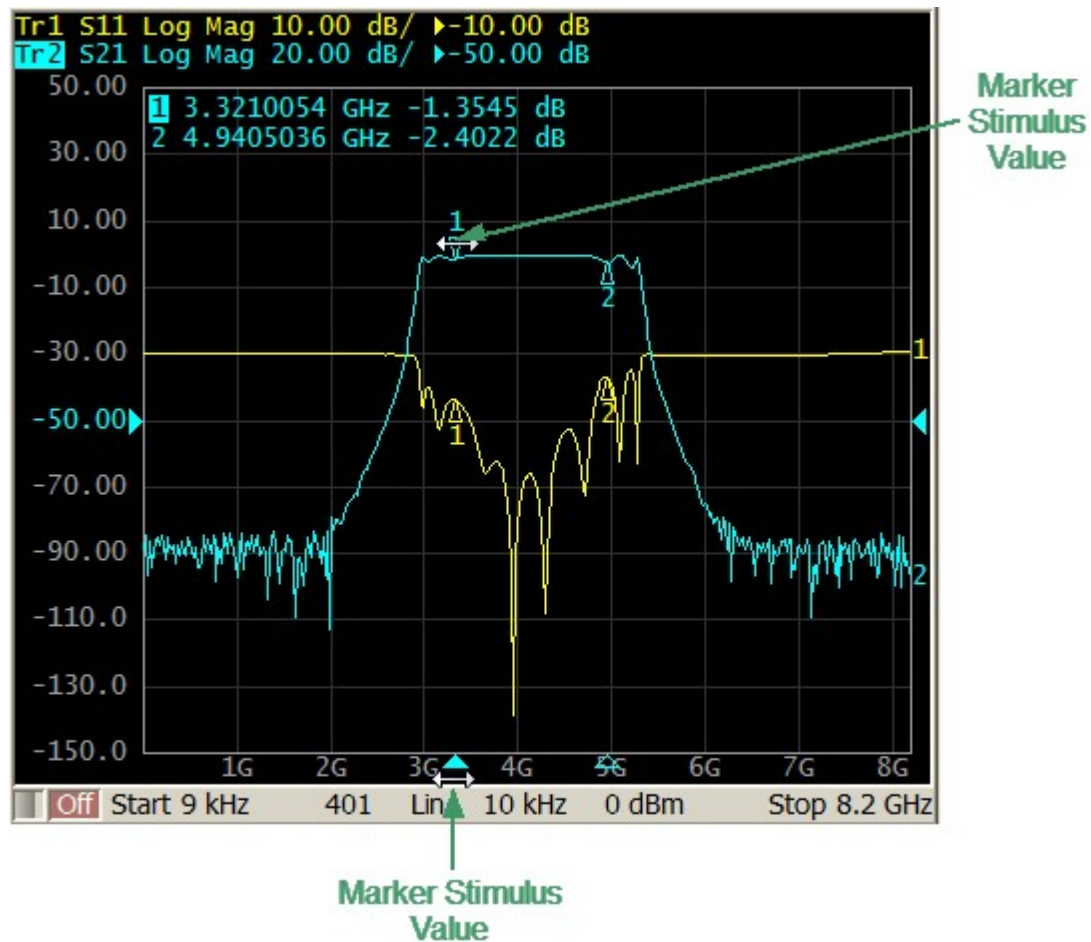
Quick Parameter Setting in the Trace Status Field



Quick Parameter Setting on the Vertical Graticule Label



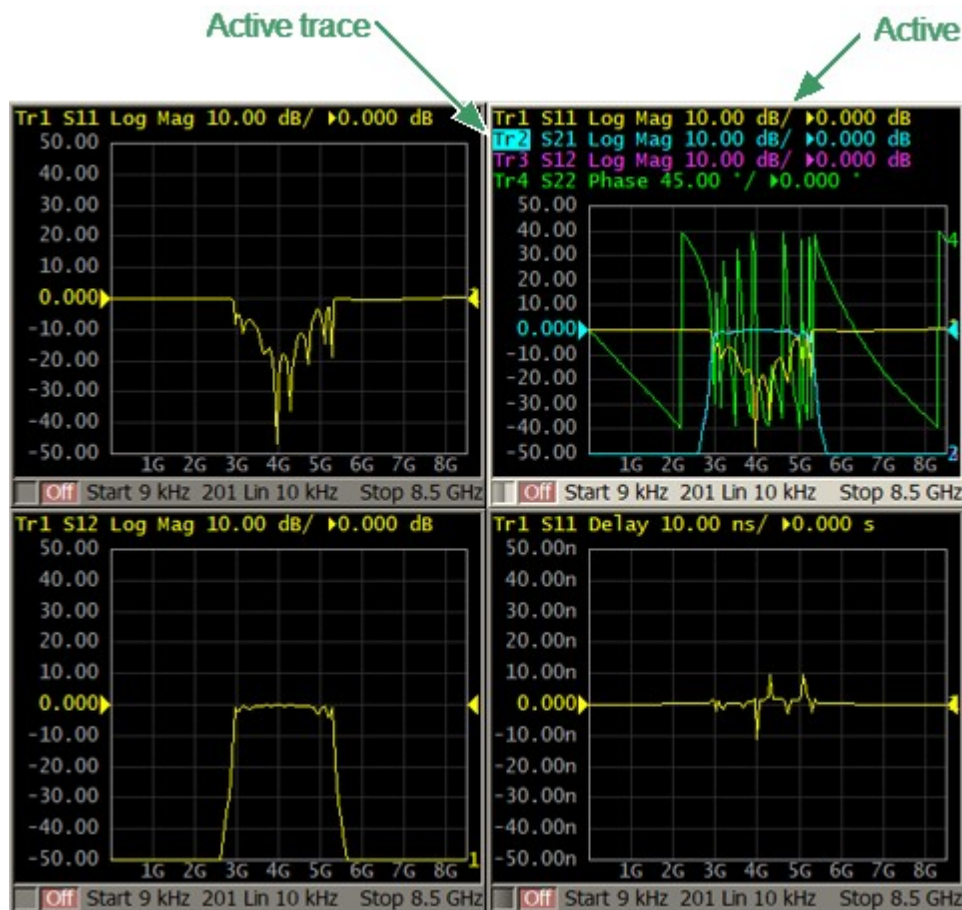
Quick Parameter Setting on the Channel Status Bar



Quick Parameter Setting on Markers

## Active Channel Selection

The active channel can be selected when two or more channel windows are open. The border line of the active window will be highlighted in a light color. To activate another window, click inside its area.

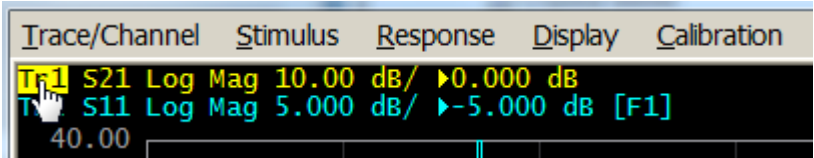


Active Trace/Channel Selection

The active channel can be selected using softkeys (See [Selection of Active Trace/Channel](#)).

# Active Trace Selection

The active trace can be selected if the active channel window contains two or more traces. The active trace name is highlighted. To activate a trace, click on the required trace status line, or on any item (trace, marker) having the same color.

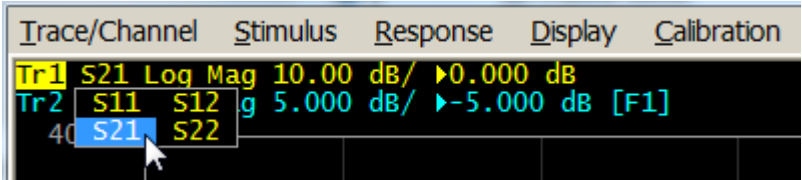


Active Trace Selection

Active trace can be selected using softkeys (See [Selection of Active Trace/Channel](#)).

# Measured Parameter Setting

To assign the measured parameters (S11, S21, S12 or S22) to a trace, click on the S-parameter name in the trace status line and select the required parameter in the drop-down menu.



Measured Parameter Setting

Measured data can be set using softkeys (See [S-Parameters](#)).

# Display Format Setting

To select the trace display format, click on the display format field in the trace status line and select the desired format in the drop-down menu.



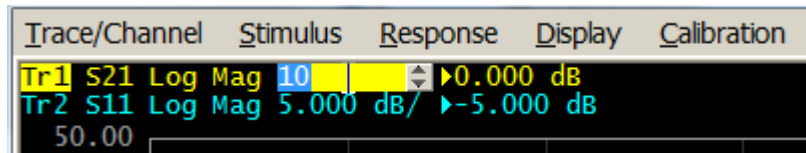
Display Format Setting

The display format can be set using softkeys (See [Format Setting](#)).

## Trace Scale Setting

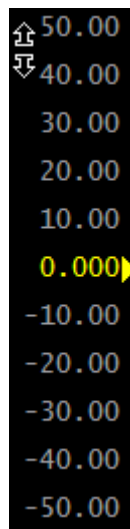
The trace scale, also known as the vertical scale division value, can be set by either of two methods.

The first method: click on the trace scale field in the trace status line and enter the required numerical value.



Trace scale setting in the trace status line

The second method: move the mouse pointer over the vertical scale until the pointer icon becomes as shown in the figure. The pointer should be placed in the top or bottom parts of the scale, at approximately 10% of the scale height from the top or bottom of the scale. Left click and drag away from the scale center to enlarge the scale, or toward the scale center to reduce the scale.



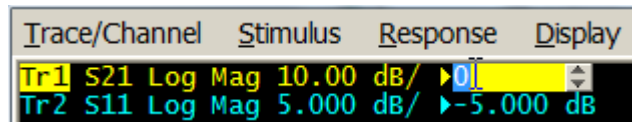
Trace scale setting on the vertical scale

The trace scale can be set using softkeys (See [Rectangular Scale](#)).

## Reference Level Setting

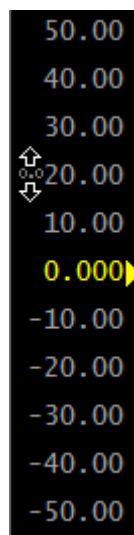
The value of the reference level, which is indicated on the vertical scale by the «►» and «◄» symbols, can be set by either of two methods.

The first method: click on the reference level field in the trace status line and enter the required numerical value.



Reference level setting in the trace status line

The second method: move the mouse pointer over the vertical scale until the pointer icon becomes as shown in the figure. The pointer should be placed in the center part of the scale. Left click and drag up to increase the reference level value, or down to reduce the value.



Reference level setting on the vertical scale

The value of the reference level can be set using softkeys (See [Rectangular Scale](#)).



## Reference Level Position

The reference level position, indicated on the vertical scale by «►» and «◄» symbols, can be set in the following way: Locate the mouse pointer on a reference level symbol until it becomes as shown in the figure, then drag and drop the reference level symbol to the desired position.

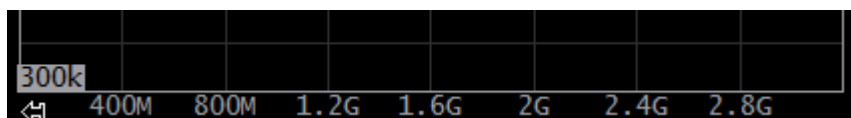


Setting Ref Level Pos on the vertical scale

The reference level position can be set using softkeys (See [Rectangular Scale](#)).

## Sweep Start Setting

Move the mouse pointer over the stimulus scale until it becomes as shown in the figure. The pointer should be placed in the left part of the scale, at approximately 10% of the scale length from the left. Left click and drag right to increase the sweep start value or left to reduce the value.

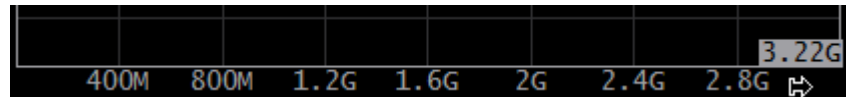


Sweep start setting on the Stimulus Scale

The start value of the sweep range can be set using softkeys (See [Sweep Range](#)).

## Sweep Stop Setting

Move the mouse pointer over the stimulus scale until it becomes as shown in the figure. The pointer should be placed in the right part of the scale, at approximately 10% of the scale length from the right. Left click and drag right to increase the sweep stop value or left to reduce the value.

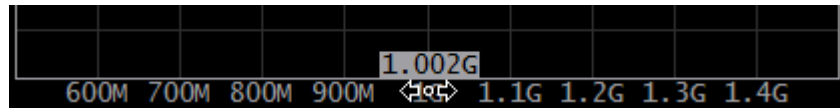


Sweep stop setting on the Stimulus scale

The stop value of the sweep range can be set using softkeys (See [Sweep Range](#)).

## Sweep Center Setting

Move the mouse pointer over the stimulus scale until it becomes as shown in the figure. The pointer should be placed in the center part of the scale. Left click and drag right to increase the sweep center value or left to reduce the value.

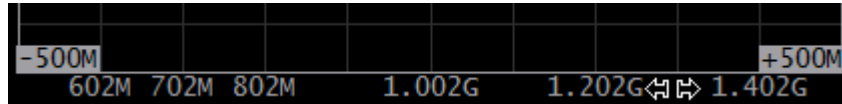


Sweep center setting on the stimulus scale

The center value of the sweep range can be set using softkeys (See [Sweep Range](#)).

## Sweep Span Setting

Move the mouse pointer over the stimulus scale until it becomes as shown in the figure. The pointer should be placed in the center part of the scale, at approximately 20% of the scale length from the right. Left click and drag to the right to increase the sweep span value, or to the left to reduce the value.

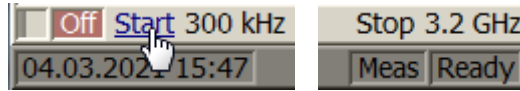


Sweep span setting on the stimulus scale

The span value of the sweep range can be set using softkeys (See [Sweep Range](#)).

## Switching Between Start/Center and Stop/Span Modes

To switch between the modes, Start/Center and Stop/Span, click on the respective field of the channel status bar. Clicking the label «Start» changes it to «Center», and the label «Stop» will change to «Span».

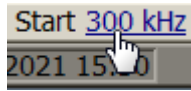


Switching between Start/Center and Stop/Span modes in channel status bar

The layout of the stimulus scale will be changed correspondingly. Switching between modes is possible using softkeys (See [Sweep Range](#)).

## Start/Center Value Setting

To enter the Start/Center values, activate the respective field in the channel status bar by clicking the numerical value.

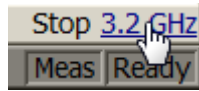


Setting the Start/Center value in the channel status bar

The Start/Center values can be set using softkeys (See [Sweep Range](#)).

## Stop/Span Value Setting

To enter the Stop/Span values, activate the respective field in the channel status bar by clicking the numerical value.



Setting the Stop/Span value in the channel status bar

The Stop/Span values can be set using softkeys (See [Sweep Range](#)).



## Number of Points Setting

To enter the number of points, activate the respective field in the channel status bar by clicking the numerical value.

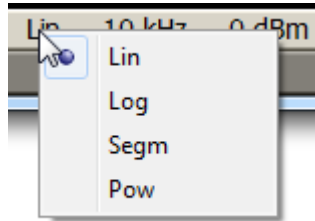


Setting the number of points value in the channel status bar

The number of points can be set using softkeys (See [Number of Points](#)).

## Sweep Type Setting

To set the sweep type, left click on the respective field in the channel status bar and select the required type in the drop-down menu.



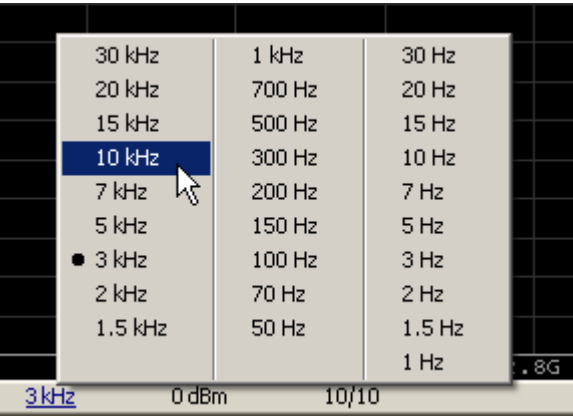
Setting the sweep type value in the channel status bar

The sweep type can be selected using softkeys (See [Sweep Type](#)).

# IF Bandwidth Setting

IF bandwidth can be set by selection in the drop-down menu, or by entering the value using numerical keys of the keyboard.

To activate the drop-down menu, right click on the IF bandwidth field in the channel status bar.



IF Bandwidth Setting in drop-down menu

To enter the IF bandwidth, activate the respective field in the channel status bar by left clicking.

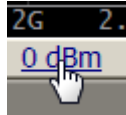


Setting the IF Bandwidth in the channel status bar

IF bandwidth can be set using softkeys (See [IF Bandwidth Setting](#)).

## Power Level/CW Frequency Setting

To enter the Power Level/CW Frequency, activate the respective field in the channel status bar by clicking the numerical value. The parameter displayed in the field depends on the current sweep type: in frequency sweep mode, the power level value can be entered; in power sweep mode, the CW frequency value can be entered.



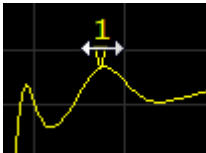
Setting the Power Level/CW Frequency in the channel status bar

The Power Level and CW Frequency can be set using softkeys (See [Stimulus Power](#) and [CW Frequency](#)).

# Marker Stimulus Value Setting

The marker stimulus value can be set by either a click and drag operation, or by entering the value using numerical keys of the keyboard.

To drag the marker, first move the mouse pointer on one of the marker indicators until it becomes as shown in the figures below:



Setting the marker value using drag and drop

To enter the numerical value of the stimulus, activate its field by clicking it in the marker data line.

Trace/Channel	Stimulus	Response	Display
Tr1 S21 Log Mag	1.500 dB/	0.000 dB	
7.500	1 3.148G	-0.4953 dB	
6.000			

Setting the marker value in the marker data line

The marker stimulus value can be set using softkeys (See [Marker Stimulus Value Setting](#)).

## Calibration and Calibration Kit

Measurement accuracy is affected by errors introduced by the Analyzer and measurement setup. The nature of these errors is varied — some are systematically repeated and some are random. Calibration is a process used to evaluate systematically repeated errors and mathematically exclude them from the measurement results in the correction process.

---

### NOTE

Be sure to properly calibrate if accurate measurements are required. Only a properly calibrated instrument provides the accuracy specified in the data sheet.

---

The section describes information about calibration, calibration kits and automatic calibration module (ACM):

- General information about calibration (See [General Information](#)).
- Working with calibration standards and calibration kits (See [Calibration Standards and Calibration Kits](#)).
- Calibration method and procedures (See [Calibration Methods and Procedures](#)).
- Power calibration with an external power meter, to maintain an accurate power level at the DUT input (See [Power Calibration](#)).
- Calibration of receivers for accurate power measurements (See [Receiver Calibration](#)).
- Mixer Calibration:
  1. [Scalar Mixer Calibration](#), requires no additional mixer. The mixer measurement is performed in frequency offset mode. This method allows for measurement of reflection parameters in vector form and transmission parameters in scalar form.
  2. [Vector Mixer Calibration](#) is performed using an additional mixer. This method allows for measurement of reflection and transmission parameters in vector form, including phase and group delay time of the transmission coefficient.
- Working with the automatic calibration module (ACM), which allows for simplification and speeding up of the analyzer calibration process (See [Automatic Calibration Module](#)).

## General Information

This section details general information about calibration:

- Guidelines for calibration (See [Basic Calibration Guidelines](#)).
- Description of measurement errors (See [Measurement Errors](#)).
- Error models (See [Error Model](#)).
- Calibration steps (See [Calibration Steps](#)).

## Basic Calibration Guidelines

Follow the guidelines below to perform calibration correctly and reduce accidental errors. Observance of the guidelines will ensure the specified accuracy of the device.

### General Guidelines

- Select all fixtures for connecting the DUT and assemble the measuring setup before starting the calibration. Perform calibration in the plane passing through the connectors to which the DUT is connected.
- Calibrate the measuring setup at the same stimulus parameters (frequency range, number of points, stimulus power) at which measurements will be performed. Changing these parameters after calibration may significantly reduce the accuracy of the measurements.
- During calibration, do not set the IF bandwidth wider than planned for measurements.
- Choose a calibration kit according to the type and gender of the DUT connectors.
- The frequency range of the selected calibration kit must correspond to the range in which the calibration is performed.
- When choosing a calibration kit, note that for SOLT calibrations the most accuracy will be provided by the calibration kit, in which the parameters of the standards are most accurately defined. For TRL calibrations, the accuracy of the calibration is mainly determined by the quality of the standards manufacturing.
- The calibration kit selected in the Analyzer software must strictly correspond to the one actually being used. The mismatch is unacceptable.
- For easy measurements, it is possible to create custom calibration kits from the available standards or specially manufactured calibration kits to solve specific measuring tasks. To include a standard in a calibration kit, calculate or measure its parameters using a high precision measuring tool. Create a description in the form of a model of standard or S-parameter table of standard, and download this description to the analyzer software.
- The choice of calibration method depends on the measurement being performed, its accuracy requirements, the permissible calibration labor intensity, and the availability of calibration kits.
- For the SOLT calibrations, it is recommended to use ACM (Automatic Calibration Module) to reduce:
  - the labor intensity of the calibration without loss of accuracy
  - wear of connectors
  - operator errors



- If an additional component (cable, attenuator, adapter) is added to the measurement setup after calibration, recalibrate. Instead of recalibration, it is possible to use the de-embedding function or the port extension function to compensate for the added electrical length (delay) and losses.

## **Recommendations for Reducing Random Measurement Errors**

- To reduce errors introduced by the instrument noise of the Analyzer, it is recommend to increase the source power of the stimulus signal, narrow the IF bandwidth, and apply averaging over several measurement sweep values.
- To reduce errors in the temperature drift of the electrical characteristics of the Analyzer and the components of the measuring setup, it is recommended:
  - To perform measurements in a room with a stable, controlled temperature, at which the technical characteristics of the analyzer are guaranteed.
  - To recalibrate if the room temperature has changed significantly after calibration.
  - To warm-up the analyzer for a time determined in the specification before starting the calibration.
  - To keep the calibration standards unpacked in the room where the measurements are taken to stabilize the parameters, before starting the calibration.
- To reduce the connector repeatability errors, it is recommended:
  - To apply proper connector care — connectors must be good and clean.
  - To use a special wrench with a standardized tightening torque, when connecting the DUT and calibration standards to measurement connectors.
  - To not change the position of the components of the measuring setup in space during or after calibration.
  - To recalibrate if setup components have been rearranged.

## Measurement Errors

S-parameter measurements are influenced by various measurement errors, which can be broken down into two categories:

- systematic errors
- random errors

Random errors comprise errors such as noise fluctuations and thermal drift in electronic components, changes in the mechanical dimensions of cables and connectors subject to temperature drift, repeatability of connections, and cable bends. Random errors are unpredictable and hence cannot be estimated and eliminated in calibration. Random errors can be reduced by having the stimulus power at the correct setting, IF bandwidth narrowing, sweep averaging, maintaining a constant environment temperature, observance of the Analyzer warm-up time, careful connector handling, and avoiding cable bending after calibration.

Random errors and related methods of correction are not mentioned further in this section.

Systematic errors are errors caused by imperfections in the components of the measurement system (see [Systematic Errors](#)). Such errors occur repeatedly, and their characteristics do not change with time. Systematic errors can be determined and then reduced by performing a mathematical correction of the measurement results.

**Calibration** is the process of measuring precision devices with predefined parameters to determine systematic errors, and such precision devices are called **calibration standards**. The most commonly used calibration standards are SHORT, OPEN, and LOAD.

The process of mathematical compensation of the systematic errors is called **error correction**.

## Systematic Errors

The systematic measurement errors of the Analyzer are divided into the following categories according to their source:

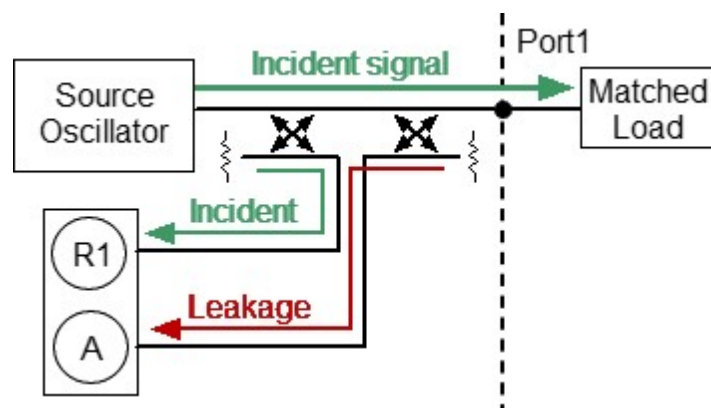
- directivity
- source match
- load match
- reflection tracking
- transmission tracking
- isolation

The measurement results before error correction are called **uncorrected**.

The residual values of the systematic measurement errors after error correction are called **effective**.

### Directivity Error

A directivity error (**Ed**) is caused by incomplete separation of the incident signal from the reflected signal by the directional coupler in the source port. In this case, part of the incident signal energy enters the receiver of the reflected signal. Directivity errors do not depend on the characteristics of the DUT, and usually have a greater effect on reflection measurements.

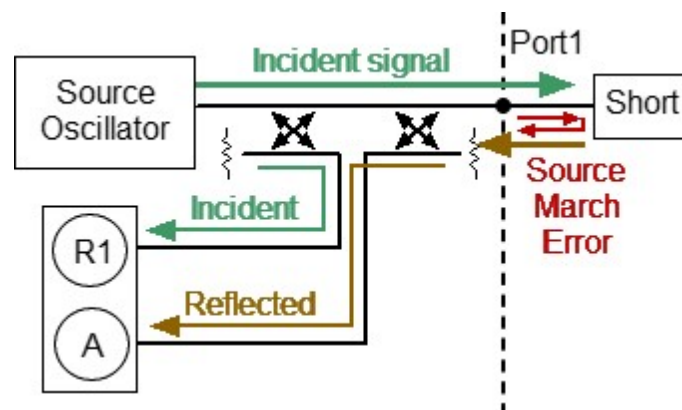


Directivity error

## Source Match Error

A source match error (**Es**) is caused by a mismatch between the source port and the input of the DUT. In this case, part of the signal reflected by the DUT reflects at the source port and re-enters the input of the DUT. The error affects both reflection measurement and transmission measurement. Source match errors depend on the difference between the input impedance of the DUT and test port impedance when it functions as a signal source.

Source match errors heavily affect measurements of a DUT with poor input matching.

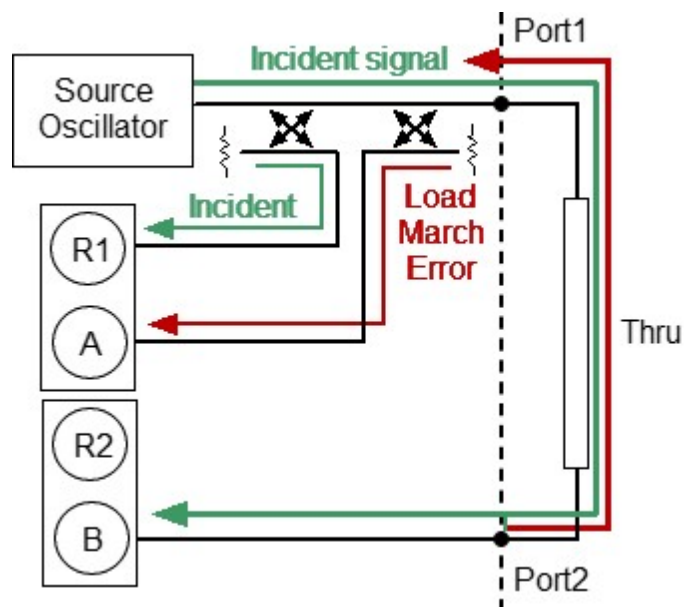


Source match error

## Load Match Error

A load match error (**EI**) is caused by a mismatch between the receiver port and the output of the DUT. In this case, part of the signal transmitted through the DUT reflects at the receiver port and returns to the output of the DUT. The error occurs during transmission measurements and reflection measurements (for a 2-port DUT). Load match errors depend on the difference between output impedance of the DUT and test port impedance when used as a signal receiver.

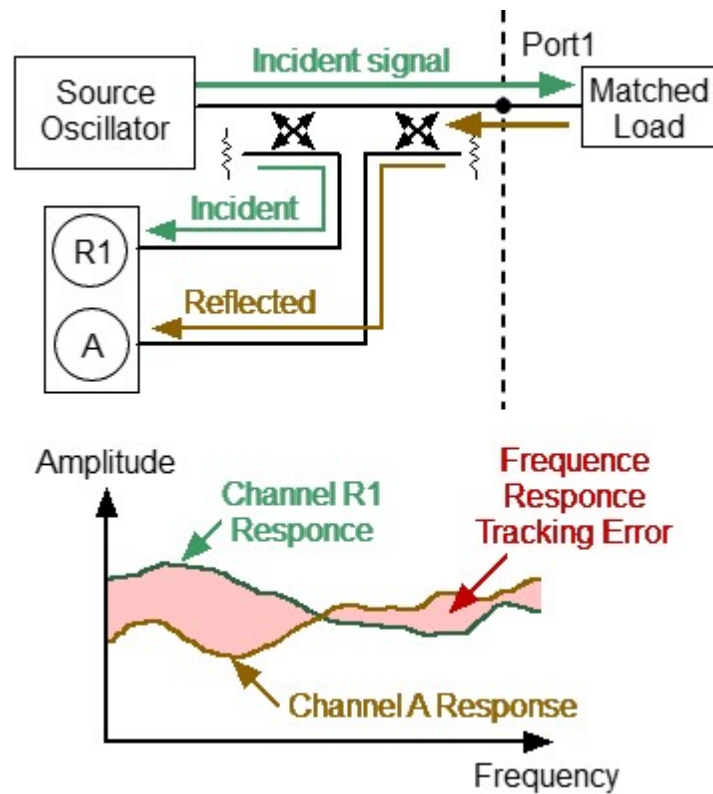
In transmission measurements, the load match error has considerable influence if the output of the DUT is poorly matched. In reflection measurements, the load match error has considerable influence in cases of poor output match and low attenuation between the output and input of the DUT.



Load match error

## Reflection Tracking Error

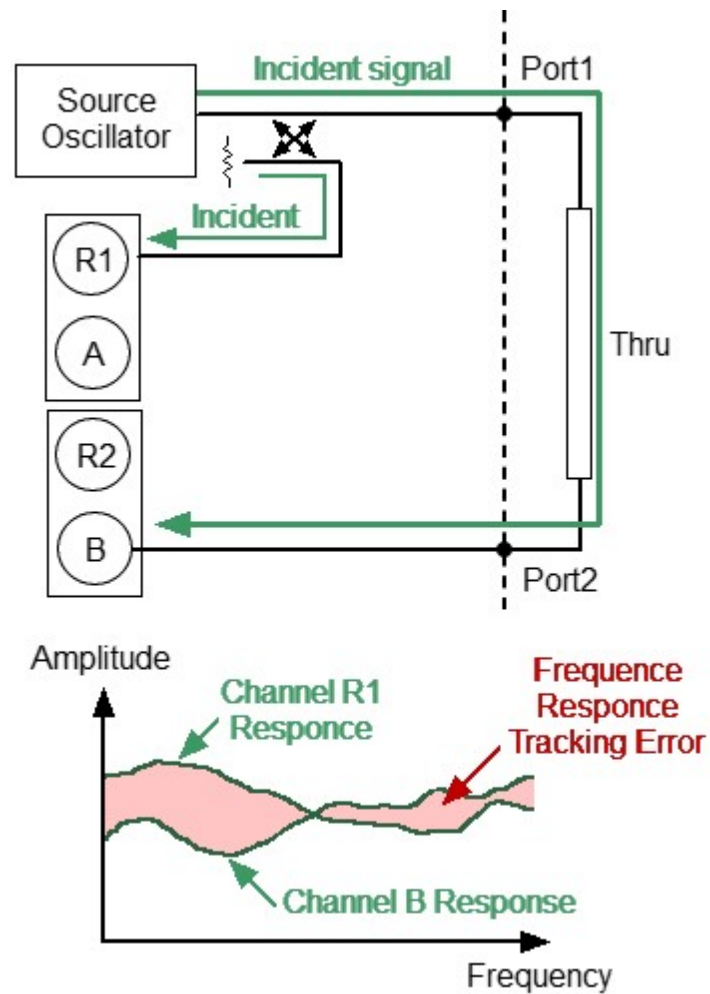
A reflection tracking error ( $E_r$ ) is caused by differences in frequency response between the test receiver and the reference receiver of the source port during reflection measurement.



Reflection tracking error

## Transmission Tracking Error

A transmission tracking error (**Et**) is caused by differences in frequency response between the test receiver of the receiver port and the reference receiver of the source port during transmission measurement.



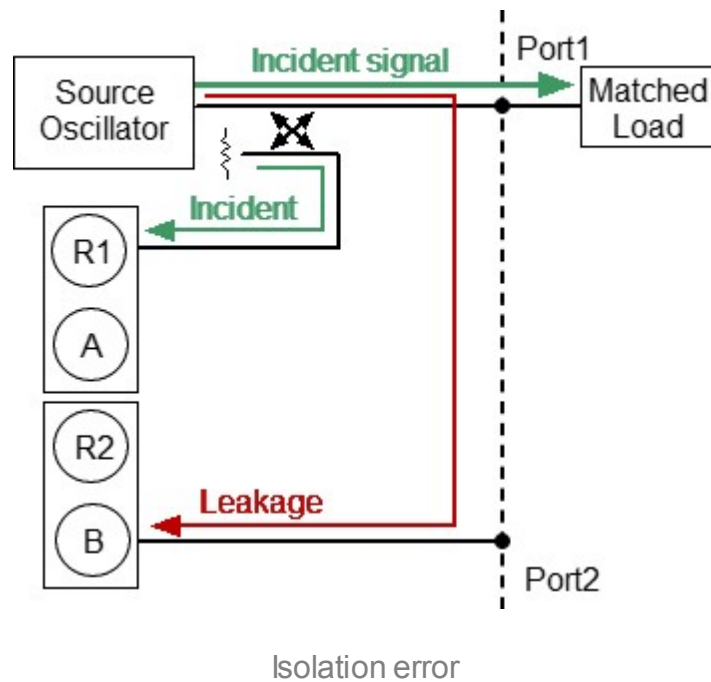
Transmission tracking error



## Isolation Error

Isolation error (**Ex**) is caused by a leakage of the signal from the source port to the receiver port bypassing the DUT.

The Analyzer has very good isolation, which allows us to ignore this error for most measurements. Isolation error measurement is an optional step in all types of calibration.



## Error Model

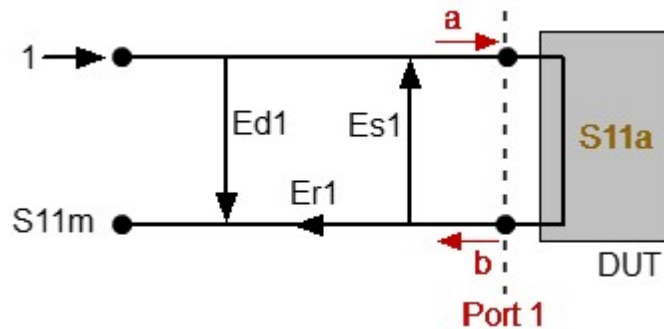
The error model in the form of signal (directed) graphs is used to analyze systematic errors of the Analyzer.

This section describes following error models:

- [One-Port Error Model](#)
- [Two-Port Error Model](#)

## One-Port Error Model

Only one port of the Analyzer is used when performing reflection measurements. The signal flow graph of errors for Port 1 is represented in the figure below. For Port 2, the signal flow graph of the errors will be similar.



a — incident wave, b — reflected wave

S11a — reflection coefficient actual value

S11m — reflection coefficient measured value

One-port error model

The measurement result at Port 1 is affected by the following three systematic error terms:

- **Ed1** — directivity.
- **Es1** — source match.
- **Er1** — reflection tracking.

For normalization, the stimulus value is taken equal to 1. All the values used in the model are complex.

After determining all the three error terms — **Ed1**, **Es1**, **Er1** — for each measurement frequency by means of a **full one-port calibration**, it is possible to calculate (mathematically eliminate the errors from the measured value S11m) the actual value of the reflection coefficient S11a.

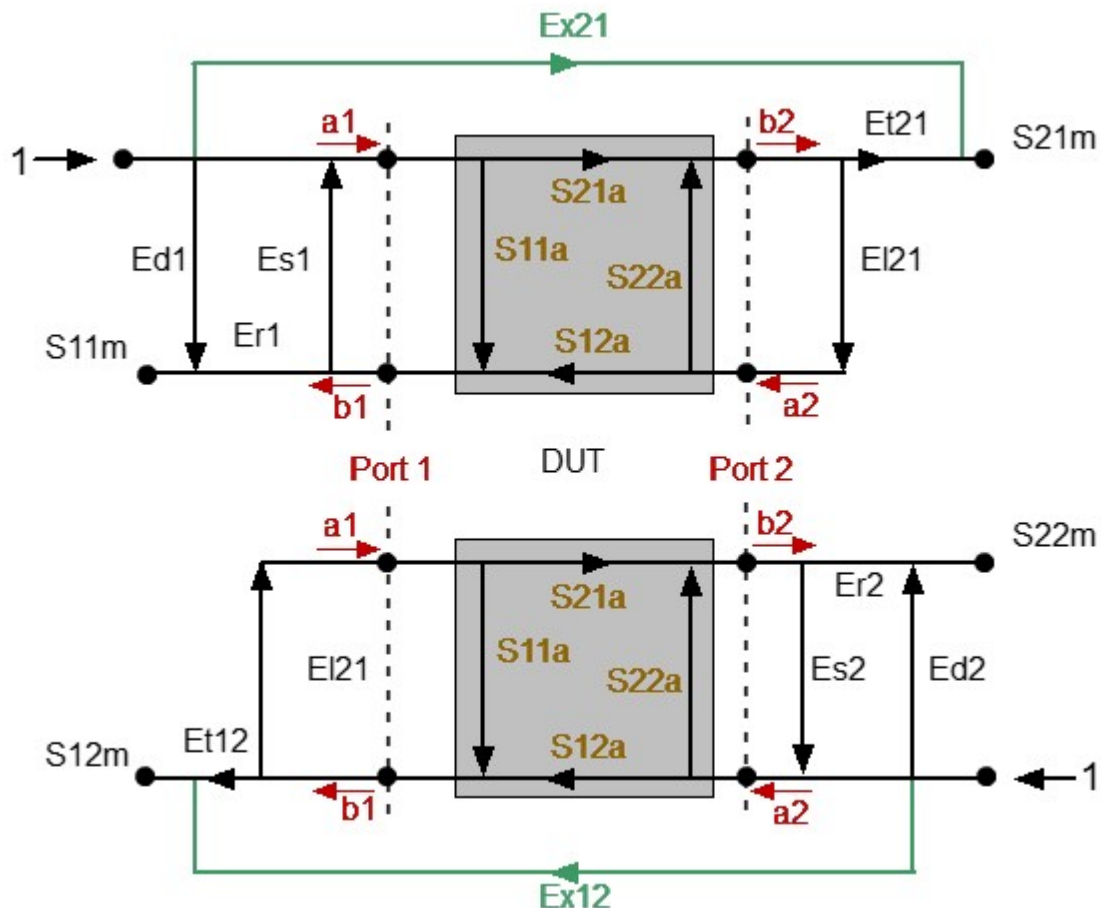
There are simplified methods, which eliminate the effects of only one or two of the three systematic errors.

For a detailed description of calibration methods, see [Calibration Methods and Procedures](#).

## Two-Port Error Model

There are two signal flow graphs considered for two-port measurements. One of the graphs describes the case where Port 1 is the stimulus source, the other graph describes the case where Port 2 is the stimulus source.

The signal flow graphs of error effects in a two-port system are represented in the figure below.



$a_1, a_2$  — incident waves,  $b_1, b_2$  — reflected waves

$S_{11a}, S_{21a}, S_{12a}, S_{22a}$  — actual value of DUT parameters

$S_{11m}, S_{21m}, S_{12m}, S_{22m}$  — measured DUT parameters values

Two-port error model

For normalization the stimulus value is taken equal to 1. All the values used in the model are complex. The measurement result in a two-port system is affected by twelve systematic error terms.

These terms are also described in the table below.

Description	Stimulus Source	
	Port 1	Port 2
Directivity	<b>Ed1</b>	<b>Ed2</b>
Source match	<b>Es1</b>	<b>Es2</b>
Reflection tracking	<b>Er1</b>	<b>Er2</b>
Transmission tracking	<b>Et1</b>	<b>Et2</b>
Load match	<b>El1</b>	<b>El2</b>
Isolation	<b>Ex1</b>	<b>Ex2</b>

After determining all twelve error terms for each measurement frequency by means of a **two-port calibration**, it is possible to calculate the actual value of the S-parameters: S11a, S21a, S12a, S22a.

There are simplified methods, which eliminate the effect of only one or several of the twelve systematic error terms.

---

#### NOTE

When using a two-port calibration, all four measurements S11m, S21m, S12m, S22m need to be known to determine any S-parameters. That is why updating one or all of the S-parameters necessitates two sweeps: first with Port 1 as a signal source, and then with Port 2 as a signal source.

---

For a detailed description of calibration methods, see [Calibration Methods and Procedures](#).

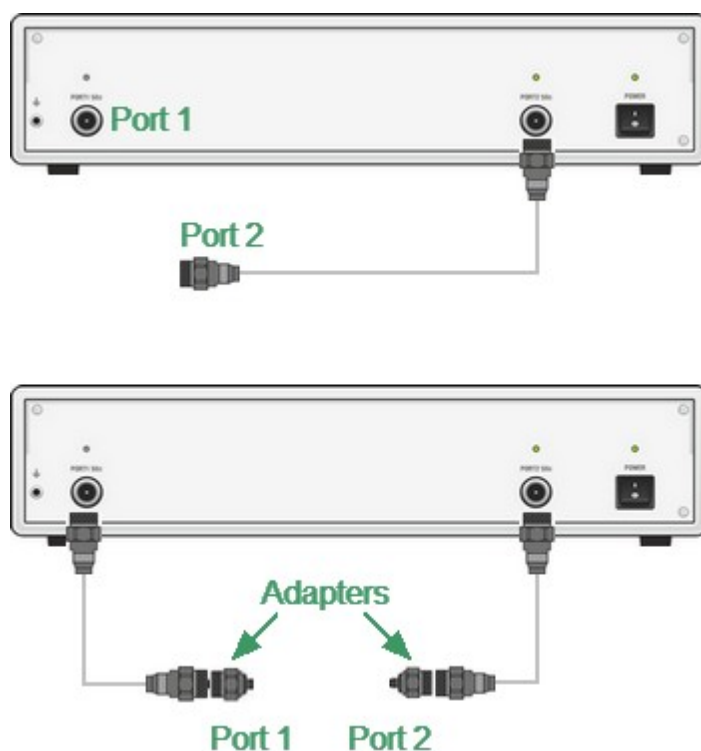
## Analyzer Test Port Definition

The test ports of the Analyzer are defined by means of calibration. The test port is a connector accepting a calibration standard in the process of calibration.

A type-N, 3.5 mm NMD, 2.4 mm NMD or 1.85 mm NMD connector on the front panel of the Analyzer will be the test port if calibration standards are connected directly to it.

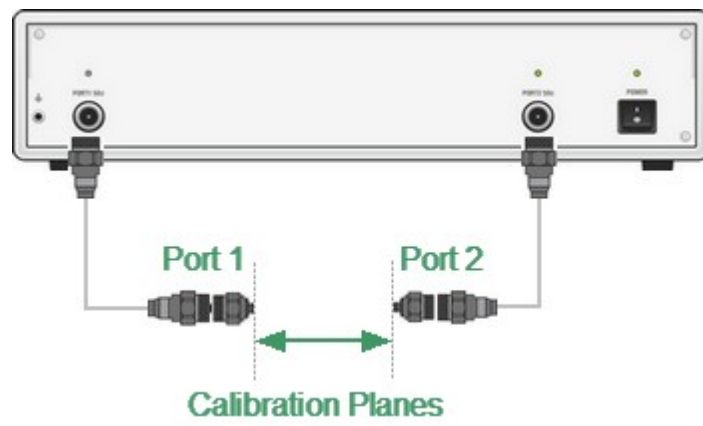
Sometimes it is necessary to connect coaxial cables and/or adapters to the connector(s) on the front panel to interface with a DUT of a different connector type. In such cases, calibration standards are connected to the connector of the cable or adapter.

The figure below represents two cases of test port definition for 2-port measurements. The use of cables and/or adapters does not affect the measurement results if they are integrated into the process of calibration.



Test port defining

The term calibration plane is used in some cases. The calibration plane is an imaginary plane located at the ends of the connectors, which accept calibration standards during calibration.



Calibration planes

## Calibration Steps

The process of calibration comprises the following steps:

- Selection of a calibration kit matching the connector type of the test port (See [Calibration Standards and Calibration Kits](#)). The calibration kit includes such standards as SHORT, OPEN, and LOAD with matched impedance. Magnitude and phase responses i.e. S-parameters of the standards are well known. The characteristics of the standards are represented in the form of an equivalent circuit model, as described in [Calibration Standards Model](#).
- Selection of a calibration method (See [Calibration Methods and Procedures](#)) is based on the required accuracy of measurements. The calibration method determines which error terms of the model (or all of them) will be compensated.
- Measurement of the standards within a specified frequency range. The number of measurements depends on the type of calibration.
- The Analyzer compares the measured parameters of the standards against their predefined values. The difference is used for calculation of the calibration coefficients (systematic errors).
- The table of calibration coefficients is saved into the memory of the Analyzer and used for error correction of the measured results of any DUT.

Calibration is always made for a specific channel, as it depends on the channel stimulus settings — particularly on the frequency span. This means that a table of calibration coefficients is being stored for each individual channel.



## Calibration Standards and Calibration Kits

### Calibration standard

Calibration standards are precision physical devices that serve as a calibration standard for the Analyzer.

Calibration standards have their own specific [type](#), specific [gender](#), specific impedance, standard definition. Calibration standard belongs to one or several classes.

Calibration standard definition is a mathematical description of its parameters (See [Calibration Standard Definition](#)). During calibration, the Analyzer measures standards and mathematically compares the results to the definitions of those standards. The comparison results are used to determine errors in the measurement system.

Calibration standard class is an application of the standard in a specific calibration method associated with a specific test port number. For example, "LOAD of Port 1" in full two-port calibration. For a detail of calibration standard classes see [Classes of Calibration Standards](#).

Calibration standards can be combined into a calibration kit.

### Calibration Kit

A calibration kit is a set of calibration standards with a specific connector type and specific impedance.

The Analyzer provides definitions of calibration kits produced by different manufacturers. The definitions of the calibration kits can be added and the predefined kits can be modified. Calibration kits editing procedure is described in [Calibration Kit Management](#).

## Types of Calibration Standards

Calibration standard type is a category of physical devices used to define the parameters of the standard. The Analyzer supports the following types of the calibration standards:

- OPEN
- SHORT
- FIXED LOAD
- SLIDING LOAD
- THRU/LINE
- UNKNOWN TRHU
- standard defined by data (S-parameters)

---

### NOTE

The type of a calibration standard should not be confused with its class. Calibration standard type is a part of the standard definition used for the calculation of its parameters.

---

## Gender of Calibration Standard

Gender of a calibration standard is typically denoted on the calibration standard label. The label and the gender of calibration standard respectively, are not accounted by the software and are used for information only. Nevertheless, it is recommended to follow some rules for calibration standard gender designation. A calibration standard can be labeled either with:

- The gender of a calibration standard itself, as **–M–** for male and **–F–** for female type of standard.
- The gender of the analyzer port, which the calibration standard is mated to, as (m) for male and (f) for female port types.

For example, same standard can be labeled as **Short –F–** or **Short (m)**.

The Analyzer software uses the first type of designation: the gender of a calibration standard itself denoted as **–M–** for male and **–F–** for female type of standards.



Female

Male

Gender of Calibration Standard

## Calibration Kit Management

This section describes how to edit the calibration kit description, and add and delete a calibration kit.

The Analyzer provides a table for 64 calibration kits. The first part of the table contains the predefined kits. The second part of the table is for calibration kit added by the user.

A calibration kit redefining can be required for the following purposes:

- To change the port assignment of a standard to ensure connector type (male, female) matching.
- To add a user-defined standard into the kit, e.g. a non-zero-length thru.
- To precise the standard parameters to improve the calibration accuracy.

A new user-defined calibration kit adding can be performed when a required kit is not included in the list of the predefined kits.

The deleting function is available for user-defined calibration kits only.

The restore function is available for predefined calibration kits only.

Any changes made to the calibration kits are automatically saved into the nonvolatile memory of the Analyzer. Clicking the Save button is not required in order to save.

---

### NOTE

Changes to a predefined calibration kit can be cancelled at any time and the initial state will be restored.

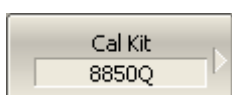
---

## Calibration Kit Selection

The calibration kit employed during a calibration should be selected according to the following procedure. If it is not specified in the list of the predefined calibration kits, it should be added. The procedure of adding and editing of the calibration kits is described in [Calibration Kit Management](#).



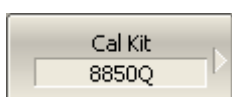
To open the list of the calibration kits (See figure below), use the following softkeys:



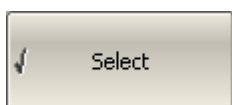
**Calibration > Cal Kit**

	Label	Description	Select	Predefined	Modified	#STDs	
2	85032B/E	Type-N 50Ω 6GHz Cal Kit (Agilent)	<input type="checkbox"/>	Yes	No	6	
3	85032F	Type-N 50Ω 9GHz Cal Kit (Agilent)	<input type="checkbox"/>	Yes	No	8	
4	85054B	Type-N 50Ω 18GHz Cal Kit with Sliding Load (Agilent)	<input type="checkbox"/>	Yes	No	8	
5	85054D	Type-N 50Ω 18GHz Cal Kit (Agilent)	<input type="checkbox"/>	Yes	No	6	
6	05CK10A-150	Type-N 50Ω 18GHz Cal Kit (Rosenberger)	<input type="checkbox"/>	Yes	No	6	
7	8850Q	Type-N 50Ω 18GHz Cal Kit (Maurly Microwave)	<input checked="" type="checkbox"/>	Yes	No	6	
8	8850C	Type-N 50Ω 18GHz Cal Kit with Sliding Load (Maurly Micr	<input type="checkbox"/>	Yes	No	8	
9	85033D/E	3.5 mm 6GHz/9GHz Cal Kit (Agilent)	<input type="checkbox"/>	Yes	No	6	
10	85052B	3.5 mm 26.5GHz Cal Kit with Sliding Load (Agilent)	<input type="checkbox"/>	Yes	No	10	

List of calibration kits



Highlight the required line in the list of the calibration kits and use the following softkey:



**Select**

Or click on the checkbox in the row "Select" using the mouse.

### NOTE

Make sure that the selected calibration kit is check marked.

## Operations on Table of Calibration Kits

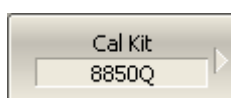
The table of calibration kits (See figure below) allows for selecting and editing of the calibration kits.

	Label	Description	Select	Predefined	Modified	#STDs	
2	85032B/E	Type-N 50Ω 6GHz Cal Kit (Agilent)	<input type="checkbox"/>	Yes	No	6	
3	85032F	Type-N 50Ω 9GHz Cal Kit (Agilent)	<input type="checkbox"/>	Yes	No	8	
4	85054B	Type-N 50Ω 18GHz Cal Kit with Sliding Load (Agilent)	<input type="checkbox"/>	Yes	No	8	
5	85054D	Type-N 50Ω 18GHz Cal Kit (Agilent)	<input type="checkbox"/>	Yes	No	6	
6	05CK10A-150	Type-N 50Ω 18GHz Cal Kit (Rosenberger)	<input type="checkbox"/>	Yes	No	6	
7	8850Q	Type-N 50Ω 18GHz Cal Kit (Maury Microwave)	<input checked="" type="checkbox"/>	Yes	No	6	
8	8850C	Type-N 50Ω 18GHz Cal Kit with Sliding Load (Maury Micr	<input type="checkbox"/>	Yes	No	8	
9	85033D/E	3.5 mm 6GHz/9GHz Cal Kit (Agilent)	<input type="checkbox"/>	Yes	No	6	
10	85052B	3.5 mm 26.5GHz Cal Kit with Sliding Load (Agilent)	<input type="checkbox"/>	Yes	No	10	

Table of calibration kits



To open the list of the calibration kits (See figure below), use the following softkeys:



**Calibration > Cal Kit**

[SENS:CORR:COLL:CKIT](#)

Sets or reads out the number of the selected calibration kit in the table of calibration kits

[SENS:CORR:COLL:CKIT:DESC](#)

Sets or reads out the calibration kit description string.

To edit a calibration kit, highlight its line in the table.

Calibration kit editing is comprised of two main procedures:

- Defining of the calibration standard (See [Calibration Standard Definition](#)).
- Assignment of classes to calibration standards (See [Classes of Calibration Standards](#)).

First, define the calibration standards, and then assign classes to them. Calibration standard definition and assignment of classes is performed in different tables.

The label of a calibration kit and its description can be edited in the table of the calibration kits (See above [figure](#)). The label appears on the calibration menu softkeys. The description is just to provide information for the user.

The table also contains display-only fields: flags of predefined and modified calibration kits and the counter of the calibration standards in a kit.

### Calibration Kit Selection for Editing

Move the highlighting to the required line in the calibration kit (See figure above) table using “↑” and “↓” arrows and click on the «**Enter**» softkey.

---

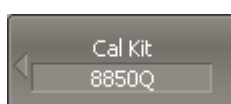
#### NOTE

The checkmark in the “Select” field does not matter for the kit selection for editing, it selects the calibration kit for calibration.

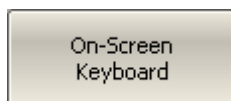
---

### Calibration Kit Label and Description Editing

Move the highlighting to the required line in the calibration kit (See figure above) table using “↑” and “↓” arrows and click on the «**Enter**» softkey. Then, enter the new text in the table.



To activate the on-screen keyboard, click the **On-Screen Keyboard** softkey.



---

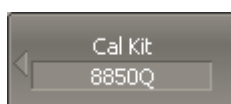
[SENS:CORR:COLL:CKIT:LAB](#)

Sets or reads out the calibration kit label.

---

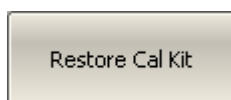
### Predefined Calibration Kit Restoration

Move the highlighting to the required line in the calibration kit (See above [figure](#)).



To cancel the user changes of a predefined calibration kit, use the following softkeys:

#### Restore Cal Kit



---

[SENS:CORR:COLL:CKIT:RES](#)

Resets the calibration kit to the factory settings.

---

---

**NOTE**

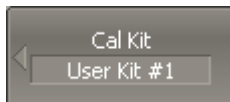
A predefined calibration kit can be restored, but cannot be erased.

---

### User-Defined Calibration Kit Deletion

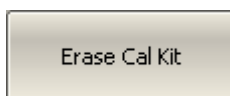
Move the highlighting to the required line in the calibration kit (See above [figure](#)).

---



To delete a user-defined calibration kit from the table, use the following softkey:

#### Erase Cal Kit



---

**NOTE**

A user-defined calibration kit cannot be restored, but can be erased.

---

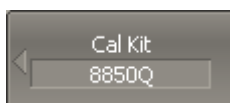
### Saving Calibration Kit to File

Saving a calibration kit to file is necessary for copying it to a different line of the table or to a different Analyzer.

This command is not necessary to save changes made by the user to the definitions of the kit, as these changes are saved automatically.

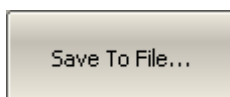
Move the highlighting to the required line in the calibration kit (See above [figure](#)).

---



To save a calibration kit to file, click the following softkey:

#### Save to File...



[MMEM:STOR:CKIT](#)

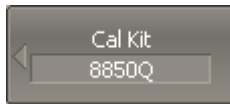
Saves the definition file for the calibration kit.



## Loading Calibration Kit from File

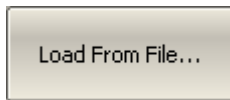
Calibration kit files that were created by the previous command can be loaded.

Move the highlighting to the required line in the calibration kit (See above [figure](#)).



To load a calibration kit from file, click the following softkey:

**Load from File...**



---

[MMEM:LOAD:CKIT](#)

Recalls the definition file for the calibration kit.

## Calibration Standard Definition

The definitions of the calibration standards included in one calibration kit are listed in the table as shown below.

	Standard			Frequency		Offset		
	No	Type	Label	Min	Max	Delay	Z0	Loss
▶	1	Open	Open -M-	0 Hz	999 GHz	37.026 ps	50 Ω	700 MΩ/s
	2	Open	Open -F-	0 Hz	999 GHz	19.42 ps	50 Ω	700 MΩ/s
	3	Short	Short -M-	0 Hz	999 GHz	42.063 ps	50 Ω	700 MΩ/s
	4	Short	Short -F-	0 Hz	999 GHz	24.512 ps	50 Ω	700 MΩ/s
	5	Load	Broadband	0 Hz	999 GHz	0 s	50 Ω	0 Ω/s
	6	Thru/Delay	Thru	0 Hz	999 GHz	0 s	50 Ω	700 MΩ/s

Terminal Impedance	$C0 \cdot 10^{-15} \text{ F}$	$C1 \cdot 10^{-27} \text{ F/Hz}$	$C2 \cdot 10^{-36} \text{ F/Hz}^2$	$C3 \cdot 10^{-45} \text{ F/Hz}^3$
	$L0 \cdot 10^{-12} \text{ H}$	$L1 \cdot 10^{-24} \text{ H/Hz}$	$L2 \cdot 10^{-33} \text{ H/Hz}^2$	$L3 \cdot 10^{-42} \text{ H/Hz}^3$
	99.14	353.6	62.23	0
	103	0	110	10.2
	0	0	0	0
	0	0	0	0
50 Ω				

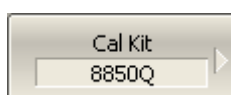
Calibration standard definition table for the standards defined by the model

The Analyzer provides two methods of defining a calibration standard:

- [Calibration standard model](#)
- [Table of S-parameters](#)

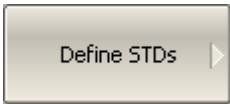
The calibration standards defined by S-parameters are called Data-Based standards. For the Data-Based standards editing, see [Data-Based Calibration Standards](#).

Each calibration standard is characterized by lower and upper values of the operating frequency. In the process of calibration, the measurements of the calibration standards outside the specified frequency range are ignored.



To open the table of calibration standard definitions, use the following softkeys:

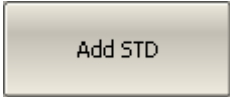
**Calibration > Cal Kit > Define STDs**



---

## Standard Adding to Calibration Kit

---



To add a calibration standard to the table of calibration standard definition (See figure above), use the following softkey:

**Calibration > Cal Kit > Define STDs > Add STD**

---

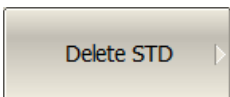
[SENS:CORR:COLL:CKIT:STAN:INS](#)

Inserts the calibration standard into the selected calibration kit.

---

## Standard Deleting from Calibration Kit

---



To delete a calibration standard from the table of calibration standard definition (See figure above), use the following softkey:

**Calibration > Cal Kit > Define STDs > Delete STD**

---

[SENS:CORR:COLL:CKIT:STAN:REM](#)

Deletes the calibration standard into the selected calibration kit.

---

## Calibration Standard Editing

Moving in the table of calibration standard definitions (See above figure) using navigation keys. Enter the parameter values for a calibration kit by using the navigation keys in the table of calibration standard definitions:

---

**N**

The calibration standard number is specified in the calibration kit data sheet (just for information).

---

**Type**

Select the standard type:

- Open
  - Short
-

	<ul style="list-style-type: none"> <li>• Load</li> <li>• Thru/Line</li> <li>• Unknown Thru</li> <li>• Sliding Load</li> <li>• Data-Based</li> </ul>
<b>Label</b>	Standard labels specified on the calibration menu softkeys.
<b>F min</b>	<p>Minimum operating frequency of the coaxial standard.</p> <p>Lower cutoff frequency of the waveguide standard.</p>
<b>F max</b>	<p>Maximum operating frequency of the coaxial standard.</p> <p>Upper cutoff frequency of the waveguide standard.</p>
<b>Delay</b>	Offset delay value in one direction (s). Can be switched to physical length (m). The parameter is used only for the calibration standard model.
<b>Z0</b>	<p>Offset characteristic impedance value (<math>\Omega</math>).</p> <p>For waveguide must be set to 1 <math>\Omega</math>.</p>
<b>Loss</b>	Offset loss value ( $\Omega/s$ ). The parameter is used only for the definition of the standard with the help of the calibration standard model.
<b>Media</b>	Coaxial or Waveguide
<b>H/W</b>	Waveguide height to width ratio.
<b>Terminal Impedance</b>	Lumped load impedance value ( $\Omega$ ). The parameter is used only for the definition of

	the standard with the help of the calibration standard model.
<b>C0 10–15 F</b>	For an OPEN standard, C0 coefficient in the polynomial formula of the fringe capacitance:  $C = C0 + C1 \cdot f + C2 \cdot f^2 + C3 \cdot f^3$
<b>C1 10–27 F/Hz</b>	For an OPEN standard, C1 coefficient in the polynomial formula of the fringe capacitance.
<b>C2 10–36 F/Hz<sup>2</sup></b>	For an OPEN standard, C2 coefficient in the polynomial formula of the fringe capacitance.
<b>C2 10–45 F/Hz<sup>3</sup></b>	For an OPEN standard, C3 coefficient in the polynomial formula of the fringe capacitance.
<b>L0 10–12 H</b>	For a SHORT standard, L0 coefficient in the polynomial formula of the residual inductance:  $L = L0 + L1 \cdot f + L2 \cdot f^2 + L3 \cdot f^3$
<b>L1 10–24 H/Hz</b>	For a SHORT standard, L1 coefficient in the polynomial formula of the residual inductance.
<b>L2 10–33 H/Hz<sup>2</sup></b>	For a SHORT standard, L2 coefficient in the polynomial formula of the residual inductance.
<b>L2 10–42 H/Hz<sup>3</sup></b>	For a SHORT standard, L3 coefficient in the polynomial formula of the residual inductance.
<a href="#"><u>SENS:CORR:COLL:CKIT:STAN:TYPE</u></a>	Sets or reads out the type of calibration standard.
<a href="#"><u>SENS:CORR:COLL:CKIT:STAN:LAB</u></a>	Sets or reads out the label for the calibration standard.

<a href="#"><u>SENS:CORR:COLL:CKIT:STAN:FMIN</u></a>	Sets or reads out the minimum frequency limit of the calibration standard.
<a href="#"><u>SENS:CORR:COLL:CKIT:STAN:FMAX</u></a>	Sets or reads out the maximum frequency limit of the calibration standard.
<a href="#"><u>SENS:CORR:COLL:CKIT:STAN:DEL</u></a>	Sets or reads out the offset delay value for the calibration standard.
<a href="#"><u>SENS:CORR:COLL:CKIT:STAN:Z0</u></a>	Sets or reads out the offset Z0 value for the calibration standard.
<a href="#"><u>SENS:CORR:COLL:CKIT:STAN:LOSS</u></a>	Sets or reads out the offset loss value for the calibration standard.
<a href="#"><u>SENS:CORR:COLL:CKIT:STAN:ARB</u></a>	Sets or reads out the value of the arbitrary impedance for the load standard.
<a href="#"><u>SENS:CORR:COLL:CKIT:STAN:C0</u></a>	Sets or reads out the C0 value for the open calibration standard.
<a href="#"><u>SENS:CORR:COLL:CKIT:STAN:C1</u></a>	Sets or reads out the C1 value for the open calibration standard.
<a href="#"><u>SENS:CORR:COLL:CKIT:STAN:C2</u></a>	Sets or reads out the C2 value for the open calibration standard.
<a href="#"><u>SENS:CORR:COLL:CKIT:STAN:C3</u></a>	Sets or reads out the C3 value for the open calibration standard.
<a href="#"><u>SENS:CORR:COLL:CKIT:STAN:L0</u></a>	Sets or reads out the L0 value for the short calibration standard.
<a href="#"><u>SENS:CORR:COLL:CKIT:STAN:L1</u></a>	Sets or reads out the L1 value for the short calibration standard.
<a href="#"><u>SENS:CORR:COLL:CKIT:STAN:L2</u></a>	Sets or reads out the L2 value for the short calibration standard.

---

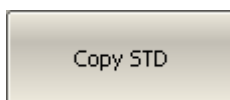
[SENS:CORR:COLL:CKIT:STAN:L3](#)

---

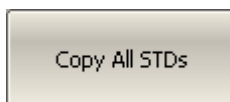
Sets or reads out the L3 value for the short calibration standard.

## Calibration Standard Copy/Paste Function

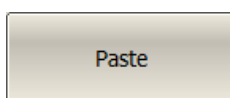
---



To save a calibration standard into clipboard, highlight the required line in the calibration standard definition table, and click the following softkey:



**Copy STD or Copy All STDs**

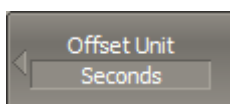


To paste the standard(s) from the clipboard, click the following softkey:

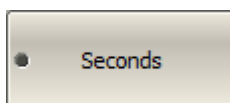
**Paste**

## Offset Delay Measurement Units Switching

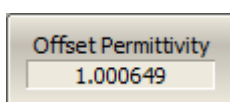
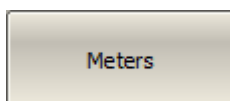
---



To switch the offset delay measurement units in the calibration standard definition table (See figure above), click the following softkey:



**Offset Unit > Seconds | Meters**



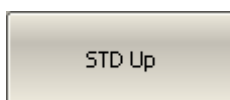
To enter the offset permittivity, click the following softkey:

**Offset Permittivity**

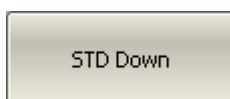
The offset permittivity is used only for the delay to length conversion. Default value equals the permittivity of air.

## Management of Sequence in Standard Table

---



To change the sequence in the table, use the following softkeys:



**STD Up or STD Down**

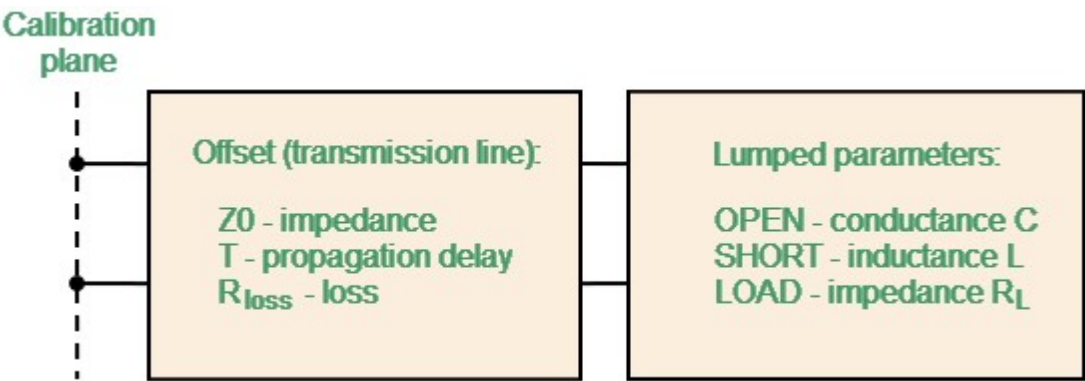




# Calibration Standard Model

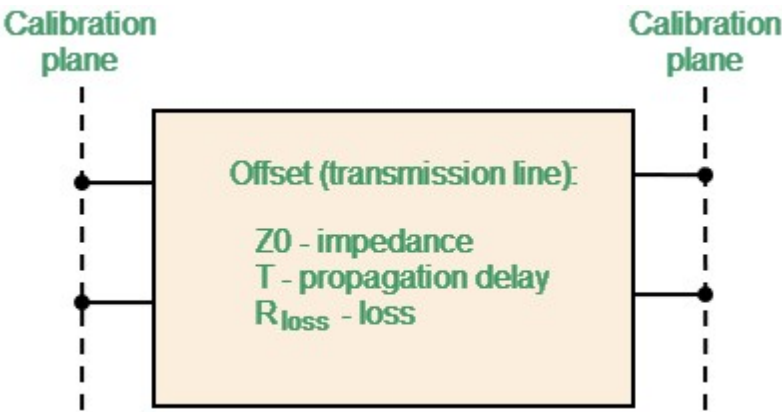
A model of a calibration standard presented as an equivalent circuit is used for determining S-parameters of the standard. The model is employed for standards of OPEN, SHORT, FIXED LOAD, THRU/LINE types.

A one-port model is used for the standards OPEN, SHORT and FIXED LOAD (See [Full One-Port Calibration](#)). This is shown in the figure below.



One-port standard model

The two-port model is used for the standard THRU/LINE (See figure below).



Two-port standard model

The description of the numeric parameters of an equivalent circuit model of a calibration standard is shown in the table below.

## Parameters of the calibration standard equivalent circuit model

Parameter (as in the program)	Parameter Definition
<b>Z0</b>  (Offset Z0)	<p>The characteristic impedance of the transmission line [<math>\Omega</math>], serving as the offset.</p> <p>For the coaxial line specified real value of characteristic impedance, usually equal to 50 <math>\Omega</math> or 75 <math>\Omega</math>.</p> <p>For waveguide calibration, the special value of 1 <math>\Omega</math> is used.</p>
<b>T</b>  (Offset Delay)	<p>The offset delay. It is defined as one-way signal propagation time in the transmission line [seconds]. The delay can be measured or mathematically determined by dividing the exact physical length by the propagation velocity in the line.</p> <p>For waveguide, delay is conventionally taken to be equal to the delay of a coaxial line of the same length. The actual signal delay in waveguide is frequency dependent and is calculated in the program.</p> <p>Instead delay, one can specify the length of the offset [meters]. The software calculates the delay according to the formula for a coaxial air line:</p> $T = \frac{\sqrt{\epsilon_r} l}{c},$ <p>where <math>l</math> — line length [m], <math>c</math> — light speed in free space 299792458 [m/s], <math>\epsilon_r</math> — relative permittivity of air 1.000649.</p> <p>The length can be specified instead of the delay provided offset of the calibration standard is a coaxial airline or a waveguide. If the calibration standard manufacturer provides a delay data, it is better to specify delay.</p> <p>Note: When the Multiline TRL calibration is used it is recommended to always specify the length of TRL lines</p>

Parameter (as in the program)	Parameter Definition
	independently of line type, dielectric, presence of propagation speed dispersion. The Multiline TRL uses for calculations physical length of lines.
<b>Rloss</b>  (Offset Loss)	<p>The offset loss in one-way propagation due to the skin effect [<math>\Omega/\text{sec}</math>].</p> <p>The loss in a coaxial transmission line is determined by measuring the delay <math>T</math> [sec] and loss <math>L</math> [dB] at 1 GHz frequency. The measured values are used in the following formula:</p> $R_{\Pi}[\Omega/s] = \frac{L[\text{dB}] \cdot Z_0[\Omega]}{4.3429[\text{dB}] \cdot T[\text{s}]}$ <p>The loss in waveguide is typically set to 0 due to its very small influence. However, the software supports a waveguide loss model. If the calibration standard manufacturer provides loss data, it is recommended to specify it.</p>
<b>Rload</b>  (Load Impedance)	<p>Load impedance of fixed load calibration standard [<math>\Omega</math>].</p> <p>For the coaxial calibration standard specified real value of characteristic impedance, usually equal to 50 <math>\Omega</math> or 75 <math>\Omega</math>.</p> <p>For waveguide calibration, the special value of 1 <math>\Omega</math> is used.</p>
<b>C</b>  (C0, C1, C2, C3)	<p>The fringe capacitance of an OPEN standard, which causes a phase offset of the reflection coefficient at high frequencies. The fringe capacitance model is described as a function of frequency, which is a polynomial of the third degree:</p> $C = C_0 + C_1 \cdot f + C_2 \cdot f^2 + C_3 \cdot f^3, \text{ where}$ <p><math>f</math> — frequency [Hz],  <math>C_0 \dots C_3</math> — polynomial coefficients.</p>

Parameter (as in the program)	Parameter Definition
	Units: C0[F], C1[F/Hz], C2[F/Hz <sup>2</sup> ], C3[F/Hz <sup>3</sup> ].
<b>L</b> (L0, L1, L2, L3)	<p>The residual inductance of a SHORT standard, which causes a phase offset of the reflection coefficient at high frequencies. The residual inductance model is described as a function of frequency, which is a polynomial of the third degree:</p> $L = L0 + L1 \cdot f + L2 \cdot f^2 + L3 \cdot f^3, \text{ where}$ <p><math>f</math> — frequency [Hz],  L0...L3 — polynomial coefficients.</p> <p>Units: L0[H], L1[H/Hz], L2[H/Hz<sup>2</sup>], L3[H/Hz<sup>3</sup>].</p>
<b>Media</b>	<p>The offset media. Allows to choose from:</p> <ul style="list-style-type: none"> <li>• coaxial</li> <li>• waveguide</li> </ul>
<b>Width to Height Ratio</b> (H/W)	The waveguide width to height ratio. Used in the waveguide loss model when the loss value is not zero.
<b>Minimum and Maximum Frequency</b> (Fmin, Fmax)	<p>The minimum and maximum standard operating frequency in the coaxial. Used for a calibration using several calibration standards, each of which does not cover entire frequency range.</p> <p>The cut off frequency and the doubled cut off frequency of the waveguide. The cutoff frequency of the waveguide is achieved at a wavelength in the waveguide equal to twice its width. Take care not to confuse this with the minimum and maximum operating frequency of the waveguide, which are usually given by the manufacturer with a margin relative to the cut off frequency.</p>

## Data-Based Calibration Standards

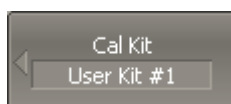
The calibration standards defined by data are set using the table of S-parameters. Each line of the table contains frequency and S-parameters of the calibration standard. For one-port standards the table contains the value of only one parameter — S11, and for two-port standards the table contains the values of all the four parameters — S11, S21, S12, S22.

The table of S-parameters can be filled in manually or downloaded from a file of Touchstone format. Files with \*.S1P extension are used for one-port standards, and files with \*.S2P extension are used for two-port standards.

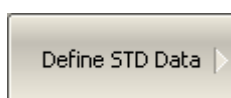
The table of calibration standard S-parameters (See figure below) allows viewing and editing of S-parameters of the calibration standards of the “Data-Based” type.

	Frequency	MLog(S11)	Arg(S11)	MLog(S21)	Arg(S21)
1	300 kHz	-67.5270094 dB	53.4023888 °	0.00394846762 dB	0.291495528 °
2	8.2997 MHz	-81.9584188 dB	-119.014476 °	0.00351271886 dB	0.322307311 °
3	16.2994 MHz	-78.8287269 dB	-128.601485 °	0.00322754052 dB	0.320729452 °
4	24.2991 MHz	-76.6945937 dB	-126.226624 °	0.00335760667 dB	0.321443452 °
5	32.2988 MHz	-75.3564463 dB	-123.320088 °	0.00336896915 dB	0.322652352 °
6	40.2985 MHz	-73.9037876 dB	-118.924822 °	0.00352190889 dB	0.325734959 °
7	48.2982 MHz	-72.3276906 dB	-114.247608 °	0.00416283402 dB	0.327349639 °
8	56.2979 MHz	-70.8755884 dB	-109.418739 °	0.00457673931 dB	0.322968328 °
9	64.2976 MHz	-69.1166427 dB	-105.957991 °	0.00452183965 dB	0.316878686 °

Table of calibration standard S-parameters



To open the table of calibration standard S-parameters, move the required line in the table (See above [figure](#)), and click the following softkeys:



**Define STD Data**

[SENS:CORR:COLL:CKIT:STAN:DATA](#)

Writes or reads out the data array of the data-based calibration standard.

### NOTE

The **Define STD Data** softkey is disabled if the type of the standard is other than “Data-Based”.

There are two different tables for one-port standards and for two-port standards. The table contains one parameter (S11) for one-port standards, and four parameters (S11, S21, S12, S22) for two-port standards. Before the user fills in the table, its type will be defined: by the Touchstone format (S1P or S2P) if the data is downloaded

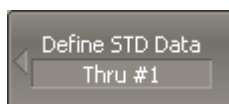
from a file, or the user will be requested to specify the type if the data is entered by the user.

The data in the table can be represented in three formats according to the user settings:

- Real part and Imaginary part.
- Linear magnitude and Phase (°).
- Logarithmic magnitude (dB) and Phase (°).

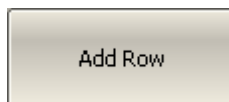
The following rule is applied for the calibration of a two-port standard: the standard is considered connected by Port 1 (S11) to the port with smallest number and by Port 2 (S22) to the port with the biggest number. If a two-port standard needs to be reversed, use the Port Reverse function (See [Port Reversing](#)).

### Adding Lines to Table



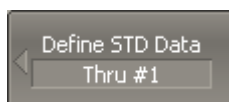
To add a line to the table of the calibration standard S-parameters (See above [figure](#)), use the following softkeys:

#### Add Row



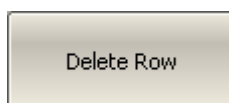
---

### Deleting Lines from Table



To delete a line from the table of the calibration standard S-parameters (See above [figure](#)), use the following softkey:

#### Delete Row



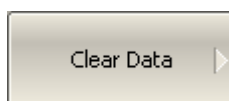
---

### Table Clearing

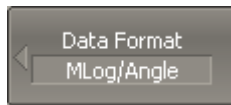


To clear the entire table of the calibration standard S-parameters (See above [figure](#)), use the following softkey:

#### Clear Data

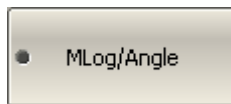
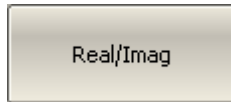


## Table Format Selection



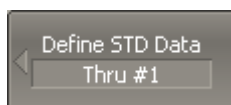
To select the format of the table of the calibration standard S-parameters (See above [figure](#)), use the following softkey:

**Format > Real/Imag | Magn/Angle | MLog/Angle**



---

## Port Reversing



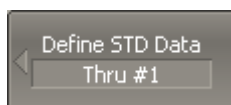
To enable/disable reversing of the ports of a two-port standard, use the following softkey:

**Reverse Ports**



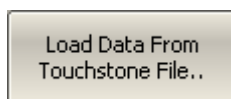
---

## Loading Data from File



To load the data from Touchstone file, use the following softkey:

**Load Data from Touchstone file...**



In the pop-up dialog select the file type (S1P or S2P) and specify the file name.

---



## Scope of Calibration Standard Definition

Different methods of calibration apply either full or partial definitions of the calibration standards.

The full two-port calibration, full one-port calibration, one-path two-port calibration, and normalization use fully defined calibration standards, i.e. the standards with known S-parameters. The S-parameters of OPEN, SHORT, LOAD, and THRU/LINE must be defined by the model or by data.

---

### NOTE

The UNKNOWN THRU and SLIDING LOAD standards are exceptional in the above calibrations. The S-parameters of these standards are defined in the process of calibration. UNKNOWN THRU is used only in full two-port calibration.

---

TRL calibration and its modifications (TRM, LRL, LRM) apply partial definition of the standards:

- TRL THRU standards must have the required value of  $Z_0$  ( $S_{11}=S_{22}=0$ ) and known length (delay).
- TRL LINE/MATCH standard must have the same value of  $Z_0$  as the first standard.
- TRL REFLECT standard must have the phase known as accurately as  $\pm 90^\circ$ .

## Classes of Calibration Standards

Along with defining a calibration standard by a calibration model or data, the standard should also be assigned a specific class. One calibration standard may belong to several classes. The class assignment is performed for each particular calibration kit.

Class assignment to a calibration standard is required for specifying such properties as the calibration method, the role of a standard in the calibration, and the number of the port(s). The Analyzer supports the following classes of the calibration standards (See table below).

Calibration Methods	Class Label	Port
Full Two-Port Calibration	OPEN	1
Full One-Port Calibration		2
One-Path Two-Port Calibration	SHORT	1
Transmission Normalization		2
Reflection Normalization	LOAD	1
		2
	THRU	1-2
TRL Calibration	TRL THRU	1-2
LRL Calibration	TRL REFLECT	1
TRM Calibration		2
LRM Calibration	TRL LINE/MATCH	1-2

For example, if the class "OPEN of Port 1" is assigned to the OPEN -F- calibration standard, it will indicate that this standard is used for calibrating the first port using the following calibration methods: full two-port, full one-port, one-path two-port, and normalization.

---

**NOTE**

The class assignment changes the labels of the calibration standards on the calibration softkeys.

---

The assignment of classes to the standards of the selected calibration kit is made in the table of standard classes (See figure below).

Class	Port	Subclass 1	Subclass 2	Subclass 3	S
Open	1	4. Open -M-			
	2	4. Open -M-			
Short	1	5. Short -M-			
	2	5. Short -M-			
Load	1	1. Lowdband	2. Sliding Load	3. Broadband	
	2	1. Lowdband	2. Sliding Load	3. Broadband	
Thru	1-2	11. Thru			
TRL Thru	1-2				
TRL Reflect	1				
	2				
TRL Line/Match	1-2				




Table of calibration standard classes

Standard labels populate the table cells by selecting them from the list of calibration kit standards.

Each row of the table corresponds to the standard class specified in the two left columns of the table.

If a single standard is assigned to the class, then it filled into the "Subclass 1" column. If several standards are assigned to the class then "Subclass 2", "Subclass 3", etc. columns are filled in.

---

**NOTE**

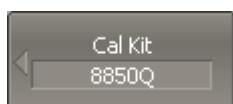
When assigning two or more subclasses to one class of calibration standards the calibration menu changes: the standard measurement softkey is replaced by the softkey, which opens the subclass menu containing the list of all the standards of this class.

---

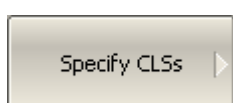
Using one subclass is appropriate in cases when it is known which standard and of which polarity, male or female, is assigned to each port, or when the mathematical models of male and female standards are the same. Using of one subclass simplifies the calibration menu structure.

Using more than one subclass allows to:

- Postpone the selection of standards of the same class available in the calibration kit to the calibration stage. It is possible to select between male and female standards, FLUSH THRU and UNKNOWN THRU.
- Perform the band split calibration as described in [Bandsplit Calibration Using Subclasses](#).



To open the table of calibration standard classes, use the following softkeys:



**Calibration > Cal Kit > Specify CLSs**

---

## Standard Class Table Editing

Moving in the table of calibration standard classes (See [above figure](#)) using navigation keys, click «Enter» in the required cell for the pop-up menu. Select the standard label in the pop-up menu to assign it the class and port number specified in the left part of the table.

---

[SENS:CORR:COLL:CKIT:ORD:SEL](#)

The subclass used to specify classes of calibration standards by the commands:

---

[SENS:CORR:COLL:CKIT:ORD:LOAD](#)

Sets or reads out the number of the calibration standard of the LOAD type used for the measurement of the specified port.

---

[SENS:CORR:COLL:CKIT:ORD:OPEN](#)

Sets or reads out the number of the calibration standard of the OPEN type used for the measurement of the specified port.

---

[SENS:CORR:COLL:CKIT:ORD:SHOR](#)

Sets or reads out the number of the calibration standard of the SHORT type used for the measurement of the specified port.

---

[SENS:CORR:COLL:CKIT:ORD:THRU](#)

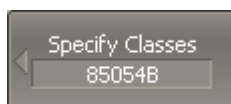
Sets or reads out the number of the calibration standard of the THRU type

---

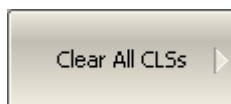
	used for the measurement between the specified ports.
<a href="#">SENS:CORR:COLL:CKIT:ORD:TRL</a>	Sets or reads out the number of the calibration standard of the TRL LINE type used for the measurement between the specified ports.
<a href="#">SENS:CORR:COLL:CKIT:ORD:TRLT</a>	Sets or reads out the number of the calibration standard of the TRL THRU type used for the measurement between the specified ports.
<a href="#">SENS:CORR:COLL:CKIT:ORD:TRLR</a>	Sets or reads out the number of the calibration standard of the TRL REFLECT type used for the measurement of the specified port.

### Deleting Standards from the Standard Class Table

Moving in the table of calibration standard classes (See [above figure](#)) using navigation keys, click «Enter» in the required cell for the pop-up menu. Select the line None in the pop-up menu to delete the standard contained in the cell.



To delete all the standards in the table of calibration standard classes, use the following softkey:

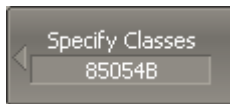


**Clear All CLSs**

## Strict Class Assignment Function

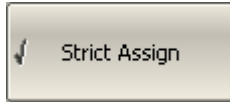
This function allows for limitation of the one standard type(s) available in each class by the feature of strict correspondence (See table below). If this function is disabled, any class can be assigned to the standard.

N	Standard Class	Standard Type
1	OPEN	Open, Data-Based (One Port)
2	SHORT	Short, Data-Based (One Port)
3	LOAD	Load, Sliding Load, Data-Based (One Port)
4	THRU	Thru/Line, Data-Based (Two Port)
5	TRL THRU	Thru/Line, Data-Based (Two Port)
6	TRL REFLECT	Open, Short, Data-Based (One Port)
7	TRL LINE/MATCH	Load, Thru/Line



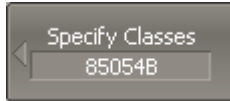
To disable/enable the function of strict class correspondence function, use the following softkey:

**Strict Assign**

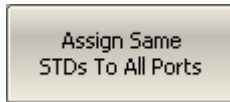


## Group Assignment of Port Number Function

This function allows for automatic assignment of one standard to all the ports of a specific class when assigned to at least one port.



To enable/disable the function of group assignment of port number, use the following softkey:



**Assign Same STDs to All Ports**

---



## Subclasses of Calibration Standards

Subclasses are used to assign one class to several calibration standards. The procedure of subclass assignment is mainly employed for calibration within a wide frequency range by several calibration standards, each of which does not cover the full frequency range. Each class of standards can contain up to 8 subclasses. The procedure of subclass assignment to the calibration standards is described in [Classes of Calibration Standards](#).

For example, suppose the LOAD standard is defined as from 0 to 2 GHz, and the sliding LOAD standard is defined as from 1.5 to 12 GHz. To perform calibration within the full frequency range, the fixed LOAD should be assigned the subclass 1, and the sliding LOAD should be assigned the subclass 2 of the "load" class.

If the standards have an overlapping frequency range (as in the example above, from 1.5 to 2 GHz), the last measured standard will be used.

---

### NOTE

Subclass assignment changes the labels of the calibration softkeys. The measurement softkey is replaced by the key, which opens the subclass menu containing the keys for measuring several calibration standards.

---

## Calibration Methods and Procedures

The Analyzer supports several methods of one-port and two-port calibration. The calibration methods vary by quantity and type of the standards being used, by type of error correction, and accuracy. The table below presents an overview of calibration methods.

Calibration Method	Parameters	Standards	Errors	Accuracy
<a href="#">Reflection Normalization</a>	S11 or S22	<ul style="list-style-type: none"> <li>• SHORT or OPEN</li> <li>• LOAD (if optional directivity is performed)</li> </ul>	Er1, Ed1 <sup>1</sup> or Er2, Ed2 <sup>1</sup>	Low
<a href="#">Transmission Normalization</a>	S21 or S12	<ul style="list-style-type: none"> <li>• THRU</li> <li>• 2 LOADs (if optional isolation calibration is performed)</li> </ul>	Et1, Ex1 <sup>2</sup> or Et2, Ex2 <sup>2</sup>	Low
<a href="#">Full One-Port Calibration</a>	S11 or S22	<ul style="list-style-type: none"> <li>• SHORT</li> <li>• OPEN</li> <li>• LOAD</li> </ul>	Er1, Ed1, Es1 or Er2, Ed2, Es1	High
<a href="#">One-Path Two-Port Calibration</a>	S11, S21 or S12, S22	<ul style="list-style-type: none"> <li>• SHORT</li> <li>• OPEN</li> <li>• LOAD</li> <li>• THRU</li> </ul>	Er1, Ed1, Es1, Et1, Ex1 <sup>2</sup> or Er2, Ed2, Es2, Et2, Ex2 <sup>2</sup>	Medium

Calibration Method	Parameters	Standards	Errors	Accuracy
		<ul style="list-style-type: none"> <li>• 2 LOADs (if optional isolation calibration is performed)</li> </ul>		
<a href="#">Full Two-Port Calibration</a>	S11, S21 S12, S22	<ul style="list-style-type: none"> <li>• SHORT</li> <li>• OPEN</li> <li>• LOAD</li> <li>• THRU</li> <li>• 2 LOADs (if optional isolation calibration is performed)</li> </ul>	Er1, Ed1, Es1, Et1, EI1, Ex1 <sup>2</sup>  Er2, Ed2, Es2, Et2, EI2, Ex2 <sup>2</sup>	High
<a href="#">Two-Port TRL Calibration</a>	S11, S21 S12, S22	<ul style="list-style-type: none"> <li>• THRU or LINE</li> <li>• REFLECT</li> <li>• LINE or 2 LOADs</li> </ul>	Er1, Ed1, Es1, Et1, EI1  Er2, Ed2, Es2, Et2, EI2	Very High
1. If optional directivity calibration is performed.  2. If optional isolation calibration is performed.				

## Reflection Normalization

Reflection normalization is the simplest calibration method used for reflection coefficient measurements (S11 or S22). Measurement of one standard (SHORT or OPEN) is sufficient to perform this type of calibration (See figure below). This method is called normalization because the measured S-parameter at each frequency point is divided (normalized) by the corresponding S-parameter of the calibration standard. Reflection normalization corrects the reflection tracking error (**Er**) only. This constrains the accuracy of the method.

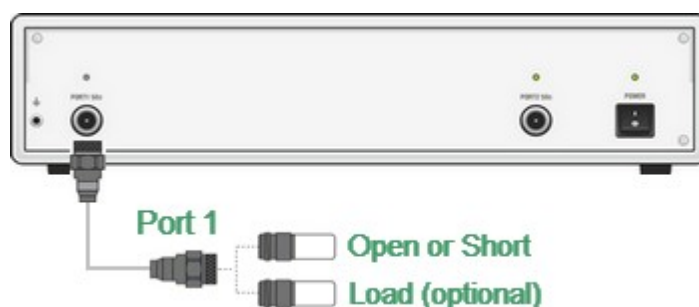
---

### NOTE

Reflection normalization can also be referred to as **response open** or **response short** calibration depending on the standard being used: OPEN or SHORT.

---

An optional LOAD standard measurement can be performed to correct the directivity error (**Ed**). The optional directivity calibration increases the accuracy of the reflection normalization.



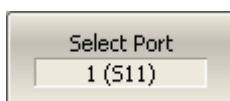
Reflection normalization

Before starting calibration perform the following settings: select active channel, set the parameters of the channel (frequency range, IF bandwidth, etc.), and select the calibration kit.



To open reflection normalization submenu, use the following softkeys:

**Calibration > Calibrate > Response (Open) | Response (Short)**



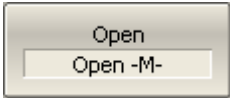


Select the test port to be calibrated using **Select Port**. Switch between the test ports (measured parameters) using this softkey.

---

[SENS:CORR:COLL:METH:OPEN](#)

Selects the port and sets the response calibration (Open) type for the calculation

---

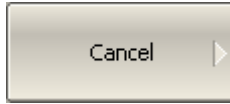
	of the calibration coefficients on completion of the calibration executed by the <a href="#">SENS:CORR:COLL:SAVE</a> command.
<a href="#">SENS:CORR:COLL:METH:SHOR</a>	Selects the port and sets the response calibration (Short) type for the calculation of the calibration coefficients on completion of the calibration executed by the <a href="#">SENS:CORR:COLL:SAVE</a> command.
	<p>Connect an OPEN or a SHORT standard to the test port as shown in above figure. Perform measurement using the <b>Open</b> or <b>Short</b> softkey respectively.</p> <p>The instrument status bar will indicate <b>Calibration in progress...</b> when the measurement is in progress. On completion of the measurement, a check mark will appear in the left part of the softkey.</p>
	<p>To perform the optional directivity calibration, connect a LOAD standard to the test port as shown in the above figure and perform measurement using <b>Load (Optional)</b> softkey.</p> <p>The instrument status bar will indicate <b>Calibration in progress...</b> when the measurement is in progress. On completion of the measurement a check mark will appear in the left part of the softkey.</p>
<a href="#">SENS:CORR:COLL:OPEN</a>	Measures the calibration data of the open standard for the specified port.
<a href="#">SENS:CORR:COLL:SHOR</a>	Measures the calibration data of the short standard for the specified port.
<a href="#">SENS:CORR:COLL:LOAD</a>	Measures the calibration data of the load standard for the specified port.
	<p>To complete the calibration procedure, click <b>Apply</b>.</p> <p>This will activate the process of calibration coefficient table calculation and saving it into the memory. The error correction function will also be automatically enabled.</p>

---

[SENS:CORR:COLL:SAVE](#)

Calculates the calibration coefficients from the calibration standards measurements depending on the selected calibration type.

---



To clear the measurement results of the standards, click Cancel.

This softkey does not cancel the current calibration. To disable the current calibration turn off the error correction function (See [Error Correction Disabling](#)).

---

[SENS:CORR:COLL:CLE](#)

Clears the measurement data of the calibration standards.

---

---

**NOTE**

The calibration status can be checked in channel status bar (See [General error correction status table](#)) or in trace status field (See [Trace error correction status table](#)).

---

## Transmission Normalization

Transmission normalization is the simplest calibration method used for transmission coefficient measurements ( $S_{21}$  or  $S_{12}$ ). Measurement of one THRU standard is sufficient to perform this type of calibration (See figure below). This method is called normalization because the measured S-parameter at each frequency point is divided (normalized) by the corresponding S-parameter of the calibration standard. Transmission normalization corrects the transmission tracking error (**Et**) only. This constrains the accuracy of the method.

---

### NOTE

Transmission normalization can also be referred to as **response thru** calibration.

---

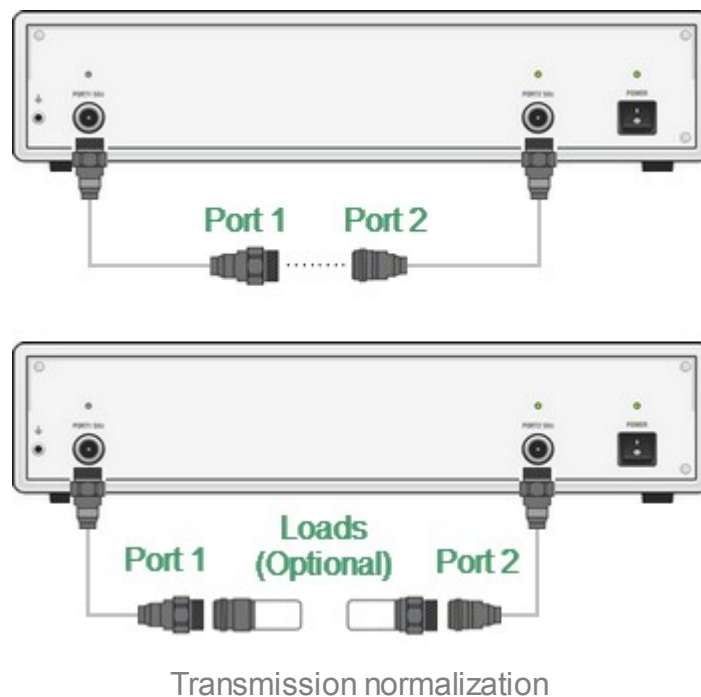
An optional isolation calibration can be performed by measurement of two LOAD standards connected to both test ports of the analyzer. In this case, the isolation error (**Ex**) is additionally corrected in the transmission normalization.

---

### NOTE

For isolation calibration, set a narrow IF bandwidth and firmly attach the cables.

---

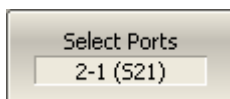


Before starting calibration perform the following settings: select active channel, set the parameters of the channel (frequency range, IF bandwidth, etc.), and select the calibration kit.



To open transmission normalization submenu, use the following softkeys:

**Calibration > Calibrate > Response (Thru)**

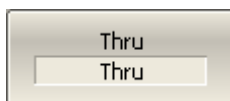


Select the direction of the calibration using the **Select Ports** softkey. The label on the softkey indicates the following: receiver port — source port (measured parameter).

---

[SENS:CORR:COLL:METH:THRU](#)

Selects the ports and sets the response calibration (Thru) type for the calculation of the calibration coefficients on completion of the calibration executed by the [SENS:CORR:COLL:SAVE](#) command.



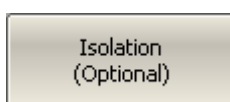
Connect a THRU standard between the test ports. If the port connectors allow through connection connect them directly (zero electrical length thru). Perform measurement using the **Thru** softkey.

The instrument status bar will indicate **Calibration in progress...** when the measurement is in progress. On completion of the measurement a check mark will appear in the left part of the softkey.

---

[SENS:CORR:COLL:THRU](#)

Measures the calibration data of the thru standard between the receiver port <rcvport> and the source port <srcport>.



To perform the optional isolation calibration, connect two LOAD standards to the test ports as shown in the above figure and enable measurement using the **Isolation (Optional)** softkey.

The instrument status bar will indicate **Calibration in progress...** when the measurement is in progress. On completion of the measurement a check mark will appear in the left part of the softkey.

---

[SENS:CORR:COLL:ISOL](#)

Measures the isolation calibration data between the receiver port <rcvport> and the source port <srcport>.





To complete the calibration procedure, click **Apply**.

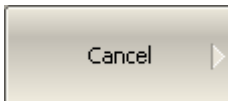
This will activate the process of calibration coefficient table calculation and saving it into the memory. The error correction function will also be automatically enabled.

---

[SENS:CORR:COLL:SAVE](#)

Calculates the calibration coefficients from the calibration standards measurements depending on the selected calibration type.

---



To clear the measurement results of the standard, click Cancel.

This softkey does not cancel the current calibration. To disable the current calibration, turn off the error correction function (See [Error Correction Disabling](#)).

---

[SENS:CORR:COLL:CLE](#)

Clears the measurement data of the calibration standards.

---

---

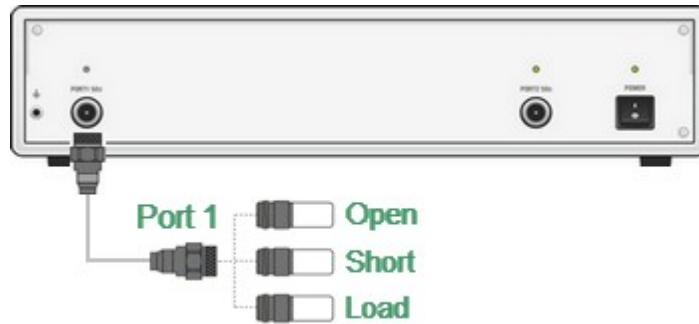
#### NOTE

The calibration status can be checked in channel status bar (See [General error correction status table](#)) or in trace status field (See [Trace error correction status table](#)).

---

## Full One-Port Calibration

Full one-port calibration (SOL) is used for reflection coefficient measurements ( $S_{11}$  or  $S_{22}$ ). The three calibration standards (SHORT, OPEN, LOAD) are measured (See figure below) in the process of this calibration. Measurement of the three standards allows for acquisition of all the three error terms (**Ed**, **Es**, and **Er**) of a one-port model. Full one-port calibration is a highly accurate method for 1-port reflection measurements.



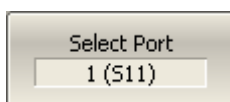
Full one-port calibration

Before starting calibration perform the following settings: select active channel, set the parameters of the channel (frequency range, IF bandwidth, etc.), and select the calibration kit.

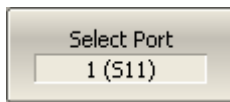


To open full one-port calibration submenu, use the following softkeys:

**Calibration > Calibrate > Full 1-Port Cal**



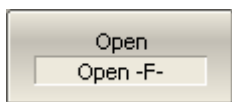
Select the test port to be calibrated using **Select Port**. Switch between the test ports (measured parameters) using this softkey.



Select the test port to be calibrated using **Select Port**. Clicking this softkey allows to switch between the test ports (measured parameters).

[SENS:CORR:COLL:METH:SOLT1](#)

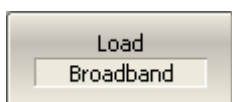
Selects the port and sets the full one-port (SOL) calibration type for the calculation of the calibration coefficients on completion of the calibration executed by the [SENS:CORR:COLL:SAVE](#) command.



Connect SHORT, OPEN, and LOAD standards to the selected test port in any consequence as shown in the above figure. Perform measurements clicking the softkey **Open**, **Short**, **Load** corresponding to the connected standard.



The instrument status bar will indicate **Calibration in progress...** when the measurement is in progress. On completion of the measurement a check mark will appear in the left part of the softkey.



---

[SENS:CORR:COLL:OPEN](#)

Measures the calibration data of the open standard for the specified port.

---

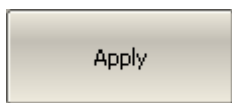
[SENS:CORR:COLL:SHOR](#)

Measures the calibration data of the short standard for the specified port.

---

[SENS:CORR:COLL:LOAD](#)

Measures the calibration data of the load standard for the specified port.



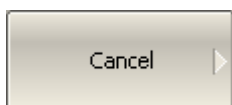
To complete the calibration procedure, click **Apply**.

This will activate the process of calibration coefficient table calculation and saving it into the memory. The error correction function will also be automatically enabled.

---

[SENS:CORR:COLL:SAVE](#)

Calculates the calibration coefficients from the calibration standards measurements depending on the selected calibration type.



To clear the measurement results of the standard, click **Cancel**.

This softkey does not cancel the current calibration. To disable the current calibration, turn off the error correction function (See [Error Correction Disabling](#)).

---

[SENS:CORR:COLL:CLE](#)

Clears the measurement data of the calibration standards.

---

**NOTE**

The calibration status can be checked in channel status bar (See [General error correction status table](#)) or in trace status field (See [Trace error correction status table](#)).

---



## One-Path Two-Port Calibration

A one-path two-port calibration combines full one-port calibration with transmission normalization. This method allows for a more accurate estimation of transmission tracking error (**Et**) than using transmission normalization.

One-path two-port calibration involves connection of the three standards to the source port of the Analyzer (as for one-port calibration) and a THRU standard connection between the calibrated source port and the other receiver port (See figure below).

One-path two-port calibration allows for correction of **Ed**, **Es**, and **Er** error terms of the source port and a transmission tracking error term (**Et**). This method does not derive source match error term (**EI**) of a [two-port error model](#).

An optional isolation calibration can be performed by measurement of two LOAD standards connected to both test ports of the analyzer. In this case, the isolation error (**Ex**) is additionally corrected in the one-path two-port calibration.

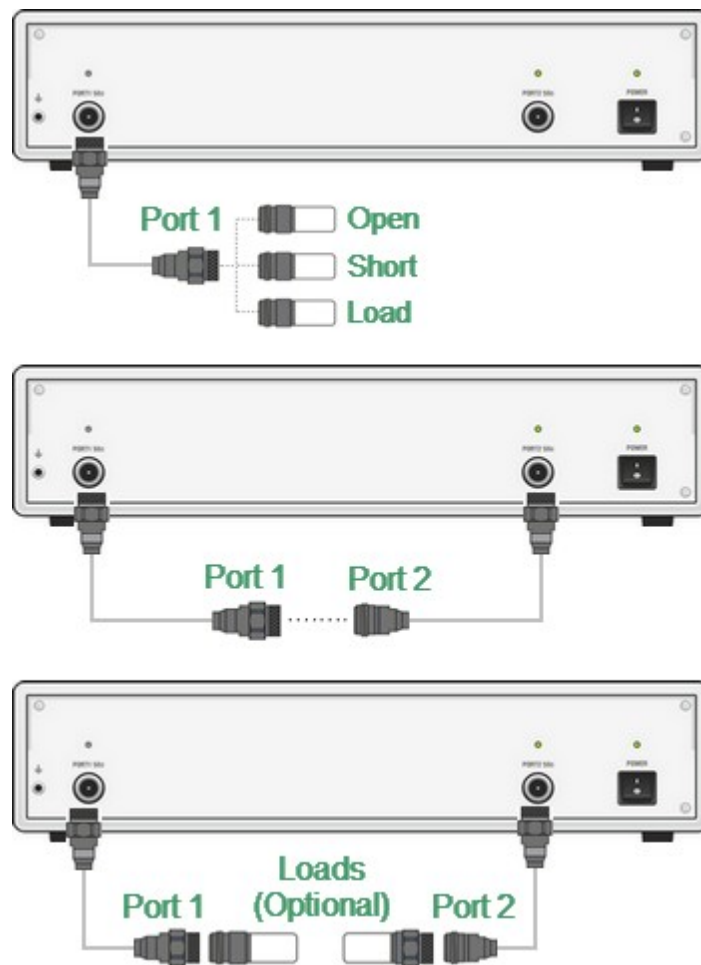
---

**NOTE**

For isolation calibration, set a narrow IF bandwidth and firmly attach the cables.

---

One-path two-port calibration is used for measurements of the parameters of a non-reciprocal DUT such as amplifiers in one direction, e.g. S11 and S21.



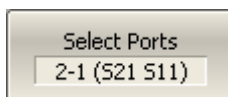
One-path two-port calibration

Before starting calibration perform the following settings: select active channel, set the parameters of the channel (frequency range, IF bandwidth, etc.), and select the calibration kit.



To open one-path two-port calibration submenu, use the following softkeys:

**Calibration > Calibrate > One Path 2-Port Cal**

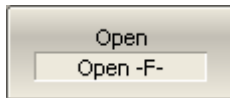


Select the direction of the calibration using the **Select Ports** softkey. The label on the softkey indicates the following: receiver port — source port (measured parameters).

---

[SENS:CORR:COLL:METH:ERES](#)

Selects the ports and sets the one path two-port calibration type for the calculation of the calibration coefficients on completion of the calibration executed by the [SENS:CORR:COLL:SAVE](#) command.



Connect SHORT, OPEN and LOAD standards to the source port in any consequence, as shown in the above figure. Perform measurements clicking the softkey **Open**, **Short**, **Load** corresponding to the connected standard.

The instrument status bar will indicate **Calibration in progress...** when the measurement is in progress. On completion of the measurement a check mark will appear in the left part of the softkey.

---

[SENS:CORR:COLL:OPEN](#)

Measures the calibration data of the open standard for the specified port.

---

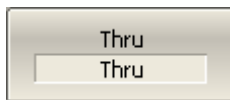
[SENS:CORR:COLL:SHOR](#)

Measures the calibration data of the short standard for the specified port.

---

[SENS:CORR:COLL:LOAD](#)

Measures the calibration data of the load standard for the specified port.



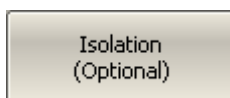
Connect a THRU standard between the test ports. If the port connectors allow through connection connect them directly (zero electrical length thru). Perform measurement using the **Thru** softkey.

The instrument status bar will indicate **Calibration in progress...** when the measurement is in progress. On completion of the measurement a check mark will appear in the left part of the softkey.

---

[SENS:CORR:COLL:THRU](#)

Measures the calibration data of the thru standard between the receiver port <rcvport> and the source port <srcport>.



To perform the optional isolation calibration, connect two LOAD standards to the test ports as shown in the above figure and enable measurement using the **Isolation (Optional)** softkey.

---

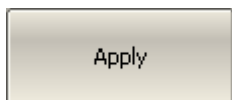
The instrument status bar will indicate **Calibration in progress...** when the measurement is in progress. On completion of the measurement a check mark will appear in the left part of the softkey.

---

[SENS:CORR:COLL:ISOL](#)

Measures the isolation calibration data between the receiver port <rcvport> and the source port <srcport>.

---



To complete the calibration procedure, click **Apply**.

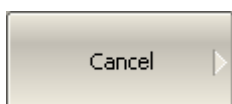
This will activate the process of calibration coefficient table calculation and saving it into the memory. The error correction function will also be automatically enabled.

---

[SENS:CORR:COLL:SAVE](#)

Calculates the calibration coefficients from the calibration standards measurements depending on the selected calibration type.

---



To clear the measurement results of the standard, click Cancel.

This softkey does not cancel the current calibration. To disable the current calibration, turn off the error correction function (See [Error Correction Disabling](#)).

---

[SENS:CORR:COLL:CLE](#)

Clears the measurement data of the calibration standards.

---

---

#### NOTE

The calibration status can be checked in channel status bar (See [General error correction status table](#)) or in trace status field (See [Trace error correction status table](#)).

---



## Full Two-Port Calibration

A full two-port calibration (SOLT) involves seven connections of standards. This calibration combines two one-port calibrations for each test port with measurement of a THRU standard in both directions (See figure below). An optional isolation calibration can be performed by measurement of two LOAD standards connected to both test ports of the Analyzer.

---

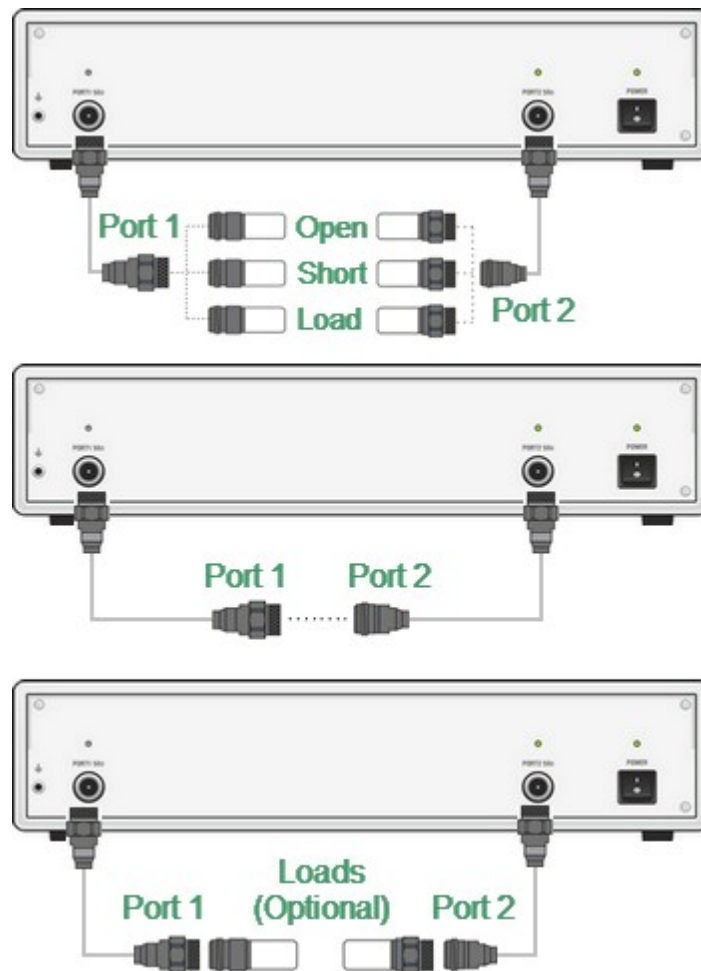
### NOTE

For isolation calibration, set a narrow IF bandwidth and firmly attach the cables.

---

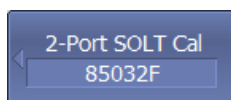
Full two-port calibration allows for correction of all the twelve error terms of a [two-port error model](#): **Ed1, Ed2, Es1, Es2, Er1, Er2, Et1, Et2, EI1, EI2, Ex1, Ex2** (correction of **Ex1, Ex2** can be omitted).

Full two-port calibration is a highly accurate method of calibration for two-port DUT measurements.



Full two-port calibration

Before starting calibration perform the following settings: select active channel, set the parameters of the channel (frequency range, IF bandwidth, etc.), and select the calibration kit.

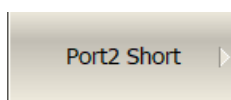
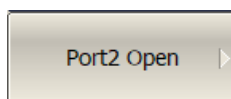
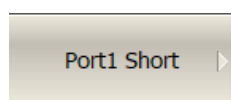
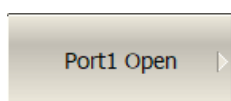


To open full two-port calibration submenu, use the following softkeys:

**Calibration > Calibrate > 2-Port SOLT Cal**

[SENS:CORR:COLL:METH:SOLT2](#)

Selects the ports and sets the full two-port calibration type for the calculation of the calibration coefficients on completion of the calibration executed by the [SENS:CORR:COLL:SAVE](#) command.



Connect SHORT, OPEN, and LOAD standards to the 1 and 2 ports in any consequence, as shown in the above figure. Perform measurements clicking the softkey **Port n Open**, **Port n Short**, **Port n Load** corresponding to the connected standard.

The instrument status bar will indicate **Calibration in progress...** when the measurement is in progress. On completion of the measurement a check mark will appear in the left part of the softkey.

[SENS:CORR:COLL:OPEN](#)

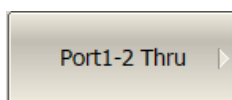
Measures the calibration data of the open standard for the specified port.

[SENS:CORR:COLL:SHOR](#)

Measures the calibration data of the short standard for the specified port.

[SENS:CORR:COLL:LOAD](#)

Measures the calibration data of the load standard for the specified port.



Connect a THRU standard between the test ports. If the port connectors allow through connection connect them directly (zero electrical length thru). Perform measurement using the **Port 1–2 Thru** softkeys.

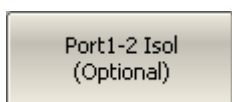
The instrument status bar will indicate **Calibration in progress...** when the measurement is in progress. On completion of the measurement a check mark will appear in the left part of the softkey.

---

[SENS:CORR:COLL:THRU](#)

Measures the calibration data of the thru standard between the receiver port <rcvport> and the source port <srcport>.

---



To perform the optional isolation calibration, connect two LOAD standards to the test ports as shown in the above figure and enable measurement using the **Port 1–2 Isol (Optional)** softkeys.

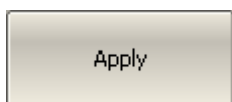
The instrument status bar will indicate **Calibration in progress...** when the measurement is in progress. On completion of the measurement a check mark will appear in the left part of the softkey.

---

[SENS:CORR:COLL:ISOL](#)

Measures the isolation calibration data between the receiver port <rcvport> and the source port <srcport>.

---



To complete the calibration procedure, click **Apply**.

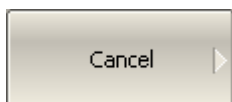
This will activate the process of calibration coefficient table calculation and saving it into the memory. The error correction function will also be automatically enabled.

---

[SENS:CORR:COLL:SAVE](#)

Calculates the calibration coefficients from the calibration standards measurements depending on the selected calibration type.

---



To clear the measurement results of the standard, click **Cancel**.

This softkey does not cancel the current calibration. To disable the current calibration, turn off the error correction function (See [Error Correction Disabling](#)).

---

[SENS:CORR:COLL:CLE](#)

Clears the measurement data of the calibration standards.

---

---

**NOTE**

The calibration status can be checked in channel status bar (See [General error correction status table](#)) or in trace status field (See [Trace error correction status table](#)).

---

## Sliding Load Calibration

In full one-port and full two-port calibrations, it is possible to employ a SLIDING LOAD calibration standard instead of a fixed one. The use of the SLIDING LOAD standard allows for significant increase in calibration accuracy at high frequencies compared to the FIXED LOAD standard.

The SLIDING LOAD calibration involves a series of measurements in different positions of the sliding element to compensate for reflection from the dissipation component.



Sliding Load Calibration

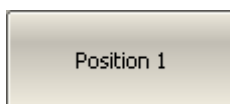
To activate the SLIDING LOAD calibration algorithm, the selected calibration kit should contain a calibration standard of SLIDING LOAD type, and it should be assigned to the "Load" class of the corresponding port. Calibration standard editing and class assignment are further described in detail in [Calibration Standard Definition](#).

If a calibration kit contains a SLIDING LOAD, the menu selection for the load will lead to a submenu for selection of the various sliding load positions.

The SLIDING LOAD calibration involves a series of measurements in different positions of the sliding element. The minimum number of measurements is 5, the maximum number of measurements is 8.



In the main menu of one-port or two-port calibration the Load softkey will open the SLIDING LOAD menu (if the above-mentioned condition is met).



Connect the SLIDING LOAD to a selected test ports and perform a series of measurements in different positions of the sliding element clicking the **Position 1**, **Position 2** ... **Position 8** softkeys.

...

---

Position 8

---

---

**NOTE**

The SLIDING LOAD calibration is not suitable for low frequencies. To eliminate this limitation, use a FIXED LOAD standard in the lower part of the frequency range. For combined calibration with SLIDING and FIXED LOADS, use the procedure of standard subclasses assigning (See [Sliding Load Calibration Example Using Subclasses](#)).

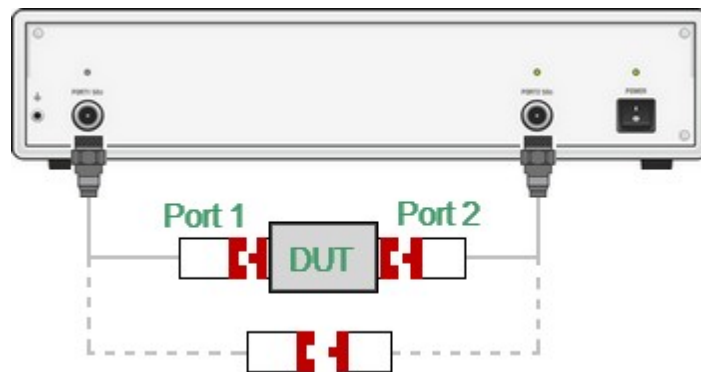
---



## Non-Insertable Device Measuring

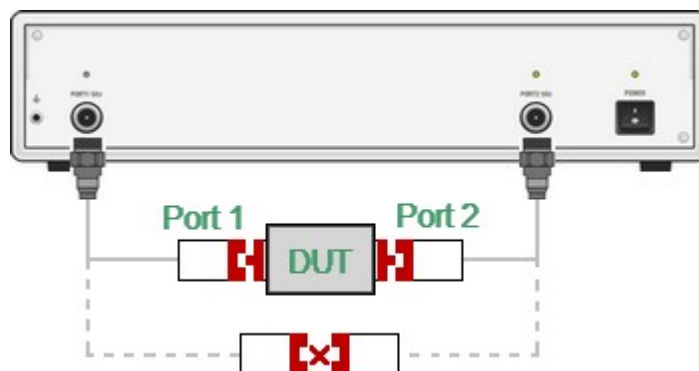
The two-port SOLT calibration procedure includes direct connection of test port cables with each other. Such connection is called Zero length THRU or Flush THRU and means that THRU has zero electrical length. However, it is not always possible to connect test port cables directly to each other. According to this criterion, DUTs are divided into insertable and non-insertable devices:

- An insertable device is one whose connectors could match together. They have the same type of connector and opposite or no gender. Test port cables can be matched together (See figures below), therefore a two-port SOLT calibration can be performed for such a measurement setup.



Insertable Device

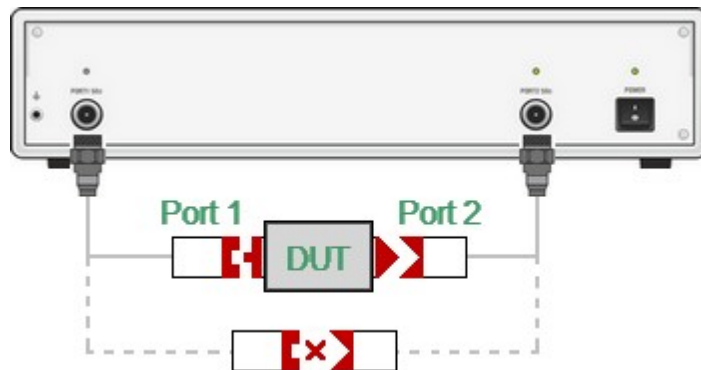
- A non-insertable device is one whose connectors could not match together. This also means that the test port cables would not match each other. In the simplest case, a non-insertable device has connectors of the same type, for example, N50, and of the same gender (See figures below).



Non-insertable Device (connectors of the same type)

In practice, there are often more complex cases of non-insertable device measurements — devices having ports of different types and/or having different

characteristic impedances, for example, N50 – 3.5, N50 – N75, N50 – Waveguide (See figures below).



Non-insertable Device (connectors of different types)

The following calibration methods are used for a non-insertable device:

- DEFINED THRU
- UNKNOWN THRU calibration
- UNKNOWN THRU Addition function
- Adapter Removal/Insertion

### Defined Thru

This method uses physical (not null) DEFINED THRU in two-port SOLT calibration (See [Full Two-Port Calibration](#)). Parameters of the DEFINED THRU must be defined in the calibration kit. If the definition of DEFINED THRU is not included in the calibration kit, it must be added manually using one of two methods: parameters of calibration standard model or S-parameters. For more details see [Calibration Standard Definition](#).

### Unknown Thru calibration

This method uses physical (not null) UNKNOWN THRU in two-port calibration. Such calibration is called SOLR (Short-Open-Load-Reciprocal).

Any two-port network satisfying the reciprocal condition ( $S_{12} = S_{21}$ ) can be used as UNKNOWN THRU. Generally speaking, most passive, linear microwave networks will turn out to be reciprocal. Combined transmission loss of UNKNOWN THRU and calibration setup should not exceed 40 dB. The UNKNOWN THRU can be a wide class of two-port network, including DUT, if it meets the specified conditions.

The only parameter of UNKNOWN THRU, which should be known in advance is approximate electrical delay. In most cases, there is no need to enter its value

manually, since the Analyzer has the function of automatic detection of the UNKNOWN THRU electrical delay. For more details see [Unknown Thru Requirements](#).

This method is applicable when both test ports can be calibrated using the same calibration kit. For example, test ports of the same type, and of the same gender. For more details see [Unknown Thru Calibration](#). For this purpose, the software includes an UNKNOWN THRU standard in the description of each predefined calibration kit.

### **Unknown Thru Addition function**

The main difference between this method and Unknown Thru calibration is that the calibration is carried out in two steps. A one-port calibration of each port must be performed in advance using a mechanical calibration kit (See [Full One-Port Calibration](#)) or ACM. Then the Unknown Thru Addition function measures UNKNOWN THRU and completes the two-port calibration. Since it is possible to select an individual calibration kit for each one-port calibration, the test ports can be of different types, up to a combination of coaxial and waveguide types. For more details see [Unknown Thru Addition](#).

### **Adapter Removal/Insertion function**

Adapter Removal function used to remove any adapter characteristics from the calibration plane.

Adapter Insertion function used to insert any adapter characteristics to the calibration plane.

The initial calibration plane is established by two-port SOLT calibration with Zero-length THRU. Then both functions use an additional measurement of the three standards (Short-Open-Load) to mathematically remove or insert the adapter.

Requirements for the adapter in the Adapter Removal/Insertion function are the same as for UNKNOWN THRU. For a typical adapter transition between different types of connectors, these requirements are easily met. For more details see [Adapter Removal/Insertion](#).

### **Accuracy of Methods**

- Unknown Thru calibration (SOLR) is potentially most accurate method and is preferable method for non-insertable device measurement.
- Unknown Thru addition accuracy is comparable to SOLR.
- The Adapter Removal/Insertion method is less accurate than Unknown Thru methods as it requires more standard connections (10 connections compared to 7 connections in SOLR).

- Defined Thru is usually more accurate than Adapter Removal, but not as accurate as Unknown Thru method.

## Unknown Thru Requirements

An arbitrary two-port device with unknown parameters can be used as an UNKNOWN THRU in 2-port SOLR (Short-Open-Load-Reciprocal) calibration. An UNKNOWN THRU should satisfy next requirements:

- The UNKNOWN THRU must be Reciprocal ( $S_{21} = S_{12}$ ), which holds for most passive linear network.
- The combined transmission loss of the UNKNOWN THRU and calibration path is not recommended to exceed 40 dB.
- The approximate electrical delay of the UNKNOWN THRU should be specified manually or set to zero in order the analyzer to detect it automatically.

### Requirements for automatic detection of electrical delay

It is usually not necessary to enter the electrical delay manually because the analyzer can automatically detect the electrical delay of the UNKNOWN THRU during the calibration procedure. In this case, the UNKNOWN THRU value should be set to zero.

For the Analyzer to correctly automatically detect the UNKNOWN THRU delay, the following condition must be met:

$$\frac{Span}{N - 1} < \frac{1}{2 \cdot \tau_0},$$

where  $Span$  — frequency span of calibration,

$N$  — number of points,

$\tau_0$  — delay of a UNKNOWN THRU,

In other words the number of points must be enough to correctly auto-detect the electrical delay of the UNKNOWN THRU:

$$N > 2 \cdot \tau_0 \cdot Span + 1$$

**Example.** An example of calculating the number of points sufficient for the correct automatic determination of UNKNOWN THRU delay by the Analyzer.

Let the UNKNOWN THRU is a coaxial cable having approximate length  $l_0 \approx 100mm$ . The approximate delay of the cable will be  $\tau_0 \approx 477ps$ , providing the velocity factor of the cable equals  $1/\sqrt{\epsilon} \approx 0.7$ .

Let us the Span is 8 GHz.

For the analyzer to correctly automatically detect the UNKNOWN THRU delay, the number of points must be:

$$N > 2 \cdot 477 \cdot 10^{-12} \cdot 8 \cdot 10^9 + 1; N > 9.$$

### Manual set of Unknown Thru delay

It is possible to manually enter either the delay or physical length of the UNKNOWN THRU. The accuracy of the UNKNOWN THRU length must be known within of 1/2 of the wavelength in the Thru media at the maximum calibration frequency. Accordingly, the accuracy of the UNKNOWN THRU delay must be known within of

$$\frac{1}{2 \cdot F_{stop}},$$

where  $F_{stop}$  — stop frequency of calibration.

When length specified the value of the Thru media permittivity must be also specified.

If the Thru media type is waveguide, the cutoff frequency must be specified.

## Unknown Thru Calibration

---

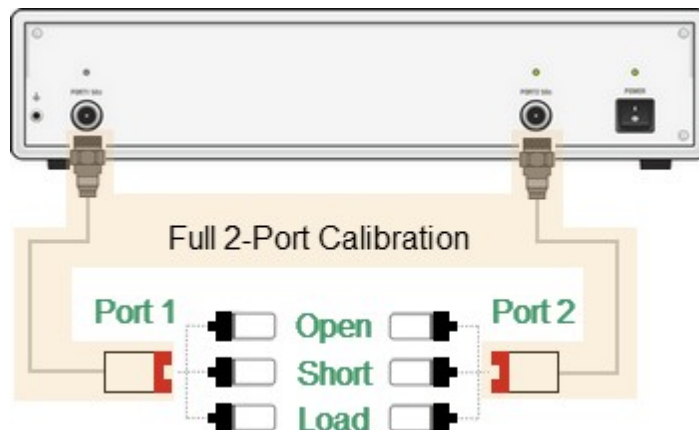
### NOTE

The availability of the Unknown Thru calibration depends on the Analyzer model (See corresponding [datasheet](#)).

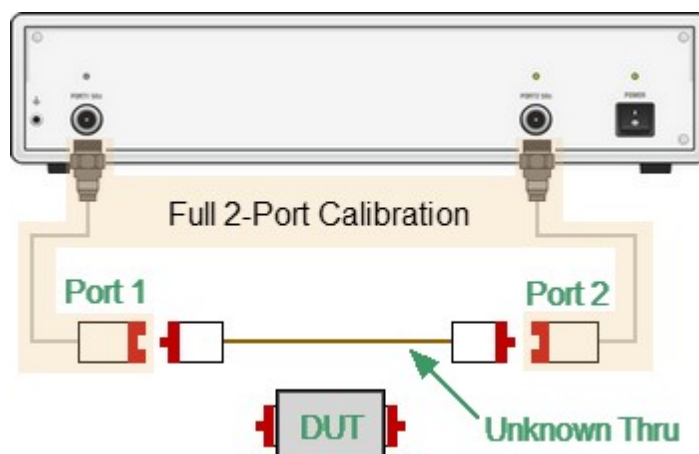
---

Unknown Thru Calibration or SOLR (Short-Open-Load-Reciprocal) is analogous to SOLT calibration, where UNKNOWN THRU is used instead of DEFINED THRU (See figures below).

For this purpose, the program includes a UNKNOWN THRU standard in the description of each predefined calibration kit. This method is used when connecting DUT with connectors of the same type, and of the same gender, when one calibration kit can be used to calibrate both ports.



First Stage of SOLR Calibration



Second Stage of SOLR Calibration

By default in all predefined calibration kits, the UNKNOWN THRU delay value is set to zero. In this case UNKNOWN THRU delay value is detected by the Analyzer automatically. In some cases, it is required to enter the UNKNOWN THRU value manually (See [Unknown Thru Requirements](#)). To do this, use the [Calibration Kits Editor](#).

---

**NOTE**

If different types of connectors are used to connect the DUT, then a single calibration kit cannot be used for two-port SOLT calibration. In this case, it is necessary to create a description of the user calibration kit, composed of standards suitable for both ports. It is more convenient to use the [Unknown Thru Addition](#) method instead.

---

The SOLR calibration procedure is similar to the SOLT calibration procedure, described in the section [Full Two-Port Calibration](#).

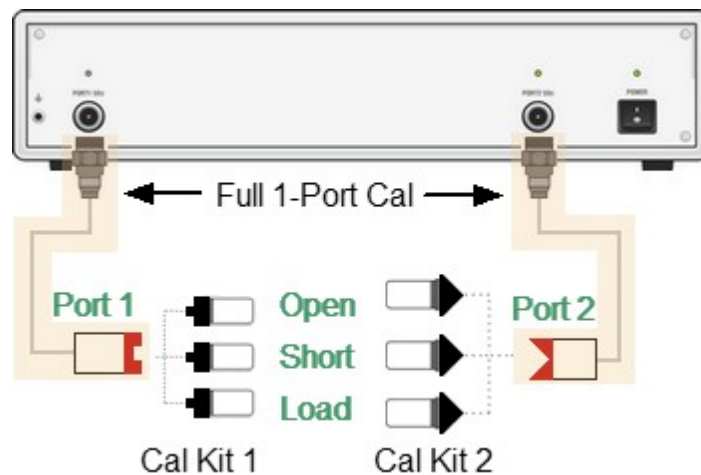


## Unknown Thru Addition

Unknown Thru Addition method is used mainly in DUT connecting with connectors of various types. In this case, one predefined calibration kit cannot be used for calibration of both ports, as in the Unknown Thru calibration method.

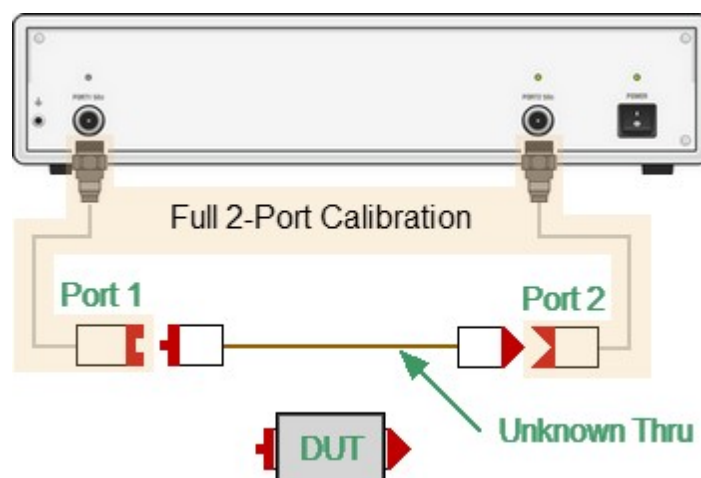
The difference between this method and Unknown Thru calibration is that the calibration is performed in two steps:

- One-port calibration of each port must be performed in advance using a mechanical calibration kit (See [Full One-Port Calibration](#)) or ACM. Since a suitable calibration kit can be selected for each one-port calibration, the test ports can be of different types, up to a combination of coaxial type and waveguide type ports.



One-port calibrations before the Thru Addition method

- The Unknown Thru Addition function measures UNKNOWN THRU and completes the two-port calibration.



Unknown Thru Addition method

To add an UNKNOWN THRU, proceed as follows:

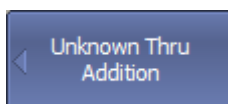
- First, select in the software the calibration kit used for the port to be calibrated. Perform full one-port calibration for each port. For more details about this procedure see [Full One-port Calibration](#).
- Go to the Unknown Thru Addition submenu. Set the delay (length) of UNKNOWN THRU, or leave it at zero value for automatic detection. If using length instead of delay, also enter permittivity. If using waveguide THRU, also set the Cutoff Frequency.
- Connect the ports directly using an appropriate UNKNOWN THRU and perform the measurements. Eventually, the full two-port calibration coefficients will be computed.

---

#### NOTE

The Unknown Thru Addition function is not accessible if the one-port calibration is interpolated or extrapolated. The status of the one-port calibration must be **[Cor]**, not **[C?]** or **[C!]**.

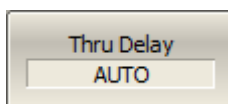
---



To open the Unknown Thru Addition submenu, use the following softkeys:

**Calibration > Calibrate > Unknown Thru Addition**

---



Enter the THRU delay or length or set 0 for AUTO using **Thru Delay**.

---

[SENS:CORR:COLL:THRU:ADD:DEL](#)

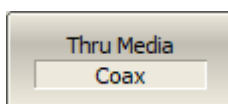
Sets or reads out the approximate delay value of an unknown thru in the thru addition function.

---

[SENS:CORR:COLL:THRU:ADD:LENG](#)

Sets or reads out the approximate value of the mechanical length of an unknown thru in the thru addition function.

---

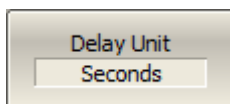


Select the THRU media: Coax or Waveguide using **Thru Media**.

---

[SENS:CORR:COLL:THRU:ADD:MED](#)

Specifies the media of the thru in the thru addition function.

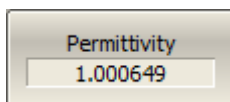


Select the desired measurement units for Delay (Length): Seconds or Meters using **Delay Unit**.

---

[SENS:CORR:COLL:THRU:ADD:UNIT](#)

Selects the display units of the thru delay (length) in the thru addition function.

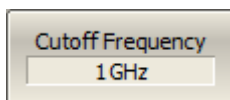


When the measurement units Meters are selected, enter the Permittivity value using **Permittivity**.

---

[SENS:CORR:COLL:THRU:ADD:PERM](#)

Sets or reads out the value of the permittivity of the thru media in the thru addition function.

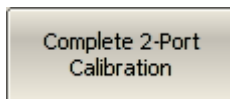


When the adapter media Waveguide is selected, enter the Cutoff Frequency value using **Cutoff Frequency**.

---

[SENS:CORR:COLL:THRU:ADD:WAV:CUT](#)

Sets or reads out the value of the cutoff frequency of the waveguide thru in the thru addition function.



To complete the Unknown Thru Addition procedure, click **Complete 2-Port Calibration**.

---

[SENS:CORR:COLL:THRU:ADD:FULL2:COMP](#)

Completes the full two-port calibration between the specified ports provided that each port was calibrated using full one-port calibration.

---

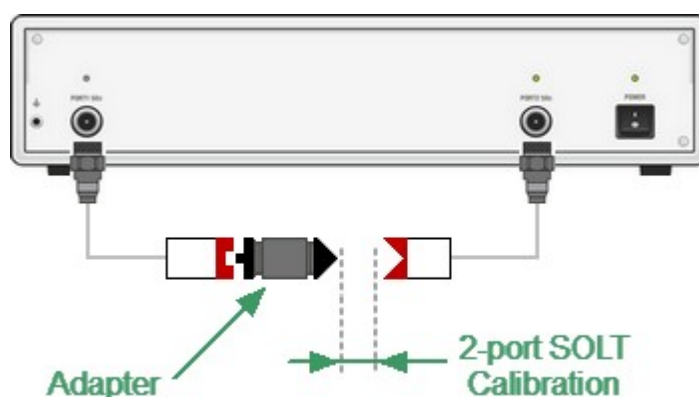
## Adapter Removal/Insertion

Adapter Removal and Adapter Insertion functions are designed to mathematically exclude adapter characteristics from the calibration plane or add adapter characteristics to the calibration plane. They adapt two-port SOLT calibration with Zero-Lenght Thru for non-insertable device measurement. Functions are performed in two steps. The first step is performing two-port SOLT calibration, the second is measuring 3 standards Short-Open-Load to exclude/include adapter.

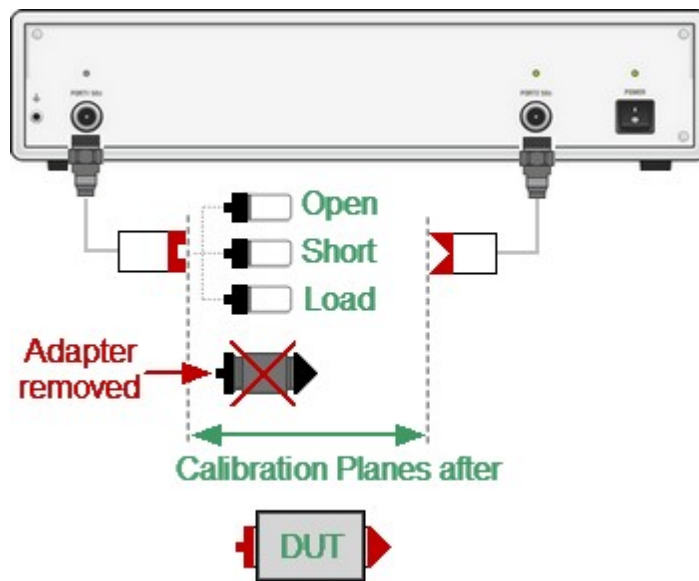
The parameters of the adapter do not need to be known. Requirements for the adapter are the same as for UNKNOWN THRU. For more details see [Unknown Thru Requirements](#). For a typical adapter-transition between different types of connectors, these requirements are easily met.

## Adapter Removal Function

Adapter Removal function used to remove any adapter characteristics from the calibration plane. The function is used when, for two-port SOLT calibration, the connection of zero length thru test port cables is only possible with an adapter. Adapter is added to measuring setup, two-port SOLT calibration is performed. After the calibration is complete, the adapter is removed from the measuring setup. Adapter characteristics are mathematically removed from the calibration plane using three Open-Short-Load standards (See figure below).



## First Stage of Adapter removal function



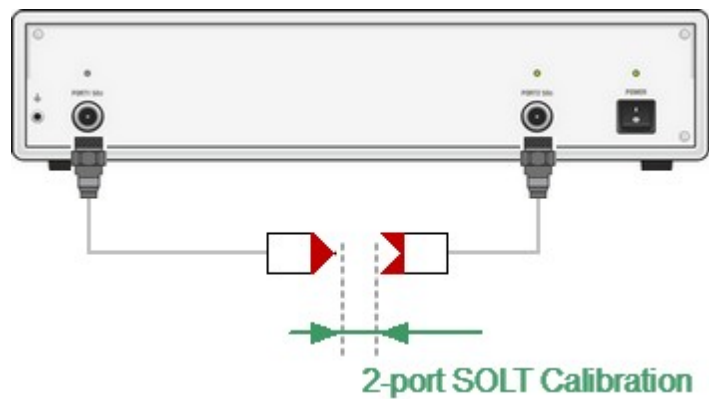
Second Stage of Adapter removal function

Adapter Removal procedure (See [Adapter removal](#) figure):

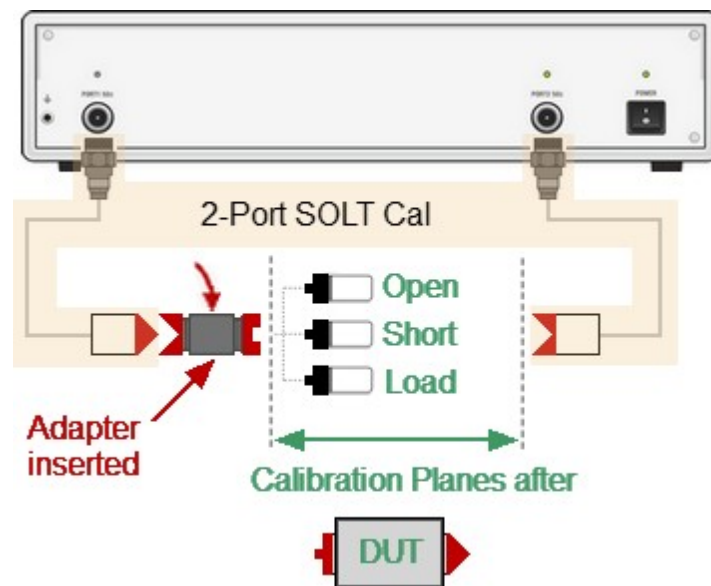
- Connect the adapter to the port.
- Perform two-port SOLT calibration.
- Remove the adapter.
- In Adapter Removal/Insertion submenu select port number to which the adapter was connected.
- Set Adapter Delay (Adapter Length), or leave it at zero to automatically detect it. If the Adapter type is waveguide, set its type and Cutoff Frequency.
- Measure three standards: OPEN, SHORT, and LOAD, for the corresponding test port.
- To complete the Adapter Remove procedure, click **Apply**.

### Adapter Insertion function

Adapter Insertion function used to insert any adapter characteristics to the calibration plane. The function is used when test port cables allow connection zero-length thru, but it is possible to connect DUT to them only with the use of an adapter. 2-port SOLT calibration is performed. After the calibration is complete, the adapter is added to the measuring setup. Adapter characteristics are mathematically added to the calibration plane using three Open-Short-Load standards (See figure below).



First Stage of Adapter insertion function



Second Stage of Adapter insertion

Adapter Insertion procedure (see [Adapter insertion](#) figure):

- Perform two-port SOLT calibration.
- Connect the adapter to the test port, which cannot be directly connected to the DUT.
- In Adapter Removal/Insertion function menu select the port number to which the adapter is connected.
- Set Adapter Delay (Adapter Length), or leave it at zero to automatically detect it. If the Adapter type is waveguide, set its type and Cutoff Frequency.
- Measure three standards: OPEN, SHORT, and LOAD, for the corresponding test port.

- To complete the Adapter Insert procedure, click **Apply**.

---

#### NOTE

The Adapter Removal/Insertion function is accessible when the status of the initial tow-port calibration is [Cor], not [C?] or [C!].

---

#### NOTE

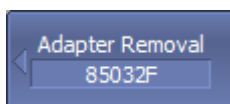
Before starting adapter removal, select the appropriate calibration kit.

---

#### NOTE

When test ports have different Z0, enable automatic Z0 selecting function (See [Automatic Z0 Selecting](#)).

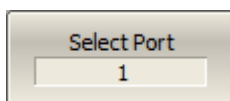
---



To open the Adapter Removal/Insertion submenu, use the following softkeys:

**Calibration > Calibrate > Adapter Removal**

---



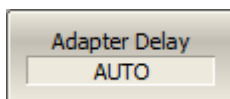
Select the port number for Adapter Removal/Insertion using **Select Port**.

---

[SENS:CORR:COLL:METH:ADAP:REM](#)

Selects the port number and sets the adapter removal/insertion function for the calculation of the calibration coefficients when the [SENS:CORR:COLL:SAVE](#) command has been executed.

---



Enter the adapter delay or length or set 0 for AUTO using **Adapter Delay**.

---

[SENS:CORR:COLL:ADAP:DEL](#)

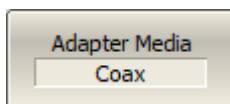
Sets or reads out the approximate delay value of an adapter in the adapter removal/insertion function.

---

[SENS:CORR:COLL:ADAP:LENG](#)

Sets or reads out the approximate value of the mechanical length of the adapter in the adapter removal/insertion function.

---

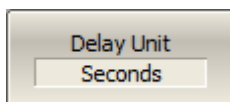


Select the Adapter media: coax or waveguide using **Adapter Media**.

---

[SENS:CORR:COLL:ADAP:MED](#)

Specifies the adapter media in the adapter removal/insertion function.

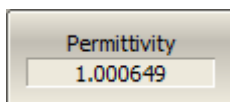


Select the desired measurement units for delay (length): seconds or meters using **Delay Unit**.

---

[SENS:CORR:COLL:ADAP:UNIT](#)

Selects the display units of the adapter delay (length) in the adapter removal/insertion function.

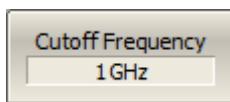


When the measurement units **Meters** are selected, enter the permittivity value using **Permittivity**.

---

[SENS:CORR:COLL:ADAP:PERM](#)

Sets or reads out the value of the permittivity of an adapter media in the adapter removal/insertion function.

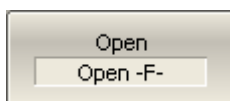


When the adapter media Waveguide is selected, enter the Cutoff Frequency value using **Cutoff Frequency**.

---

[SENS:CORR:COLL:ADAP:WAV:CUT](#)

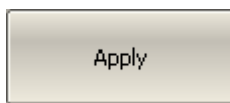
Sets or reads out the value of the cutoff frequency of the waveguide adapter.



Connect SHORT, OPEN, and LOAD standards to the selected port in any consequence as shown in [Full One-Port Calibration](#). Perform measurements clicking the softkey corresponding to the connected standard.



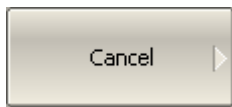
The instrument status bar will indicate **Calibration in progress...** when the measurement is in progress. On completion of the measurement a check mark will appear in the left part of the softkey.



To complete the Adapter Remove/Insert procedure, click **Apply**.

---





To clear the measurement results of the standard, click **Cancel**.

This softkey does not cancel the current calibration. To disable the current calibration, turn off the error correction function (See [Error Correction Disabling](#)).

---

## Two-Port TRL Calibration

---

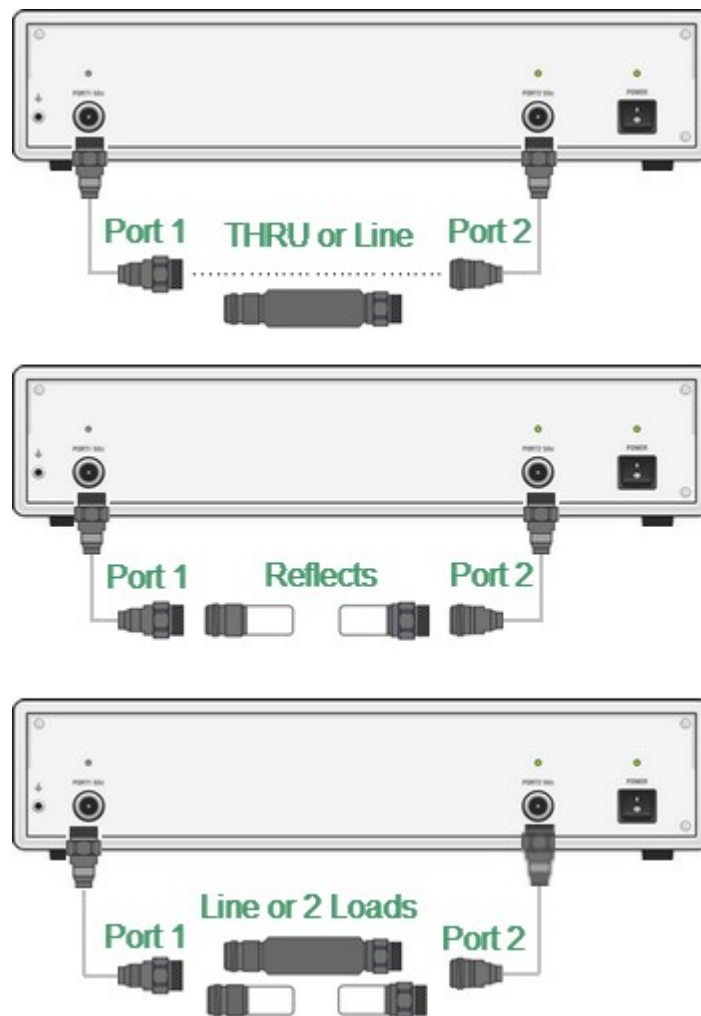
### NOTE

The availability of the TRL calibration depends on the Analyzer model (See corresponding [datasheet](#)).

---

TRL (Thru-Reflect-Line) calibration is the most accurate calibration method described herein, as it uses airlines as calibration standards. The TRL calibration requires the use of the following calibration standards (See figure below):

- THRU or REFERENCE LINE
- REFLECT (SHORT or OPEN)
- second LINE or two MATCHes



TRL Calibration

TRL is a general name for a calibration family, which comprises such calibrations as LRL, TRM, or LRM — named depending on the calibration standards used.

If a zero-length THRU is used as the first standard, the method is called TRL calibration. If a non-zero length LINE is used as the first standard, the calibration method is called LRL (Line-Reflect-Line). To denote the first standard of the TRL and LRL calibration, assign TRL-Thru class, which includes THRU and LINES. A LINE of TRL-Thru class is also called Reference Line.

An SHORT is usually used as a second standard in TRL calibration. To denote the second standard of the TRL calibration, assign TRL-Reflect class.

A second LINE is used as the third standard in TRL calibration. At low frequencies, two MATCHes is used instead of LINE, as they are an equivalent of a matched line of infinite length. In the latter case, the calibration method is called TRM (Thru-Reflect-Match) or LRM (Line-Reflect-Match) respectively. To denote the third standard of the TRL calibration, assign TRL-line/match class, which includes LINES and MATCHes.

### **Frequency Range**

TRL and LRL calibrations have a limited bandwidth, suitable for lower to upper frequency ratios up to 1:8. The band limits depend on the LINE length in TRL calibration or on the difference between the lengths of the two LINES in LRL calibration.

In theory, TRM and LRM calibrations do not have limitations in frequency; however, their practical use at higher frequencies is limited by the quality of the MATCHes. It is recommended to use the TRM and LRM calibrations up to 1 GHz.

### **Impedance of LINES and MATCHes**

All the LINES and MATCHes used for TRL calibration must have  $Z_0$  impedance values as precise as possible. TRL calibration transfers the impedance of standards into the calibrated system. Precise airlines with an accurate  $Z_0$  impedance of 50  $\Omega$  are used as LINES in coaxial paths.

### **REFERENCE LINE**

A zero-length THRU is used as the first standard in TRL calibration. In LRL calibration a LINE, which is called REFERENCE LINE, is used instead of a zero-length THRU. The shortest LINE is used as the REFERENCE LINE. Its length must to be known, so that the calibration plane positions could be calculated exactly. However, LRL calibration is also possible when the REFERENCE LINE length is not known. In this case, its length is assumed to be equal to zero, the calibration plane being in the middle of the LINE, and not at the ports' edges.

## TRL LINE

TRL LINE is an airline used in TRL calibration, or the second longest LINE used in LRL calibration. The length of TRL LINE should be known just approximately. The LINE length is used to determine the calibration bandwidth. Let  $\Delta L$  be the difference between the two LINES in LRL calibration. In TRL calibration this difference will be equal to the LINE length, as a zero-length THRU is used as a REFERENCE LINE. Then the phase difference between the TRL LINE and REFERENCE LINE or THRU should be no less than  $20^\circ$  at the lower frequency and no more than  $160^\circ$  at the upper frequency of the calibration.

$$20 < \frac{360 \cdot f \cdot \Delta L}{v} < 160,$$

where  $\Delta L = L_1 - L_0$ ,

$v$  — wave velocity in LINE (for airline it is  $c = 2.9979 \cdot 10^8$  m/sec),

$L_0$  — REFERENCE LINE length,

$L_1$  — TRL LINE length.

So, the useful frequency range for TRL/LRL calibration is 1:8. Two or more TRL LINES are used to extend the calibration frequency. For example, when using two TRL LINES, the frequency range can be increased up to 1:64. Besides, TRL/LRL calibration does not work at low frequencies, as it would require a very long LINE.

## TRL MATCH

Unlike TRL/LRL calibration, TRM/LRM calibration uses MATCHes, which are the equivalent to the infinitely long LINE, instead of a TRL LINE. Theoretically TRM/LRM calibration has no frequency limitations. However, the use of TRM/LRM calibration at higher frequencies is limited by the quality of the MATCHes. As a rule, the TRM/LRM calibration is used at lower frequencies, as it is good starting from zero frequency.

## TRL REFLECT

There are no strict requirements to the TRL REFLECT standard. Only approximate parameters of the TRL REFLECT standard should be known. The REFLECT standard should have high reflection coefficient, close to 1. The phase of the standard must be known within  $\pm 90^\circ$ . Normally, any SHORT meets this requirement. The next requirement is that the reflection coefficient must be the same for all the ports. If one standard is used for all the ports by turns, then this requirement is automatically fulfilled. If the ports have different genders or types of connectors, use special standards with the identical electrical specifications, which are available in pairs.

## TRL Calibration Frequency Extension

To extend the frequency of TRL calibration a method of dividing into several non-overlapping bands is applied. For each frequency band a separate TRL LINE of different length is used. The phase difference between each TRL LINE and the REFERENCE LINE must be from 20° to 160°, as indicated above. A MATCH standard is used in the lowest frequency band.

The Analyzer software allows up to 8 LINES to be used for calibration frequency extension. To achieve this, there are two steps of handling the calibration kits:

- Defining frequency limits to calibration standards (See [Calibration Standard Definition](#)).
- Assigning classes to calibration standards, where up to 8 calibration standards can be assigned to one class (See [Calibration Standard Class Assignment](#)).

Perform the above mentioned dividing of the calibration band into sub-bands and assign a separate TRL LINE to each of them in the calibration kit editing menu before calibration.

Before starting calibration, perform the following settings: select active channel, set the parameters of the channel (frequency range, IF bandwidth, etc.), select the calibration kit.



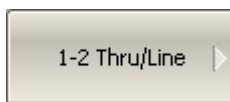
To open TRL calibration submenu, use the following softkeys:

**Calibration > Calibrate > 2-Port TRL Cal**

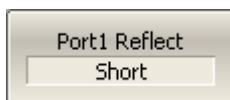
---

[SENS:CORR:COLL:METH:TRL2](#)

Selects the ports and sets the 2-port TRL calibration type for the calculation of the calibration coefficients on completion of the calibration executed by the [SENS:CORR:COLL:SAVE](#) command.



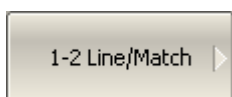
Connect a TRL THRU (THRU or LINE) standard between the test ports. Perform measurement using the **1-2 Thru/Line** softkey.



Connect a TRL REFLECT standard to the test ports in any order. Perform measurement using **Port 1 Reflect** and the **Port 2 Reflect** softkey.



Connect a TRL LINE/MATCH (LINE between the test ports and 2 LOADs to each port). Perform measurement using the **Port 1-2 Line/Match** softkey.



The instrument status bar will indicate **Calibration in progress...** when the measurement is in progress. On completion of the measurement a check mark will appear in the left part of the softkey.

---

[SENS:CORR:COLL:TRLT](#)

Measures the calibration data of the TRL thru standard between <port1> and <port2>.

---

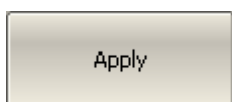
[SENS:CORR:COLL:TRLR](#)

Measures the calibration data of the TRL reflect standard for the specified port.

---

[SENS:CORR:COLL:TRL](#)

Measures the calibration data of the TRL line standard between <port1> and <port2>.



To complete the calibration procedure, click **Apply**.

This will activate the process of calibration coefficient table calculation and saving it into the memory. The error correction function will also be automatically enabled.

---

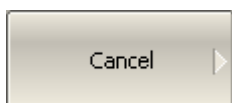
[SENS:CORR:COLL:SAVE](#)

Calculates the calibration coefficients from the calibration standards measurements depending on the selected calibration type.

---

#### NOTE

System correction will turn automatically off when **Apply** softkey is pressed to perform TRL calibration (See [System Correction Setting](#)).



To clear the measurement results of the standard, click Cancel.

This softkey does not cancel the current calibration. To disable the current calibration, turn off the error correction function (See [Error Correction Disabling](#)).

---

[SENS:CORR:COLL:CLE](#)

Clears the measurement data of the calibration standards.

---

---

**NOTE**

The calibration status can be checked in channel status bar (See [General error correction status table](#)) or in trace status field (See [Trace error correction status table](#)).

---

## Multiline TRL Calibration

---

### NOTE

The availability of the Multiline TRL calibration depends on the Aalyzer model (See corresponding [datasheet](#)).

---

Regular TRL calibration, described in the section [TRL Calibration](#) uses several LINES of different lengths for frequency extension. It is provided by the method of dividing the frequency band into separate sub-bands.

Multiline TRL calibration also uses several LINES, but it does not divide the frequency band into several sub-bands. Instead, all the LINES are used simultaneously over the whole calibration bandwidth. The redundancy of the LINES measurements allows for both extending the frequency range and increasing the calibration accuracy. The number of LINES should be no less than three. The more LINES are used, the higher the accuracy will be achieved.

To employ multiple LINES in the calibration procedure, use the same method of standards subclasses assignment as in the regular TRL calibration (See [Calibration Standard Class Assignment](#)). Defining frequency limits to calibration standards is not necessary for Multiline TRL calibration method.

The following table shows the differences between the regular and Multiline TRL calibrations when entering the data into the calibration standards editing menu.

Calibration Standard	Data in Calibration Kit Manager	
	TRL	Multiline TRL
REFERENCE LINE or THRU	1. Type: THRU/LINE 2. Min and max frequency 3. Delay <sup>1</sup> 4. Class: TRL THRU	1. Type: THRU/LINE 2. Delay <sup>1</sup> 3. Class: TRL LINE/MATCH or TRL THRU
LINE	1. Type: THRU/LINE 2. Min and max frequency 3. Class: TRL LINE/MATCH	The total number of LINES is no less than 3.



Calibration Standard	Data in Calibration Kit Manager	
	TRL	Multiline TRL
MATCH (optional)	1. Type: MATCH  2. Min and max frequency  3. Class: TRL LINE/MATCH	1. Type: MATCH  2. Class: TRL LINE/MATCH
REFLECT	1. Type: SHORT or OPEN.  2. Min and max frequency.  3. Model parameters, which allow calculating value of phase response within ±90°.  4. Class: TRL REFLECT.	

<sup>1</sup> The delay for coaxial airlines is equal to  $L/v$  , where L is the length of the LINE, and v is the wave velocity in LINE equal to 2.9979·108 m/s.

If a calibration kit for Multiline TRL has been created and edited, it is possible to switch between normal and multiline TRL calibrations a specific button in the TRL calibration menu, shown below.



To toggle between normal and Multiline TRL calibrations, use the **Multiline** softkey.



[SENS:CORR:COLL:METH:TRL:MULT](#)

Turns the multi-line TRL option ON/OFF.

## Bandsplit Calibration Using Subclasses

If the required frequency range of the calibration exceeds the operating frequency ranges of some calibration standards, use several standards to have the whole required frequency range covered. A calibration kit should contain several standards of each class (e.g. TRL LINE) with some specified frequency limits. Each of these standards will be applied for the measurements within its frequency limits. The total frequency band of all the standards should cover all the required frequency range of the calibration without “gaps”.

When several calibration standards of one class are used for calibration, subclasses should be assigned to these standards using the calibration kit editing function. The procedure of subclass assignment is described in [Calibration Standard Class Assignment](#).

---

### NOTE

When assigning two or more subclasses to one class of calibration standards the calibration menu changes: the standard measurement softkey is replaced by the softkey, which opens the subclass menu containing the list of all the standards of this class.

---

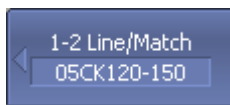
Examples of using this method are given in the sections [TRL Calibration Example Using Subclasses](#) and [Sliding Load Calibration Example Using Subclasses](#).

## TRL Calibration Example Using Subclasses

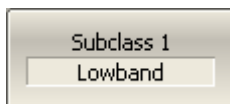
### NOTE

The availability of the TRL calibration depends on the Analyzer model (See corresponding [datasheet](#)).

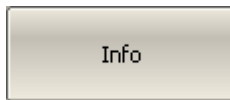
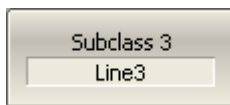
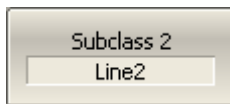
Here is an example of calibration using the calibration kit for TRL calibration, in which the "TRL LINE/MATCH" class contains 3 subclasses: load (Lowband), line 2 (TRL Line2), and line 3 (TRL Line3).



In the main menu of TRL calibration the **1-2 Line/Match** softkey will open the subclass menu (if the above mentioned condition is met).



Connect the Lowband, Line2 and Line3 to the test ports in any consequence and perform measurements clicking the softkey corresponding to the connected standard.



If two standards have an overlapping frequency range, the last measured standard will be used in the overlapping region.

To view additional information about each standard frequency range, in which its measurements are applied (See figure below), press the **Info** softkey.

1-2 Line/Match	Used in Calculations	STD Label	Applied to Range	
			Fmin	Fmax
Subclass 1	<input checked="" type="checkbox"/>	Lowband	300 kHz	680.2745 MHz
Subclass 2	<input checked="" type="checkbox"/>	Line2	3.600165 GHz	8 GHz
Subclass 3	<input checked="" type="checkbox"/>	Line3	720.273 MHz	3.5601665 GHz
Subclass 4	<input type="checkbox"/>			
Subclass 5	<input type="checkbox"/>			
Subclass 6	<input type="checkbox"/>			
Subclass 7	<input type="checkbox"/>			

Measurement: Ready

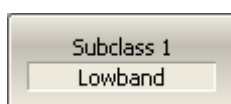
Information on calibration standard measurements

## Sliding Load Calibration Example Using Subclasses

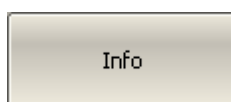
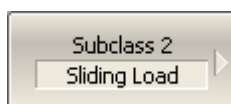
Here is an example of calibration using the calibration kit 85054B, in which the "Load" class contains 3 subclasses: fixed low-frequency load (Lowband), sliding load (Sliding), and fixed broadband load (Broadband). Only first two standards are used for calibration.



In the main calibration menu the **Load** softkey will open the subclass menu (if the above mentioned condition is met).



Connect Lowband and Sliding Load standards to the 1 port in any consequence and perform measurements clicking the softkey corresponding to the connected standard. To measure the Lowband, press the **Lowband** softkey, and to measure the Sliding Load, press the **Sliding Load** softkey. The procedure of sliding load measurement is described in detail in [Sliding Load Calibration](#).



If two standards have an overlapping frequency range, the last measured standard will be used in the overlapping region.

To view additional information about each standard frequency range, in which its measurements are applied (See figure below), press the **Info** softkey.

Port 1 Load	Used in Calculations	STD Label	Applied to Range	
			Fmin	Fmax
Subclass 1	<input checked="" type="checkbox"/>	Lowband	300 kHz	1.984114 GHz
Subclass 2	<input checked="" type="checkbox"/>	Sliding Load	2.0001125 GHz	3.2 GHz
Subclass 3	<input type="checkbox"/>	Broadband		
Subclass 4	<input type="checkbox"/>			
Subclass 5	<input type="checkbox"/>			
Subclass 6	<input type="checkbox"/>			
Subclass 7	<input type="checkbox"/>			

Measurement: Ready

Information on calibration standard measurements

## Waveguide Calibration

The Analyzer supports the following calibration methods in a waveguide environment:

- [Reflection Normalization](#) or [Transmission Normalization](#)
- [Full One-Port Calibration](#)
- [One-Path Two-Port Calibration](#)
- [Full Two-Port Calibration](#)
- [Two-Port TRL Calibration](#)

The Analyzer further supports use of a [sliding load standard](#) in the above-mentioned calibrations, except TRL.

General use and features:

- System  $Z_0$  should be set to  $1\ \Omega$  before calibration. Offset  $Z_0$  and terminal impedance in the calibration standard definition also should be set to  $1\ \Omega$  (See [System Impedance  \$Z\_0\$](#) ).
- Waveguide calibration uses two offset short standards instead of a combination of short and open standards. Typically,  $1/8\lambda_0$  and  $3/8\lambda_0$  offset sort standards are used, where  $\lambda_0$  — wave length in waveguide at the mean frequency.

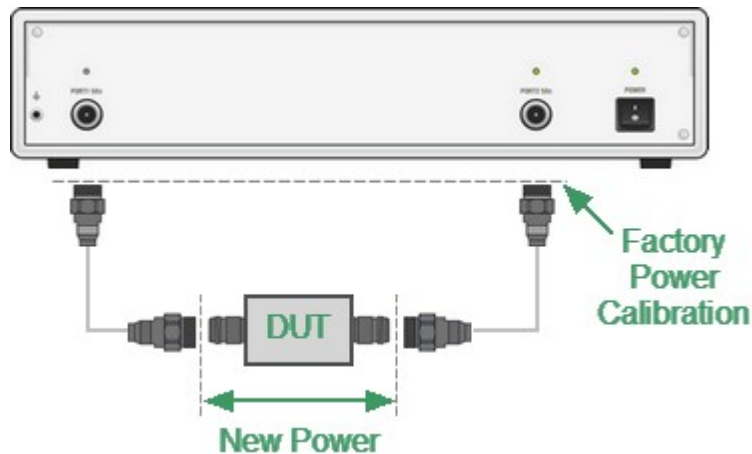


In waveguide calibration, one of two offset short standards must be assigned to the open class (see [Calibration Standard Class Assignment](#)). Consequently, the GUI will contain an Open button with the label of this short standard.

## Power Calibration

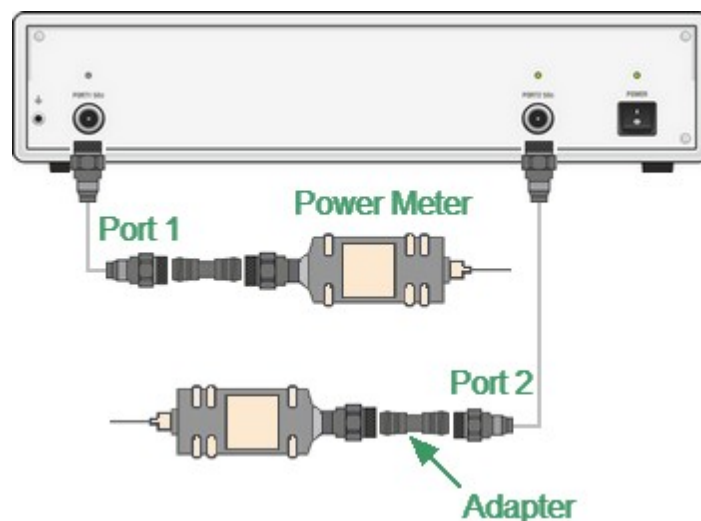
The Analyzer ensures steady power level at the test port inputs with the specified accuracy. The power level is defined between the instrument's minimum and maximum output power level.

A DUT is connected to the Analyzer by cables (see figure below), which have some losses. The power calibration allows to maintain a more accurate power level at a DUT input, adjusted to the use of the cables.



Power Calibration

The power calibration is performed by an external power meter connected to the cables' ends, which will be later connected to the DUT. After the power calibration is complete, power correction automatically turns on. Later it is possible to disable or enable again the power correction function.



Power Calibration with external power meter

---

**NOTE**

If an adapter or other accessory is used when connecting the power meter to the measurement port, the losses introduced by the adapter or accessory are compensated by the Loss Compensation function.

---

The power calibration is performed for each port and each channel individually.

---

**NOTE**

The power correction status is indicated in the trace status field (See [Trace Status Field](#)) and in the channel status bar (See [Channel Status Bar](#)).

---

### Loss Compensation Table

The loss compensation function allows to apply compensation for unwanted losses produced between the power meter and the calibrated port in the process of power calibration. Define the losses, which are needed to compensate in the table specifying frequency and losses (See figure below).

	Frequency	Loss
1	300 kHz	0.1 dB
2	1 GHz	0.2 dB
3	2 GHz	0.4 dB
4	3 GHz	0.5 dB
5		

Loss compensation table

Linear interpolation will be applied to the losses in the intermediary frequency points. The loss compensation table is defined for each port individually.

---

**NOTE**

To have the losses compensated for, enable this function and fill out the table before starting the power calibration procedure.

---

### Loss Compensation Table Editing

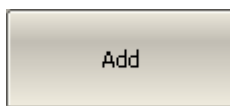
If the loss compensation needs to be applied, enable this function and fill out the table before starting the power calibration procedure. Fill out the table for each port individually.

---



To add a new row to the loss compensation table, use the following softkeys:

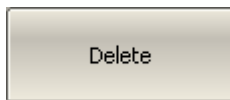
---



**Calibration > Power Calibration > Loss Compens > Add**

A new row will appear under the highlighted one.

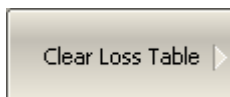
---



To delete a highlighted row, use the following softkeys:

**Calibration > Power Calibration > Loss Compens > Delete**

---



To clear all the table, use the following softkeys:

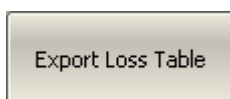
**Calibration > Power Calibration > Loss Compens > Clear Loss Table**

---

[SOUR:POW:PORT:CORR:COLL: TABL:LOSS:DATA](#)

Sets/gets the loss compensation table used when the power calibration is executed by the [SOUR:POW:PORT:CORR:COLL](#) command.

---



To save the table into a \*.lct file on the hard, use the following softkeys:

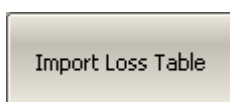
**Calibration > Power Calibration > Loss Compens > Export Loss Table**

---

[MMEM:STOR:PLOS](#)

Saves the loss compensation file.

---



To open the table from a \*.lct file from the hard, use the following softkeys:

**Calibration > Power Calibration > Loss Compens > Import Loss Table**

---

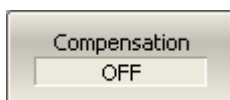
Enter frequency and loss values into the table, scrolling by navigation keys.

---

[MMEM:LOAD:PLOS](#)

Recalls the loss compensation file.

---



To enable the loss compensation function, use the following softkeys:

**Calibration > Power Calibration > Loss Compens > Compensation**



---

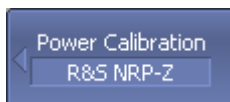
[SOUR:POW:PORT:CORR:COLL:TABL:LOSS](#)

Turns the state of the loss compensation used when the power calibration is executed by the [SOUR:POW:PORT:CORR:COLL](#) command ON/OFF.

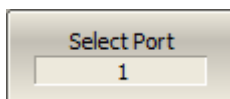
---

## Power Calibration Procedure

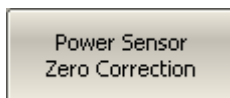
Perform connection and setting of an external power meter as described in [Power Meter Setting](#). Connect the sensor to one of the test ports of the Analyzer and perform calibration as described below. Then repeat the calibration for the other test port.



To select the calibrated port number, use the following softkeys:



**Calibration > Power Calibration > Select Port**



To zero power meter, use the following softkeys:

**Calibration > Power Calibration > Power Sensor Zero Correction**

---

[SYST:COMM:PSEN:ZERO](#)

Executes the zeroing procedure of the power sensor.

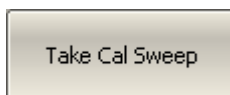
---

---

### NOTE

The power meter sensor can be connected to the port, as during zero setting the output signal of the port is turned off.

---



To execute power calibration, use the following softkeys:

**Calibration > Power Calibration > Take Cal Sweep**

---

[SOUR:POW:PORT:CORR:COLL](#)

Measures the power calibration data for the port <Pt> using the power meter controlled via USB or USB/GPIB.

---

---

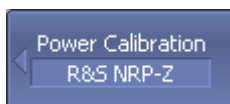
**NOTE**

After the power calibration is complete, power correction automatically turns on.

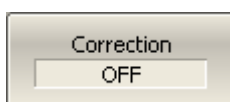
---

**Power Correction Setting**

---



To enable/disable power correction, use the following softkeys:



**Calibration > Power Calibration > Correction**

---

[SOUR:POW:PORT:CORR](#)

Turns the power correction ON/OFF.

---

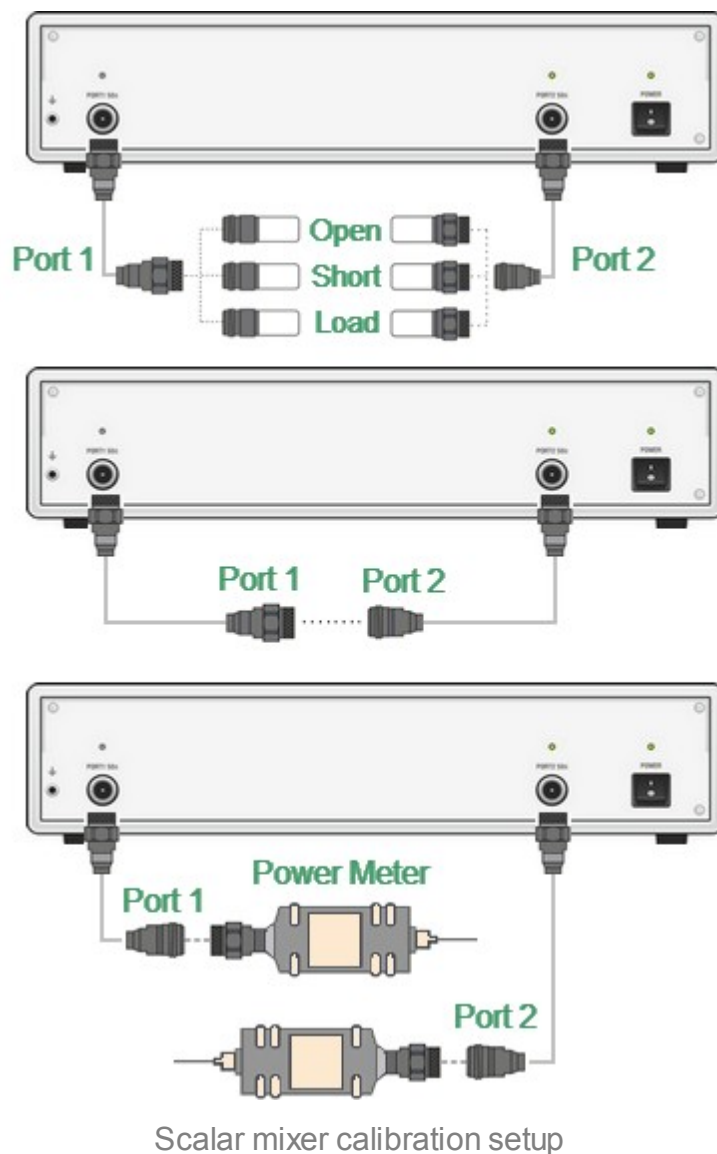
## Scalar Mixer Calibration

### NOTE

The availability of the Scalar Mixer Calibration depends on the Analyzer model (See corresponding [datasheet](#)).

Scalar mixer calibration is the most accurate method of calibration applied to measurements of mixers in frequency offset mode.

The scalar mixer calibration requires OPEN, SHORT, and LOAD standards as well as external power meter (See figure below). The power meter connection and setup are described in [Power Meter Setting](#).



The scalar mixer calibration allows the following measurements:

- Reflection S11 and S22 parameters in vector form.
- Transmission S21 and S12 parameters in scalar form.

The power meter can be connected either one port or both ports. If the power meter was connected to port 1, then S21 transmission parameter will be calibrated. If the power meter was connected to port 2, then S12 transmission parameter will be calibrated.

Before starting the calibration, perform the following settings: select active channel and set its parameters (frequency span, IF bandwidth, etc.), and define the calibration kit. Then enable the frequency offset mode and perform the port settings.

---

#### NOTE

The scalar mixer calibration can be performed without frequency offset. Frequency offset mode can be enabled later, during mixer measurements. In this case, the basic frequency range should cover the frequency range of each port in offset mode. This procedure is convenient, but less accurate as it involves interpolation.

---

### Scalar mixer calibration procedure

---



To access the scalar mixer calibration menu, use the following softkeys:

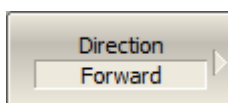
**Calibration > Mixer/Converter Calibration > Scalar Mixer Calibration**

---

[SENS:CORR:OFFS:COLL:METH:SMIX2](#)

Selects the ports and sets the scalar mixer calibration type.

---

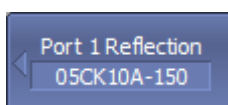


Then select the required calibration direction:

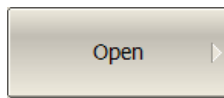
- **Forward,**
- **Reverse,**
- **Both.**

Direction means the transmission direction to be calibrated. Direction determines the power meter connection port or both ports.

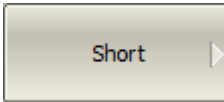
---



Click the **Reflection Port n** softkey.



Connect SHORT, OPEN and LOAD standards to Port 1 as shown in the above figure. Perform measurements for each standard using the respective standard softkeys.



The instrument status bar will indicate **Calibration in progress...** when the measurement is in progress. On completion of the measurement, a check mark will appear in the left part of the softkey.



---

[SENS:CORR:OFFS:COLL:OPEN](#)

Selects the ports and sets the scalar mixer calibration type

---

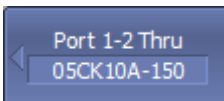
[SENS:CORR:OFFS:COLL:SHOR](#)

Measures the calibration data of the short standard of the specified port when the frequency offset feature is on for scalar mixer calibration.

---

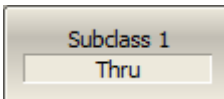
[SENS:CORR:OFFS:COLL:LOAD](#)

Measures the calibration data of the short standard of the specified port when the frequency offset feature is on for scalar mixer calibration.

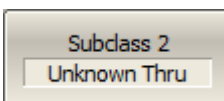


Connect a THRU standard between the test ports.

Click the **Port 1-2 Thru** softkey if a flush thru or non-zero thru is used and is strictly defined in the calibration kit definition.



Click the **Unknown Thru** softkey if a non-zero thru is used and there is no definition for it.

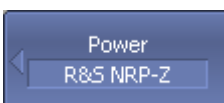


The instrument status bar will indicate **Calibration in progress...** when the measurement is in progress. On completion of the measurement, a check mark will appear in the left part of the softkey.

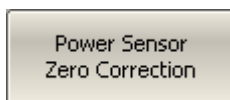
---

[SENS:CORR:OFFS:COLL:THRU](#)

Measures the calibration data of the thru standard of the specified port when the frequency offset feature is on for scalar mixer calibration.



Click the **Power** softkey.



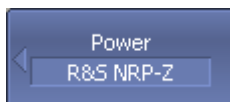
If needed, zero power meter using the **Power Sensor Zero Correction** softkey.

---

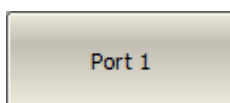
#### NOTE

The power meter sensor can be connected to the port, as during zero setting the output signal of the port is turned off.

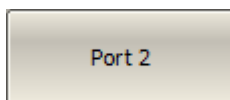
---



Connect the power meter to Port n and click the corresponding softkey.



The instrument status bar will indicate **Calibration in progress...** when the measurement is in progress. On completion of the measurement, a check mark will appear in the left part of the softkey.

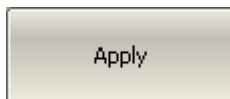


---

#### [SENS:CORR:OFFS:COLL:PMETer](#)

Measures the scalar-mixer calibration data using the power meter when the frequency offset feature is ON.

---



To complete the calibration procedure, click **Apply** softkey.

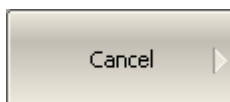
This will activate the process of calibration coefficient table calculation and saving it into the memory. The error correction function will also be automatically enabled.

---

#### [SENS:CORR:OFFS:COLL:SAVE](#)

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.

---



To clear the measurement results of the standards, click **Cancel** softkey.

This softkey does not cancel the current calibration. To disable the current calibration, turn off the error correction function.

---

#### [SENS:CORR:OFFS:COLL:CLE](#)

Clears the calibration measurement data of scalar mixer calibration when the frequency offset feature is ON.

---

---

**NOTE**

The calibration status can be checked in the channel status bar (See [General error correction status table](#)) or in trace status field (See [Trace error correction status table](#)) — **SMC** label.

---

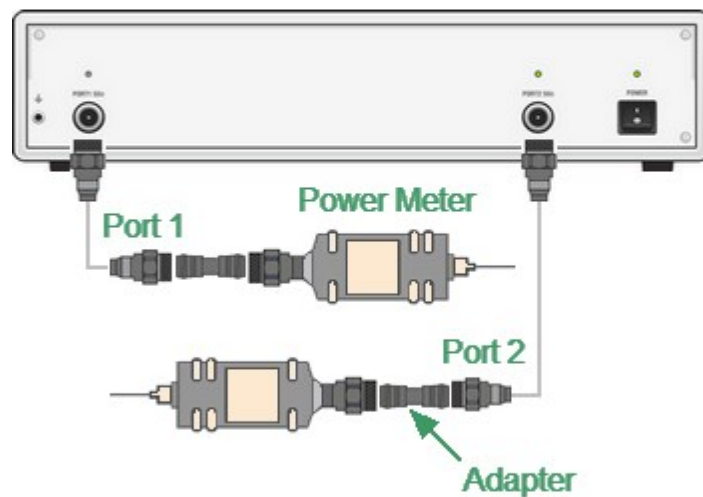
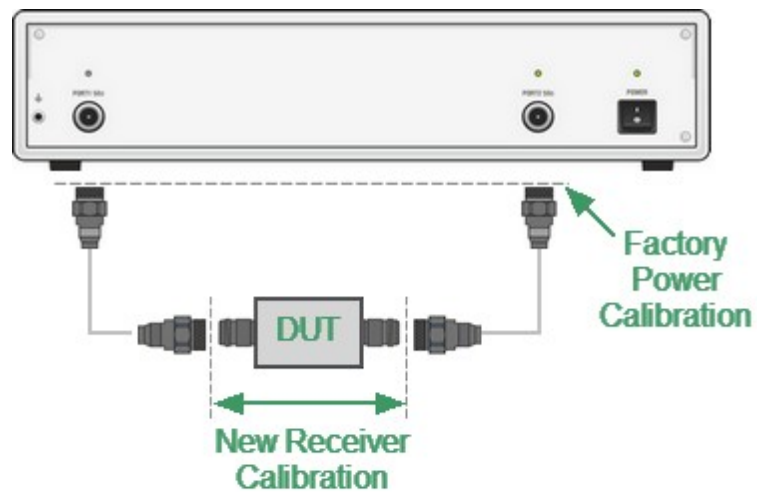
## Receiver Calibration

Receiver calibration is only used for absolute measurements. The receiver calibration is divided into the test receiver (A, B) calibration and the reference receiver (R1, R2) calibration (See [Analyzer Block Diagram](#)). The calibration procedure is different for these receivers.

### 1. Test receiver calibration

When performing absolute power measurements (See [Absolute Measurements](#)), the gain of receivers is factory calibrated to test port on the front panel.





Receiver Calibration

In practice, the power is measured at test port inputs made by the fixture producing losses. The test receiver calibration allows to measure the power at port inputs with higher accuracy.

The receiver calibration is performed by sending the calibration signal from the source port to the calibrated port input. The receiver calibration requires the connection between the both test ports using THRU connection.

To make the receiver calibration most accurate, first perform power calibration on the source port. If the source power calibration was not performed, the calibrated port needs to be connected to the source port on the front panel.

## 2. Reference receiver calibration

Unlike test receivers, the reference receiver measures the output power of its port. Supplying a signal from a different port is meaningless for a reference receiver. That is why the power source in reference receiver calibration is the reference receiver itself, no matter which port is assigned to be a source.

## 3. General comments

It is possible to specify the power offset value before calibration. As a result, the receiver readings will be offset by this value.

After the receiver calibration is complete, receiver correction automatically turns on. Later it is possible to disable or enable again the receiver correction function.

The power calibration is performed for each port and each channel individually.

---

### NOTE

The power correction status is indicated in the trace status field (See [Trace Status Field](#)) and in the channel status bar (See [Channel Status Bar](#)).

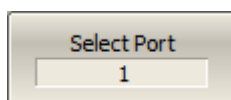
---

## Receiver Calibration Procedure

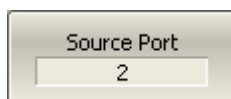
Connect a THRU between the receiver port and the source port if test receiver calibration is performed.



To select the calibrated port number, use the following softkeys:



**Calibration > Receiver Calibration > Select Port**



To select the source port number, use the following softkeys:

**Calibration > Receiver Calibration > Source Port**

---

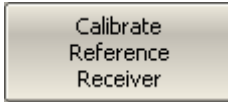
### NOTE

Source port number is valid only for test receiver calibration.

---

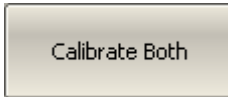


To execute the test receiver calibration, use the following softkeys:



**Calibration > Receiver Calibration > Calibrate Test Receiver**

To execute the reference receiver calibration, use the following softkeys:



**Calibration > Receiver Calibration > Calibrate Reference Receiver**

Use the **Calibrate Both** softkey to perform the calibration of the test and reference port receivers in succession.

---

[SENS:CORR:REC:COLL:TCH:ACQ](#)

Executes receiver calibration of the test receiver of the specified port <Pt>.

---

[SENS:CORR:REC:COLL:RCH:ACQ](#)

Executes receiver calibration of the reference receiver of the specified port <Pt>.

---

[SENS:CORR:REC:COLL:ACQ](#)

Executes receiver calibration of both the test receiver and the reference receiver of the specified port <Pt>.

---

---

**NOTE**

After the receiver calibration is complete, receiver correction automatically turns on.

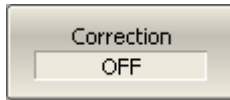
---

## Receiver Correction Setting

---



To enable/disable receiver correction, use the following softkeys:



**Calibration > Receiver Calibration > Correction**

---

[SENS:CORR:REC](#)

Executes receiver calibration of both the test receiver and the reference receiver of the specified port <Pt>.

---

## Vector Mixer Calibration

### NOTE

The availability of the Vector Mixer Calibration depends on the Analyzer model (See corresponding [datasheet](#)).

Vector scalar calibration is a calibration method applied for mixer measurements. This method allows measurement of both reflection and transmission S-parameters in vector form, including phase and group delay of the transmission coefficient.

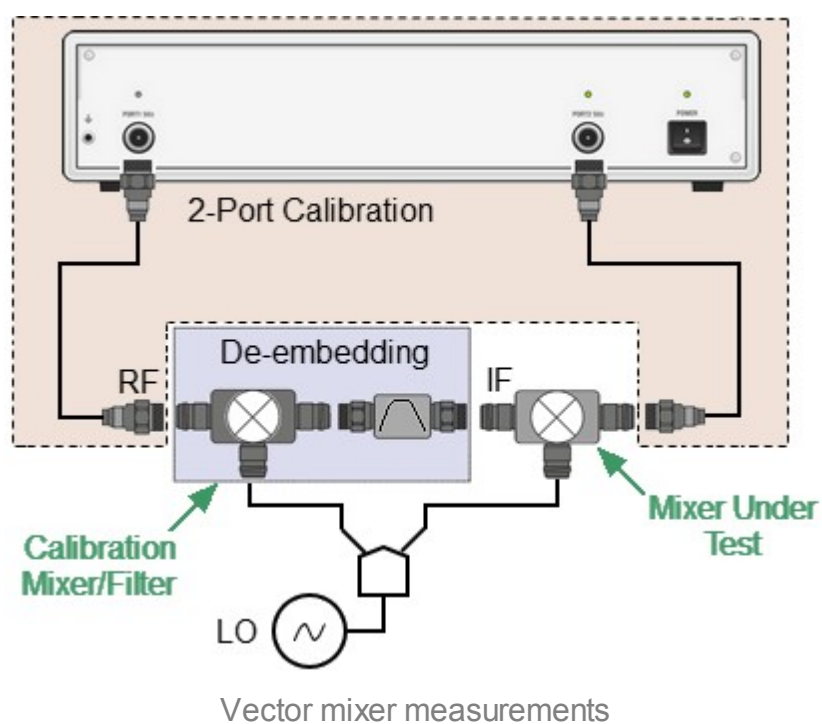
The vector mixer measurements require an additional mixer with a filter, which is called a calibration mixer.

The filter separates the IF, which is the input frequency for the mixer under test:

- $RF + LO$
- $RF - LO$
- $LO - RF$

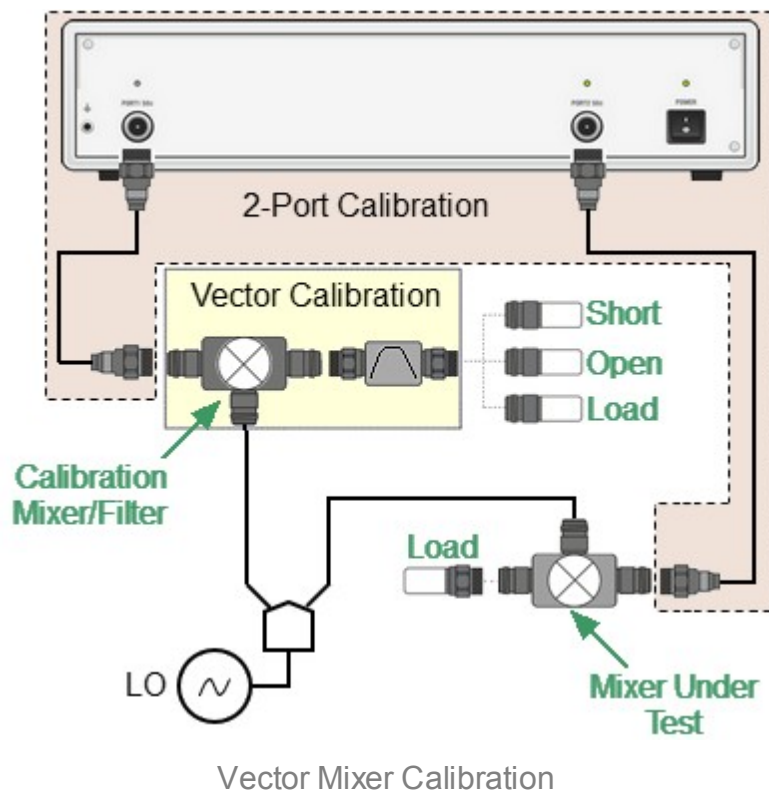
Both the calibration mixer and mixer under test are powered from one LO.

The vector mixer measurement is a combination of a two-port calibration and a de-embedding function (See figure below).



The de-embedding function requires an S-parameter file of the circuit. Acquisition of such a file for the calibration mixer/filter pair is called vector mixer calibration.

To obtain an S-parameter file of the calibration mixer/filter, use SHORT, OPEN, and LOAD calibration standards (See figure below).



## Vector Mixer Calibration Procedure (except M models)

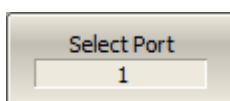
Before starting the calibration, perform the following settings:

- Activate a channel and set its parameters (frequency span, IF bandwidth, etc.), and define the calibration kit.
- Perform two-port calibration.
- Assemble vector calibration setup.
- Set frequency and power of the external LO.



To access the vector mixer calibration menu, use the following softkeys:

## Calibration > Mixer/Converter Calibration > Vector Mixer Calibration



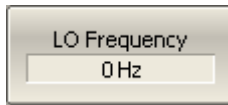
To select the number of test port connected to the calibration mixer, click **Select Port** softkey.

---

[SENS:CORR:VMC:COLL:PORT](#)

Turns the setup option in the vector mixer calibration ON/OFF.

---



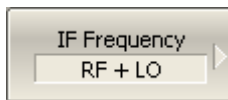
Enter the LO frequency, using the **LO Frequency** softkey.

---

[SENS:CORR:VMC:COLL:LO:FREQ](#)

Sets or reads out the LO frequency value used in the vector mixer calibration.

---



Select the frequency to be separated by the filter, using the **IF Frequency** softkey:

- RF + LO
  - RF – LO
  - LO – RF
- 

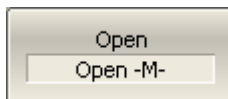
[SENS:CORR:VMC:COLL:IF:SEL](#)

Selects the IF frequency from RF+LO, RF-LO, and LO-RF, depending on the IF frequency of the calibration mixer in the vector mixer calibration.

---



Connect SHORT, OPEN and LOAD standards to IF filter output as shown in [Vector Mixer Calibration](#). Perform the measurement using the respective standard softkey.



The instrument status bar will indicate **Calibration in progress...** when the measurement is in progress. Upon completion of the measurement, a check mark will appear in the left part of the softkey.



---

[SENS:CORR:VMC:COLL:OPEN](#)

Measures the open standard in order to characterize the calibration mixer + filter in the vector mixer calibration.

---

[SENS:CORR:VMC:COLL:SHOR](#)

Measures the short standard in order to characterize the calibration mixer + filter in the vector mixer calibration.

---

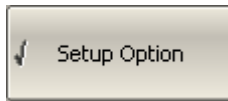
[SENS:CORR:VMC:COLL:LOAD](#)

Measures the load standard in order to characterize the calibration mixer + filter in

---

the vector mixer calibration.

---



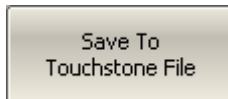
To complete the calibration procedure, click the **Save To Touchstone File** softkey.

---

[SENS:CORR:VMC:COLL:OPT](#)

Turns the setup option in the vector mixer calibration ON/OFF.

---



This will activate calculation of the calibration mixer/filter pair S-parameters and saving those into a Touchstone file. Enter the file name in the pop-up dialog.

If the **Setup Option** feature is enabled, the S-parameter file will be passed to the de-embedding function and this function will be activated.

---

[SENS:CORR:VMC:COLL:SAVE](#)

Completes the vector mixer calibration procedure.

---

---

**NOTE**

The calibration status can be checked in the channel status bar (See [General error correction status table](#)) — **F2** and **Dmb** labels (two-port calibration and de-embedding function).

---



## Automatic Calibration Module

Automatic calibration modules (ACMs) are special devices, which allow for automating the process of calibration. The ACM model is selected according to the parameters of the calibrated Analyzer: the working frequency range, the number of measuring ports, and the type of RF connectors. All ACM models and their specifications are available on the Copper Mountain Technologies [website](#). One of the models is shown in the image below.



Automatic Calibration Module

The ACM offers the following advantages over the traditional SOLT calibration, which uses a mechanical calibration kit:

- Reduces the number of connections of standards. Instead of connecting seven standards, it requires connecting only two ACM connectors.
- Reduces the calibration time.
- Reduces human error probability.
- Provides higher accuracy potentially.

The ACM has two RF connectors for connecting to the test ports of a two-port Analyzer and a USB-connector for control. The ACM contains electronic switches, which switch between different reflection and transmission impedance states, as well as memory, which stores precise S-parameters of these impedance states.

After connecting the ACM to the Analyzer, the analyzer software performs the calibration procedure automatically, i.e. switches between different ACM states, measures them, and computes calibration coefficients using the data stored in the ACM memory.

## **Automatic Calibration Module Features**

### **Calibration Types**

The ACM allows the Analyzer software to perform one-path two-port, full one-port or full two-port calibration. Calibration is performed with the click of a button.

### **Characterization**

Characterization is a table of S-parameters for all the states of the ACM switches, stored in the ACM memory. There are two types of characterization: user characterization and factory characterization. The ACM has two memory sections. The first one is write-protected and contains factory characterization. The second memory section allows to store up to three user characterizations. Factory characterization or any of the user characterizations stored in the ACM memory can be selected before calibration. The user characterization option is provided for saving new S-parameters of the ACM after connecting adapters to the ACM ports.

The software allows to perform a user characterization and save the data to the ACM with the click of a button. To do this, the Analyzer test ports should be calibrated in configuration compatible with the ACM ports.

### **Automatic Orientation**

Orientation means relating the ACM ports to the test ports of the Analyzer. While the Analyzer test ports are indicated by numbers, the ACM ports are indicated by the letters A and B.

Orientation can be defined either manually or automatically. In the case of automatic orientation, the Analyzer software determines the ACM orientation each time prior to its calibration or characterization.

### **Unknown Thru**

The Thru implemented by the electronic switches inside the ACM introduces losses. That is why the exact parameters of the Thru should be known, or an Unknown Thru algorithm should be used to achieve the specified calibration accuracy. The software allows to use both options. The ACM memory stores S-parameters of the Thru, which are used to compute calibration coefficients. If an Unknown Thru algorithm is applied, such parameters are disregarded.

### **Thermal Compensation**

The most accurate calibration is achieved if the ACM temperature is equal to the temperature at which it was characterized. When this temperature changes, certain ACM state parameters may deviate from the parameters stored in the memory. This results in reduction of the ACM calibration accuracy.

To compensate for the thermal error, the ACM features thermal compensation function. Thermal compensation is a software function of the ACM S-parameter correction based on its temperature dependence and the data from the temperature sensor inside the ACM. The temperature dependence of each ACM is determined at the factory and saved into its memory.

Thermal compensation can be enabled or disabled.

### **Confidence Check**

The ACM also implements an additional state — an attenuator, which is not used in calibration. The attenuator is used to check the current calibration performed by ACM or any other method. This is called a confidence check.

In the confidence check mode, the factory measurement of the attenuator is loaded into the memory trace, which may be compared to the measurement being performed by the active trace. The two traces may be compared, and their differences may be evaluated to determine the accuracy of the calibration performed.

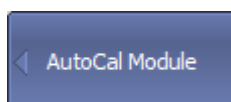
For a detailed comparison, the math (division) function can be used for data and memory.

## Automatic Calibration Procedure

### Settings Before Calibrating

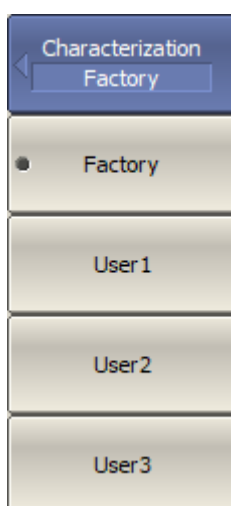
Before calibrating the Analyzer with the ACM, perform some settings, i.e. activate a channel and set channel parameters (frequency range, IF bandwidth, etc.).

Connect the ACM to the Analyzer test ports and connect the USB port of the ACM to the USB port of the PC.



To open automatic calibration submenu, use the following softkeys:

**Calibration > AutoCal**



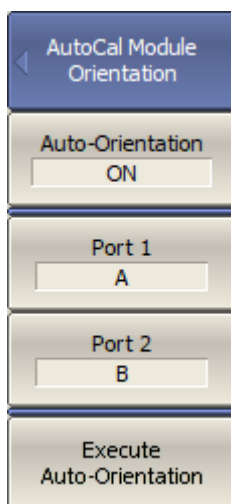
To select characterization, use the following softkeys:

**Characterization > [Factory | User1 | User2 | User3]**

---

[SENS:CORR:COLL:ECAL:UCH](#)

Sets or reads out the characterization number used when executing AutoCal (factory or user characterizations).



When selecting manual or automatic orientation for ACM, it is recommended to select the automatic orientation.

To enable auto orientation for ACM before performing each automatic calibration, use the following softkeys:

#### **Orientation > Auto-Orientation [ON | OFF]**

To manually select the orientation, turn OFF **Auto-Orientation**. Use the **Port n** softkeys to relate the analyzer test ports with the ACM ports.

Regardless of the other settings, automatic orientation is performed immediately by clicking the **Execute Auto-Orientation** softkey.

[SENS:CORR:COLL:ECAL:ORI:STAT](#)

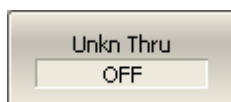
Turns the Auto-Orientation function ON/OFF when the AutoCal Module calibration is executed.

[SENS:CORR:COLL:ECAL:ORI:EXEC](#)

Executes the Auto-Orientation procedure of the AutoCal Module.

[SENS:CORR:COLL:ECAL:PATH](#)

Sets or reads out the AutoCal module port number which is connected to a specified port of the Network Analyzer.



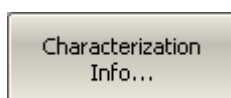
Enable or disable Unknown Thru algorithm using the **Unkn Thru [ON | OFF]** softkey.

[SENS:CORR:COLL:ECAL:UTHR:STAT](#)

Turns the Unknown Thru feature ON/OFF when the AutoCal Module calibration is executed.



Enable or disable the thermal compensation using the **Thermo compensation [ON | OFF]** softkey.



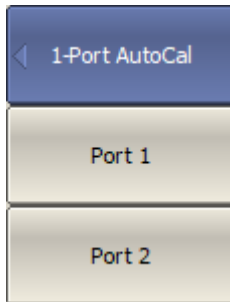
To display detailed information on characterization, use the **Characterization Info** softkey.

[SENS:CORR:COLL:ECAL:INF?](#)

Gets information on the AutoCal Module connected to the Network Analyzer.

## One/Two-Port Calibration

For a one/two-port calibration, connect any ports of the ACM to the ports to be calibrated on the Analyzer.



To perform full one-port calibration, use the **1-Port AutoCal** softkey.

Then, select the port number. Click **Port n** softkey to start the calibration.

Wait until calibration is complete.

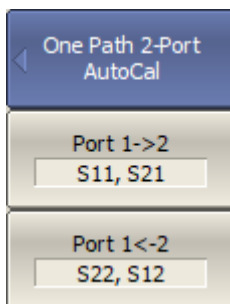
The instrument status bar will indicate **Calibration in progress...** when the measurement is in progress.

---

[SENS:CORR:COLL:ECAL:SOLT1](#)

Executes one-port calibration of the specified port of the specified channel using the AutoCal module.

---



To perform one path two-port calibration, use the **One Path 2-Port Auto** softkey.

Then, select the direction of the calibration using the **Port n->m** softkey. The label on the softkey indicates the measured parameters.

Click **Port n->m** softkey to start the calibration.

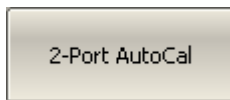
Wait until calibration is complete.

The instrument status bar will indicate **Calibration in progress...** when the measurement is in progress.

---

[SENS:CORR:COLL:ECAL:ERES](#)

Executes one path two-port calibration between the specified two ports of the specified channel using the AutoCal module.



To perform full two-port calibration, click the **2-Port AutoCal** softkey.

Wait until calibration is complete.

The instrument status bar will indicate **Calibration in progress...** when the measurement is in progress.

---

[SENS:CORR:COLL:ECAL:SOLT2](#)

Executes full two-port calibration between the specified two ports of the specified channel using the AutoCal module.

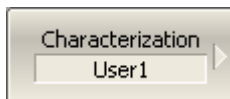
---

## User Characterization Procedure

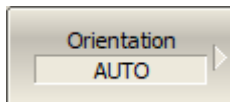
User characterization of ACM is required when modifying ACM connectors with adapters. The characterization is performed for the new ACM configuration, which includes adapters. To ensure calibration accuracy it is not recommended to disconnect and reconnect the adapters back after characterization until calibration is complete.

Before performing the user characterization of the ACM, perform full two-port calibration of the analyzer in configuration of the test ports compatible with the configuration of the ACM ports.

Connect the ACM to the Analyzer test ports and connect the USB port of the ACM to the USB port of the PC.

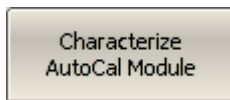


Select user characterization 1 to 3 using the **Characterization** softkey.



Select a manual or automatic orientation for the ACM using the **Orientation** softkey.

It is recommended to select the **AUTO** orientation.



Perform characterization using the **Characterize ACM** softkey.

---

After the ACM measurement is completed, the following dialog box will appear:



**User Characterization** [X]

Characterization

Number    Operator

Analyzer  Location

Connectors

Port A

Port B

Adapter Description

Port A

Port B

Fill in the following fields:

- User name
- Analyzer name
- Characterization location
- Connectors (types of adapter connectors)
- Adapter description (description of adapters)

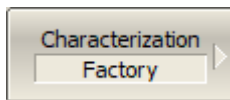
Use the **Save** softkey to complete the user characterization of the ACM.

## Confidence Check Procedure

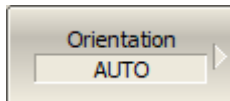
Perform a confidence check if the reliability of the current calibration needs to be verified. This function can be used to check the accuracy of either calibration performed with an ACM or with a mechanical calibration kit.

Connect the ACM to the Analyzer test ports and connect the USB port of the ACM to the USB port of the PC.

Enable the display of the data trace for the needed parameter, for example, S21. It is possible to enable several data traces simultaneously, for example, S11, S22, S21, S12.

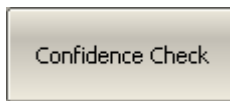


Select characterization using the **Characterization** softkey.



Select a manual or automatic orientation for the ACM using the **Orientation** softkey.

It is recommended to select AUTO orientation.



Perform a confidence check using the **Confidence Check** softkey.

---

[SENS:CORR:COLL:ECAL:CCH](#)

Executes the confidence check of the calibration coefficients of the specified channel using the AutoCal module.

---

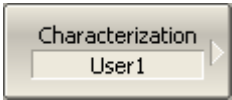
After the measurement is completed, two traces for each S-parameter will be displayed. The measured parameters will be shown as the data trace, and the ACM parameters will be shown as the memory trace.

Compare the data trace and the memory trace of the same parameter, for example S21. To perform more accurate comparison, enable the function of math operations between data and memory traces. In the logarithmic magnitude or phase format, use the Data/Memory operation (See [Mathematical Operations](#)). In the linear magnitude format, use the Data-Memory operation.

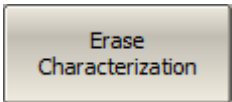
The conclusion on whether the current calibration provides sufficient accuracy or not is made by the user.

# Erasing the User Characterization

If necessary, it is possible to erase the user characterization in the ACM. The procedure erases all data of selected user characterization, overwriting it with zeros. Factory characterization cannot be erased.



Select the user characterization using the **Characterization** softkey.

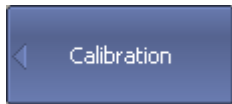


Perform erase procedure using the **Erase Characterization** softkey.

---

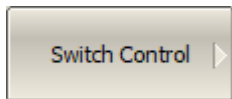
## Manual Switch Control

The software allows to manage the ACM switches directly. The ACM states are selected from the list of possible states and switched by pressing the button. The number of switches and states of each ACM are described in the block diagrams of modules in the document "Automatic Calibration Modules Operating Manual".



To manage ACM switches, use the following softkeys:

**Calibration > AutoCal > Switch Control**



Select the required ACM state from the list.

---

### CAUTION

This function is intended for advanced users, it is not used in standard ACM work.

---

## Error Correction Status

The error correction status is indicated for each trace individually. There is also a general status of error correction for all traces of a channel.

### General error correction status

The general error correction status for all S-parameter traces of a channel is indicated in the specific field on a channel status bar (See [General error correction status table](#)). For the channel status bar description, see [Channel Status Bar](#).

Symbol	Definition	Note
<b>Cor</b>	Error correction is enabled. The stimulus settings are the same for the measurement and the calibration.	If the function is active for all traces — black characters on a gray background.  If the function is active only for some of the traces (other traces are not calibrated) — white characters on a red background.
<b>C?</b>	Error correction is enabled. The stimulus settings are not the same for the measurement and the calibration. Interpolation is applied.	
<b>C!</b>	Error correction is enabled. The stimulus settings are not the same for the measurement and the calibration. Extrapolation is applied.	
<b>Off</b>	Error correction is turned off.	For all traces. White characters on a red background.
<b>---</b>	No calibration data. No calibration was performed.	

### Trace error correction status

The error correction status for each individual trace is indicated in the trace status field (See table below). For trace status field description, see [Trace Status Field](#).

Symbols	Definition
RO	OPEN response calibration
RS	SHORT response calibration
RT	THRU response calibration
OP	One-path two-port calibration
F1	Full one-port (SOL) calibration
F2	Full two-port (SOLT) or TRL calibration
SMC	Scalar mixer calibration

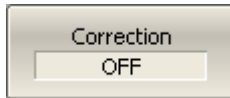
## Error Correction Disabling

This feature allows to disable the error correction function, which automatically becomes enabled after completion of calibration by any method.

---



To disable and enable the error correction function, use the following softkeys:



**Calibration > Correction**

---

[SENS:CORR:STAT](#)

Turns the S-parameter error correction ON/OFF.

---



# System Impedance Z0

Z0 is the system impedance of a measurement path. Normally, it is equal to the impedance of the calibration standards used for calibration. The Z0 value should be specified before calibration, as it is used for calibration coefficient calculations.

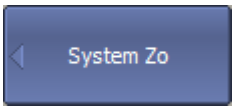
For waveguide calibration, the system impedance must be set to 1 Ω.

The impedance of both test ports is the same for most of measurement types. The Analyzer can perform measurements when Z0 values of the test ports are different, for example, Type N50 – Waveguide. For such measurements, use different impedance settings for the test ports, Z01 and Z02.

NOTE

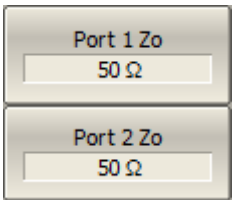
To calibrate the Analyzer with different port impedances, Z01 and Z02, the following methods are provided: Adapter Removal, Unknown Thru Addition (described in [Non-Insertable Device Measuring](#)).

## Manual Z0 Setting



To set the system impedance Z0, use the following softkeys:

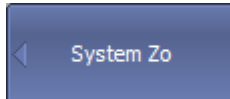
**Calibration > System Zo > Port n Zo**



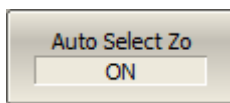
<a href="#">SENS:CORR:IMP</a>	Sets or reads out the system impedance Z0.
<a href="#">SENS:CORR:PORT:IMP</a>	Sets or reads out the impedance Z0 of port <Pt>.

## Automatic Z0 Selecting

The automatic system impedance selecting function sets Z0 during the process of calibration standard measurement, using data from the definition of the calibration standard in a calibration kit. Z0 of the corresponding port is set when measuring one-port standards. Z0 of the two ports is set when measuring two-port standards. The UNKNOWN THRU standard does not make any changes in Z0 of the ports. By default, the function is enabled, but it can be disabled.



To enable/disable the function of automatic selecting of port impedance Z0, use the following softkeys:



**Calibration > System Zo > Auto Select Zo > [ON|OFF]**

---

[SENS:CORR:IMP:SEL:AUTO](#)

Turns the auto-select Z0 function ON/OFF.

---

## Calibration Trigger Source

The function sets the trigger source to start measuring the calibration standards. If an Internal source is selected, the calibration starts immediately. If the source is System, the system trigger is used to start the calibration. The source of the system trigger is set by the soft buttons:

**Stimulus > Trigger > Trigger Source > [Internal | External | Manual | Bus]**

If a system trigger is used, the averaging trigger function and the external trigger event function set "to point" affect the start of the calibration in the same way as during standard measurements. When using a system trigger, the trigger source Bus should not be used, otherwise the program may be blocked.

---

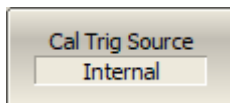
### NOTE

This function does not apply to calibration using the ACM, power calibration, receiver calibration. In these calibrations, the internal trigger is always used.

---



To set the calibration trigger source, use the following softkeys:



**Calibration > Cal Trig Source [Internal | System]**

---

[SENS:CORR:TRIG:FREE](#)

Enables/disables the internal trigger source for calibration.

---

## Measurement Data Analysis

The following section describes the process of Measurement Data Analysis using the Analyzer.

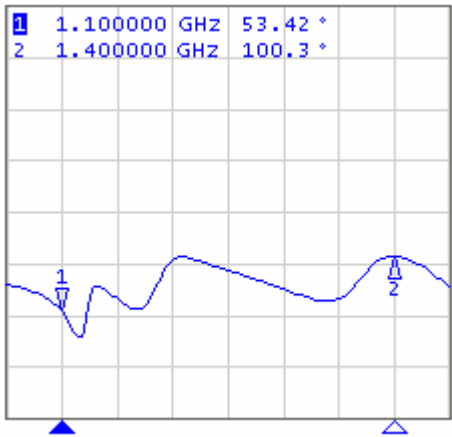
Special software marker tools are used to read and look up the numerical values of the stimulus and the measured value on selected points on the graph. For a detailed description see [Markers](#).

This section also contains information about the various functions and tools used to analyze measurements.

- [Memory Trace Function](#) is used to save data traces and perform mathematical operations between memory and data traces.
- [Trace Hold](#) is used to hold the maximum or minimum values of the trace.
- [Fixture Simulation](#) is used to simulate measurement conditions that differ from real measurement conditions.
- [Time Domain Transformation](#) is used to convert the measured characteristics in the frequency domain into the circuit response in the time domain.
- [Time Domain Gating \(except M models\)](#) is used to eliminate unwanted responses in the time domain.
- [S-Parameter Conversion](#) is used to convert the measurement results into different parameters: impedance or admittance in reflection/transmission measurement, inverse S-parameter, equivalent impedance or admittance in transmission shunt measurements, S-parameter complex conjugate.
- [General S-Parameter Conversion](#) is used for general conversion of S-parameters to Z, Y, T, H, ABCD — parameters based on the full matrix of S-parameters.
- A function of pass/fail determination for the trace of the measurement data according to various criteria:
  1. [Limit Test](#) is used to compare the trace of the measured value with the limit line.
  2. [Ripple Limit Test](#) is used to check the value of the ripple trace with user-defined ripple limits
  3. [Peak Limit Test](#) is used to check if the peak of the trace of the set polarity falls within the limits for the peak set by the user.

# Markers

A marker is a tool for numerical readout of a stimulus value and value of the measured parameter in a specific point on the trace. Up to 16 markers can be activated on each trace. A trace with two markers is shown in the figure below.



Trace with two markers

The markers allow to perform the following tasks:

- Reading absolute values of a stimulus and a measured parameter in selected points on the trace.
- Reading relative values of a stimulus and a measured parameter related to the reference point.
- Search for specific points on the trace (minimum, maximum, target level, etc.).
- Determining trace parameters (statistics, bandwidth, etc.).
- Editing stimulus parameters using markers.

Markers can have the following indicators:

1 ▽	Symbol and number of the active marker on a trace.
Δ 2	Symbol and number of the inactive marker on a trace.
▲	Symbol of the active marker on a stimulus axis.
Δ	Symbol of the inactive marker on a stimulus axis.

The marker data field contains the marker number, stimulus value, and the measured parameter value. The number of the active marker is highlighted in an inverse color.

The marker data field contents vary depending on the display format (rectangular or circular):

- In rectangular format, the marker shows the measurement parameter value plotted along Y-axis in the active format (See the table below).

Format Type Description	Label	Data Type (Y-axis)	Measurement Unit (Y-axis)
Logarithmic Magnitude	<b>Log Mag</b>	S-parameter magnitude: logarithmic $20 \cdot \log S $ , $ S  = \sqrt{a^2 + b^2}$	Decibel (dB)
Voltage Standing Wave Ratio	<b>SWR</b>	$\frac{1+ S }{1- S }$	Dimensionless value
Phase	<b>Phase</b>	S-parameter phase from $-180^\circ$ to $+180^\circ$ : $\frac{180}{\pi} \cdot \arctg \frac{b}{a}$	Degree ( $^\circ$ )
Expanded Phase	<b>Expand Phase</b>	S-parameter phase, range expanded to from below $-180^\circ$ to over $+180^\circ$	Degree ( $^\circ$ )
Group Delay	<b>Group Delay</b>	Signal propagation delay within the DUT: $-\frac{d\varphi}{d\omega}$ , $\varphi = \arctg \frac{b}{a}$ , $\omega = 2\pi \cdot f$	Second (sec.)
Linear Magnitude	<b>Lin Mag</b>	S-parameter magnitude: linear $\sqrt{a^2 + b^2}$	Dimensionless value
Real Part	<b>Real</b>	S-parameter real part: $a = \text{re}(S)$	Dimensionless value
Imaginary Part	<b>Imag</b>	S-parameter imaginary part: $b = \text{im}(S)$	Dimensionless value

- In circular format, the marker shows two or three values listed in the table below.

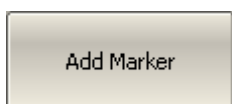
Label	Marker Readings (Measurement Unit)		
	Reading 1	Reading 2	Reading 3
Smith (Lin)	Linear magnitude	Phase (°)	—
Smith (Log)	Logarithmic magnitude (dB)	Phase (°)	—
Smith (Re/Im)	Real part	Imaginary part	—
Smith (R + jX)	Resistance ( $\Omega$ )	Reactance ( $\Omega$ )	Equivalent capacitance or inductance (F/H)
Smith (G + jB)	Conductance (S)	Susceptance (S)	Equivalent capacitance or inductance (F/H)
Polar (Lin)	Linear magnitude	Phase (°)	—
Polar (Log)	Logarithmic magnitude (dB)	Phase (°)	—
Polar (Re/Im)	Real part	Imaginary part	—

## Marker Addition



To enable a new marker, use the following softkeys:

**Markers > Add Marker**



[CALC:MARK](#)

Turns the marker ON/OFF.

## NOTE

The new marker appears as the active marker in the middle of the stimulus axis. The input field for the marker stimulus value is activated.

---

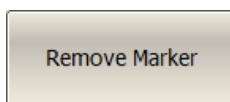
## Marker Deletion

---



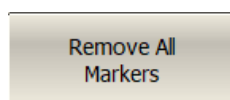
To delete a marker, use the following softkeys:

**Markers > Remove Marker**



To delete all the markers, use the following softkeys:

**Markers > Remove All Markers**



---

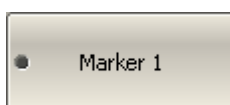
## Marker Activation

---

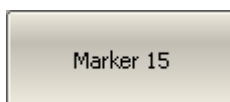


To activate a marker by its number, use the following softkeys:

**Markers > Select > Marker n**

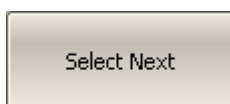


...



To activate a marker from the list of markers, use the following softkeys:

**Markers > Select Next**



---

[CALC:MARK:ACT](#)

Sets the active marker.

---

---

### NOTE

A marker can be activated by clicking on it.

---

## Marker Stimulus Value Setting

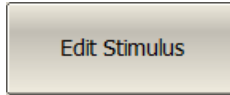


The active marker must be selected before setting the marker stimulus value. The stimulus value must be set by entering the numerical value from the keyboard, by arrows, by dragging the marker using the mouse (See [Marker Stimulus Value Setting](#)), or by enabling the search function (See [Marker Position Search Functions](#)).



To set the marker stimulus value, use the following softkeys:

**Markers > Edit Stimulus**



or click on the stimulus value field using the mouse.

Then, enter the value using the numerical keys on the keypad, by «↑», «↓» arrows.

---

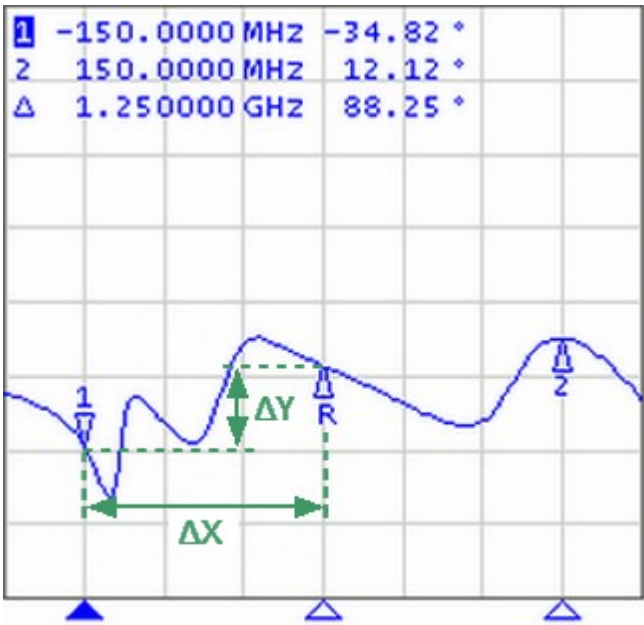
[CALC:MARK:X](#)

Sets or reads out the stimulus value of the marker.

---

# Reference Marker Feature

The reference marker feature allows to view the data relative to the reference marker. Other markers readings are represented as delta relative to the reference marker. The reference marker shows the absolute data, and is indicated with «R» symbol instead of a number (See figure below). Enabling of a reference marker turns all the other markers to relative display mode.



Reference marker

Reference marker can be indicated on the trace as follows:

R ▽	Symbol of the active reference marker on a trace.
Δ R	Symbol of the inactive reference marker on a trace.

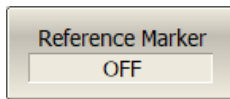
The reference marker displays the stimulus and measurement absolute values. The rest of the markers display the relative values:

- Stimulus value ( $\Delta X$  in the figure above) is the difference between the absolute stimulus values of this marker and the reference marker.
- Measured value ( $\Delta Y$  in the figure above) is the difference between the absolute measurement values of this marker and the reference marker.



To enable/disable the reference marker, use the following softkeys:

**Markers > Reference Marker**



[CALC:MARK](#)

Turns the marker ON/OFF.

---

[CALC:MARK:ACT](#)

Sets the active marker.

---

[CALC:MARK:REF](#)

Turns the reference marker ON/OFF.

---

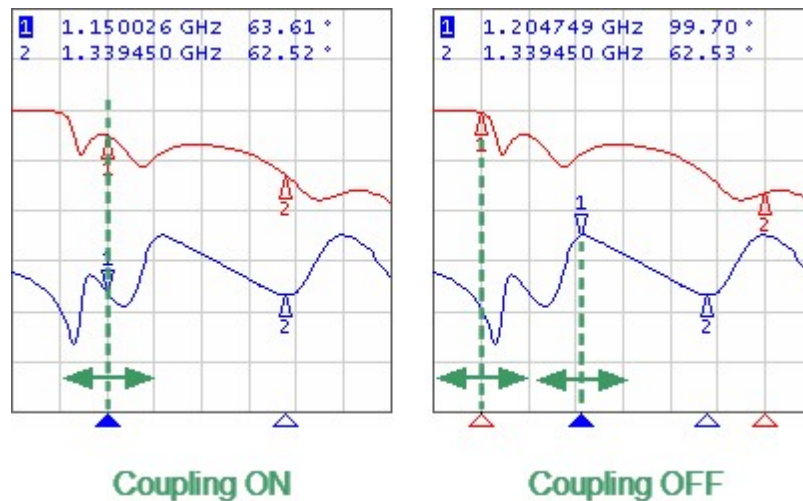
## Marker Properties

The following section describes marker properties:

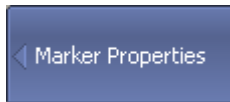
- [Marker Coupling Feature](#) is the function that determines the coupling of markers with the same numbers on different traces.
- [Marker Table](#) is used to display the marker values of all traces and channels.
- [Marker Value Indication Capacity](#) is the setting of the bit-length of numerical values on markers.
- [Multi Marker Data Display](#) is the ability to enable display of the marker data for all traces simultaneously.
- [Marker Data Arrangement](#) is the ability to rearrange the marker data display on the screen.
- [Marker Data Alignment](#) is the ability to align the marker data display on the screen.
- [Memory Trace Value Display](#) is the ability to turn on the memory trace marker values if a memory trace is available.
- [Marker Discrete Mode](#) is the moving of the marker only between actual measurement points.

## Marker Coupling Feature

The marker coupling feature enables/disables coupling of markers with the same numbers on different traces. If the feature is turned on, the markers with the same numbers will move along the X-axis synchronously on all the traces. If the coupling feature is off, the position of the markers with same numbers along X-axis will be independent (See figure below).



Marker coupling feature



To enable/disable the marker coupling feature, use the following softkeys:



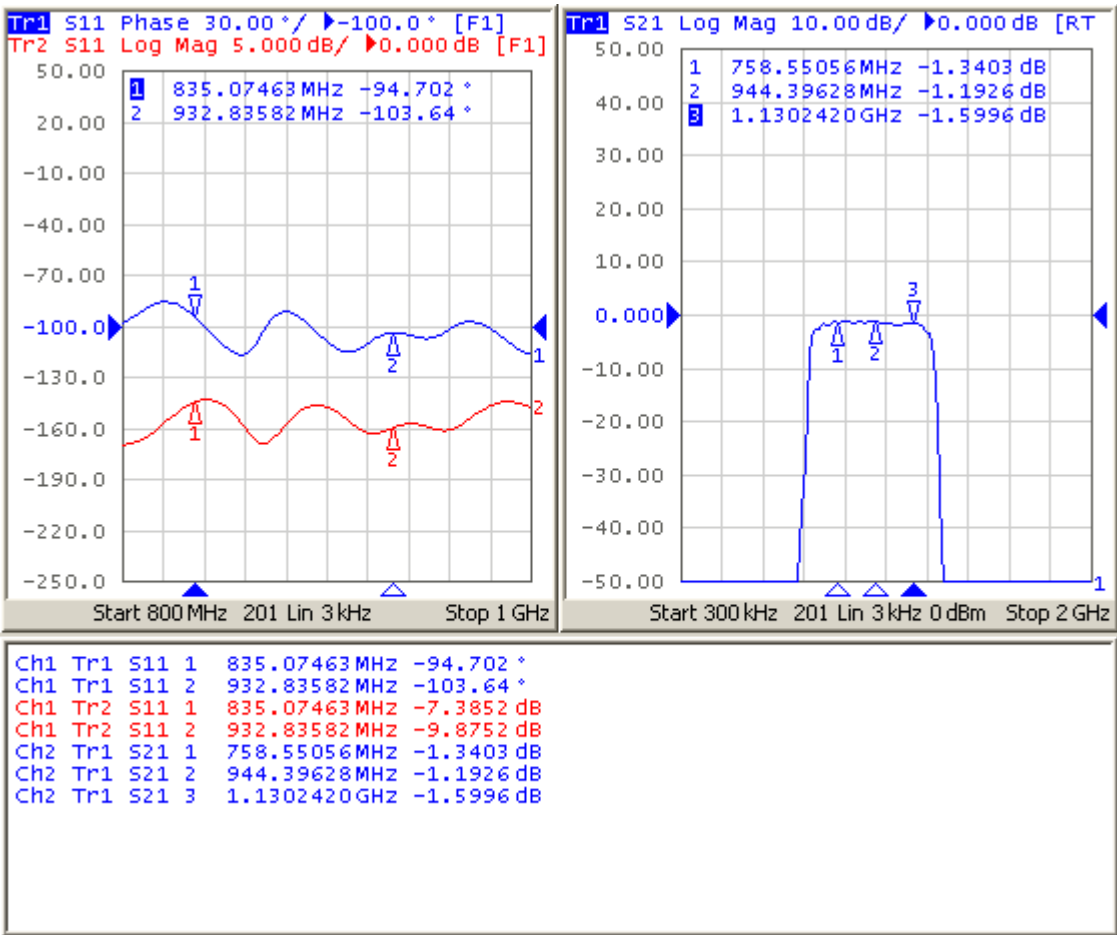
**Markers > Properties > Marker Couple**

[CALC:MARK:COUP](#)

Turns the marker coupling between traces ON/OFF.

# Marker Table

The marker table allows to view the values of the markers of all traces and channels (See figure below).



Marker table



To show/hide the marker table, use the following softkeys:

**Markers > Properties > Marker Table**



[DISP:MARKer:TABL](#)

Turns the marker table ON/OFF.

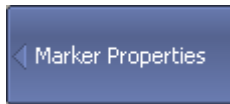
[DISP:PART:VIS](#)

Shows or hides the display partition specified by the <char> parameter.

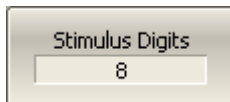


## Marker Value Indication Capacity

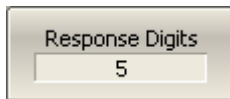
By default, the marker stimulus values are displayed with 8 decimal points and marker response values are displayed with 5 decimal points. These settings can be changed.



To set the marker value indication capacity, use the following softkeys:



**Markers > Properties > Stimulus Digits**



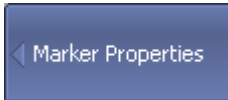
**Markers > Properties > Response Digits**

---

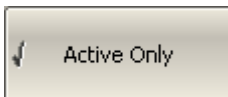


## Multi Marker Data Display

If several overlapping traces are displayed in one diagram, by default only active marker data is displayed on the screen. The display of the marker data for all traces can be enabled simultaneously. The markers for different traces can be distinguished by color. Each marker will be the same color as its trace.



To enable/disable the multi marker data display, toggle the softkey:



**Markers > Marker Properties > Active Only**

---

[DISP:WIND:ANN:MARK:SING](#)

Selects display of either the active trace markers or all trace markers.

---

### NOTE

When multi marker data display is enabled, to avoid data overlapping on the screen, arrange the marker data on the screen (See [Marker Data Arrangement](#)).

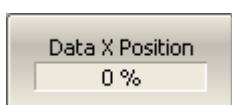
---

## Marker Data Arrangement

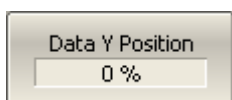
By default, marker data is displayed in the upper left corner of the screen. The marker data display can be rearranged on the screen. The marker data position on the screen is shown using two parameters - relative position on the X and Y axes, in percent. Zero percent is in the upper left corner, 100% is in the lower right corner. Marker data position for each trace is set separately. This allows to avoid data overlapping on the screen.



To arrange the marker data on the screen, enter the relative position on the X and Y axes, using the following softkeys:



**Markers > Marker Properties > Data X Position**



**Markers > Marker Properties > Data Y Position**

---

[DISP:WIND:TRAC:ANN:MARK:POS:X](#)

Sets or reads out the display position of the marker annotation on the X-axis by a percentage of the display width.

---

[DISP:WIND:TRAC:ANN:MARK:POS:Y](#)

Sets or reads out the display position of the marker annotation on the Y-axis by a percentage of the display height.

---

### NOTE

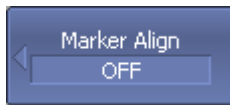
The marker data can also be drag-and-dropped using the mouse.

---

## Marker Data Alignment

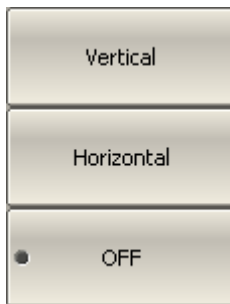
By default, marker data is displayed independently for each trace. The marker data display can be aligned on the screen. This alignment deactivates the independent marker data layout. In this case, the relative position on the X and Y axes is valid only for the first trace. The marker data of the other traces becomes aligned relatively to the first trace. Two types of alignment are available:

- Vertical — marker data of different traces are displayed one under another.
- Horizontal — marker data of different traces are displayed in line.



To set the marker data alignment, use the following softkeys:

**Markers > Marker Properties > Align > [Vertical | Horizontal | OFF]**



---

[DISP:WIND:ANN:MARK:ALIG](#)

Sets or reads out the alignment mode of the marker display position of each trace.

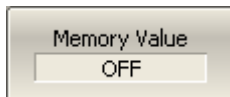
---

## Memory Trace Value Display

By default, the marker values of the data traces (not memory traces) are displayed on the screen. The display of memory trace maker values can be enabled, if a memory trace is available.



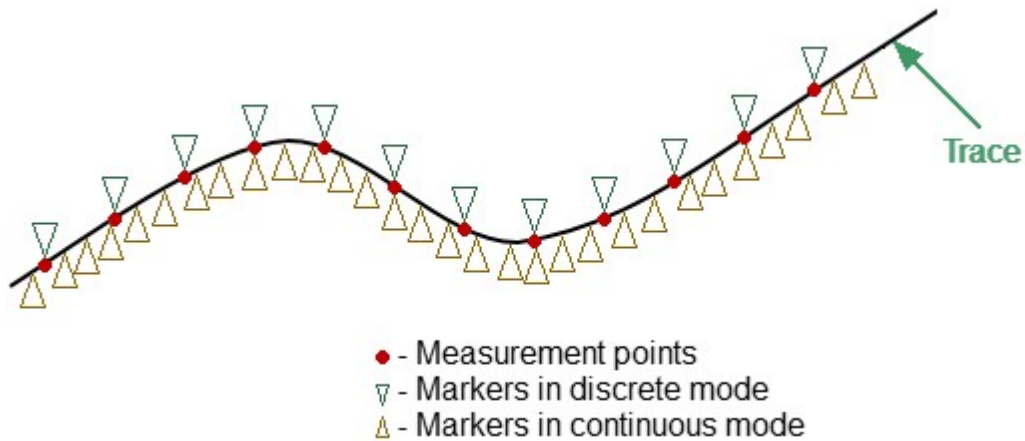
To enable/disable the display of memory trace marker values, toggle the softkey:



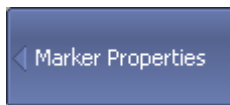
**Marker > Marker Properties > Memory Value [ON | OFF]**

## Marker Discrete Mode

By default, the marker can be moved along the values interpolated between measurement points. To move the marker only between actual measurement points, enable the marker discrete mode.

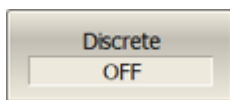


Marker Discrete and Continuous Modes



To enable / disable discrete mode, use the following softkeys:

**Marker > Marker Properties > Discrete [ON | OFF]**



[CALC:MARK:DISC](#)

Turns the marker discrete mode ON/OFF.

## Marker Position Search Functions

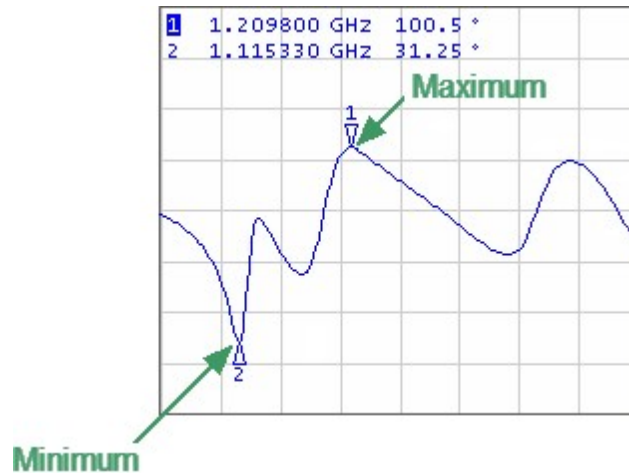
The marker position search function allows to find the following values on a trace:

- [Maximum value](#)
- [Minimum value](#)
- [Peak value](#)
- [Target level](#)

This section contains information about search tracking mode (See [Search Tracking](#)) and on the function used to set the search range of the marker position (See [Search Range](#)).

## Maximum and Minimum Search Functions

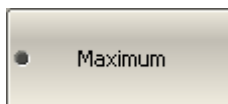
Maximum and minimum search functions are used to determine the maximum and minimum values of the measured parameter and move the marker to these positions on the trace (See figure below).



Maximum and minimum search



To find the maximum or minimum values on a trace, use the following softkeys:



**Markers > Marker Search > Maximum**



**Markers > Marker Search > Minimum**

[CALC:MARK:FUNC:EXEC](#)

Executes the marker search according to the specified criterion.

[CALC:MARK:FUNC:TYPE](#)

Selects the type of the marker search.

### NOTE

Activate the marker before starting maximum or minimum search (see [Marker Activation](#)).

In Smith chart and polar formats, the search is executed for the first marker value.



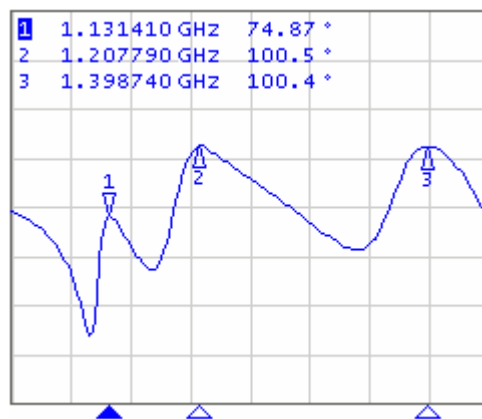


## Search for Peak

Peak search function is used to determine the peak value of the measured parameter and move the marker to this position on the trace.

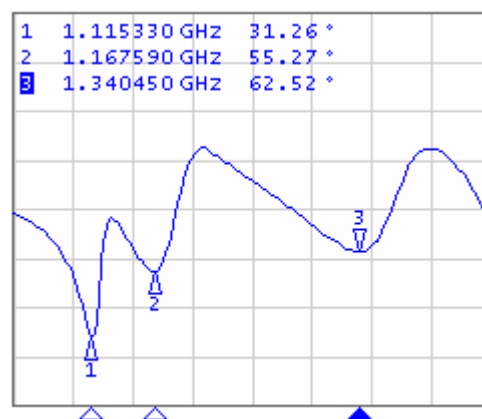
**Peak** is a local extreme of the trace.

Peak is considered **positive** if the value of the peak is greater than the values of the adjacent points (See figure below).



Positive peaks

Peak is considered **negative** if the value of the peak is smaller than the values of the adjacent points (See figure below).



Negative peaks

**Peak excursion** is the smallest of the absolute differences between the response values in the peak point and the two adjoining peaks of the opposite polarity.

The peak search is executed only for the peaks meeting the following conditions:

- The peaks must have the polarity (positive, negative, or both) specified by the user.
- The peaks must have a peak deviation no less than the value assigned by the user.

The following options for the peak search are available:

- search for nearest peak
- search for greatest peak
- search for left peak
- search for right peak

The nearest peak is a peak that is located most near to the current position of the marker along the stimulus axis.

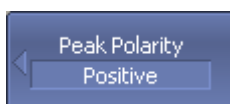
The greatest peak is a peak with maximum or minimum value, depending on the current polarity settings of the peak.

---

#### NOTE

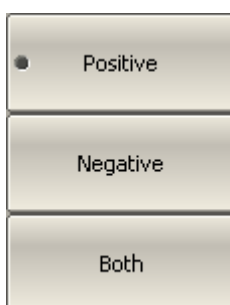
Finding the greatest peak is different from finding the maximum or minimum, as the peak cannot be located at the trace's limit points, even if those points have a maximum or minimum value.

---



To set the polarity of the peak, use the following softkeys:

**Markers > Marker Search > Peak > Peak Polarity > [ Positive | Negative | Both ]**




---

[CALC:MARK:FUNC:PPOL](#)

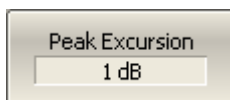
Selects the peak polarity when the marker peak search is performed by the [CALC:MARK:FUNC:EXEC](#) command.

---



To enter the peak excursion value, use the following softkeys:

**Markers > Marker Search > Peak > Peak Excursion**

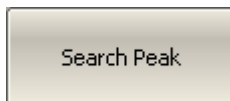


Then enter the value using the numerical keypad, or the «↑», «↓» arrows.

---

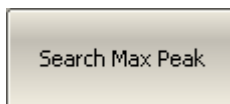
[CALC:MARK:FUNC:PEXC](#)

Sets or reads out the peak excursion value when the marker peak search is performed by the [CALC:MARK:FUNC:EXEC](#) command.



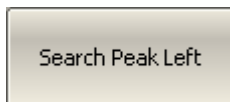
To activate the nearest peak search, use the following softkeys:

**Markers > Marker Search > Peak > Search Peak**



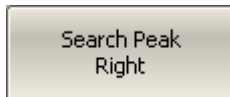
To activate the greatest peak search, use the following softkeys:

**Markers > Marker Search > Peak > Search Max Peak**



To activate the left peak search, use the following softkeys:

**Markers > Marker Search > Peak > Search Peak Left**



To activate the left peak search, use the following softkeys:

**Markers > Marker Search > Peak > Search Peak Right**

---

[CALC:MARK:FUNC:EXEC](#)

Executes the marker search according to the specified criterion.

---

[CALC:MARK:FUNC:TYPE](#)

Selects the type of the marker search, which is performed by the [CALC:MARK:FUNC:EXEC](#) command.

---

#### NOTE

Activate the marker before starting maximum or minimum search (see [Marker Activation](#)).

In Smith chart and Polar formats, the search is executed for the first marker value.

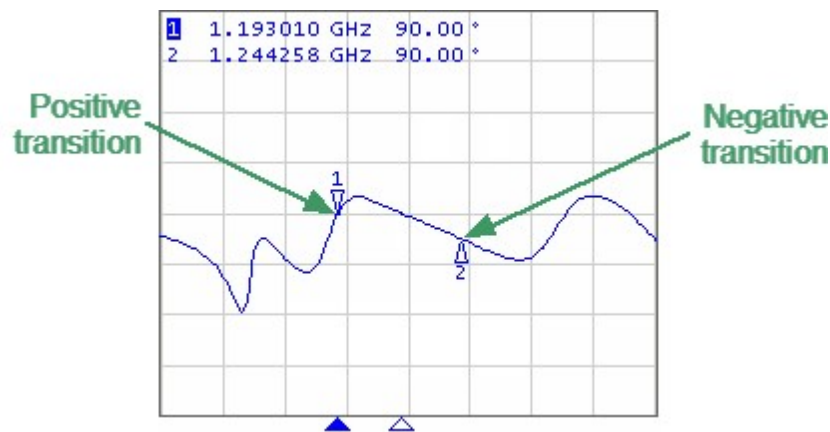
---

## Search for Target Level

The target level search function is used to locate the marker with the given level of the measured parameter (See figure below).

The trace can have two types of transition at the points where the target level crosses the trace:

- Transition type is positive if the function derivative (trace slope) is positive at the intersection point with the target level.
- Transition type is negative if the function derivative (trace slope) is negative at the intersection point with the target level.



Target level search

Target level search is performed only for intersection points that have a user-selected specific transition polarity (positive, negative, or both).

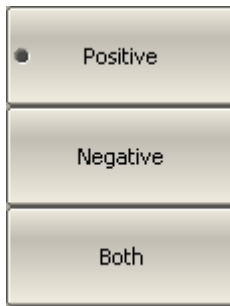
The following options for the target level search are available:

- search for nearest target
- search for left target
- search for right target



To set the transition polarity, use the following softkeys:

**Markers > Marker Search > Target > Target Transition > [ Positive | Negative | Both ]**



---

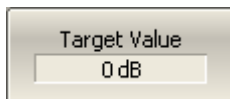
#### [CALC:MARK:FUNC:TTR](#)

Selects the type of the target transition when the marker transition search is performed by the [CALC:MARK:FUNC:EXEC](#) command.



To enter the target level value, use the following softkeys:

**Markers > Marker Search > Target > Target Value**

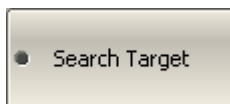


Then enter the value using the numerical keypad, or the «↑», «↓» arrows.

---

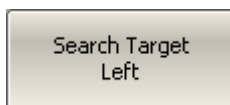
#### [CALC:MARK:FUNC:TARG](#)

Sets or reads out the target value when the marker target search is performed by the [CALC:MARK:FUNC:EXEC](#) command.



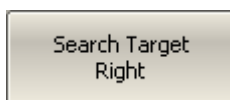
To activate the nearest target search, use the following softkeys:

**Markers > Marker Search > Target > Search Target**



To activate the left target search, use the following softkeys:

**Markers > Marker Search > Target > Search Target Left**



To activate the right target search, use the following softkeys:

**Markers > Marker Search > Target > Search Target Right**

---

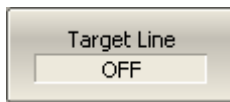
#### [CALC:MARK:FUNC:EXEC](#)

Executes the marker search according to the specified criterion.

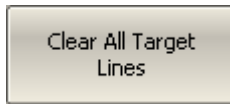
---

#### [CALC:MARK:FUNC:TYPE](#)

Selects the type of the marker search, which is performed by the [CALC:MARK:FUNC:EXEC](#) command.



To enable/disable target level indication on the screen, use the following softkeys:



**Markers > Marker Search > Target > Target Line**

The **Clear All Target Lines** softkey disables indication of target level lines of all the markers.

---

**NOTE**

Activate the marker before starting maximum or minimum search (see [Marker Activation](#)).

In Smith chart and Polar formats, the search is executed for the first marker value.

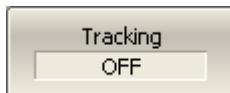
---

## Search Tracking

The marker position search function, by default, can be initiated by any press of the search key. Search tracking mode performs continuous marker position search, until this mode is disabled.



To enable/disable search tracking mode, use the following softkeys:



**Markers > Marker Search > Tracking**

---

[CALC:MARK:FUNC:TARG](#)

Sets or reads out the target value when the marker target search is performed by the [CALC:MARK:FUNC:EXEC](#) command.

---

## Search Range

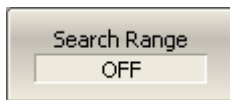
The search range for the marker position search can be set by setting the stimulus limits. This function includes the following additional features:

- Search range coupling, which allows to define the same search range for all the traces of a channel.
- Vertical line indication of the search range limits.



To enable/disable the search range, use the following softkeys:

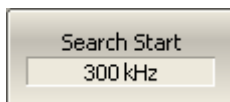
**Markers > Marker Search > Search Range**



---

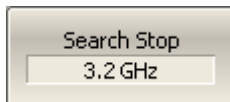
[CALC:MARK:FUNC:DOM](#)

Turns the state of the arbitrary range when executing the marker search ON/OFF.



To set the search range limits, use the following softkeys:

**Markers > Marker Search > Search Start**



**Markers > Marker Search > Search Stop**

---

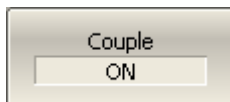
[CALC:MARK:FUNC:DOM:STAR](#)

Sets or reads out the start value of the marker search range.

---

[CALC:MARK:FUNC:DOM:STOP](#)

Sets or reads out the stop value of the marker search range.



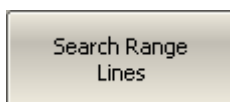
To enable/disable search range coupling, use the following softkeys:

**Markers > Marker Search > Couple**

---

[CALC:MARK:FUNC:DOM:COUP](#)

Turns the state of the marker search range coupling for different traces ON/OFF.



To enable/disable search range limits indication, use the following softkeys:





## Marker Math Functions

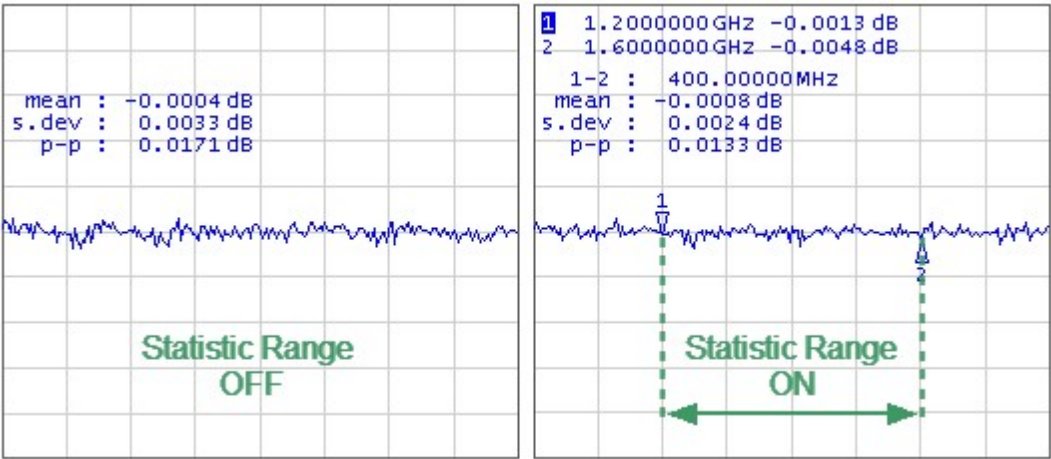
Marker math functions use markers to calculate various trace characteristics. Four marker math functions are available:

- [Statistics](#)
- [Bandwidth Search](#)
- [Flatness](#)
- [RF Filter](#)

# Trace Statistics

The trace statistics feature allows to determine and view trace parameters, such as mean, standard deviation, and peak-to-peak.

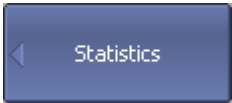
The range of trace statistics can be defined by two markers (See figure below).



Trace statistics

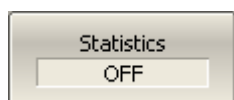
## Trace Statistics parameter

Symbol	Definition	Formula
mean	Arithmetic mean	$M = \frac{1}{N} \cdot \sum_{i=1}^N x_i$
s.dev	Standard deviation	$\sqrt{\frac{1}{N-1} \cdot \sum_{i=1}^N (x_i - M)^2}$
p-p	Peak-to-Peak: difference between the maximum and minimum values	Max – Min



To enable/disable trace statistics function, use the following softkeys:

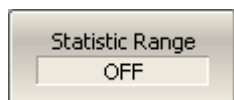
**Markers > Marker Math > Statistics > Statistics**



---

[CALC:MST](#)

Turns the math statistics display ON/OFF.



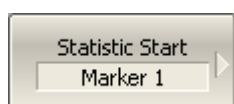
To enable/disable trace statistics range, use the following softkeys:

**Markers > Marker Math > Statistics > Statistic Range**

---

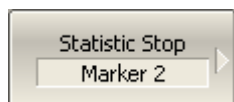
[CALC:MST:DOM](#)

Selects either the partial frequency range or the entire frequency range to be used for math statistic calculation.



To set the start/stop markers of the statistics range, use the following softkeys:

**Markers > Marker Math > Statistics > Statistic Start**



**Markers > Marker Math > Statistics > Statistic Stop**

---

[CALC:MST:DOM:STAR](#)

Sets or reads out the number of the marker, which specifies the start frequency of the math statistics range.

---

[CALC:MST:DOM:STOP](#)

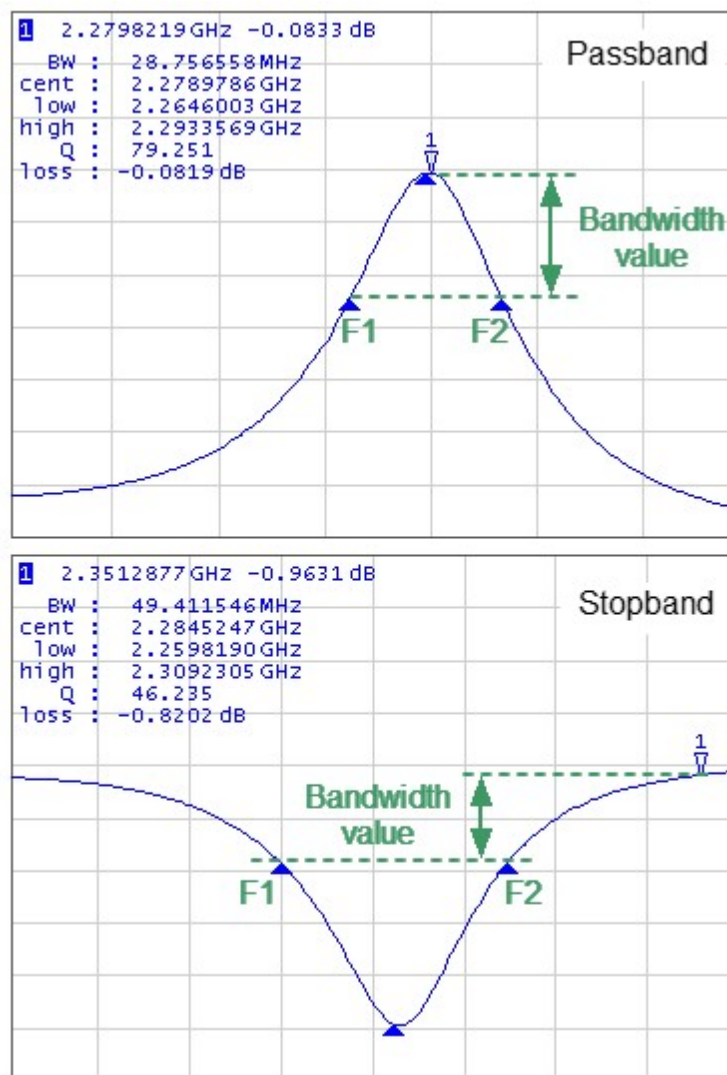
Sets or reads out the number of the marker, which specifies the stop frequency of the math statistics range.

---

## Bandwidth Search

The bandwidth search function allows to determine and view the following parameters of a passband or a stopband: bandwidth, center frequency, lower frequency, higher frequency, Q value, and insertion loss (See figure below).

The bandwidth search is executed from the reference point. The active marker or the maximum trace value can be selected as the reference. The bandwidth search function detects lower and higher cutoff frequencies that differ from the reference point response by a user-specified bandwidth value (usually – 3 dB).



F1 and F2 are the lower and higher cutoff frequencies of the band respectively

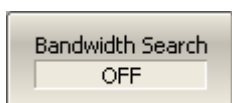
Bandwidth search

## Bandwidth parameters

Parameter Description	Symbol	Definition	Formula
Bandwidth	<b>BW</b>	The difference between the higher and lower cutoff frequencies.	$F2 - F1$
Center Frequency	<b>cent</b>	The midpoint between the higher and lower cutoff frequencies.	$(F1+F2)/2$
Lower Cutoff Frequency	<b>low</b>	The lower frequency point of the intersection of the bandwidth cutoff level and the trace.	$F1$
Higher Cutoff Frequency	<b>high</b>	The higher frequency point of the intersection of the bandwidth cutoff level and the trace.	$F2$
Quality Factor	<b>Q</b>	The ratio of the center frequency to the bandwidth.	$\text{cent}/\text{BW}$
Loss	<b>loss</b>	The trace measured value in the reference point of the bandwidth search.	—



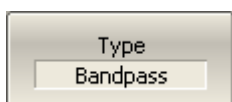
To enable/disable bandwidth search function, use the following softkeys:



**Markers > Marker Math > Bandwidth Search > Bandwidth Search**

[CALC:MARK:BWID](#)

Turns the bandwidth search function ON/OFF.



Set the bandwidth search type by softkeys:

**Markers > Marker Math > Bandwidth Search > Type**

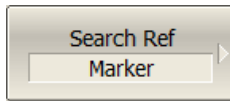
The type and the softkey label toggle between **Bandpass** and **Notch** settings.

---

[CALC:MARK:BWID:TYPE](#)

Sets the type of the bandwidth search function.

---



To set the search reference point, use the following softkeys:

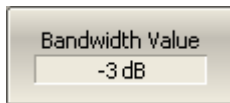
**Markers > Marker Math > Bandwidth Search > Search Ref > > [Marker | Maximum | Minimum]**

---

[CALC:MARK:BWID:REF](#)

Selects the reference point for the bandwidth search function: reference marker or absolute maximum value of the trace.

---



To enter the bandwidth value, use the following softkeys:

**Markers > Marker Math > Bandwidth Search > Bandwidth Value**

---

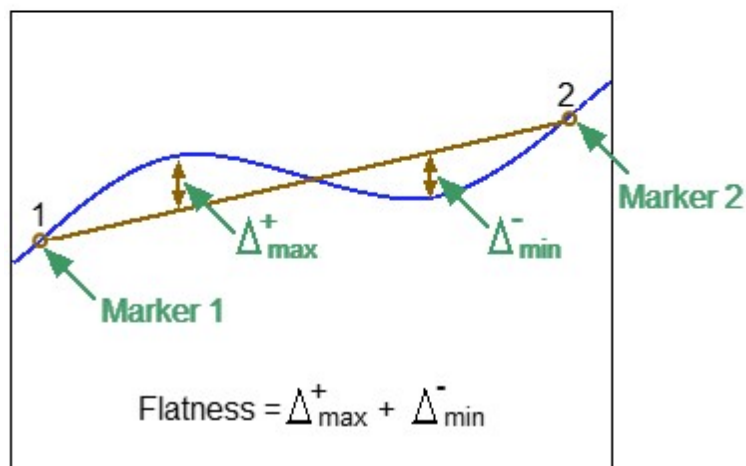
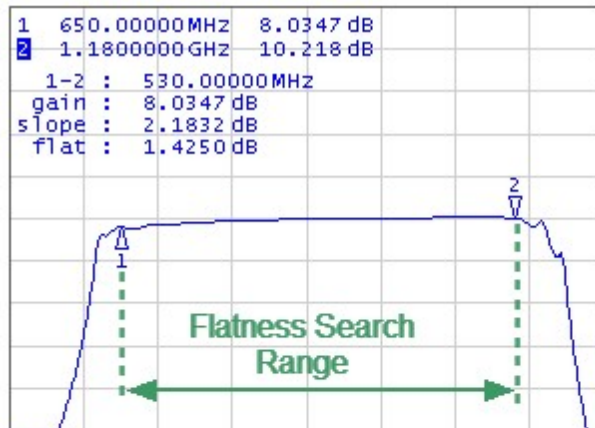
[CALC:MARK:BWID:THR](#)

Sets the bandwidth definition value.

---

## Flatness

The flatness search function allows to determine and view the following trace parameters: gain, slope, and flatness. Two markers to specify the flatness search range should be set (See figure below).

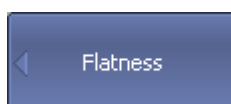


Flatness search



## Flatness parameters

Parameter Description	Symbol	Definition
Gain	<b>gain</b>	Marker 1 value.
Slope	<b>slope</b>	Difference between marker 2 and marker 1 values.
Flatness	<b>flat</b>	Sum of “positive” and “negative” peaks of the trace, which are measured from the line connecting marker 1 and marker 2 (See above figure).



To enable/disable the flatness search function, use the following softkeys:



**Markers > Marker Math > Flatness > Flatness**

---

[CALC:MARK:MATH:FLAT:STAT](#)

Turns the marker flatness function ON/OFF.



To select the markers specifying the flatness search range, use softkeys:



**Markers > Marker Math > Flatness > Flatness Start**

**Markers > Marker Math > Flatness > Flatness Stop**

---

[CALC:MARK:MATH:FLAT:DOM:STAR](#)

Sets or reads out the number of the marker, which specifies the start frequency of the flatness function domain.

---

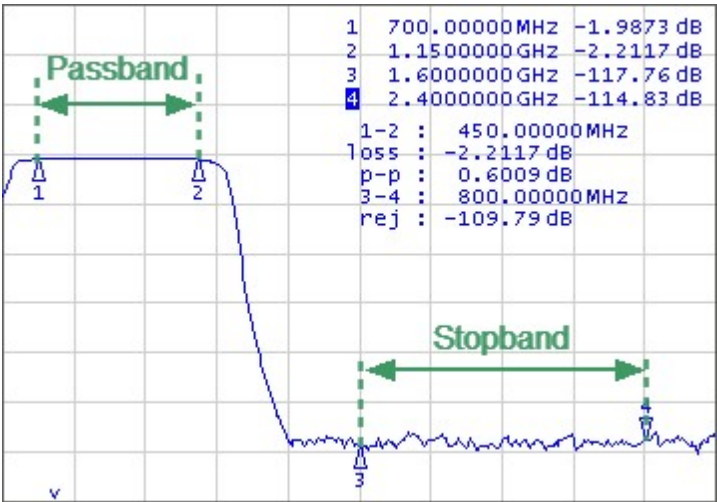
[CALC:MARK:MATH:FLAT:DOM:STOP](#)

Sets or reads out the number of the marker, which specifies the stop frequency of the flatness function domain.

---

# RF Filter Statistics

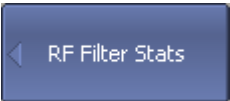
The RF filter statistics function allows to determine and view the following filter parameters: loss, peak-to-peak in a passband, and rejection in a stopband. The passband is specified by the first pair of markers, and the stopband is specified by the second pair of markers (See figure below).



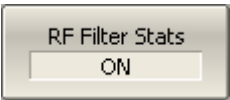
RF filter statistics

## RF filter statistics parameters

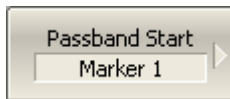
Parameter Description	Symbol	Definition
Loss in passband	loss	Minimum value in the passband.
Peak-to-peak in passband	p-p	Difference between maximum and minimum in the passband.
Reject	rej	Difference between maximum in stopband and minimum in passband.



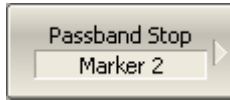
To enable/disable the RF filter statistics function, use the following softkeys:



**Markers > Marker Math > RF Filter Stats > RF Filter Stats**



To select the markers specifying the passband, use the following softkeys:

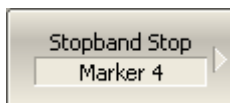


**Markers > Marker Math > RF Filter Stats > Passband Start**

**Markers > Marker Math > RF Filter Stats > Passband Stop**



To select the markers specifying the stopband, use the following softkeys:



**Markers > Marker Math > RF Filter Stats > Stopband Start**

**Markers > Marker Math > RF Filter Stats > Stopband Stop**

---

## Marker Functions

Using the current position of a marker, the following parameter settings can be set:

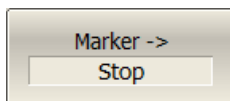
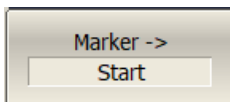
- stimulus start
- stimulus stop
- stimulus center
- reference level
- electrical delay

Activate the marker before adjusting these settings (see [Marker Activation](#)).



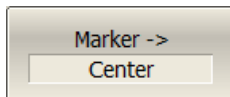
To set the stimulus start, use the following softkeys:

**Markers > Marker Functions > Marker→Start**



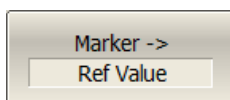
To set the stimulus stop, use the following softkeys:

**Markers > Marker Functions > Marker→Stop**



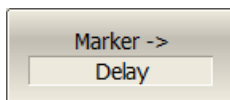
To set the stimulus center, use the following softkeys:

**Markers > Marker Functions > Marker→Center**



To set the reference level, use the following softkeys:

**Markers > Marker Functions > Marker→Ref Value**



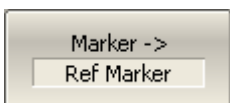
To set the electrical delay, use the following softkeys:

**Markers > Marker Functions > Marker→Delay**

---

[CALC:MARK:SET](#)

Sets the value of the specified item to the value of the position of the marker.



To set reference marker to the active marker point, use the following softkeys:

**Markers > Marker Functions > Marker→Ref Marker**

---



## Memory Trace Function

An associated memory trace can be created for each data trace. The memory trace is saved at the moment when the corresponding softkey is pressed or a program command is received. After saving the memory trace, the screen displays two traces — data and memory. The following settings of the memory and traces display can be performed:

Display	Trace status field
Data and memory	D&M
Memory only	M
Data only	Dat
Data and memory OFF	Off

---

### NOTE

Up to 8 memory traces can be created for each data trace. For a detail description see [Memory FIFO](#).

---

The memory trace is displayed in the same color as the main data trace, but it is half as bright (color and brightness of data and memory traces can be customized, see [User Interface Setting](#)).

The memory trace is used for displaying and mathematical operations with data trace. For a detail description see [Mathematical Operations](#).

In fact, complex measurement data is saved in memory, not their graphical representation. Consequently:

- Mathematical operations are carried out between the current and stored S-parameters.
- The memory trace changes similar to an associated data trace when the settings are changed, such as [Format](#), [Electrical delay](#), [Time domain](#), etc.

---

### NOTE

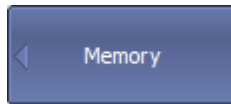
The memory trace cannot be extrapolated or interpolated, so when the frequency range or sweep type are changed, the memory contents become incorrect. When the number of points is changed, the memory is automatically cleared.

---

## Saving Data Trace into Memory

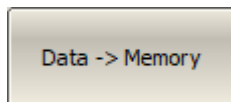
The function of saving data traces into memory is applied to individual trace or to all traces of the channel at once.

The trace to which the function is applied must be preselected as active (See [Selection of Active Trace/Channel](#)).

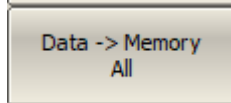


To save an active data trace into the memory, use the following softkeys:

**Display > Memory > Data->Memory**



To save all data traces into memory, use the following softkeys:



**Display > Memory > Data->Memory All**

---

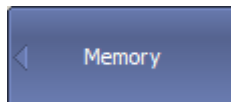
[CALC:MATH:MEM](#)

Copies the measurement data to the memory trace.

---

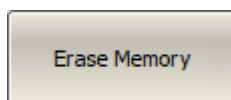
## Erasing Memory

The trace to which the function is applied must be preselected as active (See [Selection of Active Trace/Channel](#) ).



To erase the memory of the active trace, use the following softkeys:

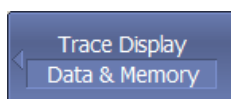
**Display > Memory > Erase Memory**



---

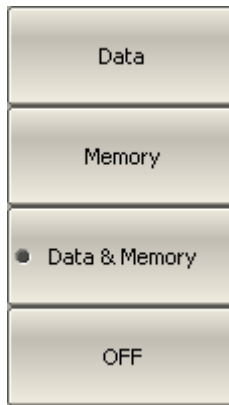
## Trace Display Setting

The trace to which the function is applied must be preselected as active (See [Selection of Active Trace/Channel](#)).



To set the type of traces to be displayed on the screen, use the following softkeys:

**Display > Trace Display > [ Data | Memory | Data & Memory | OFF ]**



---

[DISP:WIND:TRAC:MEM](#)

Turns the memory trace display ON/OFF.

---

[DISP:WIND:TRAC:STAT](#)

Turns the data trace display ON/OFF.

---



# Memory FIFO

The memory FIFO function increases the number of memory traces up to 8 for each data trace. Memory traces are saved in a FIFO (first-in-first-out) queue.

By default, the memory FIFO function is disabled, the queue depth is 1, so there is only 1 memory trace associated with each data trace. It is [Memory Trace Function](#).

When the memory FIFO function is enabled, the queue depth increases to 8, so it is possible to record up to 8 memory traces for each data trace.

Memory traces are saved in chronological order by pressing the **Data -> Memory** softkey. The new memory trace is numbered 1, and the numbers of the previous memory traces are increased by one. If the number of memory traces in the memory FIFO exceeds 8, the oldest trace is discarded.

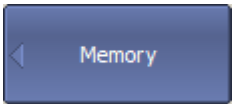
The number of traces currently saved in FIFO is displayed in the trace status bar (See figure below).



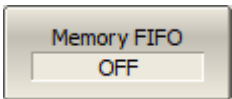
The number of traces saved in Memory FIFO

All memory traces contained in the memory FIFO are displayed simultaneously.

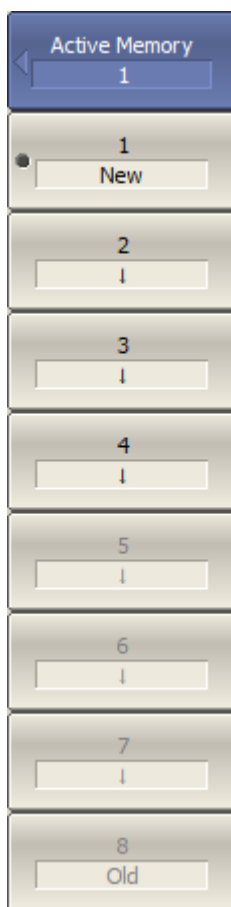
For math operations, only one of memory FIFO trace is used (such a trace is called active). By default, the newest memory trace is active. If necessary, any trace in the memory FIFO can be activated.



To enable / disable the function of saving to memory FIFO, use the following softkeys:



**Display > Memory > Memory FIFO > [ON | OFF]**



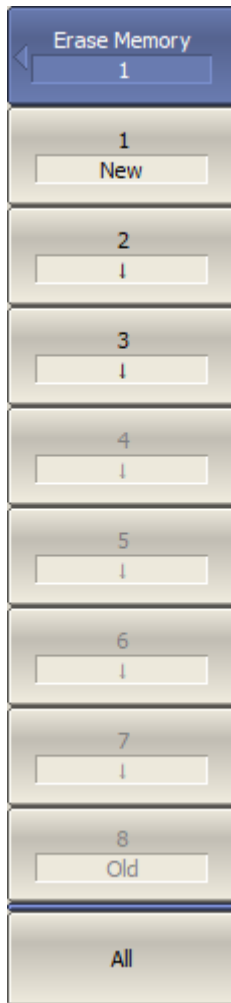
To assign a memory trace as active for math operations, use the following softkeys:

**Display > Memory > Active Memory > [ 1 | 2 | 3 ... 8 ]**

The memory traces in the FIFO are arranged in chronological order, where 1 is the newest save, 8 is the oldest.

## Erasing FIFO Memory

---



To erase the memory trace in FIFO, use the following softkeys:

**Display > Memory > Erase Memory > [ 1 | 2 | 3 ...8 | All ]**

## Mathematical Operations

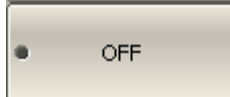
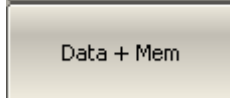
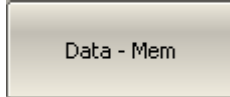
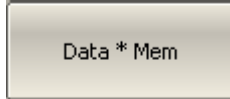
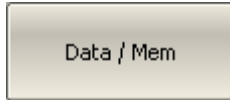
The memory trace can be used for mathematical operations with the data trace. The mathematical operations are performed on complex values before they are formatted for display. The result of math operation replaces the data trace. The following mathematical operations can be performed:

<b>Data/ Memory</b>	<p>Divides the measured data by the memory data.</p> <p>The trace status field indicates: <b>D/M</b>.</p>
<b>Data* Memory</b>	<p>Multiplies the measured data by the memory data.</p> <p>The trace status field indicates: <b>D*M</b>.</p>
<b>Data– Memory</b>	<p>Subtracts a memory data from the measured data.</p> <p>The trace status field indicates: <b>D–M</b>.</p>
<b>Data+ Memory</b>	<p>Adds the measured data to the memory data.</p> <p>The trace status field indicates: <b>D+M</b>.</p>
<b>Normalization</b>	<p>Pressing the <b>Normalization</b> softkey performs 3 steps in sequence:</p> <ol style="list-style-type: none"><li>1. Saves the current data into memory.</li><li>2. Turns on the math operation <b>Data/ Memory</b> (normalizes the measured data).</li><li>3. Turns on "data only" display type.</li></ol> <p>The trace status field indicates: <b>D/M Dat</b>.</p>



To access math operations, use the following softkeys:

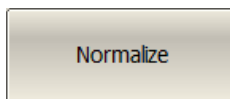
**Display > Memory > Data Math > [ Data / Mem | Data \* Mem | Data – Mem | Data + Mem | OFF ]**



---

#### CALC:MATH:FUNC

Selects the math operation between the measured data and the memory data.



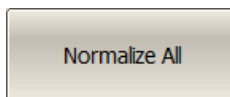
Normalization function can be done using the following softkeys:

- for one trace:

**Display > Memory > Normalize**

- for all traces of the active channel:

**Display > Memory > Normalize All**



---

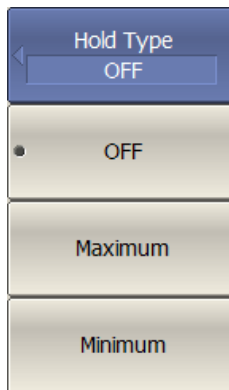
#### NOTE

If the memory FIFO function is turned on, check if the active memory trace is the desired trace for math operation (See [Memory FIFO](#))

---

## Trace Hold

The trace hold function is used to hold the maximum or minimum values of the trace. When the function is enabled, the inscription **[Max hold]** or **[Min hold]** appears in the trace status bar (See [Trace Status Field](#)).



To turn ON/OFF trace hold function press the following softkeys:

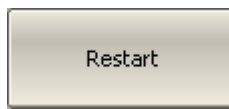
**Display > Trace Hold > Hold Type**

Select the hold type:

- **OFF**
- **Maximum**
- **Minimum**

[CALC:HOLD:TYPE](#)

Sets the type of the trace hold function.



The **Restart** softkey in the **Display> Trace Hold** menu is used to restart the trace hold.

[CALC:HOLD:CLEar](#)

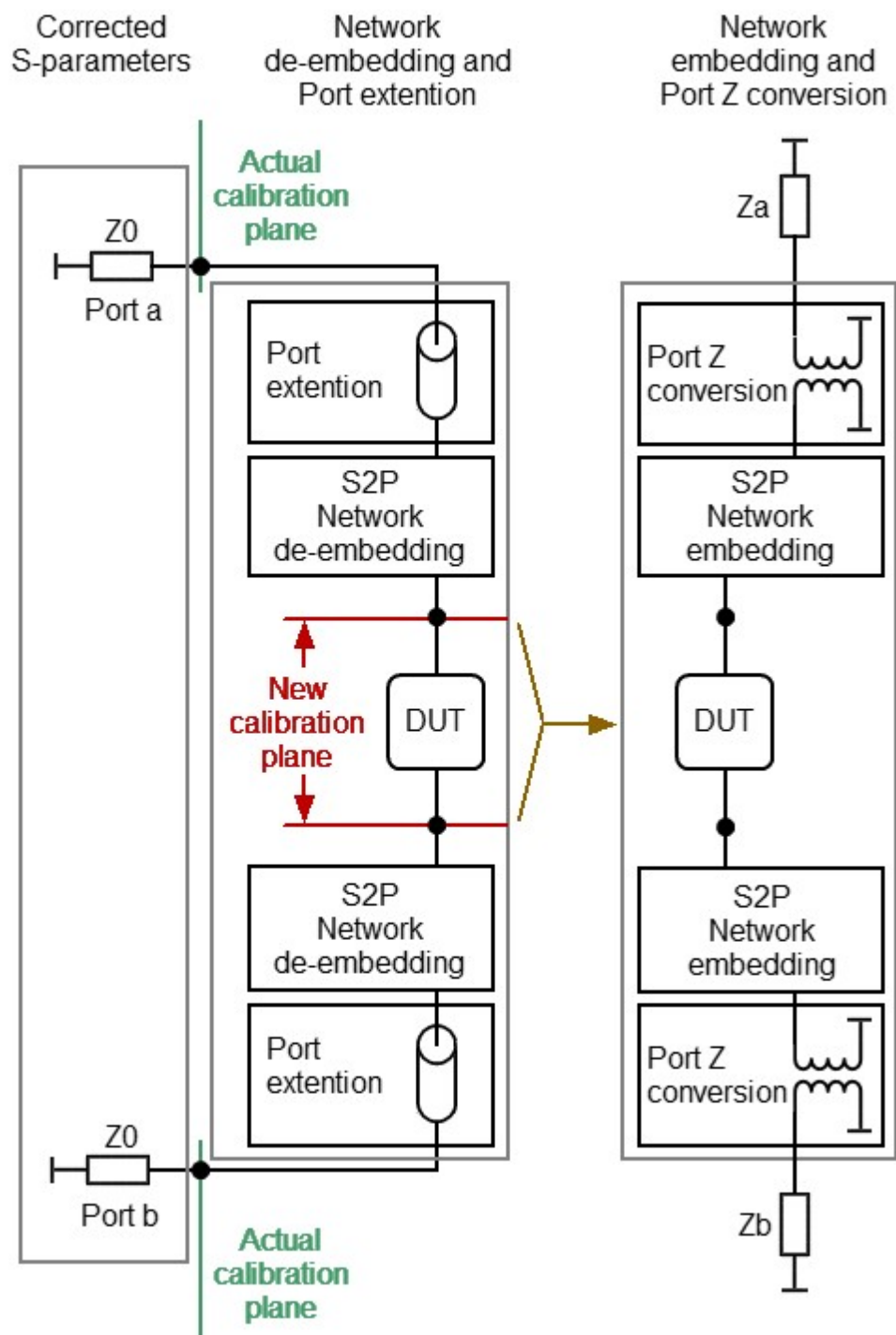
This command resets the trace hold function.

## Fixture Simulation

The fixture simulation functions are a set of software functions for mathematically simulating measurement conditions that are different from the actual measurement conditions. The following conditions can be simulated:

- [Port Extension](#) or [Automatic Port Extension](#)
- [Port reference impedance conversion](#)
- [Circuit de-embedding](#)
- [Circuit embedding](#)

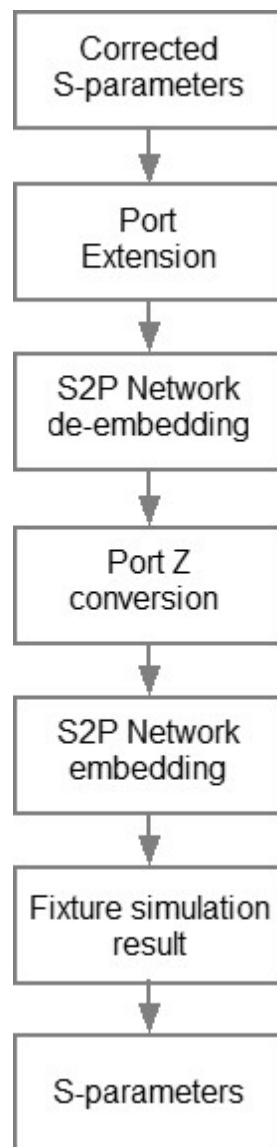
The logic diagram of the fixture simulation function is shown in the figure below.



Logic diagram of fixture simulation function

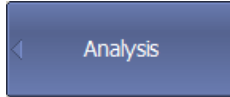
The data processing flow diagram of the fixture simulation feature is shown in the the figure below.



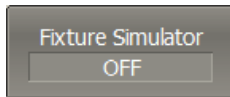


Data processing flow diagram of fixture simulation function

The channel to which the function is applied must be preselected as active (See [Selection of Active Trace/Channel](#)). Fixture simulation functions affect all the traces of the channel.



To enable fixture simulation functions , use the following softkeys:



**Analysis > Fixture Simulator**

---

[CALC:FSIM:STAT](#)

Turns the fixture simulation function ON/OFF.

---

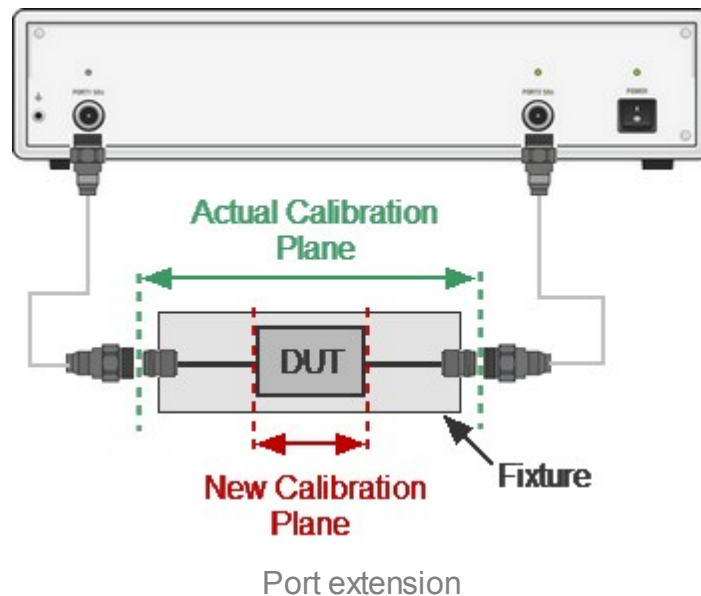
**NOTE**

The fixture simulator softkey enables or disables fixture simulation in general. Each fixture simulator function has its own on/off softkey.

---

## Port Extension

The port extension function moves the calibration plane toward the DUT terminals by the specified electrical delay value. The function is useful when a fixture is used for the DUT connecting and the calibration cannot be performed at the DUT terminals. The calibration plane can be established at coaxial connectors of the fixture and then moved to the DUT terminals using the port extension function (See figure below).



The function uses the model of the perfectly matched transmission line with loss with parameters:

- The phase incursion in the line is

$$\Delta\varphi = e^{-j \cdot 2\pi \cdot f \cdot \tau},$$

where  $f$  - frequency, Hz,

$\tau$  - electrical delay, sec.

- The loss of the line  $L(f)$  can be specified by one of the following methods:

1. Frequency-independent loss at DC ( $L_0$ )

$$L(f) = L_0.$$

2. Loss determined by the losses in two frequency points ( $L_0$  at DC, and  $L_1$  at frequency  $F_1$ )

$$L(f) = L_0 + (L_1 - L_0) \sqrt{\frac{f}{F_1}}$$

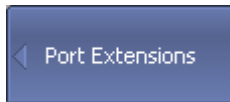
3. Loss determined by the losses in three frequency points ( $L_0$  at DC,  $L_1$  at frequency  $F_1$  and  $L_2$  at frequency  $F_2$ )

$$L(f) = L_0 + (L_1 - L_0) \left( \frac{f}{F_1} \right)^n$$

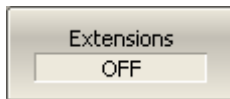
$$n = \frac{\log \left| \frac{L_1}{L_2} \right|}{\log \frac{F_1}{F_2}}$$

#### NOTE

The accuracy of the port extension method depends on the fixture used. The closer the fixture parameters are to the model of a perfectly matched transmission line, the higher the accuracy.



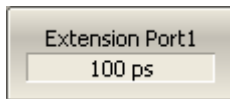
To enable the port extension function, use the following softkeys:



**Calibration > Port Extensions > Extensions [ ON|OFF]**

[SENS:CORR:EXT](#)

Turns the port extension function ON/OFF.

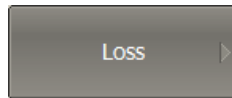


To set the electrical delay for each port, use the following softkeys:

**Extension Port1 or Extension Port2**

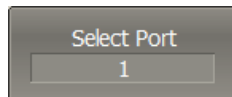
[SENS:CORR:EXT:PORT:TIME](#)

Sets or reads out the electrical delay value for the port extension function.



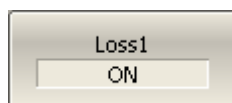
To open the menu of the losses, use the following softkeys:

**Calibration > Port Extensions > Loss**



To select the port in menu, use softkey:

**Select Port**



Enter the  $L_1$ ,  $F_1$  values and enable the use of these values in further calculations using the following softkeys:

**Loss1 [ ON | OFF ]**

Loss1  
-1 dB

**Loss1**

Freq1  
1 GHz

**Freq1**

Perform the same steps for  $L_2, F_2$ .

Loss at DC  
0 dB

Enter the  $L_0$  value using the following softkey:

**Loss at DC**

---

[SENS:CORR:EXT:PORT:INCL](#)

Turns the loss compensation of loss 1 and loss 2 for the port extension function ON/OFF.

---

[SENS:CORR:EXT:PORT:FREQ](#)

Sets or reads out the values of the frequency 1 and frequency 2 to calculate the loss for the port extension function.

---

[SENS:CORR:EXT:PORT:LOSS](#)

Sets or reads out the values of loss 1 and loss 2 for the port extension function.

---

[SENS:CORR:EXT:PORT:LDC](#)

Sets or reads out the loss value at DC for the port extension function.

---

## Automatic Port Extension

The auto port extension function allows for automatic calculation of port extension parameters by measuring a SHORT or an OPEN standard. It is also possible to measure both standards; in this case the average value will be used.

The auto port extension function can be used simultaneously for any number of ports from 1 to the number of actual instrument ports. First select the number of ports and then connect SHORT or OPEN standards to the chosen ports.

In the auto port extension menu, specify the frequency range, which will be taken into account when calculating the port extension parameter. There are three methods of setting the frequency range:

- Current frequency range.
- User-defined frequency range (within current range).
- User-defined frequency point (selected with a marker).

The result of the auto port extension function is the calculation of the electrical delay value. After auto port extension completes, this delay value appears in the corresponding field of the **Port Extension** menu, and the [port extension](#) function is automatically enabled, if it was disabled.

If the option **Include Loss** is enabled prior to the auto port extension function running, the loss values **Loss1**, **Loss2** at the respective frequency values **Freq1**, **Freq2** will be calculated and applied. The **Freq1**, **Freq2** values are calculated as  $\frac{1}{4}$  and  $\frac{3}{4}$  of the frequency range set by one of the following two methods: current or user defined. If the frequency range is defined by a marker, frequency point **Freq2** is not calculated.

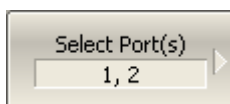
If the **Adjust Mismatch** option is enabled prior to the auto port extension function running, the frequency-independent loss at DC, the **Loss at DC** value, is also set. The value of loss at the lower frequency of the current range is used as the **Loss at DC** value.



To open the menu of the auto port extension function, use the following softkeys:

**Calibration > Port Extensions > Auto Port Extension**

---



Then select the number of ports:

**Select Port(s)**



Select the frequency range:

---

**Method [Current Span | Active Marker | User Span]**

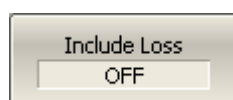
---

[SENS:CORR:EXT:AUTO:PORT](#)

Turns the status of the auto port extension for the port number <Pt> ON/OFF.

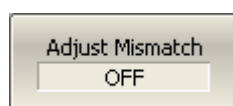
[SENS:CORR:EXT:AUTO:CONF](#)

Specifies the frequency range used for calculation of the results of the auto port extension function.



Enable the include loss function **Loss1**, **Loss2**, if required:

**Include Loss [ON | OFF]**



Enable adjust mismatch function **Loss at DC**, if required:

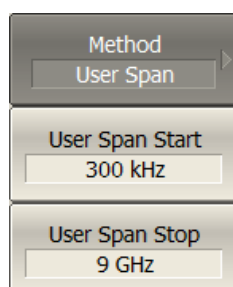
**Adjust Mismatch [ON | OFF]**

[SENS:CORR:EXT:AUTO:LOSS](#)

Turns the usage of "Loss1" and "Loss2" values for the results of the auto port extension function ON/OFF.

[SENS:CORR:EXT:AUTO:DCOF](#)

Turns the usage of "Loss at DC" value for the results of the auto port extension function ON/OFF.



When using **User Span** method, select frequency range, using the following softkeys:

**User Span Start**

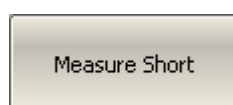
**User Span Stop**

[SENS:CORR:EXT:AUTO:STAR](#)

Sets or reads out the start value of the user span of the auto port extension function.

[SENS:CORR:EXT:AUTO:STOP](#)

Sets or reads out the stop value of the user span of the auto port extension function.



Execute the auto port extension function after connecting SHORTs or OPENs to the ports:

**Measure Short or Measure Open**



If both measurements have been performed, the result will appear as the average value of the two.

---

[SENS:CORR:EXT:AUTO:RES](#)

Deletes the finished measurement data of the OPEN and SHORT standards of the auto port extension function.

---

[SENS:CORR:EXT:AUTO:MEAS](#)

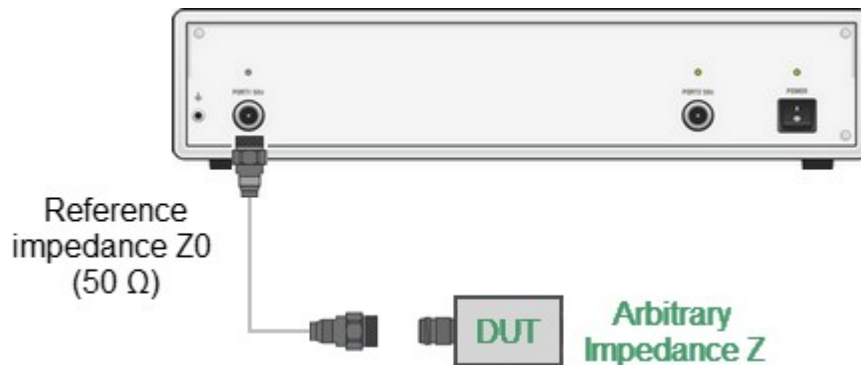
Performs measurement of the standard SHORT or OPEN, automatically calculates and sets the parameters of the port extension.

---



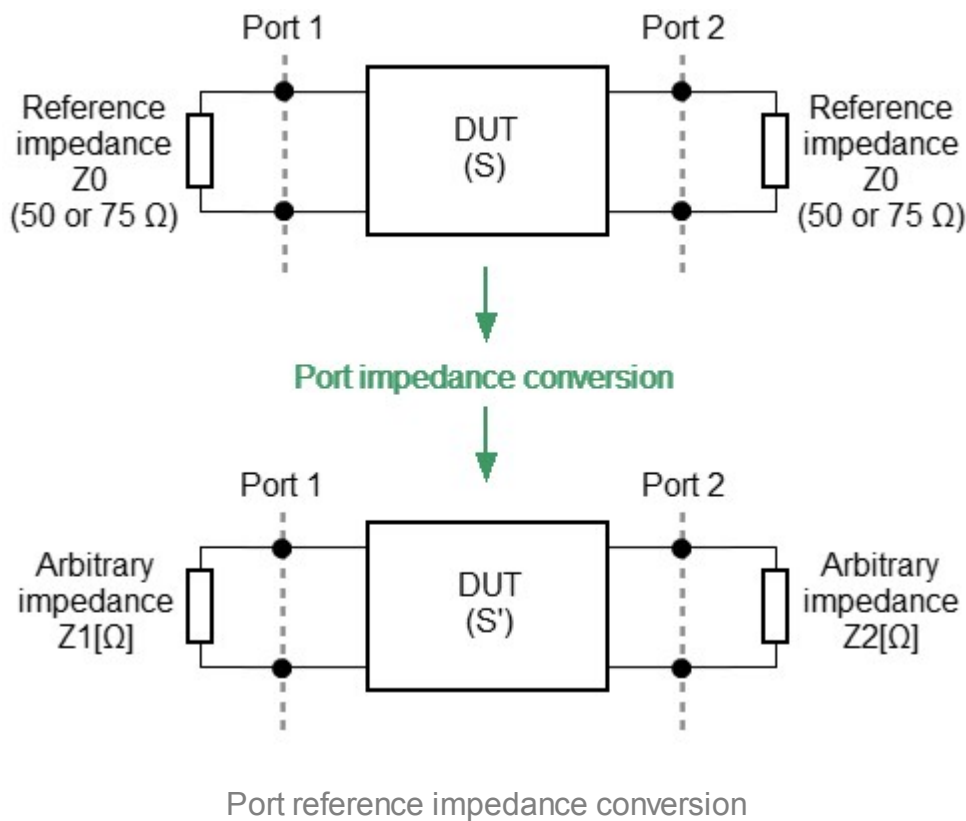
## Port Reference Impedance (Z) Conversion

The default reference impedance of a port is equal to the reference impedance of the connectors (50 or 75  $\Omega$ ). But in the process, it is often required to measure DUT with arbitrary resistance (See example in the figure below), not equal to the reference impedance of a port. In this case, it is possible to convert the reference impedance to an arbitrary impedance value using the program.



Example of measuring a DUT with an arbitrary impedance by the Analyzer with reference impedance 50  $\Omega$

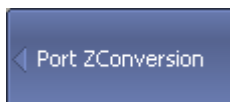
Port reference impedance conversion is a function that mathematically converts the matrix of S-parameters measured at the reference impedance of port Z0 to the matrix of S-parameters measured at an arbitrary impedance of port Z1 (See figure below). The function is also referred to as the renormalization transformation of S-parameters.



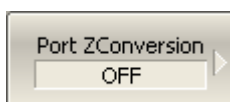
Renormalization can be based on two alternative microwave circuit theories, whose conversion formulas may yield different results if the reference impedance of at least one test port has a non-zero imaginary part. The first theory is "A General Waveguide Circuit Theory" (R.B.Marks and D.F.Williams). The second theory is the "Power waves and the Power Scattering Matrix" (K.Kurokawa).

#### NOTE

The source value of the  $Z_0$  port reference impedance (commonly 50  $\Omega$ ) is defined in the process of the calibration. It is determined by the characteristic impedance of the calibration kit and its value is entered as described in [System Impedance  \$Z\_0\$](#) .



To enable/disable the port reference impedance conversion function, use the following softkeys:

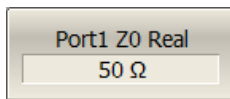


**Analysis > Fixture Simulator > Port Z Conversion > Port Z Conversion [ON | OFF]**

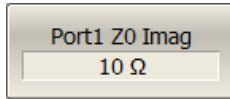
[CALC:FSIM:SEND:ZCON:STAT](#)

Turns the port impedance conversion function ON/OFF.

---

A rectangular softkey with a light gray background. The top half contains the text "Port1 Z0 Real" in a small, dark font. The bottom half contains a text entry field with "50 Ω" inside.

To enter the value of the simulated impedance of Port n, use the **Port n Z0 Real** and **Port n Z0 Imag** softkeys.

A rectangular softkey with a light gray background. The top half contains the text "Port1 Z0 Imag" in a small, dark font. The bottom half contains a text entry field with "10 Ω" inside.

---

[CALC:FSIM:SEND:ZCON  
:PORT:Z0](#)

Sets or reads out the value of the impedance of the port impedance conversion function.

---

[CALC:FSIM:SEND:ZCON  
:PORT:Z0:REAL](#)

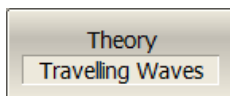
Sets or reads out the real part of the impedance of the port impedance conversion function.

---

[CALC:FSIM:SEND:ZCON  
:PORT:Z0:IMAG](#)

Sets or reads out the imaginary part of the impedance of the port impedance conversion function.

---

A rectangular softkey with a light gray background. The top half contains the text "Theory" in a small, dark font. The bottom half contains a text entry field with "Travelling Waves" inside.

To choose the theory according to which the renormalization of S-parameters is performed, use the **Theory** softkey.

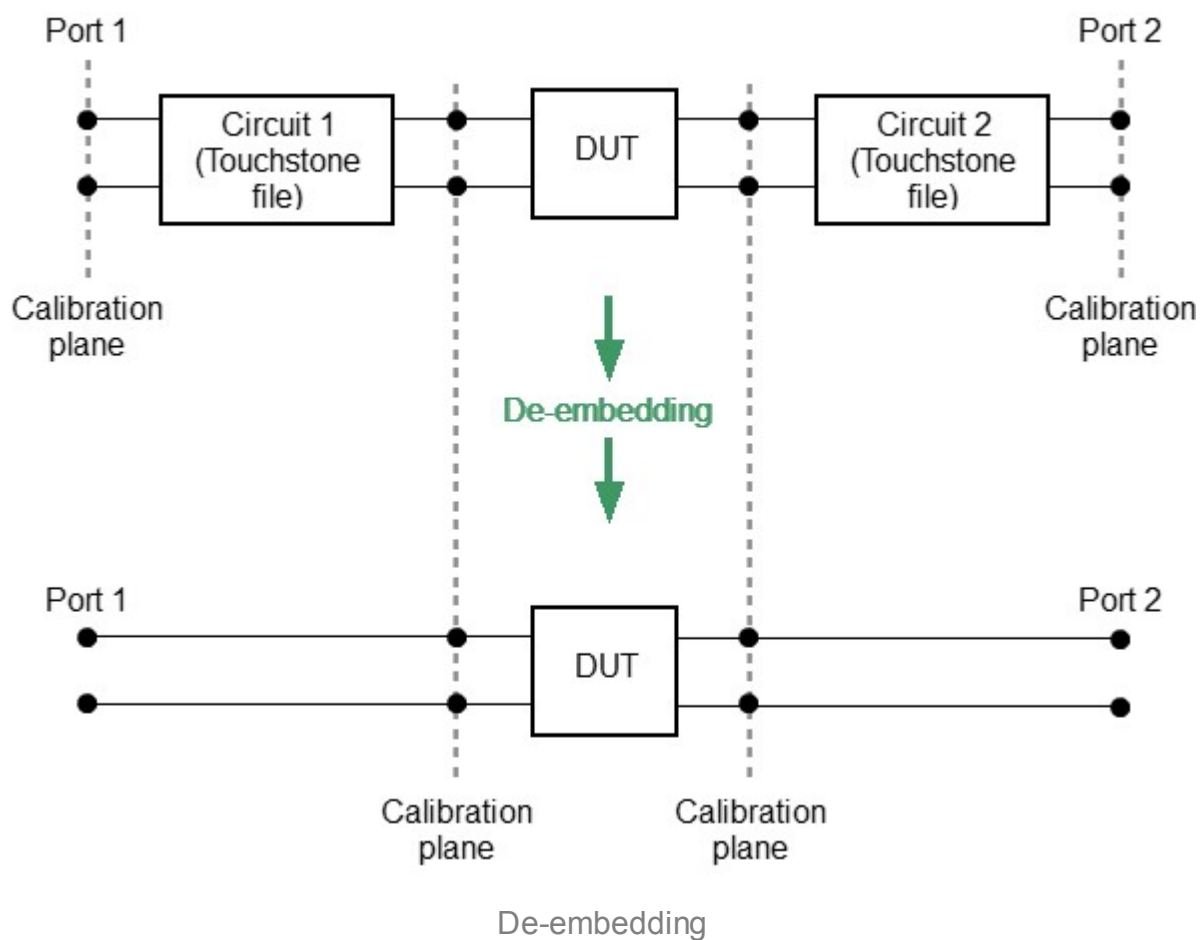
---

## De-embedding

De-embedding is a function of transforming the S-parameter by eliminating some circuit effect from the measurement results.

The de-embedding function allows to mathematically exclude the effect of the fixture circuit existing between the calibration plane and the DUT in the real network from the measurement results. The fixture is used for the DUTs, which cannot be directly connected to the test ports.

The de-embedding function shifts the calibration plane closer to the DUT, so as if the calibration has been executed on the network with this circuit removed (See figure below).



The circuit being removed should be defined in the data file containing S-parameters of that circuit. The circuit should be described as 2-port in Touchstone file (extension \*.S2P), which contains the S-parameter table: S11, S21, S12, S22 for a number of frequencies.

---

**NOTE**

The S-matrices of all de-embedding circuits are oriented so that the S11 is directed to the Analyzer port and S22 directed to the DUT.

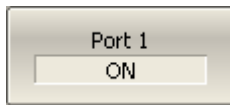
---



To enable/disable the de-embedding function, use the following softkeys:

**Analysis > Fixture Simulator > De-Embedding > De-Embedding [ON | OFF]**

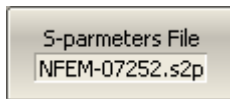
---



To enable/disable the de-embedding function for Port n, use the following softkeys:

**Analysis > Fixture Simulator > De-Embedding > Port n [ON | OFF]**

---



To enter the file name of the de-embedded circuit S-parameters of Port n, use the following softkeys:

**Analysis > Fixture Simulator > De-Embedding > S-parameters File**

---

[CALC:FSIM:SEND:DEEM  
:STAT](#)

Turns the 2-port network de-embedding function ON/OFF.

---

[CALC:FSIM:SEND:DEEM  
:PORT:USER:FIL](#)

Sets or reads out the name of the \*.S2P file of the de-embedded circuit of the 2-port network de-embedding function.

---

[CALC:FSIM:SEND:DEEM  
:PORT:STAT](#)

Turns the 2-port network de-embedding function for specified port ON/OFF.

---

---

**NOTE**

If the S-parameters file is not specified, the softkey will be grayed out.

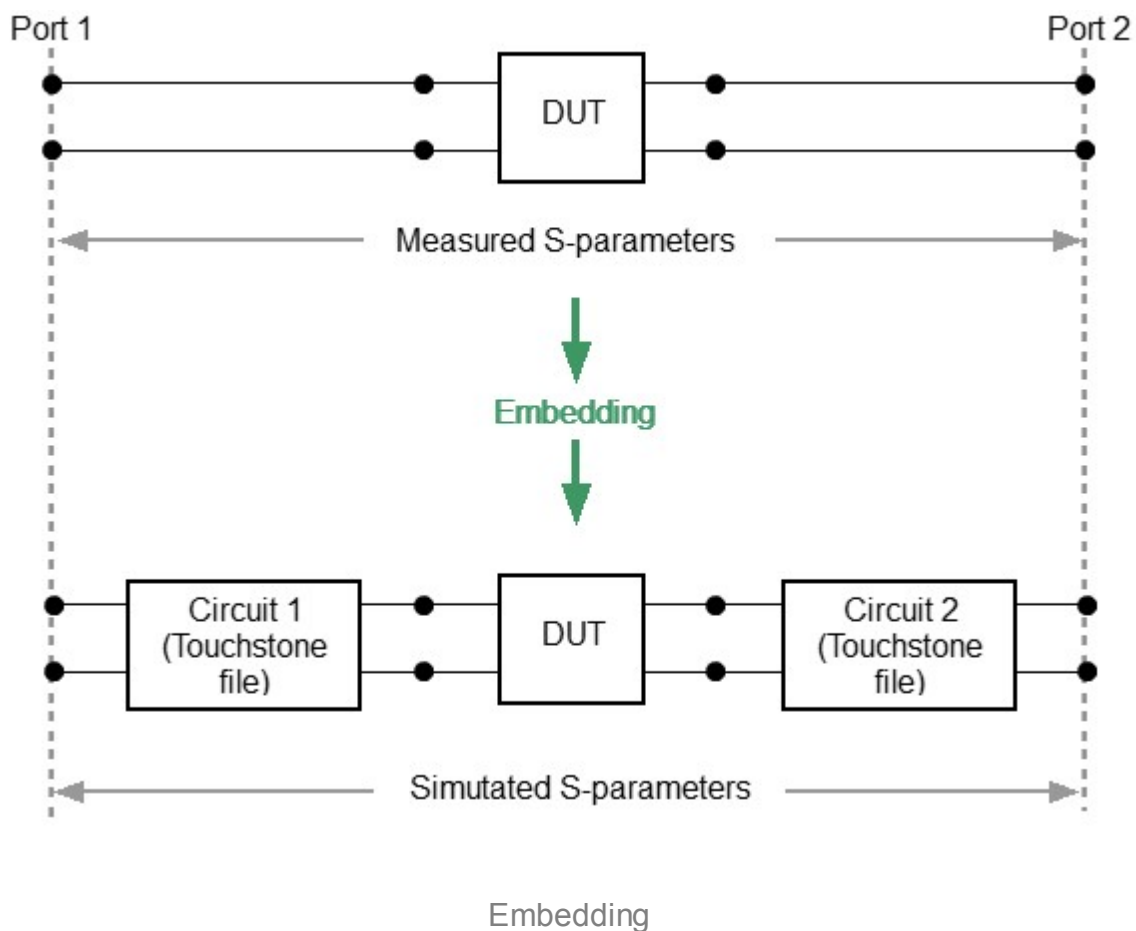
---

## Embedding

Embedding is a function of the S-parameter transformation via integration of some virtual circuit into the real network (See figure below).

The embedding function allows to mathematically simulate the DUT parameters after adding the fixture circuits.

The embedding function is an inverted [de-embedding function](#).



The circuit being integrated should be defined in the data file containing S-parameters of that circuit. The circuit should be described as a 2-port in Touchstone file (extension \*.S2P), which contains the S-parameter table: S11, S21, S12, S22 for a number of frequencies.

---

### NOTE

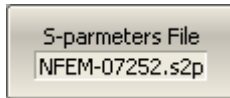
The S-matrices of all embedding circuits are oriented so that the S11 is directed to the Analyzer port and S22 directed to the DUT.

---

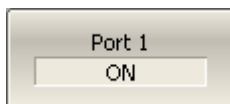


To enable/disable the embedding function, use the following softkeys:

**Analysis > Fixture Simulator > Embedding > Embedding [ON | OFF]**



If the S-parameters file is not specified, the softkey for Port n activation will be grayed out.



To enter the file name of the embedded circuit S-parameters of Port n, use the following softkeys:

**Analysis > Fixture Simulator > Embedding > S-parameters File**

To enable/disable the embedding function for Port n, use the following softkeys:

**Analysis > Fixture Simulator > Embedding > Port n.**

---

[CALC:FSIM:SEND:PMC:STAT](#)

Turns the 2-port network embedding function ON/OFF.

---

[CALC:FSIM:SEND:PMC:PORT:USER:FILE](#)

Sets or reads out the name of the \*.S2P file of the embedded circuit of the 2-port network embedding function.

---

[CALC:FSIM:SEND:PMC:PORT:STAT](#)

Turns the 2-port network embedding function for each port ON/OFF.

---

## Time Domain Transformation

---

### NOTE

The availability of this feature depends on the Analyzer model (See corresponding [datasheet](#)).

---

The Analyzer measures parameters of the DUT in the frequency domain. Time domain transformation is a function of mathematical transformation of the measured parameters in order to obtain the time domain representation.

The time domain function allows to select the following transformation types:

- **Bandpass** mode simulates the response of the bandpass network to the impulse.
- **Lowpass impulse** mode simulates the response of the lowpass network to the impulse.
- **Lowpass step** mode simulates the response of the lowpass network to the unit step function.

Bandpass mode is applied to the DUTs that do not operate with DC current such as band pass filters. The frequency settings can be arbitrary.

Lowpass mode is applied to the DUTs that operate with DC current such as cables. The DC value is extrapolated from the first few frequency points, however a user has ability to set the DC value manually. The frequency settings is required to be a harmonic frequency grid, where the frequency value at each frequency point is an integer multiple of the start frequency.

The time domain resolution in the lowpass mode is twice as high as in the bandpass mode.

The time domain domain is a periodic function due to the discrete nature of the frequency response. The time domain ambiguity range is determined by the step in the frequency domain:

$$\Delta T = \frac{1}{\Delta F}, \quad \Delta F = \frac{F_{max} - F_{min}}{N - 1}$$

The time domain response has a ringing due to the finite nature of the frequency response. To reduce the ringing the windowing is applied to the frequency response. The time domain transformation function applies the Kaiser window function. The window function selection is a tradeoff between the ringing reducing and the time domain resolution.



The Kaiser window is defined by the  $\beta$  parameter, which smoothly fine-tunes the window shape from minimum (rectangular) to maximum. The user can fine-tune the window shape, or select one of the three pre-programmed windows:

- **Minimum** (rectangular)
- **Normal**
- **Maximum**

#### Preprogrammed window types

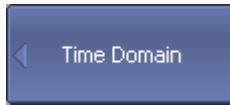
Window	Lowpass Impulse		Lowpass Step	
	Side Lobes Level	Pulse Width	Side Lobes Level	Edge Width
Minimum	– 13 dB	$\frac{0.6}{F_{max}-F_{min}}$	– 21 dB	$\frac{0.45}{F_{max}-F_{min}}$
Normal	– 44 dB	$\frac{0.98}{F_{max}-F_{min}}$	– 60 dB	$\frac{0.99}{F_{max}-F_{min}}$
Maximum	– 75 dB	$\frac{1.39}{F_{max}-F_{min}}$	– 70 dB	$\frac{1.48}{F_{max}-F_{min}}$

#### NOTE

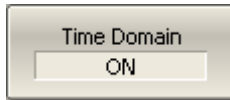
As the time domain transformation can be applied for separate traces of a channel, the stimulus axis label will be displayed depending on the active trace, in frequency or time units.

## Time Domain Transformation Activation

---



To enable/disable time domain transformation function, use the following softkeys:



**Analysis > Time Domain > Time Domain [ ON|OFF]**

---

[CALC:TRAN:TIME:STAT](#)

Turns the time domain transformation function ON/OFF.

---

---

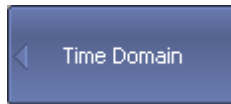
### NOTE

Time domain transformation function is accessible only in linear frequency sweep mode.

---

## Time Domain Transformation Span

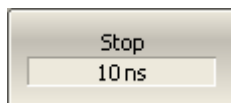
To define the span of time domain representation, its start and stop, or center and span values can be set.



To set the start and stop limits of the time domain range, use the following softkeys:



**Analysis > Time Domain > Start**



**Analysis > Time Domain > Stop**

---

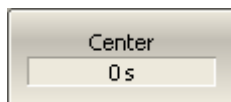
### [CALC:TRAN:TIME:STAR](#)

Sets or reads out the time domain start value when the time domain transformation function is turned ON.

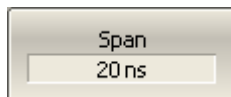
---

### [CALC:TRAN:TIME:STOP](#)

Sets or reads out the time domain stop value when the time domain transformation function is turned ON.



To set the center and span of the time domain, use the following softkeys:



**Analysis > Time Domain > Center**

**Analysis > Time Domain > Span**

---

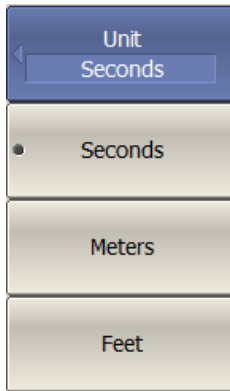
### [CALC:TRAN:TIME:CENT](#)

Sets or reads out the time domain center value when the time domain transformation function is turned ON.

---

### [CALC:TRAN:TIME:SPAN](#)

Sets or reads out the time domain span value when the time domain transformation function is turned ON.



To set the unit of the time domain, use the following softkeys:

**Analysis > Time Domain > Unit > [ Seconds | Meters | Feet ]**

---

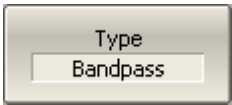
[CALC:TRAN:TIME:UNIT](#)

Selects the transformation unit for the time domain transformation function: seconds, meters, feet.

---

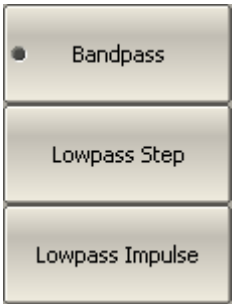
**Time Domain Transformation Type**

---



To set the time domain transformation type, use the following softkeys:

**Analysis > Time Domain > Type > [ Bandpass | Lowpass Impulse | Lowpass Step ]**



---

[CALC:TRAN:TIME](#)

Selects the transformation type for the time domain transformation function: band-pass or low-pass.

---

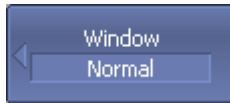
[CALC:TRAN:TIME:STIM](#)

Selects the stimulus type for the time domain transformation function: impulse or step.

---

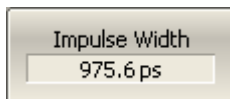
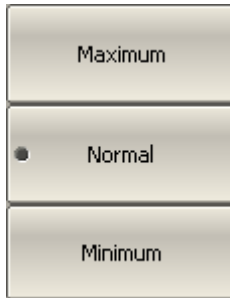
## Time Domain Transformation Window Shape Setting

---



To set the window shape, use the following softkeys:

**Analysis > Time Domain > [ Window > Minimum | Normal | Maximum ]**



To set the window shape for the specific impulse width or front edge width, use the following softkeys:

**Analysis > Time Domain > Window > Impulse Width**

---

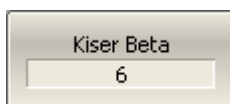
[CALC:TRAN:TIME:IMP:WIDT](#)

Sets or reads out the impulse width (time domain transformation resolution), coupled with the Kaiser–Bessel window shape  $\beta$  parameter.

---

[CALC:TRAN:TIME:STEP:RTIM](#)

Sets or reads out the rise time of the step signal (time domain transformation resolution), coupled with the Kaiser–Bessel window shape  $\beta$  parameter.



To set the window shape for the specific  $\beta$ -parameter of the Kaiser–Bessel filter, use the following softkeys:

**Analysis > Time Domain > Window > Kaiser Beta**

The available  $\beta$  values are from 0 to 13:

- "0" corresponds to minimum window.
- "6" corresponds to normal window.
- "13" corresponds to maximum window.

---

[CALC:TRAN:TIME:KBES](#)

Sets or reads out the  $\beta$  parameter, which controls the Kaiser–Bessel window shape when performing the time domain transformation.

---

---

**NOTE**

The impulse width and  $\beta$  of the Kaiser-Bessel filter are the dependent parameters. When setting one of the parameters the other one will be adjusted automatically.

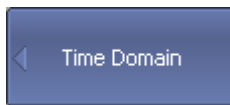
---

## Cable Correction Setting

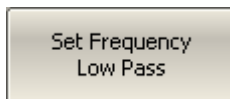
When the length units are selected the velocity factor setting of the Cable correction function affects the X-axis scale. See [Cable Correction Function](#).

## Frequency Harmonic Grid Setting

If lowpass mode is used, the frequency range must be set to a harmonic grid. The frequency values in measurement points are integer multiples of the start frequency  $F_{min}$ .



To create a harmonic grid for the current frequency range, use the following softkeys:



**Analysis > Time Domain > Set Frequency Low Pass**

---

[CALC:TRAN:TIME:LPFR](#)

Changes the frequency range to match with the low-pass type of the time domain transformation function.

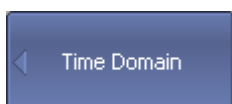
---



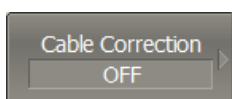
## Cable Correction Function

Cable correction function allows to take into account the influence of cable characteristics when taking measurements. The function is disabled by default.

### Cable Correction Activation



To enable/disable cable correction function of the time domain transformation function, use the following softkeys:



**Analysis > Time Domain > Cable Correction > Cable Correction**

[SENS:CORR:TRAN:TIME:STAT](#)

Turns the cable correction ON/OFF when the time domain transformation function is turned ON.

### Cable Table

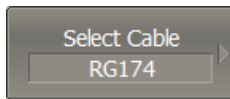
By default, the program contains a list of cables in the form of a cable table (See figure below). Each row of the table contains the cable name and the required parameters: velocity factor, cable loss and frequency.

All table fields can be edited. Changes are saved automatically.

If there is no cable description in the table, it is possible to add it. To do this, create a new row in the table using the **Add New Cable** button and enter its name and parameters.

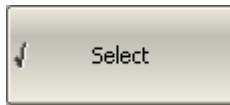
	Select	Type	Velocity Factor	Loss	Frequency
1	<input type="checkbox"/>	RG142	0.69	0.443 dB/m	1 GHz
2	<input type="checkbox"/>	RG17, 17A	0.659	0.18 dB/m	1 GHz
3	<input checked="" type="checkbox"/>	RG174	0.66	0.984 dB/m	1 GHz
4	<input type="checkbox"/>	RG178B	0.69	1.509 dB/m	1 GHz
5	<input type="checkbox"/>	RG178, 188	0.69	1.017 dB/m	1 GHz
6	<input type="checkbox"/>	RG213/U	0.66	0.292 dB/m	1 GHz
7	<input type="checkbox"/>	RG214	0.659	0.292 dB/m	1 GHz
8	<input type="checkbox"/>	RG223	0.659	0.165 dB/m	1 GHz
9	<input type="checkbox"/>	RG55, 55A, 55B	0.659	0.541 dB/m	1 GHz
10	<input type="checkbox"/>	RG58, 58B	0.659	1.574 dB/m	1 GHz
11	<input type="checkbox"/>	RG58A, 58C	0.659	0.787 dB/m	1 GHz
12	<input type="checkbox"/>	RG8, 8A, 10, 10A	0.659	0.262 dB/m	1 GHz
13	<input type="checkbox"/>	RG9, 9A	0.659	0.289 dB/m	1 GHz

Cable Table



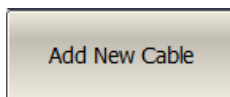
To open the cable table, use the following softkeys:

**Analysis > Time Domain > Cable Correction > Select Cable**

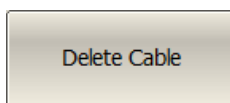


To select the cable in table, use the **Select** softkey.

**NOTE:** Make sure that the selected cable is check marked.

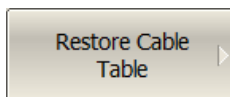


To add the new cable in the table, use the **Add New Cable** softkey:



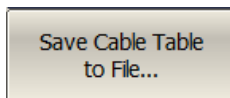
**NOTE:** A new cable can be added in the table by specifying its name and parameters in the empty field at the end of the table.

To delete cable table, use the **Delete Cable** softkey.

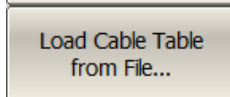


To restore cable table, use the following softkeys:

**Restore Cable > OK**



To save cable table in file, use the **Save Cable Table to File...** softkey.



To load cable table in file, use the **Load Cable Table from File...** softkey.

---

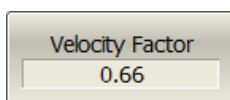
## Velocity Factor

Time and distance are related by velocity. To obtain the accurate mismatch location, it is important to set the right velocity factor of the transmission medium.

By default, the software assumes it to be equal to 1. But in practice, this can be different depending on the characteristics of the transmission line. If the velocity factor of a transmission line is not known, it can be calculated from the dielectric constant value:

$$v_f = \frac{1}{\sqrt{\epsilon_r}}$$

where  $\epsilon_r$  — the dielectric constant value.



To set the velocity cable, use the following softkeys:

**Analysis > Time Domain > Cable Correction > Velocity Factor**

---

[SENS:CORR:TRAN:TIME:RVEL](#)

Sets or reads out the cable relative wave speed velocity for the cable correction function, when the time domain transformation function is turned ON.

---

---

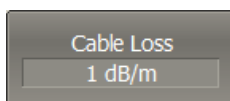
**NOTE**

The velocity cable value can also be set in the cable table.

---

## Cable Loss

The cable loss is the attenuation a wave experiences when traveling through the cable. The cable loss value is set per meter.



To set the cable loss, use the following softkeys:

**Analysis > Time Domain > Cable Correction > Cable Loss**

---

[SENS:CORR:TRAN:TIME:LOSS](#)

Sets or reads out the cable loss value for the cable correction function when the time domain transformation function is turned ON.

---

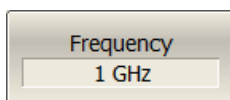
---

**NOTE**

The cable loss value can also be set in the cable table.

---

## Frequency



To set the frequency, on which the measurements are performed, use the following softkeys:

**Analysis > Time Domain > Cable Correction > Frequency**

---

[SENS:CORR:TRAN:TIME:FREQ](#)

Sets or reads out the frequency value at which the cable loss is specified for the cable

---

---

correction function when the time domain transformation function is turned ON.

---

## Time Domain Gating

---

**NOTE**

The availability of this feature depends on the Analyzer model (See corresponding [datasheet](#)).

---

Time domain gating is a function that mathematically removes unwanted responses in the time domain. The function performs a time domain transformation, selects the region in the time domain, deletes the response inside (or outside) the region and transforms back to the frequency domain. The function allows the user to remove spurious effects of the fixture devices in the frequency response, if the useful signal and spurious signal are separable in the time domain.

The recommended procedure is as follows:

- Use the time domain function for viewing the layout of useful and spurious responses.
- Enable the time domain gating and set the gate position to remove as much of spurious response as possible.
- Disable the time domain function and view the response without spurious effects in frequency domain.

The function involves two types of time domain gating:

- Bandpass — removes the response outside the gate span.
- Notch — removes the response inside the gate span.

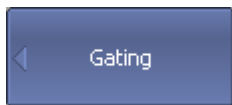
The sharp gate shape leads to ringing effect in the frequency domain. To reduce the ringing the gate shape can be smoothed. The following gate shapes are offered:

- Maximum
- Wide
- Normal
- Minimum

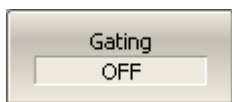
The minimum window has a sharp shape. The maximum window has a more smoothed shape. From minimum to maximum window shape, the sidelobe level increases and the gate resolution decreases. The choice of the window shape is always a trade-off between the gate resolution and the level of spurious sidelobes. The parameters of different window shapes are represented in the table below.

Window Shape	Bandpass Sidelobe Level	Gate Resolution (Minimum Gate Span)
Minimum	– 48 dB	$\frac{2.8}{F_{max}-F_{min}}$
Normal	– 68 dB	$\frac{5.6}{F_{max}-F_{min}}$
Wide	– 57 dB	$\frac{8.8}{F_{max}-F_{min}}$
Maximum	– 70 dB	$\frac{25.4}{F_{max}-F_{min}}$

## Time Domain Gate Activation



To enable/disable the time domain gating function, toggle the following softkey:



**Analysis > Gating > Gating [ ON|OFF ]**

[CALC:FILT:TIME:STAT](#)

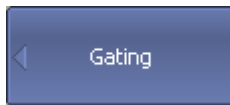
Turns the gating function ON/OFF.

## NOTE

Time domain gating function is accessible only in linear frequency sweep mode.

## Time Domain Gate Span

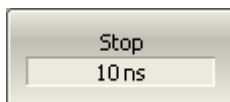
---



To the start and stop of the time domain gate, use the following softkeys:



**Analysis > Gating > Start**



**Analysis > Gating > Stop**

---

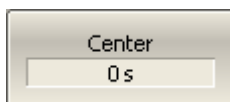
[CALC:FILT:TIME:STAR](#)

Sets or reads out the gate start value of the gating function.

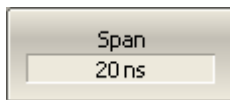
---

[CALC:FILT:TIME:STOP](#)

Sets or reads out the gate stop value of the gating function.



To set the center and span of the time domain gate, use the following softkeys:



**Analysis > Gating > Center**

**Analysis > Gating > Span**

---

[CALC:FILT:TIME:CENT](#)

Sets or reads out the gate center value of the gating function.

---

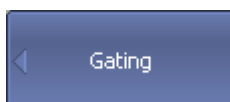
[CALC:FILT:TIME:SPAN](#)

Sets or reads out the gate span value of the gating function.

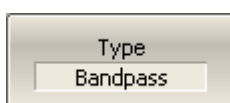
---

## Time Domain Gate Type

---



To select the type of the time domain window, use the following softkeys:



**Analysis > Gating > Type**

Toggle the type between **Bandpass** and **Notch**.

---

[CALC:FILT:TIME](#)

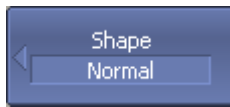
---

Sets or reads out the gate type of the gating function.



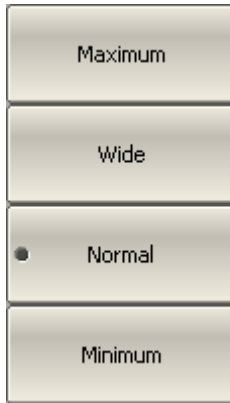
## Time Domain Gate Shape Setting

---



To set the time domain gate shape, use the following softkeys:

**Analysis > Gating > Shape > [ Minimum | Normal | Wide | Maximum ]**



---

[CALC:FILT:TIME:SHAP](#)

Sets or reads out the gate shape of the gating function.

---

## S-Parameter Conversion

The S-parameter conversion function allows for the conversion of measurement results ( $S_{ab}$ ) to the following parameters:

Parameter	Equation
Impedance in reflection measurement ( $Z_r$ )	$Z_r = Z_{0a} \cdot \frac{1+S_{aa}}{1-S_{aa}}$
Admittance in reflection measurement ( $Y_r$ )	$Y_r = \frac{1}{Z_r}$
Impedance in transmission measurement ( $Z_t$ )	$Z_t = \frac{2 \cdot \sqrt{Z_{0a} \cdot Z_{0b}}}{S_{ab}} - (Z_{0a} + Z_{0b})$
Admittance in transmission measurement ( $Y_t$ )	$Y_t = \frac{1}{Z_t}$
Inverse S-parameter	$\frac{1}{S_{ab}}$
Equivalent admittance in transmission shunt measurements ( $Y_{tsh}$ )	$Y_{tsh} = \frac{2 \cdot \sqrt{Y_{0a} \cdot Y_{0b}}}{S_{ab}} - (Y_{0a} + Y_{0b})$
Equivalent impedance in transmission shunt measurements ( $Z_{tsh}$ )	$Z_{tsh} = \frac{1}{Y_{tsh}}$
S-parameter complex conjugate	$S_{ab}^*$
<p><math>Z_{0a}</math> is characteristic impedance of Port a.</p> <p><math>Z_{0b}</math> is characteristic impedance of Port b.</p> <p><math>S_{ab}</math> is measured S-parameter (a and b are the port identifiers).</p> <p><math>Y_{0a} = \frac{1}{Z_{0a}}, Y_{0b} = \frac{1}{Z_{0b}}</math></p>	

---

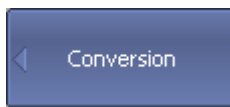
**NOTE**

Equations for  $Z_r$ ,  $Z_t$ ,  $Y_r$ ,  $Y_t$  are approximate. The general method of converting S-parameters to Z, Y, H, T, ABCD-parameters is presented in the next section. The reason for using the approximate method is the measurement speed, as only one S-parameter is used in the calculations, whereas the general method requires measurement of the full matrix of S-parameters.

---

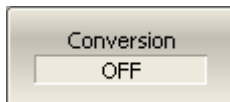
The S-parameter conversion function can be applied to an individual trace of a channel. The trace to which the function is applied must be preselected as active (See [Selection of Active Trace/Channel](#) ).

---



To enable/disable the conversion, use the following softkeys:

**Analysis > Conversion > Conversion**



---

**CALC:CONV**

Turns the S-parameter conversion function ON/OFF.

---



To select the conversion type, use the following softkeys:

**Analysis > Conversion > Function**

Then select the required function:

- **Zr: Reflection**
- **Zt: Transmission**
- **Yr: Reflection**
- **Yt: Transmission**
- **1/S: Inverse**
- **Ztsh: Trans-Shunt**
- **Ytsh: Trans-Shunt**
- **Conjugation**

---

<input checked="" type="radio"/> Z: Reflection
Z: Transmission
Y: Reflection
Y: Transmission
1/S: Inverse
Z: Trans-Shunt
Y: Trans-Shunt
Conjugation

---

[CALC:CONV:FUNC](#)

Sets or reads out the S-parameter conversion function type.

---

---

**NOTE**

All conversion types are indicated in the trace status field, when enabled.

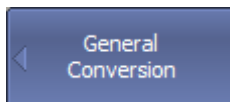
---

## General S-Parameter Conversion

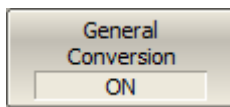
This section describes the most common method of transformation of the S-parameters to Z, Y, T, H, ABCD - parameters. The method is described in: Dean A. Frickey's "Conversions Between S, Z, Y, h, ABCD, and T Parameters which are Valid for Complex Source and Load Impedances".

The port impedances Z01 and Z02 are real numbers determined by the System Z0 menu (**Calibration > System Z0**) when the [Port Z Conversion function](#) is not active. If the Port Z Conversion function is active, then the port impedances Z01 and Z02 are complex numbers determined by the Port Z Conversion menu (Analysis > Fixture Simulator > Port Z Conversion).

The function is applied to all S-parameters of a channel. The channel to which the function is applied must be preselected as active (See [Selection of Active Trace/Channel](#) ).



To enable/disable the general conversion, use the following softkeys:

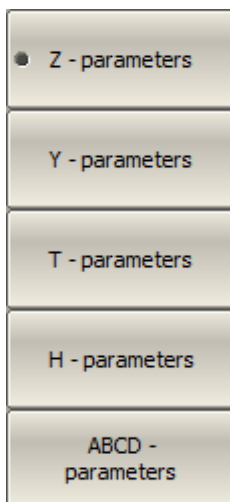


**Analysis > General Conversion > General Conversion [ON|OFF]**



To select the conversion type, use the following softkeys:

**Analysis > General Conversion > Conversion Type**



Then select the required function:

- **Z - parameters**
  - **Y - parameters**
  - **T - parameters**
  - **H - parameters**
  - **ABCD - parameters**
-

---

**NOTE**

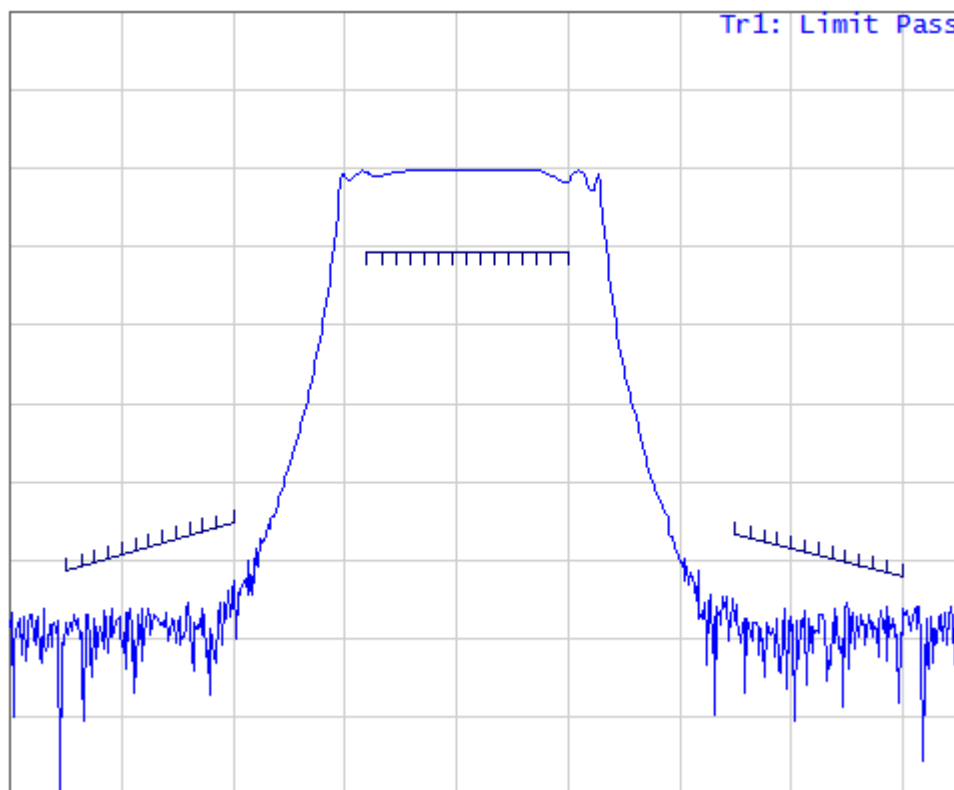
The **[Conv]** symbols are indicated in the trace status field, when general conversion is enabled.

---

## Limit Test

The limit test is a function of automatic pass/fail judgment for the trace of the measurement result. The judgment is based on the comparison of the trace to the limit line set by the user.

The limit line can consist of one or several segments (See figure below). Each segment checks the measured value for failure, whether it is an upper or lower limit. The limit line segment is defined by specifying the coordinates of the beginning (X0, Y0) and the end (X1, Y1) of the segment, and the type of the limit. The MAX or MIN limit types check if the trace falls outside of the upper or lower limit respectively.



Limit line

The limit line is set by the user in the limit table. Each row in the table describes one segment of the line. Limit table editing is described below. The table can be saved into a \*.lim file.

The display of the limit lines on the screen can be turned on/off independently of the status of the limit test function.

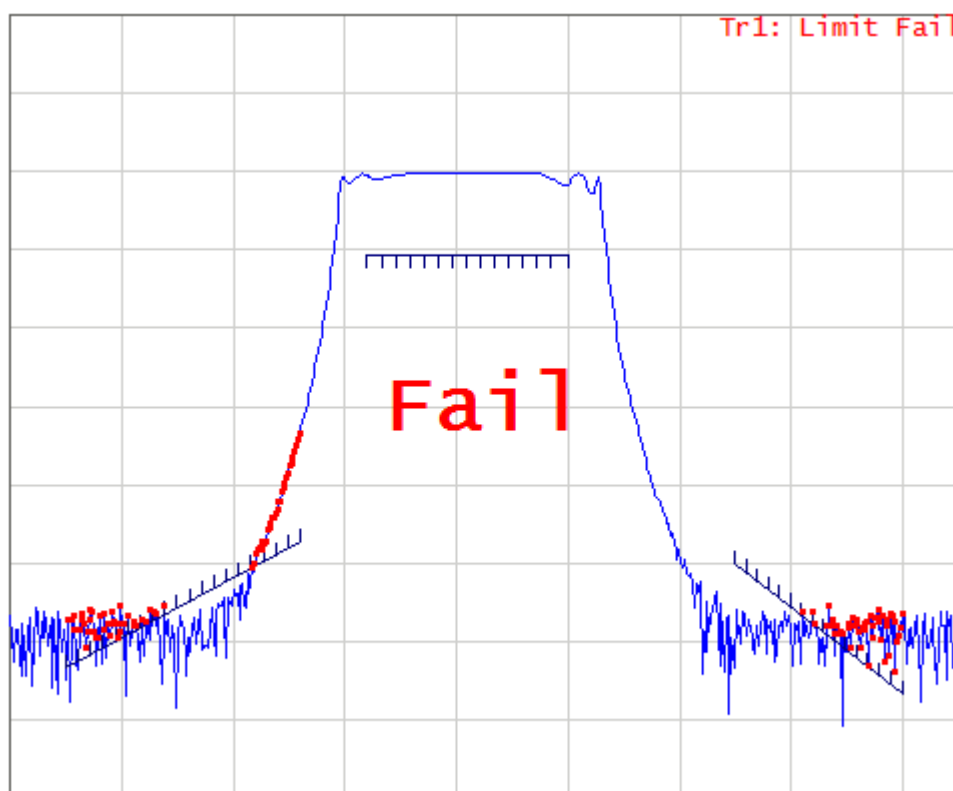
The result of the limit test is indicated in the upper right corner of the diagram:

- If the measurement result passed the limit test, the trace number and the result will be seen: **Tr1: Limit Pass** (See figure above).

- If the measurement result failed, the result will be indicated in the following ways (See figure below):

1. **Tr1:Limit Fail** will be displayed in upper right corner of the diagram.
2. Fail sign will be displayed in red in the center of the window.
3. The points of the trace, which failed the test will be highlighted in red.
4. A beep will be heard.

The fail sign and the beep can be disabled using the **Fail Sign** softkey. For beep deactivation see [Beeper Settings](#).

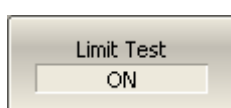


Test fail indication

### Limit Test Enabling/Disabling



To enable/disable limit test function, use the following softkeys:



**Analysis > Limit Test > Limit Test [ ON|OFF ]**



## Limit Line Editing

In the editing mode the limit table will appear in the lower part of the screen (See figure below). The limit table will be hidden when quitting the submenu.

	Begin Stimulus	End Stimulus	Begin Response	End Response
1	500 MHz	2 GHz	-90 dB	-80 dB
2	3.2 GHz	5 GHz	-10 dB	-10 dB
3	6.5 GHz	8 GHz	-80 dB	-100 dB

Limit line table

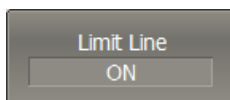
Navigating within the table to enter the values of the following parameters of a limit test segment:

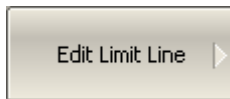
<b>Type</b>	Select the segment type among the following: <ul style="list-style-type: none"><li>• <b>MAX</b> — upper limit.</li><li>• <b>MIN</b> — lower limit.</li><li>• <b>OFF</b> — segment not used for the limit test.</li></ul>
<b>Begin Stimulus</b>	Stimulus value in the beginning point of the segment.
<b>End Stimulus</b>	Stimulus value in the ending point of the segment.
<b>Begin Response</b>	Response value in the beginning point of the segment.
<b>End Response</b>	Response value in the ending point of the segment.



To enable/disable limit line, use the following softkeys:

**Analysis > Limit Test > Limit Line [ ON|OFF ]**

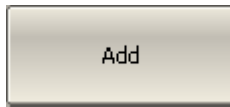




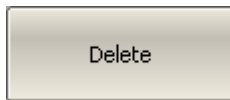
To access the limit line editing mode, use the following softkeys:

**Analysis > Limit Test > Edit Limit Line**

---



To add a new row in the table, click **Add**. The new row will appear below the highlighted one.



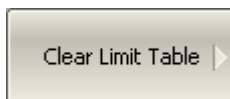
To delete a row from the table, click **Delete**. The highlighted row will be deleted.

---

[CALC:LIM:DATA](#)

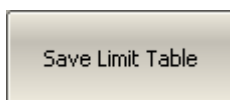
Sets the data array, which is the limit line in the limit test function.

---



To clear the entire table, use the **Clear Limit Table** softkey.

---



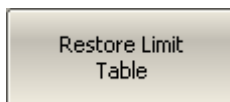
To save the table into \*.lim file, use the **Save Limit Table** softkey.

---

[MMEM:STOR:LIM](#)

Saves the ripple limit table into a file.

---



To open the table from a \*.lim file, use the **Restore Limit Table** softkey.

---

[MMEM:LOAD:LIM](#)

Recalls the limit table file. The file must be saved using the [MMEM:STOR:LIM](#) command.

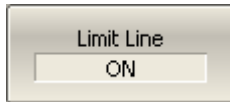
---

## Limit test display management

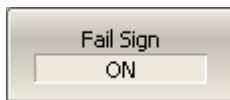
---



To enable/disable display of a limit line, use the following softkeys:



**Analysis > Limit Test > Limit Line [ ON|OFF ]**



To enable/disable display of fail sign in the center of the diagram, use **Fail Sign** softkey.

---

[CALC:LIM:DISP](#)

Turns the limit line display of the limit test function ON/OFF.

---

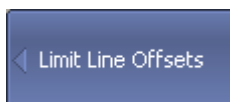
[DISP:FSIG](#)

Turns the "fail" sign display ON/OFF when performing limit test or ripple limit test.

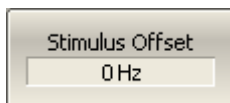
---

## Limit Line Offset

The limit line offset function allows the user to shift the segments of the limit line by the specified value along X and Y axes simultaneously.



To define the limit line offset along X-axis, use the following softkeys:

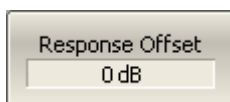


**Analysis > Limit Test > Limit Line Offsets > Stimulus Offset**

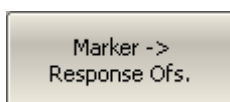
---

[CALC:LIM:OFFS:STIM](#)

Sets and reads out the value of the limit line offset along the X-axis.



To define the limit line offset along Y-axis, use the following softkeys:



**Analysis > Limit Test > Limit Line Offsets > Response Offset**

Response offset can be set to the active marker position, using the following softkeys:

---

**Analysis > Limit Test > Limit Line Offsets > Marker – >  
Response Ofs**

---

[CALC:LIM:OFFS:AMPL](#)

Sets and reads out the value of the limit line offset along the Y-axis.

---

[CALC:LIM:OFFS:MARK](#)

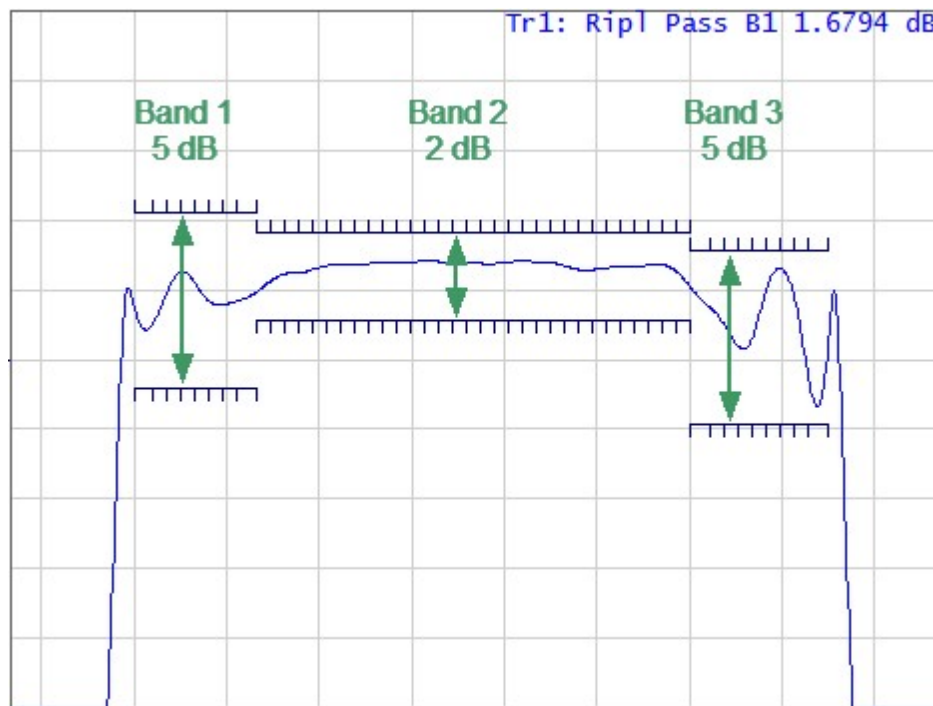
Sets the value of the limit line offset along the Y-axis to the active marker value.

---

## Ripple Limit Test

The ripple limit test is an automatic pass/fail check of the measured trace data. The trace is checked against the maximum ripple value (ripple limit) defined by the user. The ripple value is the difference between the maximum and minimum response of the trace in the trace frequency band.

The ripple limit can include one or more segments (See figure below). Each segment provides the ripple limit for the specific frequency band. A segment is set by the frequency band and the ripple limit value.



Ripple limits

The ripple limit settings are set in the ripple limit table. Each row of the table describes the frequency band of the ripple limit value. The process of ripple limit table editing is described below. The table can be saved into a \*.lim file.

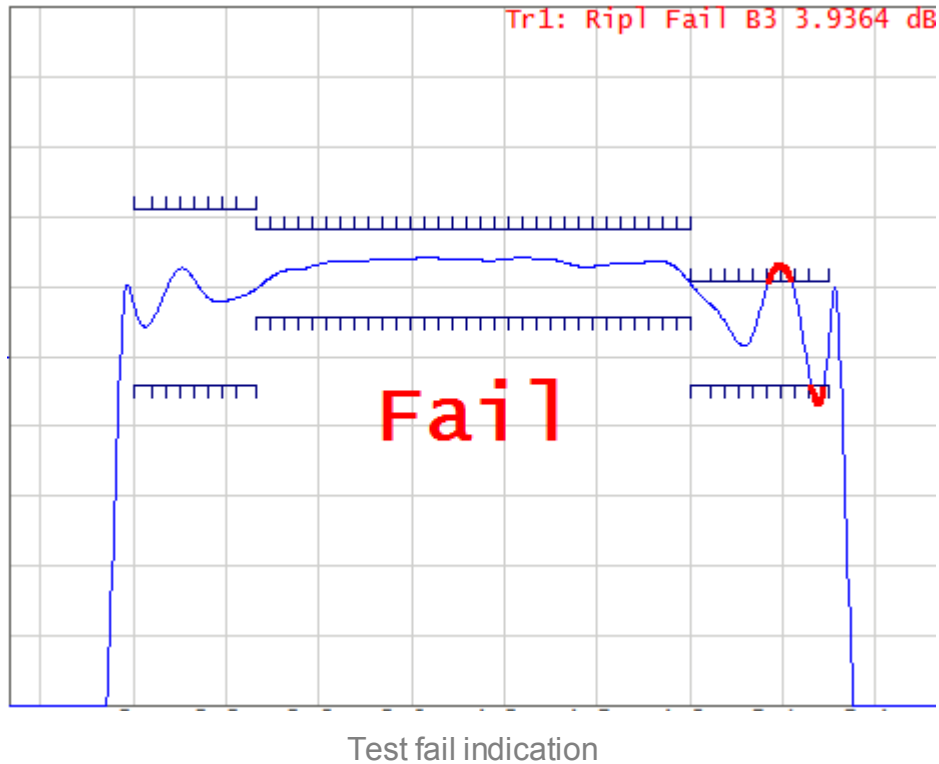
The display of the limit lines on the screen can be turned on/off independently of the status of the ripple limit test function.

The result of the ripple limit test is indicated in the upper right corner of the diagram:

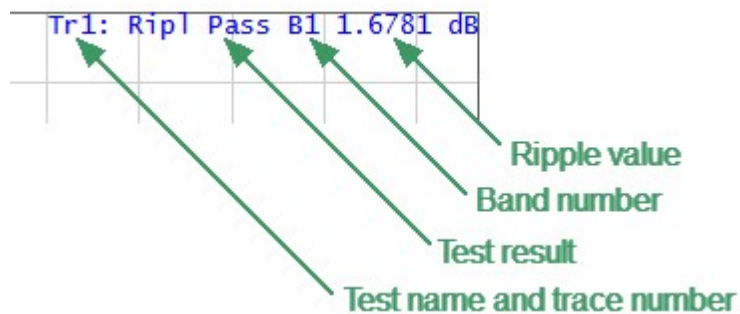
- If the measurement result passed the limit test, the trace number and the result will be seen: **Tr1: Ripl Pass**.
- If the measurement result failed, the result will be indicated in the following ways (See figure below):
  1. **Tr1: Ripl1: Fail** will be displayed in upper right corner of the diagram.

2. Fail sign will be displayed in red in the center of the window.
3. A beep will be heard.

The fail sign and the beep can be disabled using the **Fail Sign** softkey. For beep deactivation see [Beeper Settings](#).

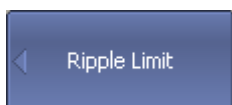


The display of the ripple value can be enabled/disabled in the ripple limit test status line in the upper right corner of the diagram (See figure below). The ripple value is displayed for the band selected by the user. The ripple value can be represented as an absolute value or as a margin to the limit.

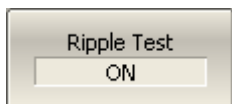


Ripple limit test status line

## Ripple Limit Enabling/Disabling



To enable/disable ripple limit test function, use the following softkeys:



**Analysis > Ripple Limit > Ripple Test [ ON|OFF ]**

[CALC:RLIM](#)

Turns the ripple limit test ON/OFF.

## Ripple Limit Editing

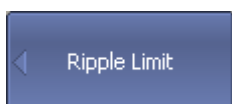
In the editing mode, the limit table will appear in the lower part of the screen (See figure below). The limit table will be hidden when exiting the submenu.

	Type	Begin Stimulus	End Stimulus	Ripple Limit
1	ON	3 GHz	3.4 GHz	4 dB
2	ON	3.4 GHz	4.8 GHz	1 dB
3	ON	4.8 GHz	5.25 GHz	4 dB

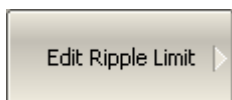
Ripple limit table

Navigating within the table to enter the values of the following parameters of a ripple limit test segment:

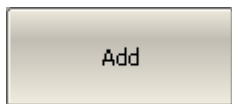
<b>Type</b>	Select the segment type among the following: <ul style="list-style-type: none"><li>• <b>ON</b> — band used for the ripple limit test.</li><li>• <b>OFF</b> — band not used for the limit test.</li></ul>
<b>Begin Stimulus</b>	Stimulus value in the beginning point of the segment.
<b>End Stimulus</b>	Stimulus value in the ending point of the segment.
<b>Ripple Limit</b>	Ripple limit value.



To access the ripple limit editing mode, use the following softkeys:



**Analysis > Ripple Limit > Edit Ripple Limit**



To add a new row in the table, click **Add**. The new row will appear below the highlighted one.



To delete a row from the table, click **Delete**. The highlighted row will be deleted.

---

[CALC:RLIM:DATA](#)

Sets the data array, which is the limit line for the ripple limit function.



To clear the entire table, use the **Clear Ripple Limit Table** softkey.

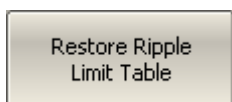


To save the table into \*.rlm file, use the **Save Ripple Limit Table** softkey.

---

[MMEM:STOR:RLIM](#)

Saves the limit table into a file.



To open the table from a \*.rlm file, use the **Recall Ripple Limit Table** softkey.

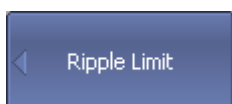
---

[MMEM:LOAD:RLIM](#)

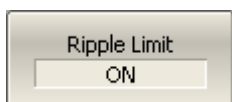
Recalls the limit table file.

---

## Ripple Limit Test Display Management



To enable/disable display of the ripple limit line, use the following softkeys:



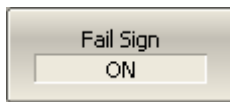
**Analysis > Ripple Limit > Ripple Limit [ ON|OFF ]**

---

[CALC:RLIM:DISP:LINE](#)

Turns the ripple limit line display ON/OFF.



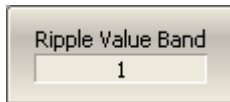


To enable/disable display of fail sign in the center of the diagram, use **Fail Sign** softkey.

---

[DISP:FSIG](#)

Turns the "fail" sign display ON/OFF when performing limit test or ripple limit test.



To enter the number of the band, whose ripple value should be displayed, use the following softkeys:

**Analysis > Ripple Limit > Ripple Value Band**

---

[CALC:RLIM:DISP:SEL](#)

Sets or reads out the number of the ripple limit test band selected for the ripple value display.



To enable/disable display of the ripple value, use the following softkeys:

**Analysis > Ripple Limit > Ripple Value > [ OFF | Absolute | Margin ]**



---

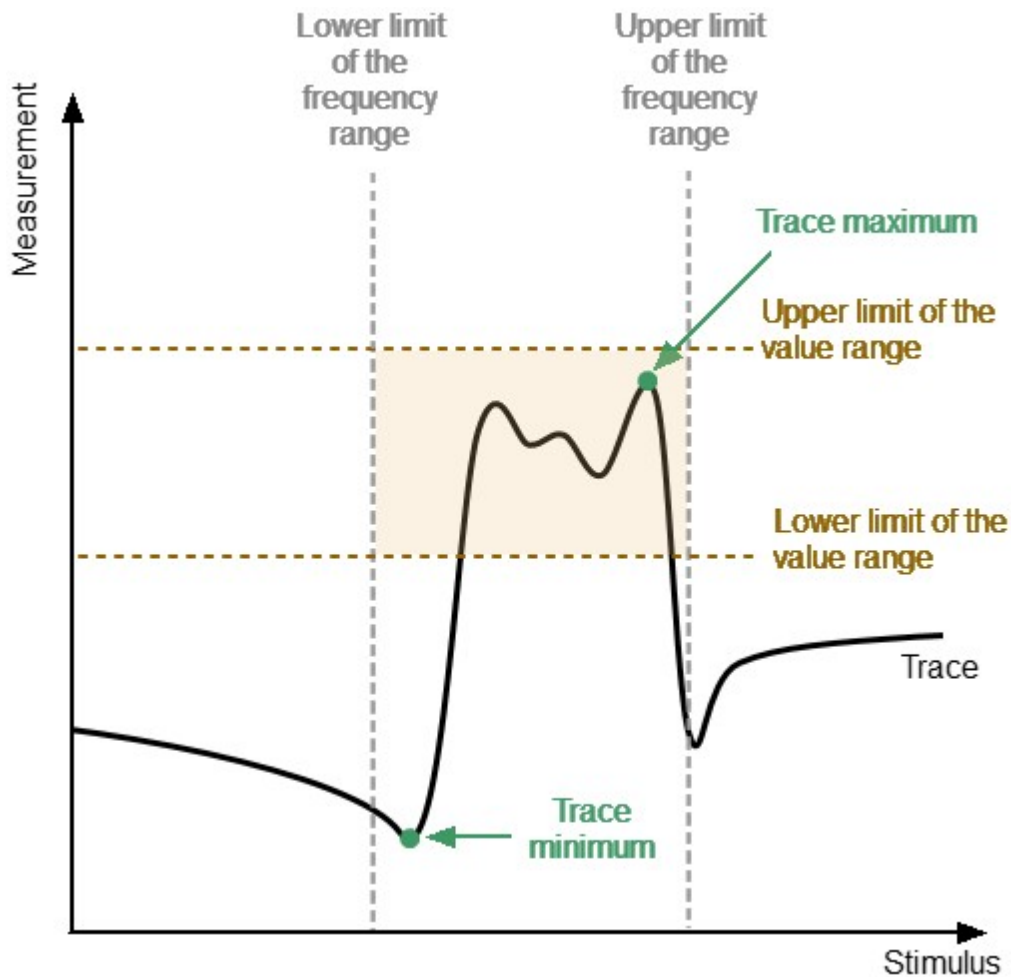
[CALC:RLIM:DISP:VAL](#)

Sets or reads out the number of the ripple limit test band selected for the ripple value display.

---

## Peak Limits Test

The peak limits test function checks whether or not the trace point with the minimum (or maximum) value of the measured value falls within the specified limits of the frequency range and/or value range (see figure below). If the trace point minimum (or maximum) falls within the specified limits, the test is passed (test result "pass"). Otherwise, the test is failed (test result "fail").



Peak limits test

## Peak Limits Test parameter

A description of the parameters of the peak limits test function in the program is shown in the table below.

Parameter	Definition
<b>Limit Type</b>	Selects the type of limit (one of the limits or both at the same time). Possible values: <ul style="list-style-type: none"><li>• <b>Stimulus</b> — checks whether or not the maximum (or minimum) point of the trace is within the specified frequency bandwidth.</li><li>• <b>Response</b> — checks whether or not the maximum (or minimum) point of the trace falls within the value range.</li><li>• <b>All</b> — checks the maximum (or minimum) trace point simultaneously within the frequency band and within the value range of the measured value.</li></ul>
<b>Begin Stimulus</b>	Lower bandwidth limit.
<b>End Stimulus</b>	Upper bandwidth limit.
<b>Begin Response</b>	Lower limit of the value range.
<b>End Response</b>	Upper limit of the value range.
<b>Peak Polarity</b>	Selects a trace point for inspection: <ul style="list-style-type: none"><li>• <b>Positive</b> — trace maximum.</li><li>• <b>Negative</b> — trace minimum.</li></ul>

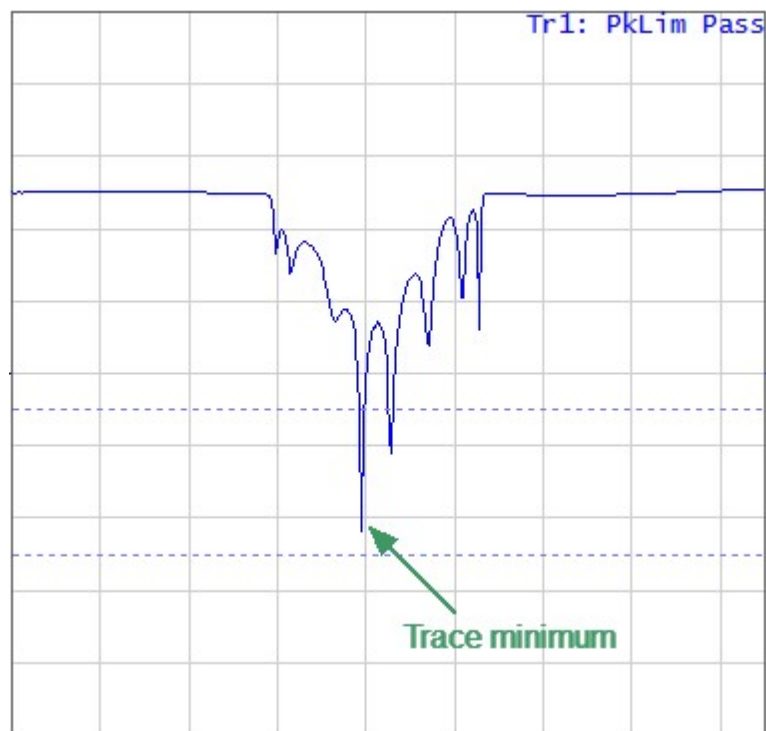
---

### NOTE

The peak limit test for the **Stimulus** limit type can only be performed in the frequency domain.

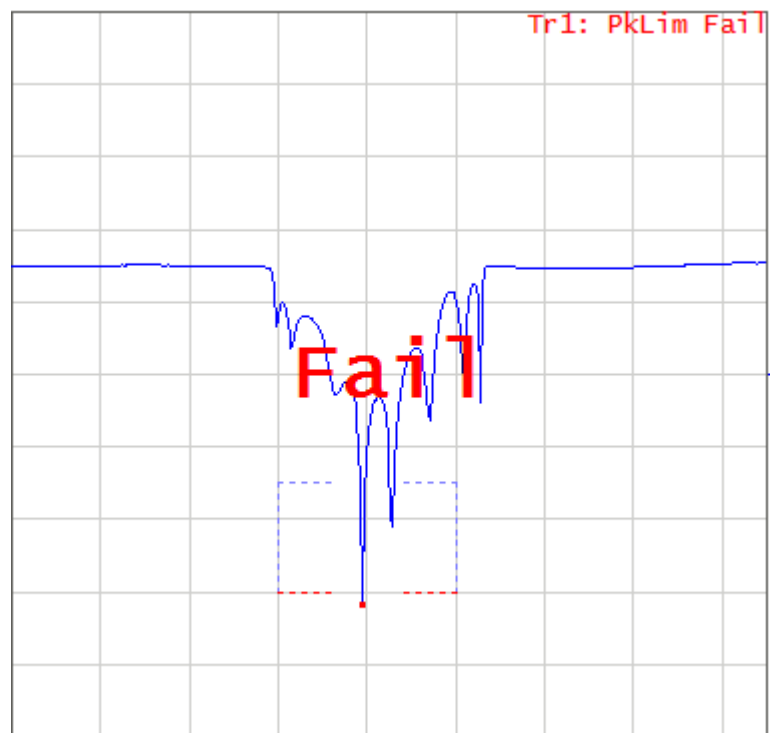
---

The figure below shows an example of the trace minimum point falling within the value range (test result "pass").



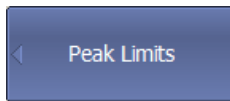
Example of the trace minimum value falling within the range of values (Limit type: Response)

The figure below shows an example of the trace minimum point being out of the value range (test result "fail").

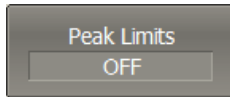


Example of the minimum trace point value being outside the value range (Limit type: All)

## Peak Limits Enabling/Disabling

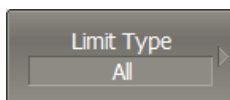


To enable/disable peak limits test function, use the following softkeys:



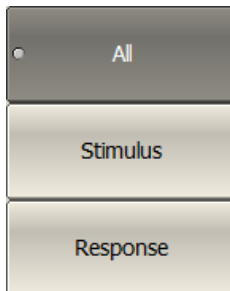
**Analysis > Peak Limits > Peak Limits [ ON|OFF ]**

## Editing Search Parameters for Peak Limits



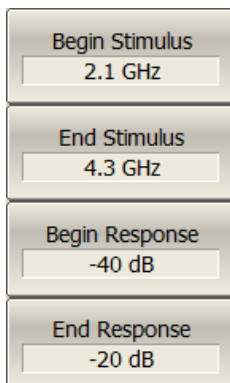
To select the limit type, use the following softkeys:

**Analysis > Peak Limits > Limit Type**



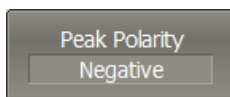
Then select the required type:

- **All**
- **Stimulus**
- **Response**



To enter the lower bandwidth limit, use the **Begin Stimulus** softkey. To enter the upper bandwidth limit, use the **End Stimulus** softkey.

To enter the lower limit of the value range, use the **Begin Response** softkey. To enter the upper limit of the value range, use the **End Response** softkey.



To select a trace point to check, use the **Peak Polarity** softkey.

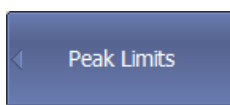
## Peak Limits Test Display Management

The display of the peak limits on the screen can be turned on/off.

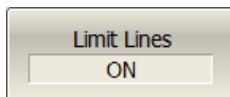
The result of the peak limits test is indicated in the upper right corner of the diagram:

- If the measurement result passed the peak limit test, the trace number and the result will be seen: **Tr1: PkLim Pass**.
- If the measurement result failed, the result will be indicated in the following ways (See figure [above](#)):
  1. **Tr1: PkLim Fail** will be displayed in upper right corner of the diagram.
  2. Fail sign will be displayed in red in the center of the window.
  3. A beep will be heard.

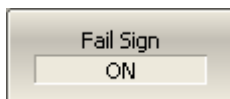
The fail sign and the beep can be disabled using the **Fail Sign** softkey. For beep deactivation see [Beeper Settings](#).



To enable/disable display of the ripple limit line, use the following softkeys:



**Analysis > Peak Limits > Limit Lines [ ON|OFF ]**



To enable/disable display of the fail sign in the center of the diagram, use the following softkeys:

**Analysis > Ripple Limit > Fail Sign [ ON|OFF ]**

---

## Special Measurement Modes

This section describes the measurements that use optional equipment, as well as special function available in select Analyzer models:

- Measurement of frequency conversion devices (See [Mixer Measurements](#)).
- Frequency range extension using special extension modules for the Cobalt Series Analyzers (See [Frequency Extension System](#)).
- Built-in dual-channel voltmeter, designed to measure DC voltage synchronously with stimulus frequency adjustment. The function is available for the Cobalt Series Analyzers with the special HW-C-AUX option (See [DC Measurement](#)).
- Measurement with the introduction of additional external components (amplifiers, attenuators, various filtering or matching circuits) into the test signal generator and receiver path. The option is available for analyzers with additional direct access to port receivers C2209 and C2220 (See [Direct Receiver Access](#)).



## Mixer Measurements

---

### NOTE

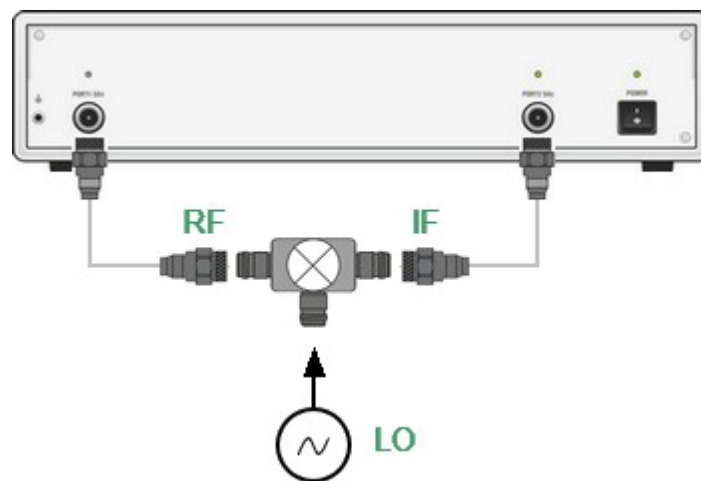
The availability of mixer measurements and their settings depends on the Analyzer model (See corresponding [datasheet](#)).

---

### Mixer Measurement Methods

The Analyzer allows to perform measurements of mixers and other frequency converting devices using scalar and vector methods.

The **scalar method** allows measurement of the scalar transmission S-parameters of frequency converting devices. Phase and group delay measurements are not accessible in this mode. The advantage of this method is the simplicity of measurement setup (no additional equipment necessary). See figure below.



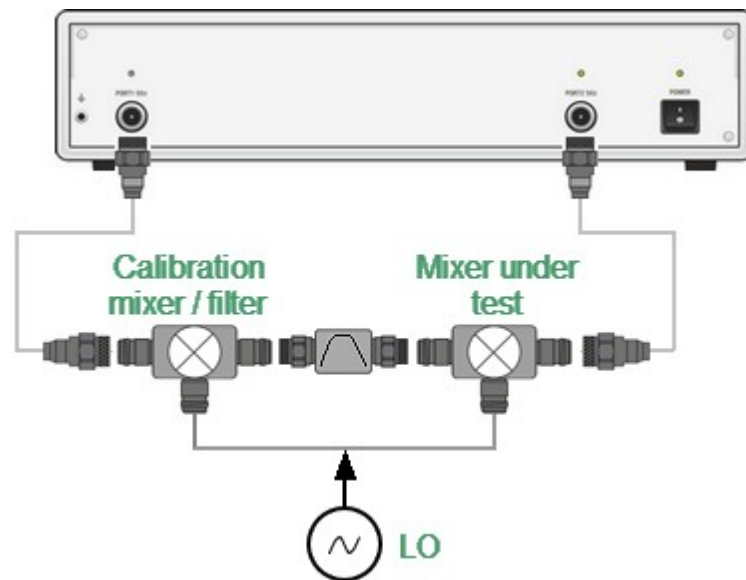
Scalar mixer measurement setup

The scalar measurement method is based on frequency offset mode. Frequency offset mode enables a frequency offset between the Analyzer test ports as described in detail in [Frequency Offset Mode](#). Frequency offset mode can be combined with various calibration methods.

When performing scalar measurements of a mixer, the most accurate method of calibration is scalar mixer calibration (See [Scalar Mixer Calibration](#)).

An easier but less accurate method is using absolute measurements in combination with receiver calibration and power calibration (See [Absolute Measurements](#), [Power Calibration](#), [Receiver Calibration](#)). This method often results in transmission S-parameter ripples due to mixer input and output mismatch. This can be partially compensated by using matching attenuators of 3-10 dB at the mixer input and output.

The **vector mixer calibration method** allows measurement of mixer transmission complex S-parameters including phase and group delay. The method requires additional equipment (See figure below): an external mixer with filter, which is called a calibration mixer, and an LO common for both the calibration mixer and the mixer under test.



Vector mixer measurement setup

The vector mixer calibration method doesn't use frequency offset. This method ensures the same frequency at both test ports of the Analyzer in normal operation mode. The vector mixer calibration procedure is described in [Vector Mixer Calibration](#).

## Frequency Offset Mode

The frequency offset mode allows for S-parameter measurement of frequency converting devices, including vector reflection measurements and scalar transmission measurements. In this context, frequency converting devices include both frequency shifting devices such as mixers and converters, as well as devices dividing or multiplying frequency.

This measurement mode is based on a frequency offset between the ports. The frequency offset is defined for each port using three coefficients: multiplier, divider, and offset. These coefficients allow for calculation of a port frequency relative to the basic frequency range:

$$F_{port} = \frac{M}{D} F_{base} + F_{ofs}$$

where:

$M$  — multiplier,

$D$  — divider,

$F_{ofs}$  — offset,

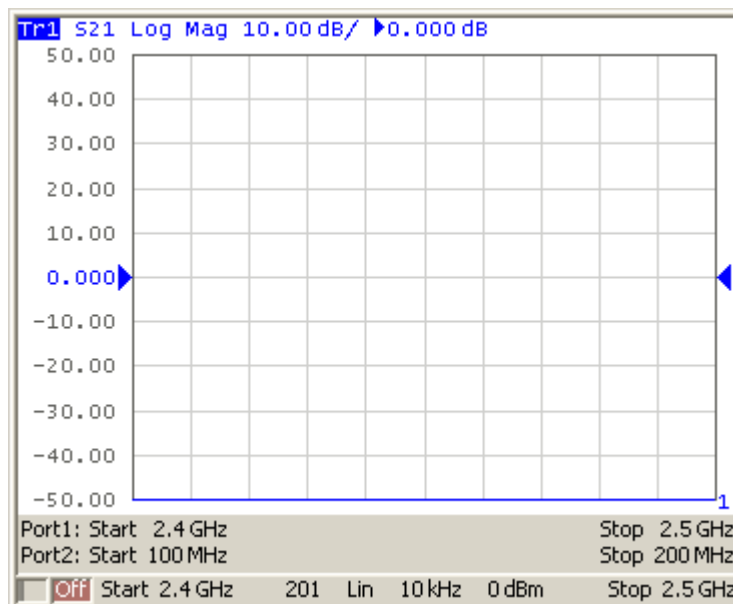
$F_{base}$  — basic frequency.

In most cases, it is sufficient to apply an offset to only one of the ports, leaving the other one at the basic frequency ( $M=1$ ,  $D=1$ ,  $F_{ofs}=0$ ).

Below are some examples of offset coefficient calculation for different types of frequency conversion. Here the mixer RF input is connected to Port 1, and the mixer IF output is connected to Port 2. The basic frequency range is set to the mixer RF frequency range, and the first port of the Analyzer does not use frequency offset. The second port of the Analyzer is set to the IF frequency range and uses frequency offset mode as follows:

<b>1. IF = RF – LO</b>	Port 2: $M = 1, D = 1, F_{ofs} = -LO$ .
<b>2. IF = LO – RF</b>	Port 2: $M = -1, D = 1, F_{ofs} = LO$ .
<b>3. IF = RF + LO</b>	Port 2: $M = 1, D = 1, F_{ofs} = LO$ .

In frequency offset mode, the bottom part of the channel window will indicate each port's frequency span (See figure below).



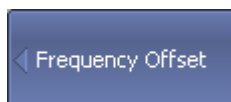
Channel window in frequency offset mode

**Start** and **Stop** frequency can be set for each port directly instead of using **Multiplier**, **Divider** and **Offset** values. Using **Start/Stop** values will set **Multiplier** and **Offset**, which can be determined from the specified frequency and the base frequency while maintaining the preset **Divider**.

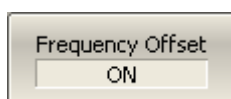
### Source/Receivers Frequency Offset Feature

Conventional frequency offset mode uses frequency offset between the ports, while the source and receivers of each port operate at a common frequency. Frequency offset between the ports allows for S-parameter measurement of frequency converting devices, including vector reflection measurements and scalar transmission measurements.

The source/receivers frequency offset feature introduces a frequency offset between the source and receivers within a single port. Frequency offset between the source and receivers allows for absolute measurements only.



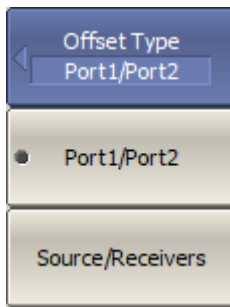
To enable/disable frequency offset mode, use the following softkeys:



**Stimulus > Frequency Offset > Frequency Offset [ ON| OFF ]**

[SENS:OFFS](#)

Turns the frequency offset feature ON/OFF.

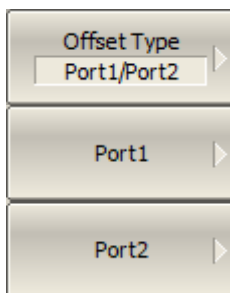


The **Offset Type** sets to **Port1/Port2** or **Source/Receivers**.

---

[SENS:OFFS:TYPE](#)

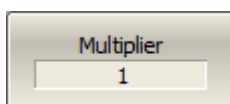
Sets or reads out the frequency offset type when the frequency offset feature is ON.



If conventional frequency offset mode uses and **Offset Type** is set to **Port1/Port2**, enter offset coefficients for Ports.



If a frequency offset is introduced between the source and the receivers within a single port and **Offset Type** is set to **Source/Receivers**, enter offset coefficients for Source and Receivers.



To enter offset coefficients of multiplier, use the **Multiplier** softkey.

---

[SENS:OFFS:PORT:MULT](#)

Sets or reads out the basic frequency range Multiplier of Port n when offset type is **Port1/Port2**.

---

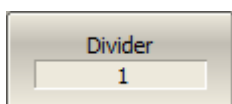
[SENS:OFFS:SOUR:MULT](#)

Sets or reads out the basic frequency range Multiplier to get the Source frequency when offset type is **Source/Receivers**.

---

[SENS:OFFS:REC:MULT](#)

Sets or reads out the basic frequency range Multiplier to get the Receiver frequency when offset type is **Source/Receivers**.



To enter offset coefficients of divider, use the **Divider** softkey.

---

[SENS:OFFS:PORT:DIV](#)

Sets or reads out the basic frequency range Divider of Port n when offset type is **Port1/Port2**.

---

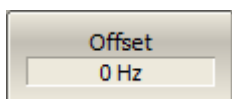
[SENS:OFFS:SOUR:DIV](#)

Sets or reads out the basic frequency range Divider to get the Source frequency when offset type is **Source/Receivers**.

---

[SENS:OFFS:REC:DIV](#)

Sets or reads out the basic frequency range Divider to get the Receiver frequency when offset type is **Source/Receivers**.



To enter the basic frequency range offset, use the **Offset** softkey.

---

[SENS:OFFS:PORT:OFFS](#)

Sets or reads out the basic frequency range Offset of Port n when offset type is **Port1/Port2**.

---

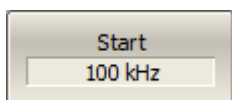
[SENS:OFFS:SOUR:OFFS](#)

Sets or reads out the basic frequency range Offset to get the Source frequency when offset type is **Source/Receivers**.

---

[SENS:OFFS:REC:OFFS](#)

Sets or reads out the basic frequency range Offset to get the Receiver frequency when offset type is **Source/Receivers**.



To set the start frequency range, use the **Start** softkey.

---

[SENS:OFFS:PORT:STAR](#)

Sets or reads out the frequency sweep Start of port <Pt> when offset type is **Port1/Port2**.

---

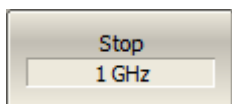
[SENS:OFFS:SOUR:STAR](#)

Sets or reads out the frequency sweep Start of the Source when offset type is **Source/Receivers**.

---

[SENS:OFFS:REC:STAR](#)

Sets or reads out the frequency sweep Start of the Receivers when offset type is **Source/Receivers**.



To set the stop frequency range, use the **Stop** softkey.

---

[SENS:OFFS:PORT:STOP](#)

Sets or reads out the frequency sweep Stop of port <Pt> when offset type is **Port1/Port2**.

---

[SENS:OFFS:SOUR:STOP](#)

Sets or reads out the frequency sweep Stop of the Source when offset type is **Source/Receivers**.

---

[SENS:OFFS:REC:STOP](#)

Sets or reads out the frequency sweep Stop of the Receivers when offset type is **Source/Receivers**.

---

# Automatic Adjustment of Offset Frequency

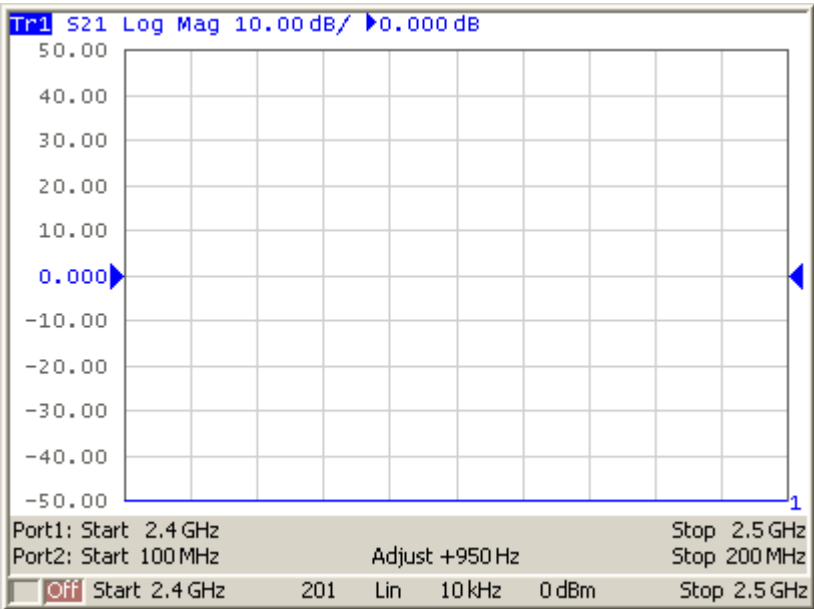
**NOTE** The availability of this feature depends on the Analyzer model (See corresponding [datasheet](#)).

When performing mixer measurements in frequency offset mode, the offset frequency must be set so that it is equal to the LO frequency. A small difference between the frequencies of the analyzer and the external LO source (frequency error) reduces the measurement accuracy.

To reduce the frequency error, it is common to synchronize the analyzer and the external LO source with a common 10 MHz reference (see [Reference Frequency Oscillator Selection](#)).

If for some reason it is not possible to synchronize the analyzer and an external source, then the **automatic offset adjustment function** can be used.

The function measures the frequency error and sets the adjust value. The analyzer uses a pair of ports (path) in the automatic offset adjustment procedure: one port as source and another port as receiver. The frequency offset between ports is adjusted for maximum response. The resulting adjust value is then applied to one of the ports. The offset adjust value is indicated in the line of the respective port in the channel window (See figure below). The function can be started by the user by pressing a button or programmed to run periodically.



Channel window in frequency offset mode with enabled automatic adjustment function of the offset frequency



Automatic adjustment is made within a  $\pm 500$  kHz range from the offset frequency set by the user. The typical residual error of automatic offset adjustment depends on the current IF filter bandwidth (See table below).

### Typical residual error of automatic offset adjustment

IF filter bandwidth	Typical residual error of automatic offset adjustment
10 kHz	500 Hz
3 kHz	50 Hz
1 kHz	15 Hz
300 Hz	5 Hz
100 Hz	2 Hz

### Settings of Automatic Offset Adjustment function

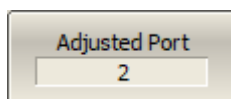


To enable/disable automatic offset adjustment function, use the following softkeys:

**Stimulus > Frequency Offset > Offset Adjust > Offset Adjust [ ON|OFF ]**

[SENS:OFFS:ADJ](#)

Turns the frequency offset adjust function ON/OFF.

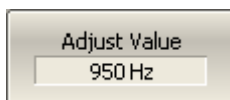


To select the port number to which the frequency adjust is applied, use the following softkeys:

**Stimulus > Frequency Offset > Offset Adjust > Adjusted Port**

[SENS:OFFS:ADJ:PORT](#)

Sets or reads out the port number to which frequency adjust is applied when the frequency offset adjust function is active.



To enter the offset adjustment value manually (typically not needed), use the following softkeys:

**Stimulus > Frequency Offset > Offset Adjust > Adjust Value**

---

[SENS:OFFS:ADJ:VAL](#)

Sets or reads out the value of the offset adjust.

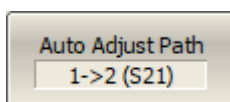
---

---

**NOTE**

Or click **Auto Adjust**, as described below.

---



To select the adjust path, i.e number of the source and receiver ports, use the following softkeys:

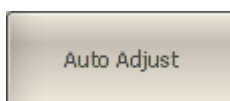
**Stimulus > Frequency Offset > Offset Adjust > Auto Adjust Path**

---

[SENS:OFFS:ADJ:PATH](#)

Sets or reads out the number of the source and receiver ports used during the offset adjust procedure.

---



To start adjustment once, use the following softkeys:

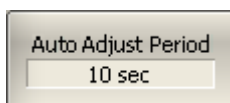
**Stimulus > Frequency Offset > Offset Adjust > Auto Adjust**

---

[SENS:OFFS:ADJ:EXEC](#)

Executes the offset adjust procedure and sets the value of the offset adjust.

---



To enable continuous adjustment enter the time interval other than zero, use the following softkeys:

**Stimulus > Frequency Offset > Offset Adjust > Auto Adjust Period**

To disable continuous adjustment set the time interval equal to zero.

---

[SENS:OFFS:ADJ:CONT:PER](#)

Sets or reads out the adjust period in seconds when the frequency offset adjust function is active.

---





## Frequency Extension System

---

**NOTE**

Frequency Extension System is available for C4209 and C4220 models.

---

Frequency extension modules (Modules) are designed to expand the frequency range of the Cobalt Series Analyzers. Modules are operated in conjunction with the Analyzer only. They cannot be used for measurements without a connection to the analyzer. The modules include the following basic elements: test/RF and LO signal frequency multipliers, a wide-band power amplifier, directional couplers, frequency converters. They also include power supply circuits and a control board. The measurement system includes:

- The Cobalt Series Analyzer (C4209 and C4220) with the configurable front panel for connection to Modules.
- Frequency Extension Modules (See table below).
- Cables for connecting Module to Analyzer.

The measurement system is controlled by the Analyzer software. The number of Modules used in the measuring system is determined by the configuration of the Analyzer in use.

Supported frequency extension modules are represented in the table below.

Module	Frequency range	Connector type
FET1854	18 GHz to 54 GHz	NMD 1,85mm, male
FEV-15	50 GHz to 75 GHz (V band)	WR-15
FEV-12	60 GHz to 90 GHz (E band)	WR-12
FEV-10	75 GHz to 110 GHz (W band)	WR-10
Custom	–	–

## Principle of Operation

The diagram of the measurement system is shown in the figure below.

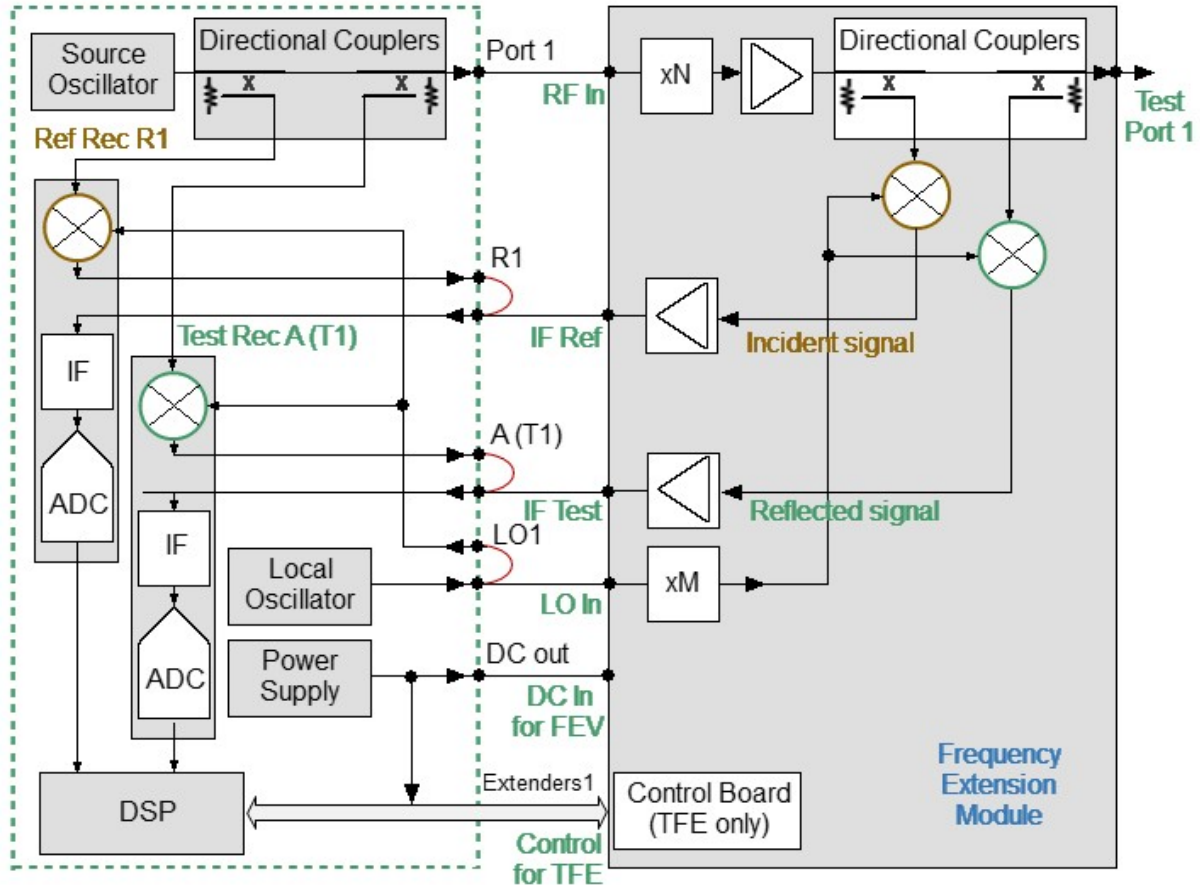


Diagram of Module connection to Analyzer

The Analyzer generates RF and LO signals in specified frequency and power ranges. The signals are transmitted to the Module through RF and LO cables.

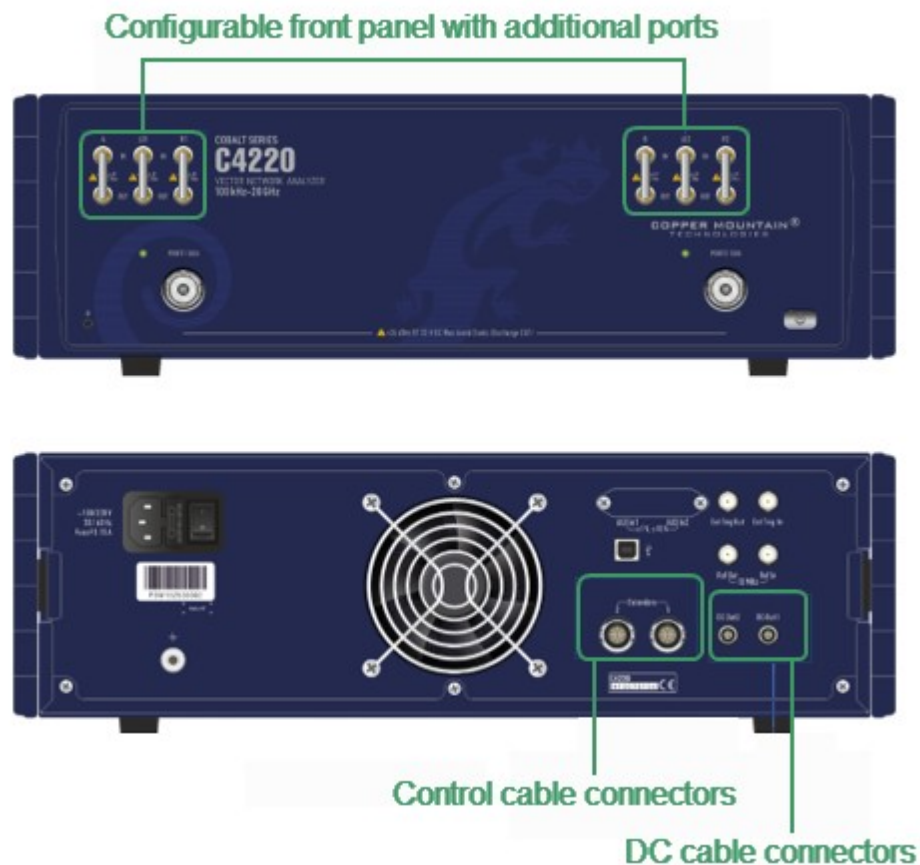
The frequency of the RF signal transmitted from the Analyzer is multiplied, filtered, and scaled in the Module to level depending on user settings.

Then, the generated RF signal is transmitted to the test port via directional couplers. The directional couplers extract the incident wave, the wave transmitted through the DUT, and the reflected wave. These signals are supplied to frequency converters for the test and reference channels. The LO signal from the Analyzer is used for conversion. The converted IF signals are amplified and passed to the Module output and then the Analyzer input via IF cables.

In turn, the Analyzer performs digital signal processing of the IF signal. An external PC uses the Analyzer software to calculate and display measurement results for complex transmission and reflection coefficients.

The C4209 and C4220 Cobalt series Analyzers have additional ports with jumper cables on the configurable front panel for connecting frequency extension systems. Connectors for controlling and powering Modules are located on the rear panel of the analyzers. Location of the connectors for the Cobalt C4220 model is shown in the figure above.

### Configurable front panel and control/DC cable connectors on rear panel



Location of connectors for the Frequency Extension Modules on the front and rear panels of the Cobalt C4220



Frequency extension modules connect to additional ports on a configurable front panel:

- «A (B) IN» — test receiver input.
- «LOx OUT» — Signal output LO (Local Oscillator).
- «Rx IN» — reference receiver input.

The number in the name (LO1 ... LO2, R1 ... R2, A, B) associates the additional port with the Test Port number.

## FET1854 Frequency Extension Module

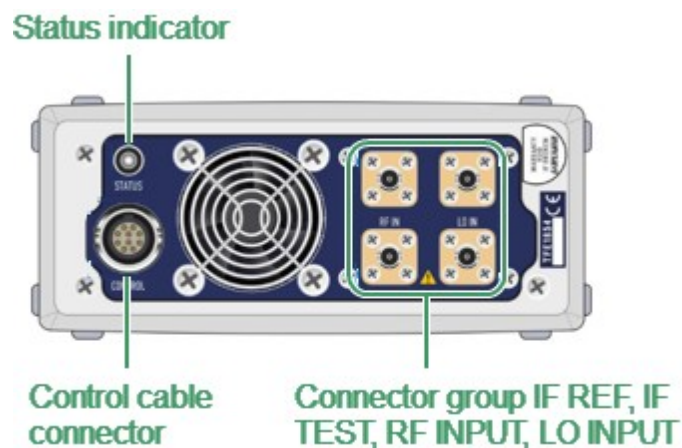
FET1854 frequency extension modules allow for measurement of the DUT S-parameters in the frequency range between 18 to 54 GHz. See the FET Operating Manual for a detailed description of the module.

The front and rear panels of the Module are shown in the figure below. The test port and ground terminal are located on the front panel. The rear panel contains a status indicator, a control cable connector, and a group of connectors for connecting the module to the Analyzer:

- Test signal (RF IN)
- LO signal (LO IN)
- IF signal of the reference channel (IF REF)
- IF signal of the test channel (IF TEST)



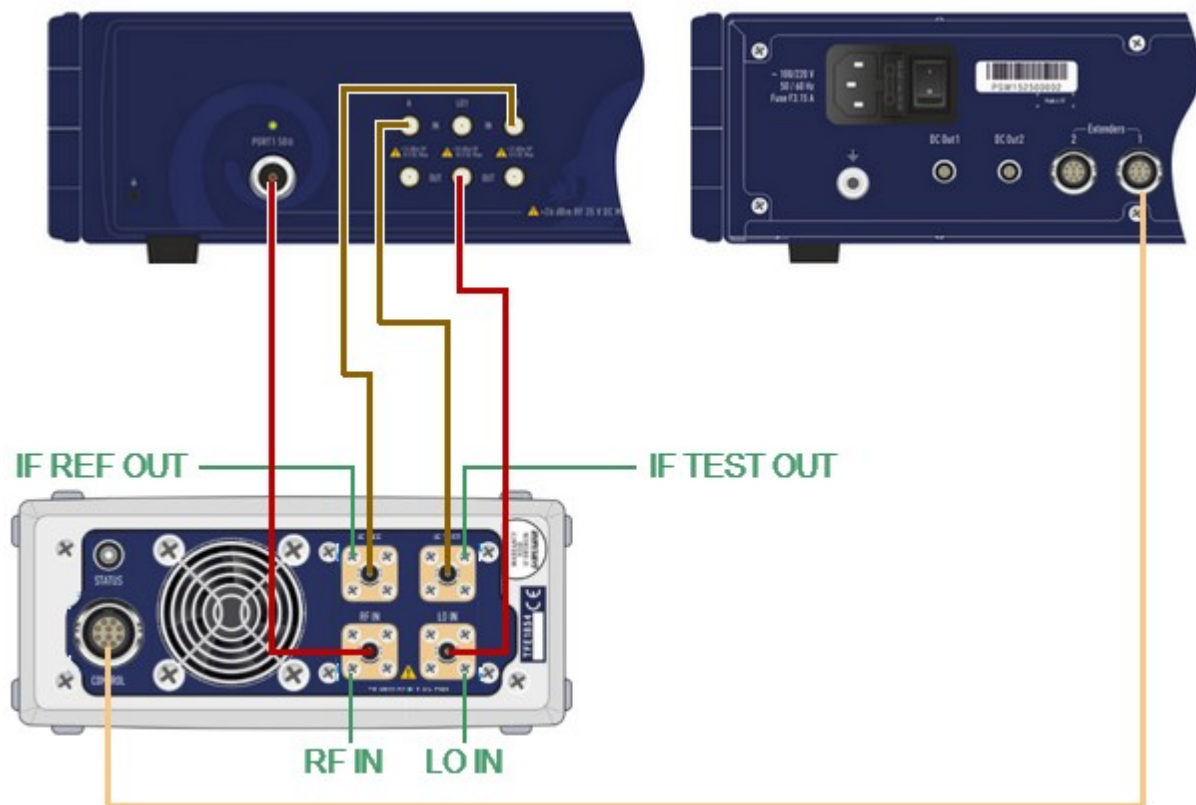
Front panel



Rear panel

The connection of the FET1854 module to the analyzer is shown in the figure below.





Connection diagram of the FET1854 module to the analyzer

Measurement system components	Connection	
	Module	Analyzer
C4209 Vector Network Analyzer PC with S2VNA Software 1 or 2 Frequency Extension Modules 1 or 2 RF cables (N, male – SMA, male) 1 or 2 LO cables (SMA, male – SMA, male)	RF IN	PORT 1 PORT 2
	LO IN	LO 1 OUT LO 2 OUT

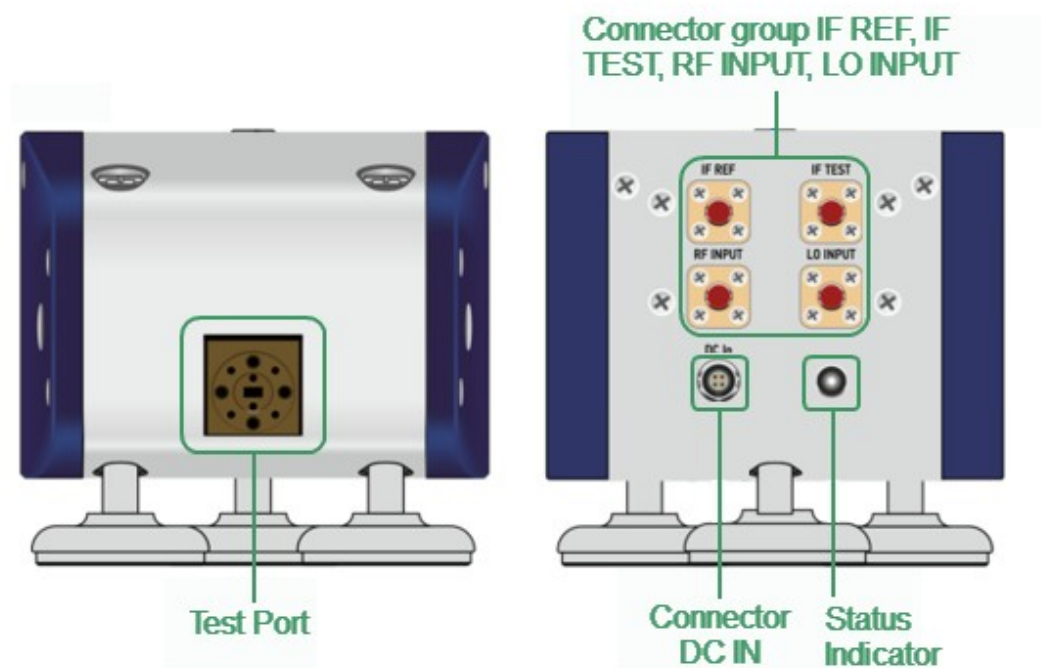
Measurement system components	Connection	
	Module	Analyzer
2 or 4 IF cables (SMA, male – SMA, male)	IF REF	R1 IN
1 or 2 control cables		R2 IN
Power supply and USB cables for Analyzer	IF TEST	A IN
Set of calibration standards, test cables, and adapters		B IN

## Frequency Extension Module FEV

FET1854 frequency extension modules allow for measurement of the DUT S-parameters in the frequency range between 50 to 110 GHz.

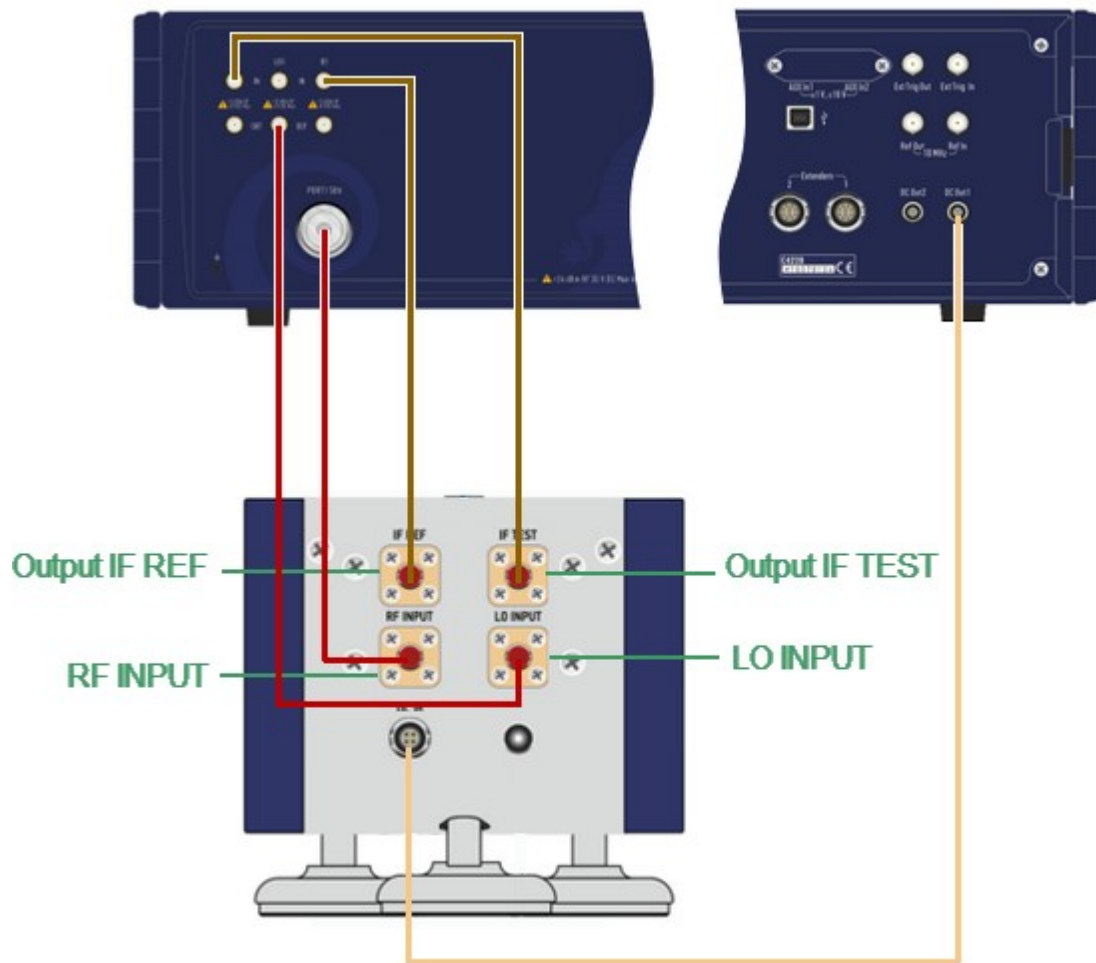
The front and rear panels of the Module are shown in the figure below. The test port is located on the front panel. The rear panel contains a status indicator, a DC in cable connector, and a group of connectors for transmission:

- Test signal (RF INPUT)
- LO signal (LO INPUT)
- IF signal of the reference channel (IF REF)
- IF signal of the test channel (IF TEST)



Front and Rear panels of FEV module

The connection of the FEV module to the analyzer is shown in the figure below.



Connection diagram of the FEV module to the analyzer

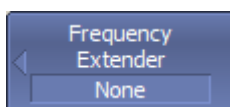
Measurement components	system	Connection	
		Module	Analyzer
C4220 Vector Network Analyzer  PC with S2VNA Software  1 or 2 Frequency Extension Modules		RF IN	PORT 1  PORT 2
		LO IN	LO 1 OUT  LO 2 OUT

Measurement components	system	Connection	
		Module	Analyzer
1 or 2 RF cables (SMA, male – SMA, male)		IF REF	R1 IN
1 or 2 LO cables (SMA, male – SMA, male)			R2 IN
2 or 4 IF cables (SMA, male – SMA, male)		IF TEST	
1 or 2 control cables			A IN
Power supply and USB cables for Analyzer			B IN
Set of calibration standards, test cables, and adapters			

## Selection of Modules in Software

By default, measurements with frequency extension modules (Module) are disabled in the Analyzer control program. Select the model of the connected Module in the software to start the Analyzer with connected extenders.

The control software will restart after selecting the model. The Analyzer is ready to work with the extender after restarting the software.



To open the frequency extender menu, use the following softkeys:

### System > Misc Setup > Frequency Extender



Then select the required Module(s):

- **FEV-15** — extender with a frequency range between 50 to 75 GHz, test port — waveguide WR-15.
- **FEV-12** — extender with a frequency range between 60 to 90 GHz, test port — waveguide WR-12.

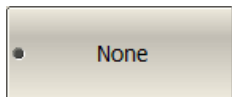


- **FEV-10** — extender with a frequency range between 75 to 110 GHz, test port — waveguide WR-10.
- **FET1854** — extender with a frequency range between 18 to 54 GHz, test port — coaxial connector NMD 1.85 mm.
- **Custom** — user defined module.

---

[SYST:FREQ:EXT:TYPE](#)

Selects or reads the frequency extender type.



To disable the work with modules, use the **None** softkey.

---

**NOTE**

The software will restart automatically when the module is turned on/off. The **Ready** message in the instrument status bar indicates that the measurements can be continued after restart.

---

---

**NOTE**

The manual setup menu opens if a **Custom** module is selected. This menu is described in [Custom Frequency Extender Setup](#).

---

---

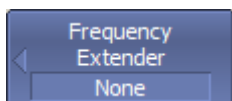
**WARNING**

All the used Modules must be of the same model with identical parameters.

---

## Configuring Modules Parameters

The output power levels of the test and LO signals are set using the Analyzer software. The insertion loss of the RF and LO cables used is also specified in the software. Settings will be available after selecting the Module model in the frequency extender menu.

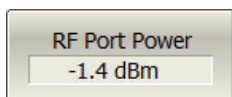


Select the required Module in the frequency extender menu (for example, FET1854):

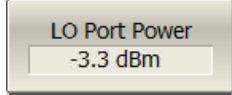


### **System > Misc Setup > Frequency extender > FET-1854**

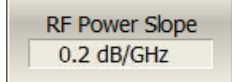
Settings the test and LO signal output power levels, and RF and/or LO cables insertion losses, will be available:



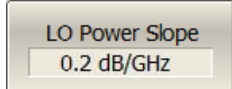
- **RF Port Power** — test signal output power level.



- **LO Port Power** — LO signal output power level.



- **RF Power Slope** — RF cable insertion loss.



- **LO Power Slope** — LO cable insertion loss.

---

### [SYST:FREQ:EXT:RFP:POW](#)

Sets or reads out the RF Port Power when the analyzer is configured to work with a frequency extender.

---

### [SYST:FREQ:EXT:RFP:PSL](#)

Sets or reads out the RF Port Power Slope when the analyzer is configured to work with a frequency extender.

---

### [SYST:FREQ:EXT:LOP:POW](#)

Sets or reads out the LO Port Power when the analyzer is configured to work with a frequency extender.

---

### [SYST:FREQ:EXT:LOP:PSL](#)

Sets or reads out the LO Port Power Slope when the analyzer is configured to work with a frequency extender.

---



---

**NOTE**

If the Module is connected using RF and LO cables those indicated in the measurement system, the following values are recommended:

Settings	FEV	FET1854
RF Port Power	+1 dBm	-1.4 dBm
LO Port Power	-4 dBm	-3.3 dBm
RF Power Slope	-0.2 dB/GHz	-0.2 dB/GHz
LO Power Slope	-0.2 dB/GHz	-0.2 dB/GHz

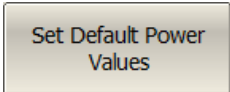
---

**NOTE**

If the Module is connected using RF and LO cables other than those indicated in the measurement system, make sure that the test and LO signal output power level at the Module input were in the ranges:

Input	FEV	FET1854
RF IN	-2 to +2 dBm	-5 to -1 dBm
LO IN	-7 to -3 dBm	-7 to -3 dBm

---

A rectangular button with a light gray background and a thin black border. The text "Set Default Power Values" is centered in a black, sans-serif font.

To set default parameters, use the following softkeys:

**System > Misc Setup > Frequency extender > Set Default Power Values**

---

**NOTE**

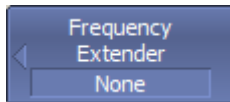
Use the status indicator on the rear panel to check the Module connection status.

---



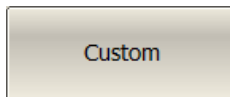
## Custom Frequency Extender Setup

The Analyzer software allows to connect custom modules. The frequency extender setup window will open after selecting the custom module in the frequency extender menu. Set the frequency range of the module and set the values of the LO multiplier (LO IN) and test signal multiplier (RF IN).

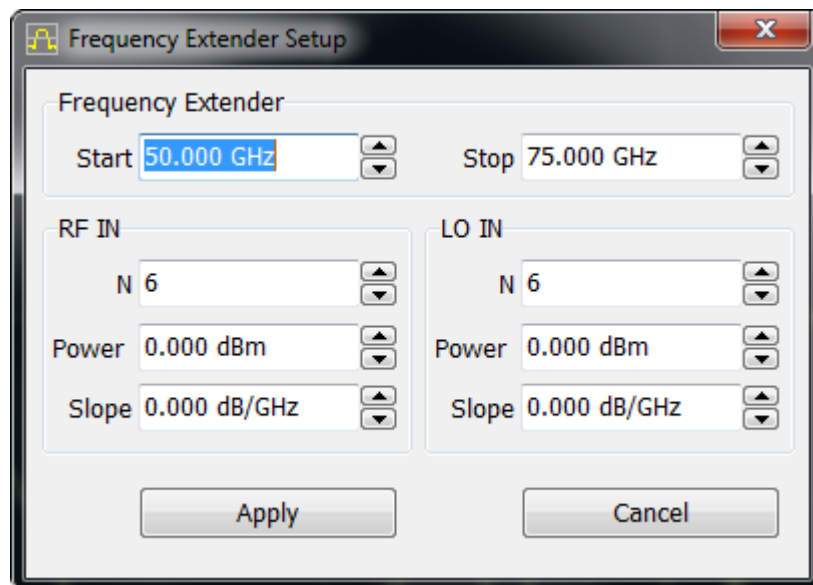


Select the custom module in the frequency extender menu:

**System > Misc Setup > Frequency Extender > Custom**



The "Frequency Extender Setup" window will open after pressing the **Custom** softkey (See figure below).



Frequency Extender Setup window

Set the parameters of the module in the window:

- Start and end frequency of the range (**Start/Stop**).
- Values of the test signal multiplier (**RF IN - N**).
- RF Port Power (**RF IN - Power**).
- RF Power Slope (**RF IN - Slope**).
- Values of the LO multiplier (**LO IN - N**).
- LO Port Power (**LO IN - Power**).

- LO Power Slope (**LO IN - Slope**).

Apply

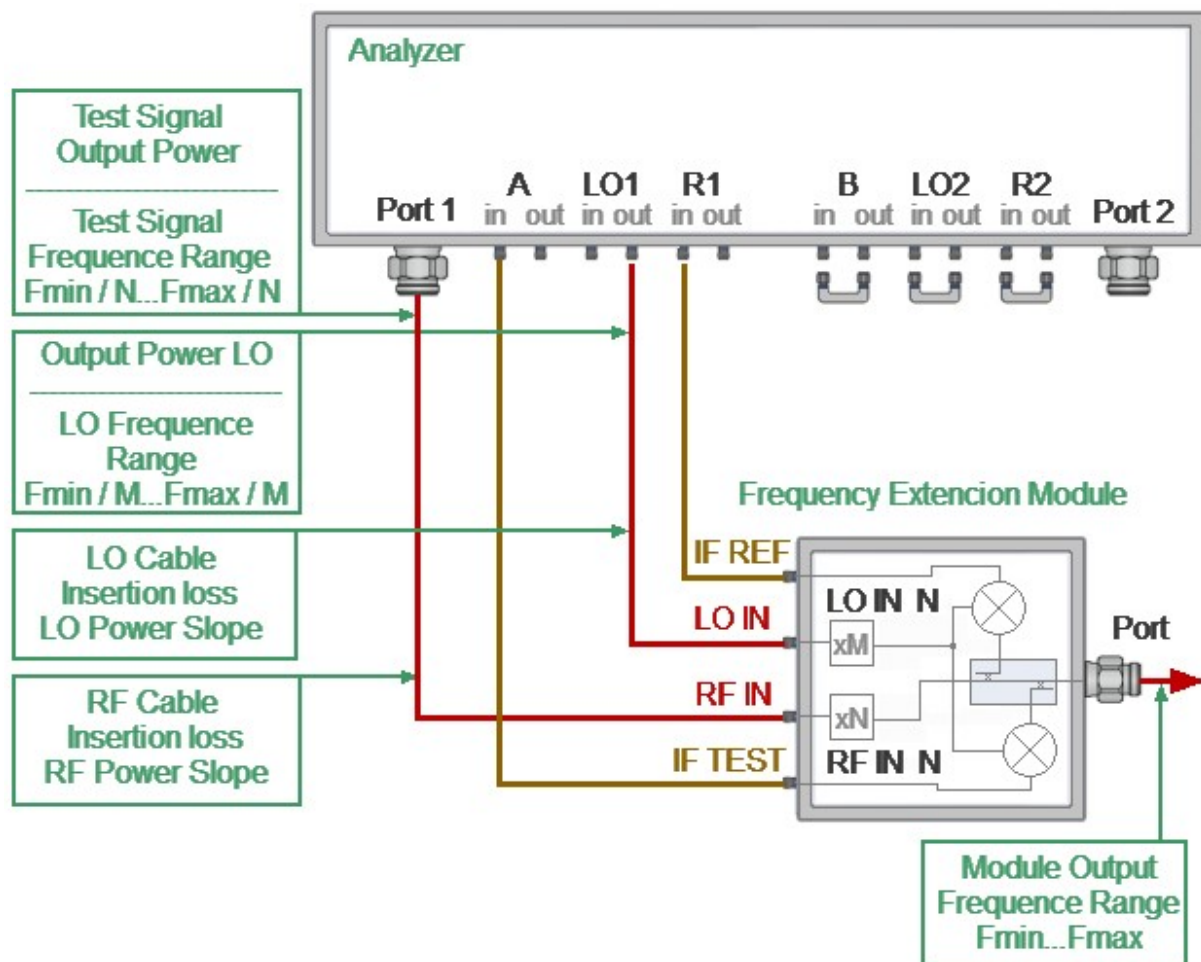
Press the **Apply** softkey to save a custom module with the configured parameters.

Cancel

Press the **Cancel** softkey to return to the extender menu without saving changes.

After selecting the Module and saving its settings, the user can change the output power level of the test and LO signals, as well as the insertion loss in the cables, using the frequency extender menu shown in [Configuring Module Parameters](#).

Module parameter settings are shown in the figure below.



Module parameter setting diagram

The module multiplies frequency of the test and LO analyzer signals. The multipliers and the frequency ranges of the module's input and output signals are shown in the following three tables.

### Frequency Extension Module FET1854

Input frequency range, GHz		Multiplier RF IN N/ LO IN N	Output frequency range, GHz
RF IN – test signal input			
Range 1	4.5 to 8.0	4	18 to 32
Range 2	4.00 to 6.25	8	32 to 50
Range 3	6.25 to 6.75	8	50 to 54
LO IN – LO signal input			
Range 1	4.5 to 9.0	4	18 to 36
Range 2	4.00 to 6.75	8	36 to 54

### Frequency Extension Module FEV-12

Input frequency range, GHz	Multiplier RF IN N/ LO IN N	Output frequency range, GHz
RF IN – test signal input		
5.0 to 7.5	12	60 to 90
LO IN – LO signal input		
5.0 to 7.5	12	60 to 90

## Frequency Extension Module FEV-10

Input frequency range, GHz	Multiplier RF IN N/ LO IN N	Output frequency range, GHz
RF IN – test signal input		
6.25 to 9.17	12	75 to 110
LO IN – LO signal input		
4.688 to 6.875	16	75 to 110

## DC Measurement

### NOTE

The DC Measurement is only available for Cobalt Series Analyzers with the HW-C-AUX option.

Some measurement applications require making DC voltage measurements in addition to standard S-parameter measurements. Cobalt Series Analyzers configured with HW-C-AUX option incorporate two auxiliary analog voltage input ports to measure DUT voltages synchronously with the VNA sweep. Voltage is measured discretely at each frequency point. The voltage measurement can be performed in the analyzer's logical channel alone or together with the measurement of complex transmission and reflection coefficients. The measured voltage values are displayed in the form of a trace in the logical channel window. This trace is assigned the name of the measured parameter in the trace status field: VAUXIn (Sweep port), where AUXIn is the number of the auxiliary input port, and Sweep port is the number of the stimulus signal output port. For example, V1(2) - AUXIn1 input, the stimulus signal output port - 2.

Voltmeter channels are independent -- the voltage can be measured simultaneously at two points in the circuit. Either input may be configured for +/- 1.0 VDC or +/- 10.0VDC measurement range.

An example of connecting a DUT to the analyzer is shown in the figure below.

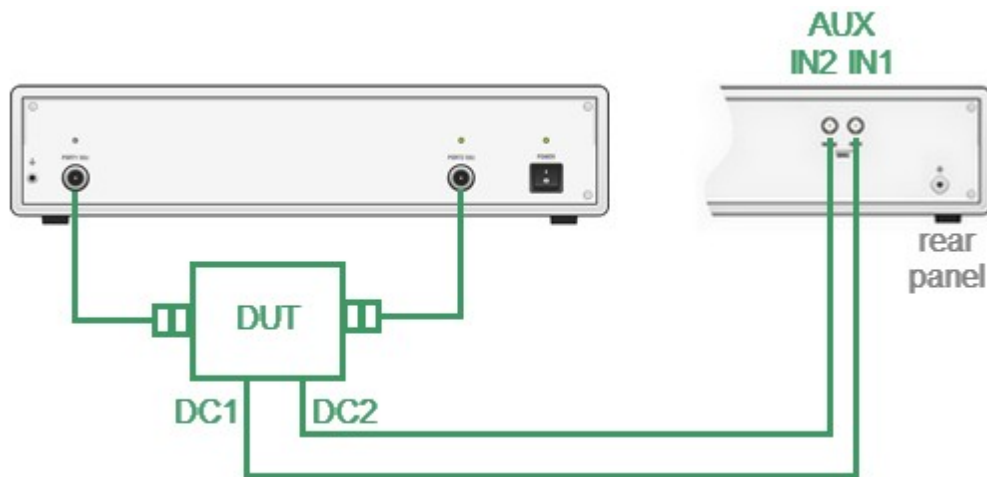
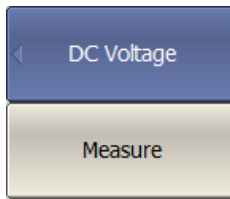


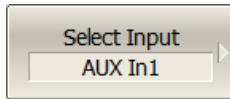
Diagram of a possible connection of a DUT to a voltmeter

Select an existing or open new window for DC voltage measurement. Make it active. Create an active trace using the Real format in this window.



To enable the DC voltage measurement mode, use the following softkeys:

**Measurement > DC Voltage > Measure**



To select input for measuring DC voltage, use the following softkeys:

**Measurement > DC Voltage > Select Input**

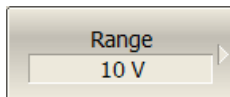
Then select the required input:

- **AUX In1**
- **AUX In2**

---

[CALC:PAR:DEF](#)

Selects the measurement parameter of the trace.



To select the range of the measured voltage (1 V or 10 V), use the following softkeys:

**Measurement > DC Voltage > Range**

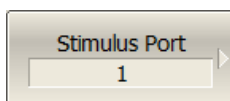
Then select the required range:

- **1V**
- **10V**

---

[SENS:VOLT:DC:RANG:UPP](#)

Sets or reads out the DC voltage range at the connector AUX1 or AUX2.



To select stimulus output port number, use the following softkeys:

**Measurement > DC Voltage > Stimulus Port**

Then select the required port [ 1|2 ]

DC voltage measurement will be performed only when the stimulus is present in the selected port.

---

[CALC:PAR:SPOR](#)

Sets or reads out the number of the stimulus port when performing DC Voltage measurements.



---

---

**NOTE**

The trace from the DC voltage measurement mode can be switched to another mode. To switch to another mode, change the measured trace parameter.

---



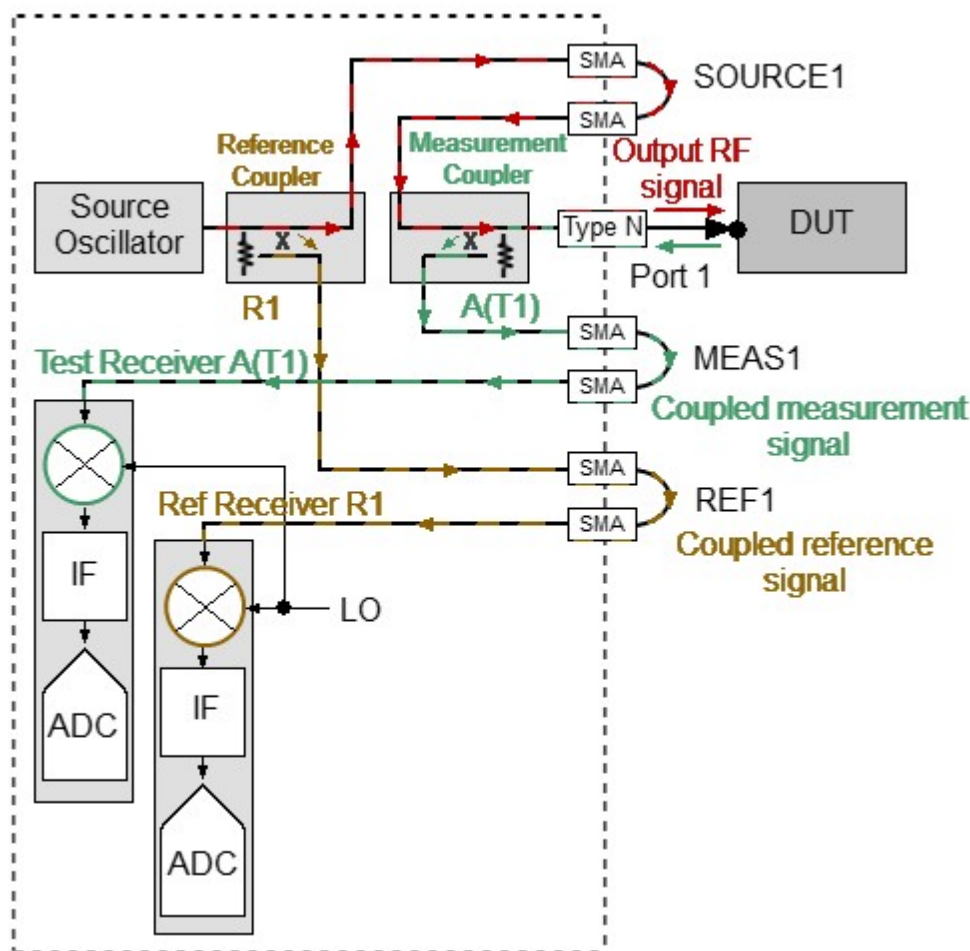
## Direct Receiver Access

### NOTE

Direct Receiver Access is available for C2209 and C2220 models.

The Cobalt C2209 and C2220 Analyzers have adjustable port configurations with direct access to receivers. This option allows for a variety of test applications that require a wider dynamic and power range. Direct receiver access enables testing of high-power devices. Additional amplifiers, attenuators, various filters and matching pads for each port can be introduced into the path of reference oscillator and the receiver path to provide optimal, near-realistic operation of the receivers and the DUT. In common mode, when direct access to the receivers is not used, additional ports are connected by jumper cable assemblies (See [Cobalt-Series](#)).

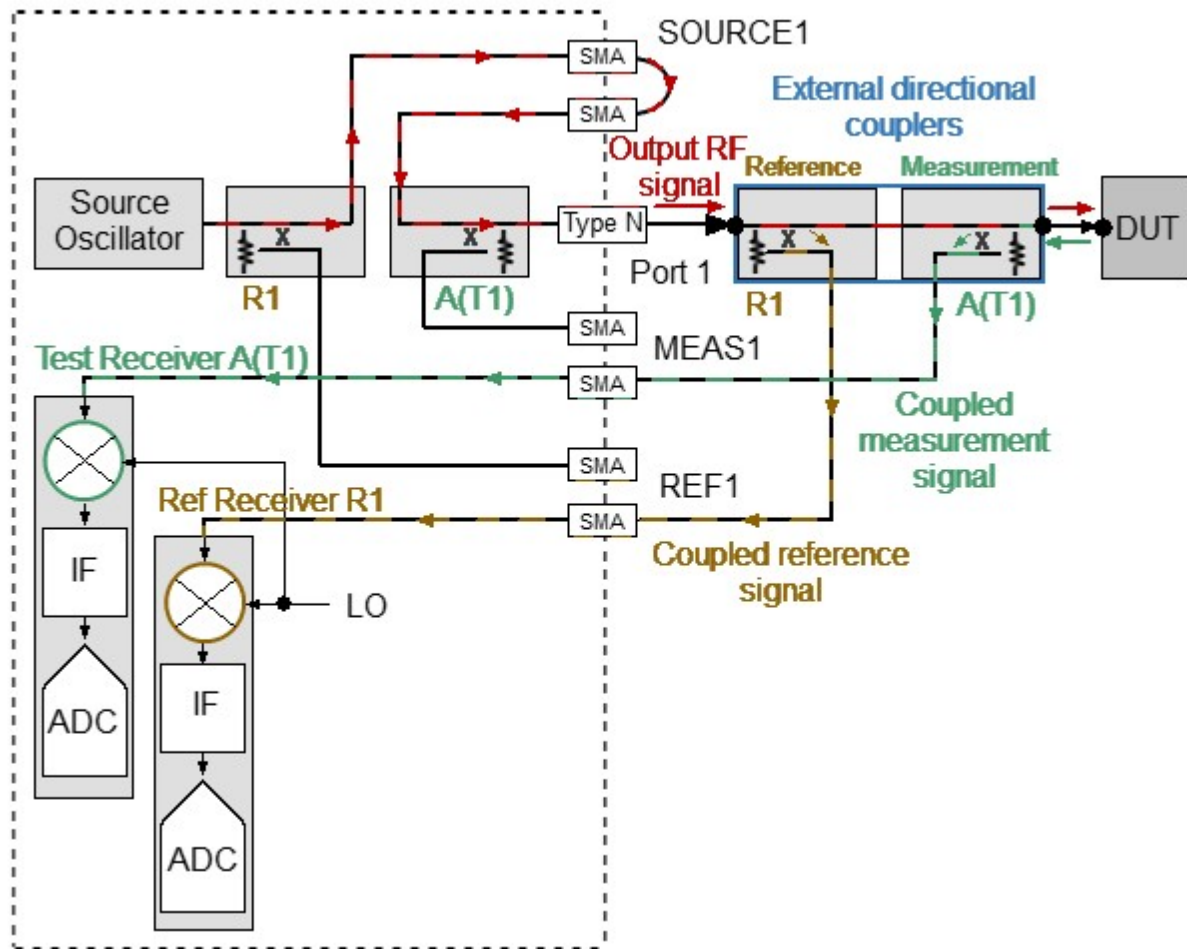
Analyzers with direct access to the receivers have different functional schemes depending on the range of operating frequencies. The C2209 Analyzer, with a frequency range up to 9 GHz, has the following functional diagram (See figure below).



C2209 Analyzer signal propagation in common mode

In common mode, all additional port cable jumpers are connected. The Analyzer takes measurements using built-in modules (see above figure). If the jumpers are removed, external directional couplers, bridges, or amplifiers can be connected to the adjustable ports. The signal propagation corresponding to the direct access mode to the receivers is shown in the figure below.

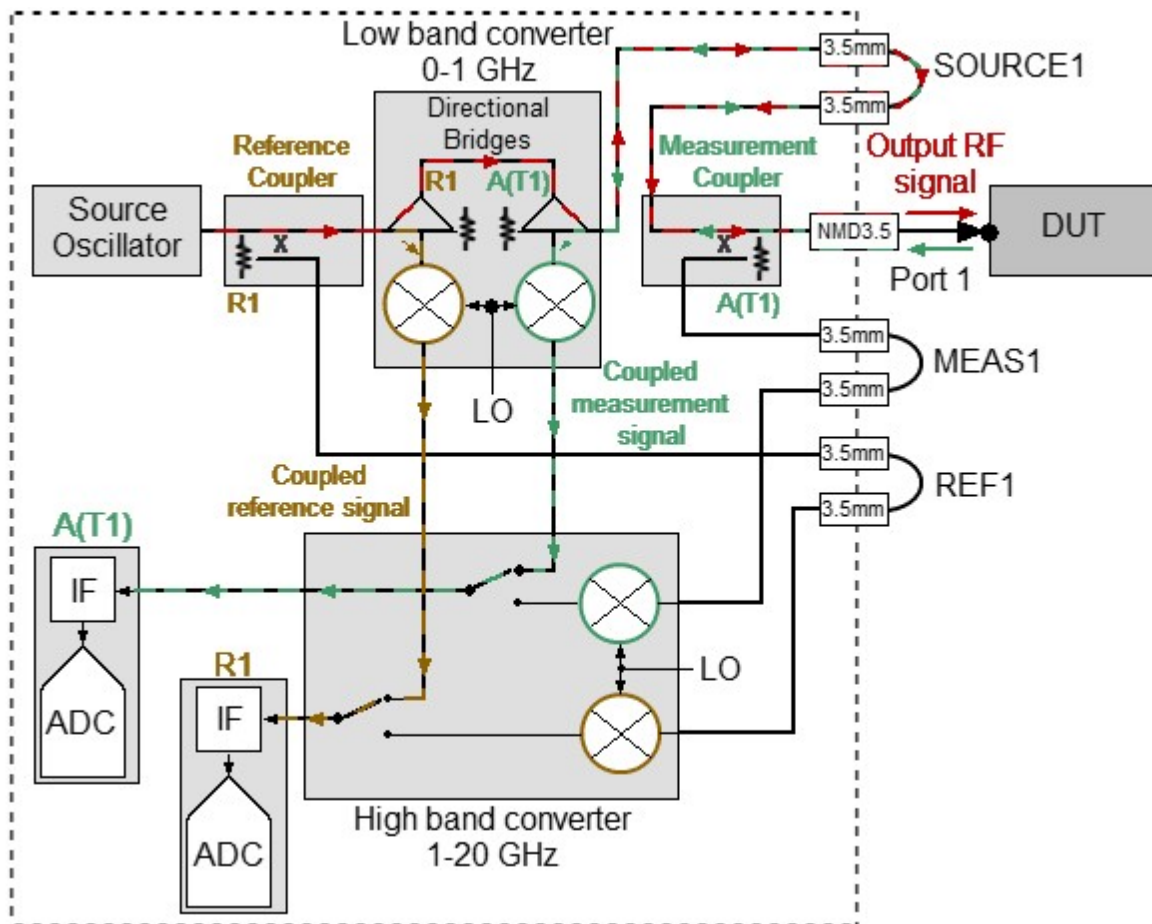
For the C2209 Analyzer with a frequency range up to 9 GHz, there is no need to switch in S2VNA software to enable direct mode.



C2209 Analyzer signal propagation in direct receiver access mode

The C2220 Analyzer, with a frequency range of up to 20 GHz, has a more complicated functional diagram (See figure below). This analyzer is a broadband instrument that uses two different internal devices for signal separation — the directional bridge and the directional coupler for each port respectfully. Both devices operate together to provide the frequency coverage beginning from 100 kHz up to 20 GHz for reference and measurement paths separately.

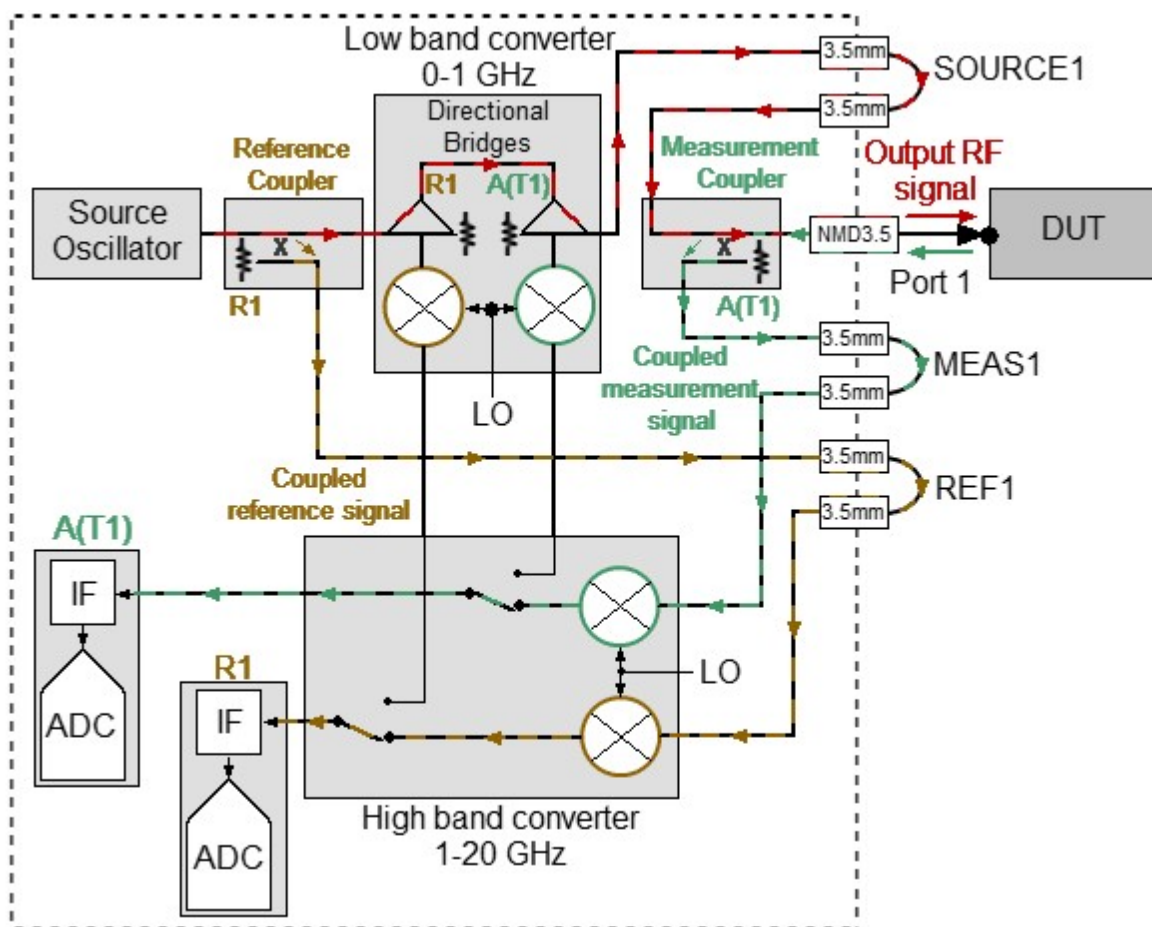
The receiver path of the Analyzer includes low and high band converter units which work independently. The first of those operates in the frequency band below 1 GHz. For signal converting, it uses output signals transmitted from directional bridges (See figure below). The low band converter with bridges is merged into one physical module.



Signal propagation in common mode while the analyzer operates in the frequency range between 100 kHz and 1 GHz

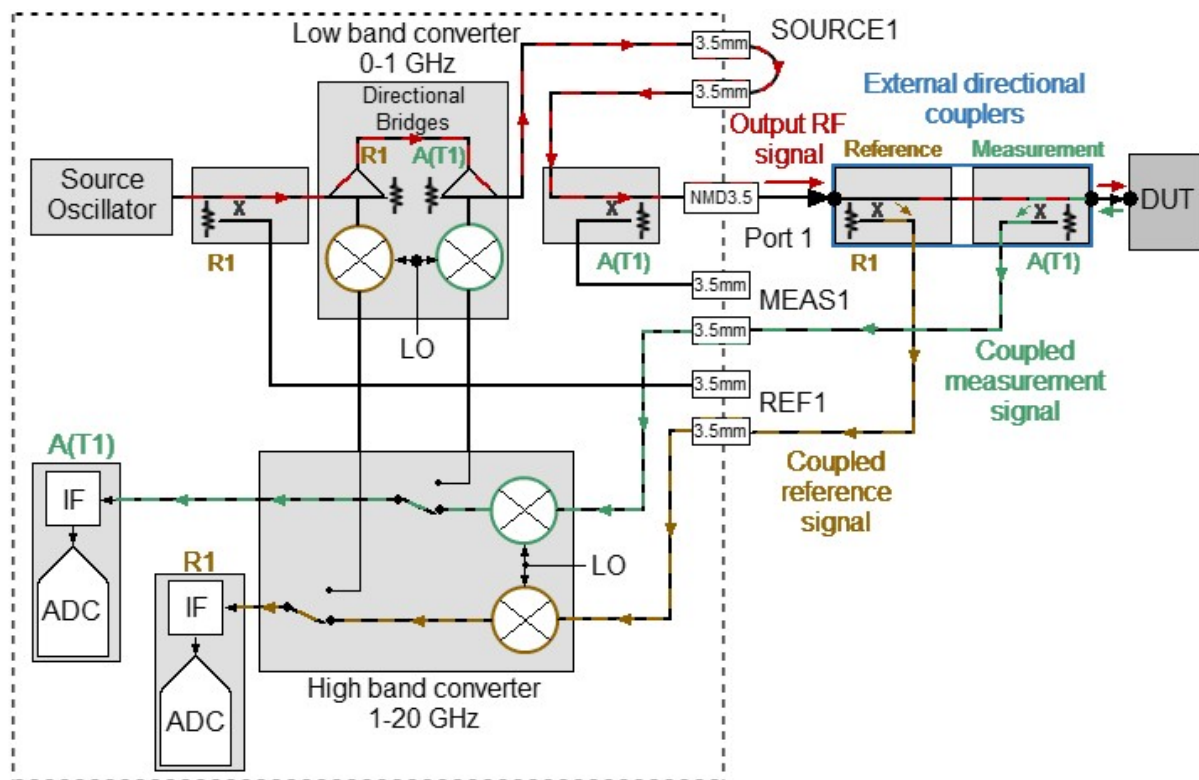
The other module — the high band converter, receives signals from directional couplers as shown in the figure below, and covers the rest of the operating frequency range.

In common mode, when all loops are connected, the Analyzer manages these converters and gathers reference and measurement signals in the entire frequency range for further analysis. This receiver design allows the user to achieve optimal raw (uncorrected) parameters such as directivity, source and load match, as well as providing higher dynamic range.



Signal propagation in common mode while the analyzer operates in the frequency range between 1 GHz and 20 GHz

Typical configuration for direct receiver access mode is demonstrated below.



Typical signal propagation in direct receiver access mode

The Analyzer operates over the entire frequency range using only the high band converter.

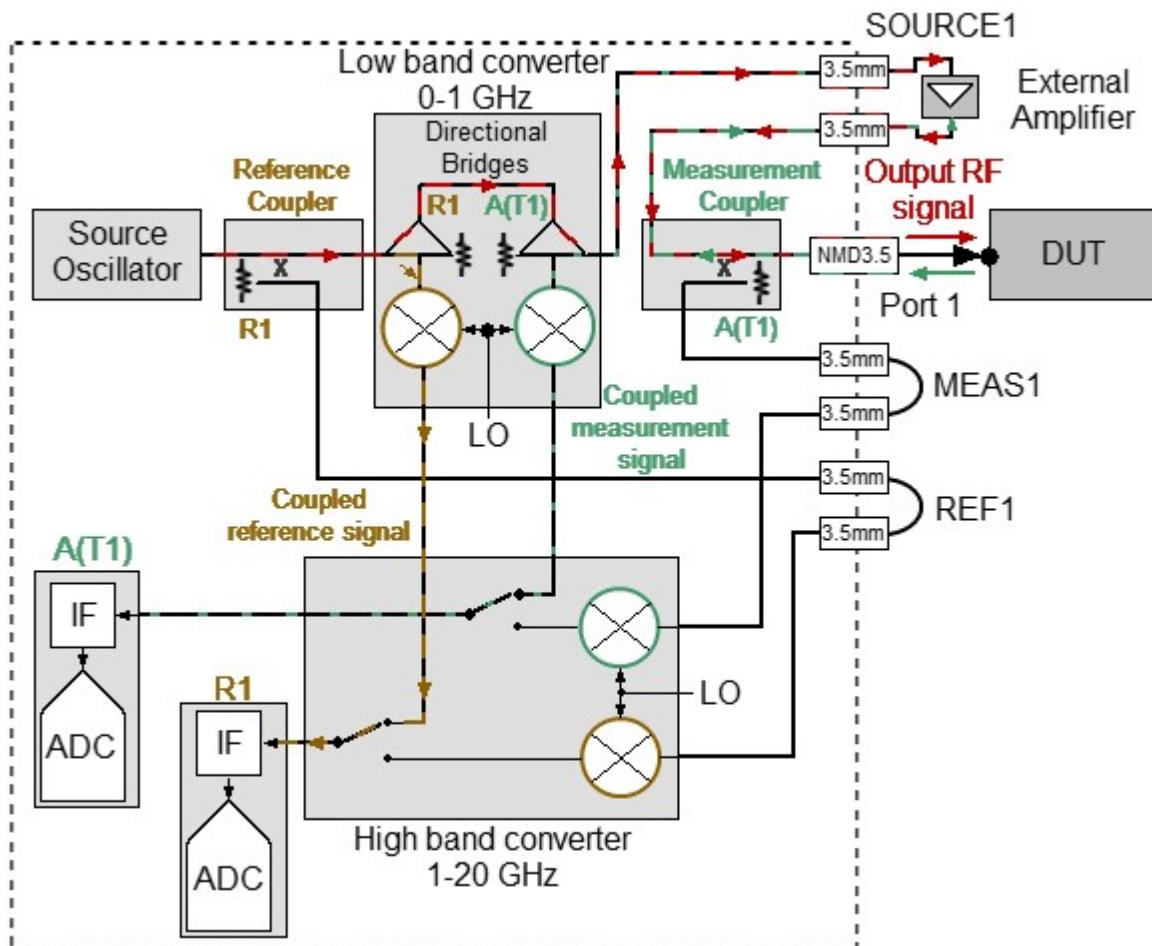
It's the most popular setup for applications requiring wider dynamic and power ranges. The above figure does not show external units, such as amplifiers, attenuators, directional devices (couplers or bridges) and so on, which are required for regular measurements. It is possible to use custom external directional devices for signal separation, which have specified parameters and work in the appropriate frequency range.

The C2220 has a special type of high band converter. It's able to process and convert input signals in the entire frequency range. In order to stop switching between low and high converters during signal sweep or receiver treatment, the analyzer offers an extra mode. This mode has a similar name, "Direct access to receivers", in the Analyzer software. It enables the user to manage the measurement switching process by pressing one single softkey.

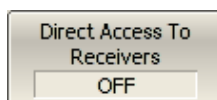
It is recommended to apply this mode when performing measurements with any external directional devices. It will be particularly valuable when an external amplifier is used together with VNA internal directional devices. Any amplifier will block direct access to internal bridge from the DUT side (refer to the figure below). It will



decrease both the system effectiveness and measurement accuracy. In this case, the VNA can analyze signals derived from internal directional coupler only using the above-mentioned mode.



Signal propagation with external amplifier installed on oscillator path while the analyzer operates in the frequency range between 100 kHz and 1 GHz



The mode allows the user to carry out measurements with signals transmitted from external directional devices within the entire frequency range, eliminating the need for switching between low and high frequency ranges of the Analyzer (for C2220 only).

Use the following softkeys to enable/disable direct receiver access:

**System > Misc Setup > Direct Access To Receivers [ON | OFF]**

---

[SYST:REC:DIR:ACC](#)

Turns the direct access to the receiver function ON/OFF (C2220model only).

---

## State Saving and Data Output

The following section describes the processes of saving and recalling:

- The set parameters of the Analyzer, calibration, measured, and memorized data are stored in the Analyzer status file and can be reloaded (See [Analyzer States](#)).
- The states of the individual channels are stored in the internal memory. Up to 4 states can be stored while the Analyzer is running. When the Analyzer is powered off, the contents of the state files are destroyed (See [Channel States](#)).
- Individual trace data in a \*.CSV file (See [Trace Data CSV Files](#)).
- Device S-parameters in a Touchstone file (See [Trace Data Touchstone Files](#)).

## Analyzer State

The Analyzer state, calibration and measured data can be saved on the hard disk to an Analyzer state file and later uploaded back into the Analyzer program. The following four types of saving are available:

<b>State</b>	The Analyzer settings.
<b>State &amp; Cal</b>	The Analyzer settings and the table of calibration coefficients.
<b>State &amp; Trace</b>	The Analyzer settings and data traces <sup>1</sup> .
<b>All</b>	The Analyzer settings, table of calibration coefficients, and data traces and memory <sup>1</sup> .
<b>State &amp; Cal &amp; Mem</b>	The Analyzer settings, table of calibration coefficients and memory <sup>1</sup> .
<p>1 When recalling the state with saved data traces, the trigger mode will be automatically set to «Hold» so that the recalled traces are not erased by currently measured data.</p>	

The Analyzer settings that are saved into the Analyzer state file are parameters that can be set in the following submenus of the softkey menu:

- All the parameters in the **Stimulus** submenu.
- All the parameters in the **Measurement** submenu.
- All the parameters in the **Format** submenu.
- All the parameters in the **Scale** submenu.
- All the parameters in the **Average** submenu.
- All the parameters in the **Display** submenu except for **Properties**.
- All the parameters of the **Markers** submenu.
- All the parameters of the **Analysis** submenu.
- **Ref Source** and **System Correction** parameters in the **System** submenu.

To save and recall a state file, ten softkeys labeled **State01**, ... **State10** can be used. Each of the softkeys correspond to a \*.STA file with the same name.

To have the Analyzer state automatically recalled after each start of the instrument use the *Autorecall.sta* file. Use the **Autorecall** softkey to save the corresponding file and thus enable this function.

To disable the automatic recall of the Analyzer state, delete the *Autorecall.sta* file using the specific softkey.

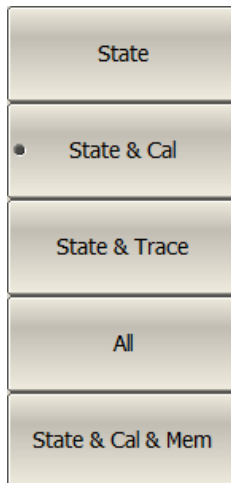
The files can be saved and recalled with arbitrary names. To save, use the **File...** softkey, which will open the **Save as** dialog box.

## Analyzer State Saving



To set the type of saving, use the following softkeys:

### Save/Recall > Save Type



Then select the required save type:

- **State**
- **State & Cal**
- **State & Trace**
- **All**
- **State & Cal & Mem**

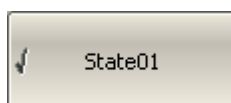
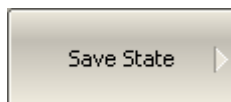
### [MMEM:STOR:STYP](#)

Selects the type of the Analyzer or channel state saving using the [MMEM:STOR](#) or [MMEM:STOR:CHAN](#) command.

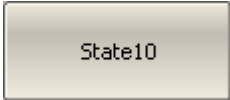
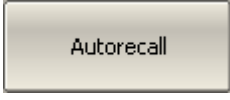
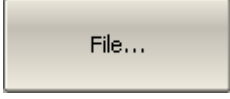


To save the state, use the following softkeys:

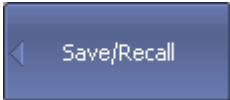



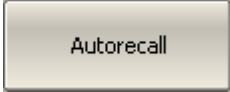

### Save/Recall > Save State



To save a state into one of the ten files, use the **State01...** **State10** softkeys.

...	A check mark in the left part of the softkey indicates that the state with the corresponding number is already saved.
	
	To save the state, which will be automatically recalled after each start of the Analyzer, use the <b>Autorecall</b> softkey.
	A check mark on the softkey indicates that such a state is already saved.
	To save a state into the file with an arbitrary name use the <b>File...</b> softkey.
<a href="#">MMEM:STOR</a>	Saves the Analyzer state into a file.

## Analyzer State Recalling

	To recall the state from an Analyzer state file, use the following softkeys:
	<b>Save/Recall &gt; Recall State</b>
	Click the required softkey of the available <b>State01...State10</b> .
...	If the state with a certain number was not saved, the corresponding softkey will be grayed out.
	
	The state automatic recall file can be selected by clicking the <b>Autorecall</b> softkey.
	To recall a state from the file with an arbitrary name, use the <b>File...</b> softkey.

---

[MMEM:LOAD](#)

Recalls the specified Analyzer state file.

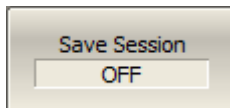
---

## Session Saving

When enabled, this function automatically saves a session upon exiting the program and resumes it when the Analyzer is turned on next time. The stored session parameters include the Analyzer settings, table of calibration coefficients, and data and memory traces.



To enable the Save Session function, use the following softkeys:



**Save/Recall > Save Session [ ON|OFF ]**

---

## Channel State

A channel state and channel calibration can be saved into the Analyzer memory.

The channel state saving procedure is similar to that of Analyzer state saving, and the same saving types (See [Analyzer State](#)) are applied to the channel state saving function.

Unlike the Analyzer state, the channel state is saved into the Analyzer's inner volatile memory (not to the hard disk), and is cleared when the Analyzer is turned off. For channel state storage, there are four memory registers: **A, B, C, D**.

The channel state saving function allows to copy easily the settings of one channel to another one.

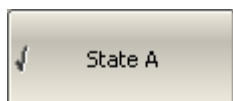
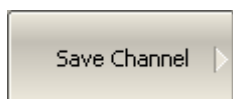
Channel calibrations are saved to an \*.STA file. The state saving function copies the calibration to another channel. By default, the \*.STA file is saved in the \State subdirectory of the main application directory.

### Channel State Saving



To save the active channel state, use the following softkeys:

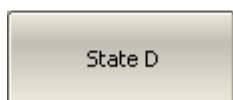
**Save/Recall > Save Channel**



To save a state into one of the four memory registers, use the **State A...State D** softkeys.

• • •

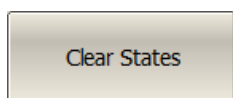
A check mark in the left part of the softkey indicates that the state with the corresponding number is already saved.



---

[MMEM:STOR:CHAN](#)

Saves the Analyzer state of the items set for the active channel into one of the four memory registers.



To clear the all channel state, use the **Clear States** softkeys.



---

[MMEM:STOR:CHAN:CLE](#)

Clears the memory of the channel state saved using the [MMEM:STOR:CHAN](#) command.

---

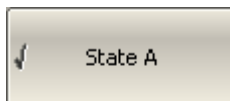
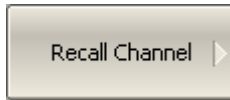
## Channel State Recalling

---



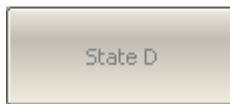
To recall the active channel state, use the following softkeys:

**Save/Recall > Recall Channel**



Click the required softkey from those available: **State A...**  
**State D.**

• • •



If the state with a certain number was not saved, the corresponding softkey will be grayed out.

---

[MMEM:LOAD:CHAN](#)

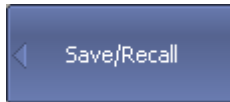
Recalls the Analyzer state for the active channel from the memory register.

---

## Calibration Saving/Recalling

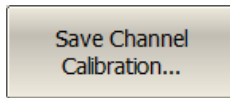
The calibration of a channel can be saved to a file. The file contains the frequency data, calibration coefficients and calibration info. The files have \*.CAL extension and are saved in the \State subdirectory of the main application directory.

### Channel Calibration Saving



To save the channel calibration use the following softkeys:

**Save/Recall > Save Channel Calibration...**



---

[MMEM:STOR:CHAN:CAL](#)

Stores the calibration of the specified channel to the file.

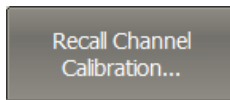
---

### Channel Calibration Recalling



To recall the channel calibration, use the following softkeys:

**Save/Recall > Recall Channel Calibration...**



---

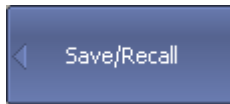
[MMEM:LOAD:CHAN:CAL](#)

Recalls the calibration for the specified channel from the file.

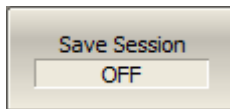
---

## Session Resuming

When enabled, this function automatically saves the *Autorecall.sta* file upon exiting the program, so when the program starts next time the Analyzer settings, calibration and memory traces are resumed.



To enable the save session function, use the following softkeys:



**Save/Recall > Save Session [ ON|OFF ]**

## Trace Data CSV File

Trace data can be saved as a \*.CSV file (comma separated values). The \*.CSV file contains comment and trace data lines. Comments start from the «!» symbol.

Before saving the \*.CSV file, set the trace type, value delimiter type, and other parameters in the **Save Trace Data** submenu (see the table below). Then, click the **Save...** button to save the values to the file.

Parameter	Definition
<b>Scope</b>	Type of trace to be saved: <ul style="list-style-type: none"><li>• <b>Active Trace</b>.</li><li>• <b>All Traces of Chan</b> — all traces of the active channel.</li></ul>
<b>Format</b>	Data save format: <ul style="list-style-type: none"><li>• <b>Displayed</b> — the format in which the trace is set (see <a href="#">Format Setting</a>).</li><li>• <b>Real-Imag</b> — real and imaginary parts.</li><li>• <b>db-Angle</b> — logarithmic magnitude in dB and phase in degrees.</li></ul>
<b>Comment</b>	Enable/disable the entry in the comment file. The comment contains 3 lines: <ol style="list-style-type: none"><li>1. Model, serial number, software version.</li><li>2. Save date (in the dd.mm.yyyy hh:mm:ss format).</li><li>3. The name of the saved parameters and their dimensionality.</li></ol>
<b>Stimulus</b>	Enable/disable recording to the file frequency at measurement point.
<b>Decimal Separator</b>	The type of delimiters between stored values, as well as the type of decimal separator: <ul style="list-style-type: none"><li>• <b>Local</b> — decimal separator — comma, value separator — semicolon.</li><li>• <b>Point</b> — decimal separator — point, value separator — comma.</li></ul>

The trace data is saved to \*.CSV in the following format:

! Comment		
F[0],	Data1,	Data2
F[1],	Data1,	Data2
...		
F[N],	Data1,	Data2

**F[n]** — frequency at measurement point n.

**Data1** — trace response in rectangular format, real part in Smith chart and polar format.

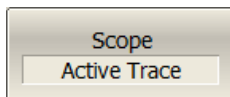
**Data2** — zero in rectangular format, imaginary part in Smith chart and polar format.

### Editing saving parameters



To open save trace submenu , use the following softkeys:

**Save/Recall > Save Trace Data**



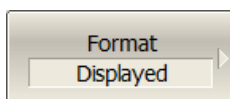
To select the type of trace, use the **Scope** softkey.

Then select the required type:

- **Active Trace**
- **All Traces of Chan**

[MMEM:STOR:FDAT:SCOP](#)

Sets whether the active trace or all active traces will be saved using the [MMEM:STOR:FDAT](#) command.



To select the format for saving data, use the **Format** softkey.

Then select the required format:

- **Displayed**
- **Real-Imag**

---

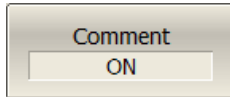
- **db-Angle**

---

[MMEM:STOR:FDAT:FORM](#)

Sets the data format when the \*.CSV file is saved using the [MMEM:STOR:FDAT](#) command.

---



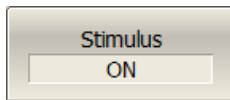
To enable/disable recording in the header file, use the **Comment** softkey.

---

[MMEM:STOR:FDAT:COMM](#)

Turns the comment strings at the beginning of the \*.CSV file saved with the [MMEM:STOR:FDAT](#) command ON/OFF

---



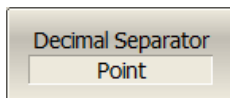
To enable/disable writing to a file frequency at measurement point., use the **Stimulus** softkey.

---

[MMEM:STOR:FDAT:STIM](#)

Turns the column with the stimulus data in the \*.CSV file saved with the [MMEM:STOR:FDAT](#) command ON/OFF.

---



To select the type of separators, use the **Decimal Separator** softkey.

Then select the required format:

- **Local**
  - **Point**
- 

[MMEM:STOR:FDAT:SEP](#)

Sets the separators used when the \*.CSV file is saved with the [MMEM:STOR:FDAT](#) command.

---

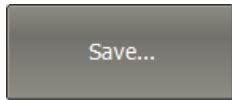
## CSV File Saving

---



To save the trace data, use the following softkeys:

**Save/Recall > Save...**



Enter the file name in the dialog that appears.

---

[MMEM:STOR:FDAT](#)

Saves the data of one or several traces to a CSV file.

---



## Trace Data Touchstone File

The Analyzer allows to save S-parameters to a Touchstone file. Files in this format are typical for most circuit simulator programs. The Touchstone file contains frequency values and S-parameters. Only one (active) trace is saved to the file (See [Active Trace Selection](#)).

The Touchstone file saving function is applied to individual channels. Activate the channel to use this function.

The \*.S1P files are used for saving S11 and S22 parameters of a one-port device.

The \*.S2P files are used for saving all four S-parameters of a two-port device.

---

### NOTE

If a channel does not have all the S-parameter traces, only available S-parameter responses will be represented. For example, if one S11 trace is enabled, the S21 response will be represented, and the S12 and S22 responses will not be represented. The missing S-parameters are displayed as zeroes in the file.

If full two-port calibration is active, all four S-parameters in a channel are measured, independently of the number of the traces.

---

The Touchstone file contains comments, header, and trace data lines. The header starts from the «#» symbol. Comments start from the «!» symbol. Comment contains following strings:

- Model, serial number, software version.
- Save date (in dd.mm.yyyy hh:mm:ss format).
- The name of the saved parameters and their units.

The \*.S1P Touchstone file for one-port measurements:

<b>! Comments</b>		
<b># Hz S FMT R Z0</b>		
<b>F[0]</b>	{S11}'	{S11}"
<b>F[1]</b>	{S11}'	{S11}"
	...	
<b>F[N]</b>	{S11}'	{S11}"

The \*.S2P Touchstone file for two-port measurements:

<b>! Comments</b>								
<b># Hz S FMT R Z0</b>								
<b>F[0]</b>	{S1 1}'	{S11} "	{S2 1}'	{S2 1}"	{S1 2}'	{S1 2}"	{S2 2}'	{S22}"
<b>F[1]</b>	{S1 1}'	{S11} "	{S2 1}'	{S2 1}"	{S1 2}'	{S1 2}"	{S2 2}'	{S22}"
				...				
<b>F[N]</b>	{S1 1}'	{S11} "	{S2 1}'	{S2 1}"	{S1 2}'	{S1 2}"	{S2 2}'	{S22}"

**Hz** — frequency measurement units (**kHz**, **MHz**, **GHz**);

**FMT** — data format:

- **RI** — real and imaginary parts;
- **MA** — linear magnitude and phase in degrees;
- **DB** — logarithmic magnitude in dB and phase in degrees;

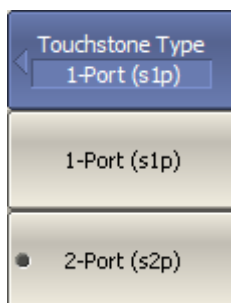
**Z0** — reference impedance value;

**F[n]** — frequency at measurement point n;

**{...}'** — {real part (RI) | linear magnitude (MA) | logarithmic magnitude (DB)};

**{...}''** — {imaginary part (RI) | phase in degrees (MA) | phase in degrees (DB)}.

## Touchstone File Saving



To select the saving type, use the following softkeys:

**Save/Recall > Save Data To Touchstone File > Type**

Then select the required Touchstone type:

- **1-Port (s1p)**
- **2-Port (s2p)**

[MMEM:STOR:SNP:TYPE:S1P](#)

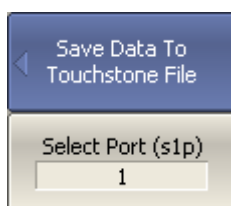
Sets and reads out the one-port Touchstone file type (\*.S1P) and the port number.

[MMEM:STOR:SNP:TYPE:S2P](#)

Sets and reads out the two-port Touchstone file type (\*.S2P) and the ports number.

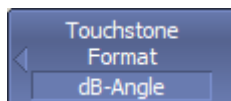
[MMEM:STOR:SNP:TYPE?](#)

Reads out the type of Touchstone file.



For one-port saving type, select the port number using the following softkeys:

**Save/Recall > Save Data To Touchstone File > Select Port (s1p).**

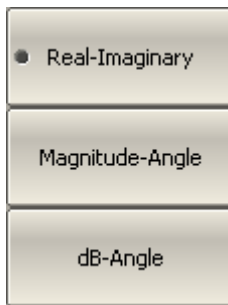


To select the data format, use the following softkeys:

**Save/Recall > Save Data To Touchstone File > Format**

Then select the required Touchstone format:

- **Real-Imaginary**

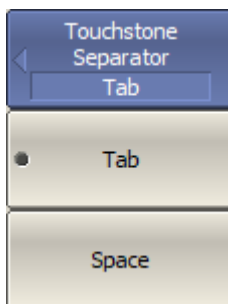


- **Magnitude-Angle**
- **dB-Angle**

---

[MMEM:STOR:SNP:FORM](#)

Sets and reads out the data format for the S-parameter.



Select the Touchstone separator type using the following softkeys:

**Save/Recall > Save Data To Touchstone File > Separator**

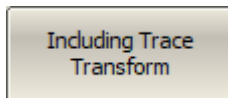
Then select the required Touchstone separator:

- **Tab**
- **Space**

---

[MMEM:STOR:SNP:SEP](#)

Sets and reads out the Touchstone file separator symbol.



If various transformations are applied to the active trace (for example, [time domain gating](#), [S-Parameter conversion](#), etc), use the following softkeys to enable/disable the inclusion of trace transform:

**Save/Recall > Save Data To Touchstone File > Including Trace Transform**

**NOTE:** If the function is not enabled, the data is written to the file without conversion.

---

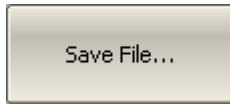
[MMEM:STOR:SNP:TRAC:TRAN](#)

Determines whether the S-parameters include the transformation of the active trace or not when saving the Touchstone file.



To save file to the hard disk, use the following softkeys:

**Save/Recall > Save Data To Touchstone File > Save File...**



Enter the file name in the dialog that appears.

---

[MMEM:STOR:SNP](#)

Saves the measured S-parameters of the active channel into a Touchstone file.

---

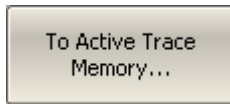
## Touchstone File Recalling

The Analyzer allows to recall data from the Touchstone files. Data can be loaded to memory traces or to data traces. When loading data to data traces, the Analyzer switches to hold mode to avoid writing over the recalled data with current data. When loading data to the memory traces, the sweep hold does not occur.

If the frequency scale of the Touchstone file does not correspond with the current analyzer frequency settings, the data is interpolated during recalling.

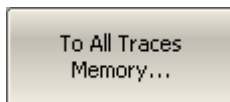


Load the data from the Touchstone file using one of the softkeys:

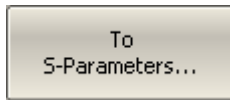


### Save/Recall > Load Data From Touchstone File

Then select the required data loading method:



- **To Active Trace Memory** — loading data to the active trace memory.



- **To All Traces Memory** — loading data to the memory of all traces.

- **To S-parameters** — loading data to all data traces of the channel.

---

### [MMEM:LOAD:SNP](#)

Loads the Touchstone file with the specified name to the measured S-parameters of the active channel.

---

### [MMEM:LOAD:SNP:TRAC:MEM](#)

Loads the Touchstone file with the specified name to the memory trace.

---

## System Settings

### Analyzer Presetting

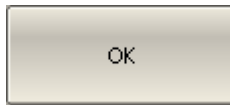
The Analyzer presetting feature allows to restore the default settings of the Analyzer.

The default settings of the analyzer are specified in [Annex 1](#).



To preset the Analyzer, use the following softkeys:

**System > Preset > OK**



---

[SYST:PRES](#)

Resets the Analyzer to factory settings.

---

## Graph Printing

This section describes the print/save procedures for graph data.

The print function is provided with the preview feature, which allows to view the image to be printed on the screen, and/or save it to a file.

The graphs can be printed using three different applications:

- MS Word (Windows only).
- Image Viewer for Windows (Windows only).
- Print Wizard of the Analyzer (Windows & Linux).

---

### NOTE

The MS Word application must be installed on the Windows system.

---

---

### NOTE

The Print Wizard requires at least one printer to be installed in Windows.

---

Print color can be selected before the image is transferred to the printing application:

- color (no changes)
- gray scale
- black & white.

The image can also be inverted before it is transferred to the printing application.

The current date and time can be added before the image is transferred to the printing application.

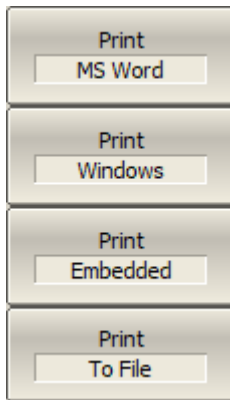


To open a print menu, use the following softkeys:

**System > Print**

---





Then select the printing application, using one of the following softkeys:

- **Print: MS Word**
- **Print: Windows**
- **Print: Embedded**
- **Print: To File**

---

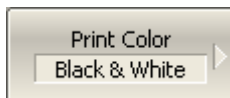
#### [HCOP](#)

Prints out the image displayed on the screen without previewing.

---

#### [MMEM:STOR:IMAG](#)

Saves the display image in BMP or PNG format into a file.



To set the print color, using the **Print Color** softkey.

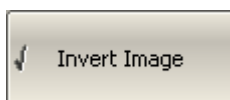
Then select the required color, using one of the following softkeys:

- **Color**
- **Gray Scale**
- **Black & White**

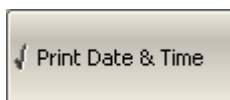
---

#### [HCOP:PAIN](#)

Sets or reads out the color chart for the image printout.



If necessary, invert the image by using the **Invert Image** softkey.



If necessary, select printing of date and time by using the **Print Date & Time** softkey.

---

#### [HCOP:DATE:STAM](#)

Turns the date and time printout in the upper right corner of the image ON/OFF.

---

#### [HCOP:IMAG](#)

Sets or reads out the color chart for the image printout.

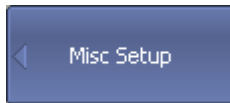
---



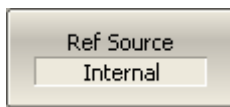
## Reference Frequency Oscillator Selection

The Analyzer can operate either with an internal or external reference frequency (10 MHz) oscillator. Initially, the Analyzer is set to operate using the internal source of the reference frequency. An external high stability oscillator can be used if more accuracy and frequency stability is required. Connect the external oscillator through the 10MHz Reference Input connector on the rear panel. Select the source of reference frequency oscillator in the software.

These two modes can be toggled in the softkey menu.



To select the reference frequency oscillator, use the following softkeys:



**System > Misc Setup > Ref Source [ Internal|External ]**

---

[SENS:ROSC:SOUR](#)

Sets or reads out an internal or external source of the 10 MHz reference frequency.

---

## System Correction Setting

The Analyzer is supplied by the manufacturer calibrated with calibration coefficients stored in its non-volatile memory. The factory calibration is used by default for the initial correction of the measured S-parameters. Such calibration is referred to as system calibration, and error correction is referred to as system correction.

The system correction ensures initial values of the measured S-parameters before the Analyzer is calibrated by the user. System calibration is performed on the plane of the physical port connectors and does not account for the cables and other fixtures used to connect the DUT. The measurement accuracy of the Analyzer with a user setup prior to calibration is not rated.

Normally, disabling the system correction setting is **not required** for calibration and further measurements.

System correction can be disregarded only when a proper calibration has been performed for the Analyzer. If user calibration has been performed, the measurement accuracy of the Analyzer is determined by user calibration and does not depend on the system correction status. If the user calibration has not been performed, then enabling/disabling system calibration would have an impact on measurements. If system correction is disabled, this is indicated in the [instrument status bar](#).

---

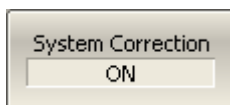
### NOTE

TRL calibration is not compatible with system correction. The system correction will be automatically turned off when TRL calibration is performed.

---



To disable/enable the system correction, use the following softkeys:



**System > Misc Setup > System Correction [ ON|OFF ]**

---

[SYST:CORR](#)

Turns the system correction ON/OFF.

---

## Power Trip Function

---

### NOTE

The availability of this feature depends on the Analyzer model (See corresponding [datasheet](#)).

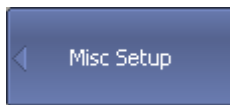
---

Port overload can occur when testing active devices. The power trip function is a safety feature to keep the analyzer's port from overloading. The function is triggered when the port safety power level is exceeded. When triggered, this function disables the stimulus signal and displays the following message in the [instrument status bar](#):

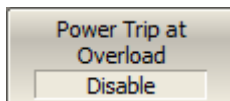
**Port <n> Power Trip at Overload!** (where <n> is the number of the port). The message has a red background.

After the overload trips, resolve the issue causing the overload, and then manually re-enable the stimulus via the submenu **Stimulus > Power > RF output [On]**.

The power trip function can be enabled or disabled by the user. By default, it is disabled. The ON/OFF state of this function is retained in subsequent sessions and does not depend on the **Preset** softkey.



To enable the power trip function, use the following softkeys:



**System > Misc Setup > Power Trip at Overload [ Enable| Disable ]**

---

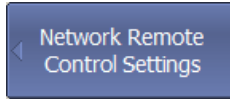
[SYST:REC:OVER:POW](#)

Turns the Power Trip at Overload function ON/OFF.

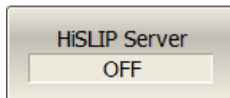
---

## Network Settings

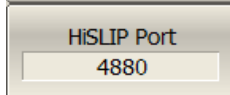
Network settings are used to enable remote control of the Analyzer.



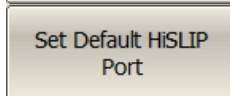
To enable/disable remote control of the Analyzer via a network using HiSLIP protocol on, use the following softkeys:



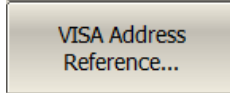
**System > Misc Setup > Network Remote Control Settings > HiSLIP Server > [ON | OFF]**



If necessary, specify the port number use the following softkeys:



**System > Misc Setup > Network Remote Control Settings > HiSLIP Port**



Set port number.

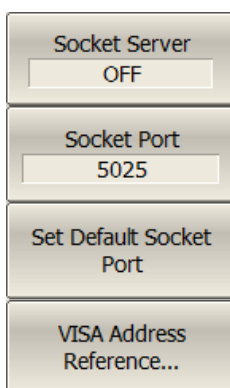
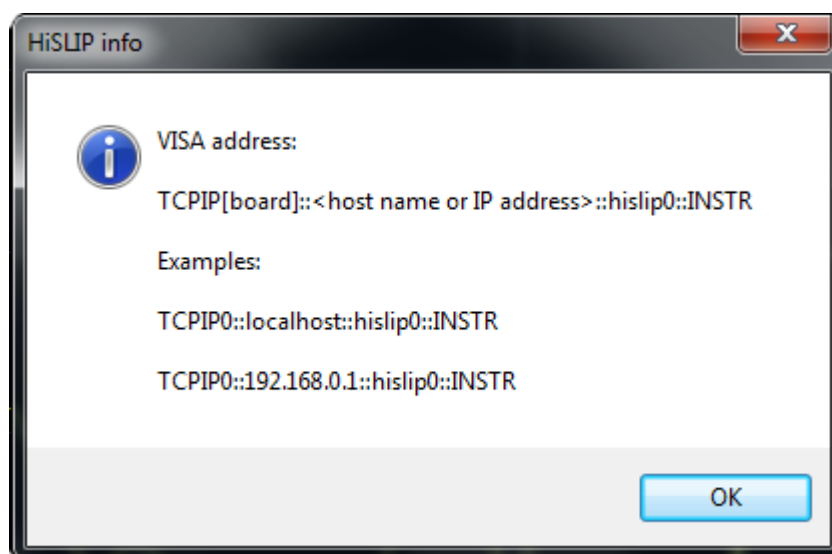
If necessary, the HiSLIP port number can be reset to the default value 4880, use the following softkeys:

**System > Misc Setup > Network Remote Control Settings > Set Default HiSLIP Port**

To query the analyzer for its VISA addresses use the following softkeys:

**System > Misc Setup > Network Remote Control Settings > VISA Address Reference...**

Such a window will be displayed in response:



To enable/disable remote control of the Analyzer via a network using TCP/IP Socket protocol on, use the following softkeys:

**System > Misc Setup > Network Remote Control Settings > Socket Server > [ON | OFF]**

If necessary, specify the port number use the following softkeys:

**System > Misc Setup > Network Remote Control Settings > Socket Port**

Set port number.

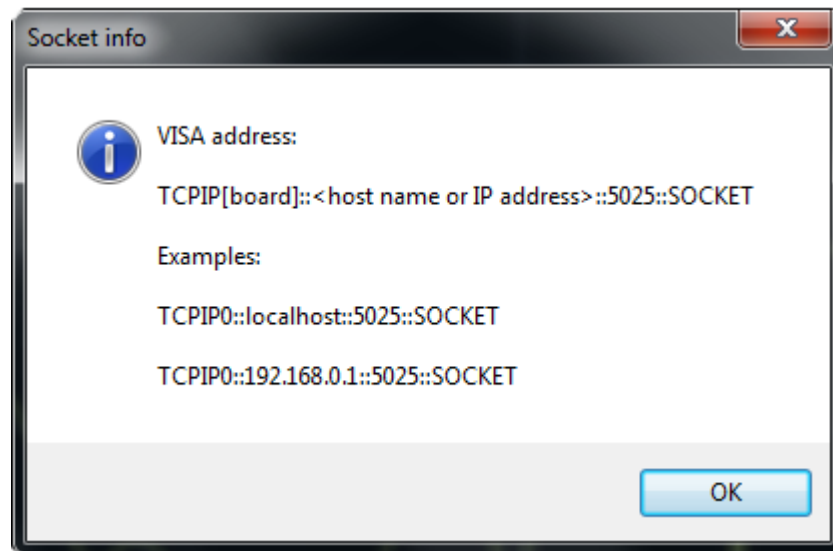
If necessary, the HiSLIP port number can be reset to the default value 4880, use the following softkeys:

**System > Misc Setup > Network Remote Control Settings > Set Default HiSLIP Port**

To query the analyzer for its VISA addresses use the following softkeys:

**System > Misc Setup > Network Remote Control Settings > VISA Address Reference...**

Such a window will be displayed in response:



**NOTE**

Remote control of the Analyzer is not possible using two interfaces simultaneously. A Socket or network must be selected.

When specifying the port number, make sure that it is not busy performing another process.

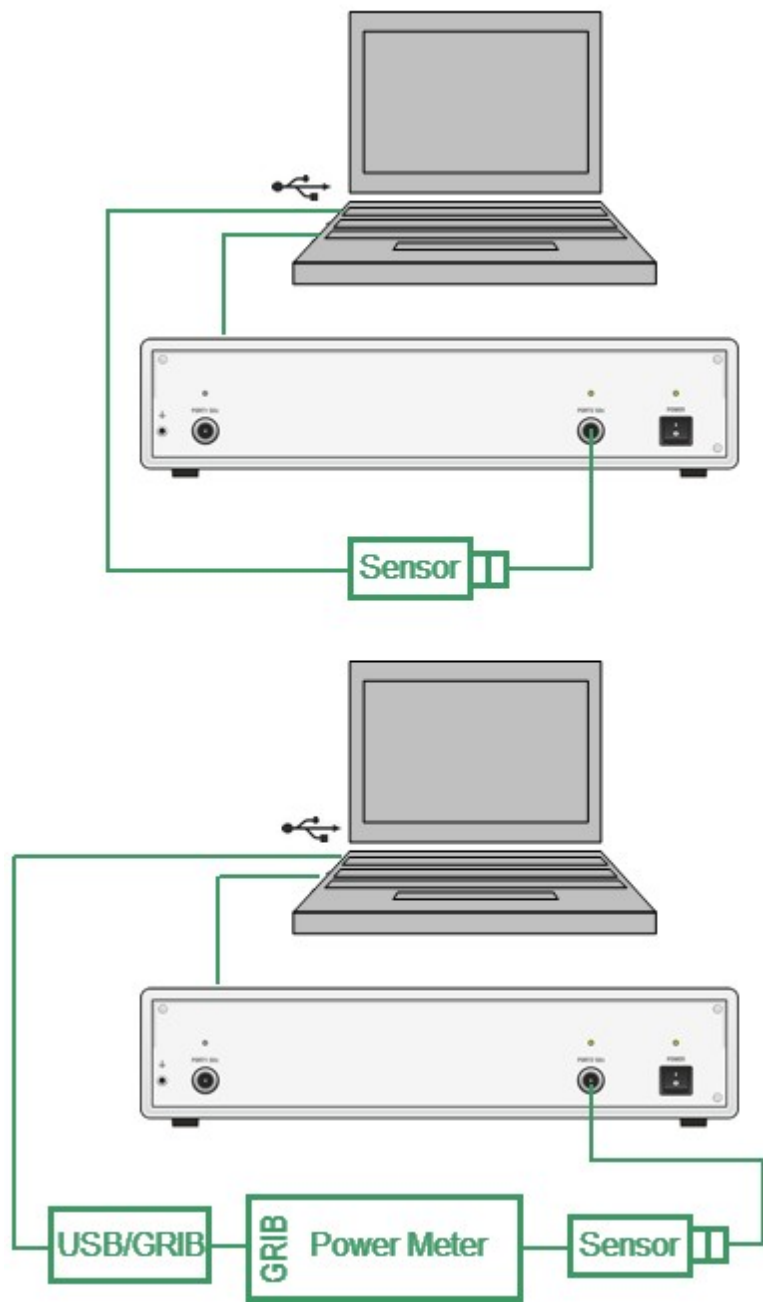
For more information about remote control of the Analyzer, see in [Programming](#).

---



# Power Meter Settings

An external power meter can be connected to the Analyzer to perform a power calibration of the test ports. Connect the power meter to the PC directly via USB port or via USB/GPIB adapter. Then, install the power meter software. The list of power meters supported by the Analyzer is shown in the table below.



Power meter setup example

## Supported power meters

<b>Power Sensor</b>	<b>Designation in the VNA Software</b>	<b>Connection Type</b>	<b>Additional Software</b>
R&S®NRP-Z Power Sensors	R&S NRP-Z USB Power Sensors	USB	<ul style="list-style-type: none"> <li>• R&amp;S®NRP-Toolkit for Windows</li> <li>• VXiplug&amp;play x64 or x86 driver rsnrpz</li> </ul>
R&S®NRV S Power Meter plus R&S®NRP-Z Power Sensors	R&S NRVS GRIB Power Meter	GPIO or USB via GPIO/USB Adapter	<ul style="list-style-type: none"> <li>• VISA Library from any vendor (visa32.dll)</li> <li>• GPIO/USB Adapter driver (if needed)</li> </ul>
R&S®NRPxxT Thermal Power Sensor	R&S NRPxT USB Power Sensor	USB	<ul style="list-style-type: none"> <li>• R&amp;S®NRP-Toolkit for Windows</li> <li>• VISA Library from any vendor (visa32.dll)</li> </ul>
NI USB-568x RF Power Sensors	NI USB-568x Power Sensors	USB	NI USB-568x driver (ni568x.dll)
LadyBug USB Power Sensors (LB478A, LB479A, LB480A, LB559A, LB579A, LB589A)	LadyBug LBxxxx USB Power Sensors	USB	Not needed (included in the VNA software installer)
LadyBug LB59XX USB Power Sensors	LadyBug LB59xx USB Power Sensors	USB	VISA Library from any vendor (visa32.dll)

Power Sensor	Designation in the VNA Software	Connection Type	Additional Software
Keysight U848x Power Sensors	Keysight U848x USB Power Sensors	USB	VISA Library from any vendor (visa32.dll)
Keysight U200x Power Sensors	Keysight U200x USB Power Sensors	USB	VISA Library from any vendor (visa32.dll)



To select the power meter, use the following softkeys:

**System > Misc Setup > Power Meter Setup > Power Meter**

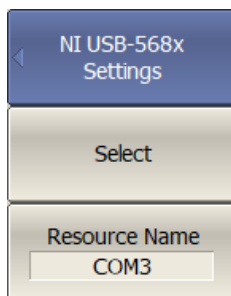
Then select the required power meter:

- **R&S NRPxT USB Sensor**
- **R&S NRPxZ USB Sensor**
- **Keysight U848x USB Sensor**
- **Keysight U200x USB Sensor**
- **NI USB-568x Power Sensor**
- **LadyBug LB59xx USB Sensor**
- **R&S NRVs GRIB Power Meter**

[SYST:COMM:PSEN:TYPE](#)

Selects the power sensor type to be used in a source power calibration.

---



NI USB-568x  
Settings

Select

Resource Name  
COM3

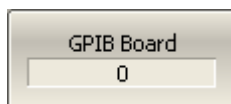
If an **NI USB-568x Power Sensor** is selected, set its resource name to VISA using **Resource Name** and confirm the selection with the **Select** button. The **Resource name** for this power sensor must be carried over from NI Measurement & Automation Explorer (MAX).

---



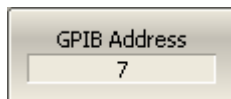
Power Meter  
Setup

If an R&S NRVS GPIB power meter is selected, set the GPIB board address and the power meter address in the bus, using the following softkeys:



GPIB Board  
0

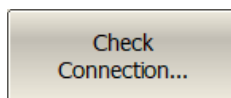
**System > Misc Setup > Power Meter Setup > GPIB Board**



GPIB Address  
7

**System > Misc Setup > Power Meter Setup > GPIB Address**

---



Check  
Connection...

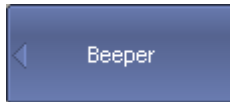
The **Check Connection...** softkey checks the connection of the power meter. It provides sensor type, if the communication between the Analyzer and the power meter has been successfully established.

---

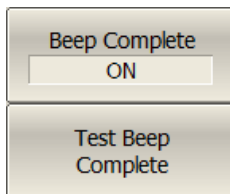
## Beeper Settings

The Analyzer features two available beeper settings, which can be toggled on/off independently from each other:

- Operation complete beeper — informs about normal completion of standard measurements during calibration.
- Warning beeper — informs about an error or a fail limit test result.



To toggle the operation complete beeper, use the following softkeys:



**System > Misc Setup > Beeper > Beep Complete**

To test the operation complete beeper, use the following softkeys:

**System > Misc Setup > Beeper > Test Beep Complete**

---

[SYST:BEEP:COMP:STAT](#)

Turns the beeper denoting completion of the operation ON/OFF.

---

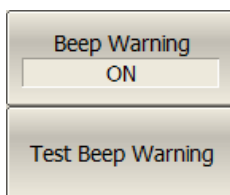
[SYST:BEEP:COMP:IMM](#)

Generates a beep to notify of the completion of the operation.



To toggle the warning beeper, use the following softkeys:

**System > Misc Setup > Beeper > Beep Warning**



To test the warning beeper, use the following softkeys:

**System > Misc Setup > Beeper > Test Beep Warning**

---

[SYST:BEEP:WARN:STAT](#)

Turns the beeper signifying a warning ON/OFF.

---

[SYST:BEEP:WARN:IMM](#)

Generates a beep to signify a warning.

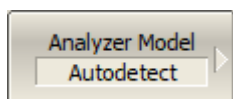
---

## Analyzer Model

The analyzer model is detected automatically when connected. If necessary, the analyzer model can be set manually.



To manually enter the analyzer model, use the following softkeys:



**System > Misc Setup > Analyzer Model**

Select an analyzer model from the list.

---

### NOTE

The message **Not Ready** appears in the state status bar if the manually entered analyzer model does not match the connected device model. Further operation of the Analyzer is not possible.

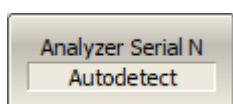
---

## Analyzer Serial Number

The serial number of the Analyzer is read automatically when connected. If necessary, the serial number can be set manually.



To manually enter the analyzer serial number, use the following softkeys:



**System > Misc Setup > Analyzer Serial N**

Enter the 10-digit serial number of the Analyzer.

---

### WARNING

The message **Not Ready** appears in the state status bar if the manually entered analyzer model does not match the connected device model. Further operation of the Analyzer is not possible.

---

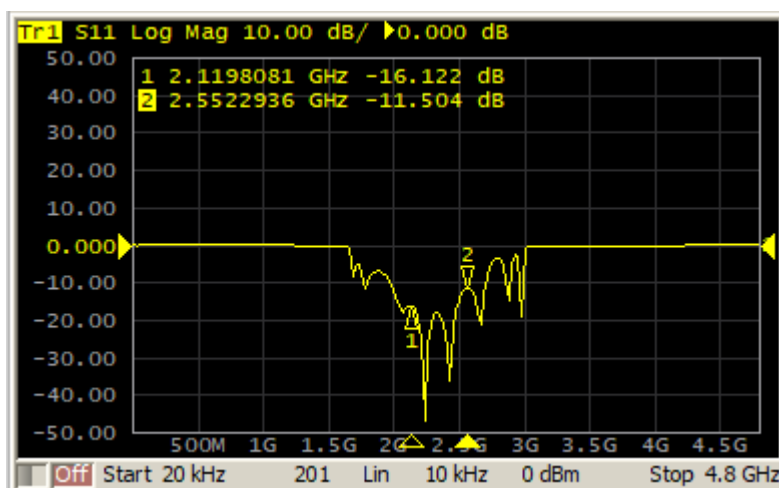
## Security Level

The analyzer software provides three levels of security:

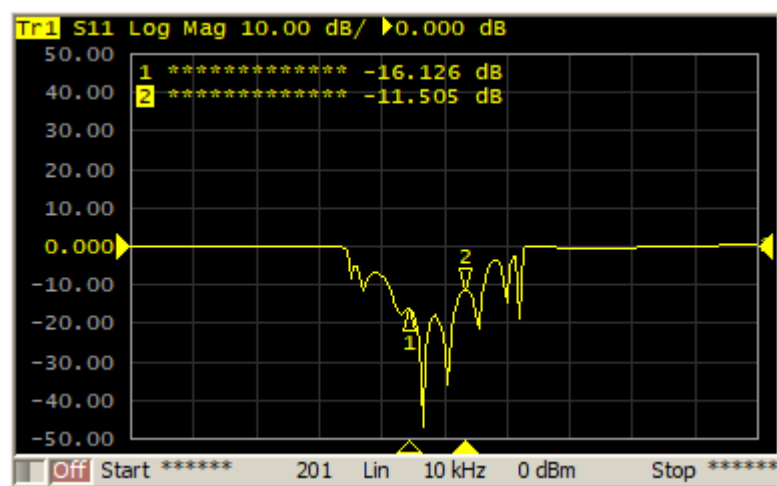
- None
- Low
- High

The frequency indication is disabled when the security level is **Low** or **High**.

When the security level is **High**, it is only possible to turn on the frequency indication after a complete reset of the Analyzer or by loading the analyzer state file, which also leads to a complete reset of the previous Analyzer state.



Security Level OFF



Security Level ON

Security Level  
None

☒ None

☐ Low

☐ High

Select the level of security in the menu:

**System > Misc Setup > Security Level > [None | Low | High]**



## Language

The default language for software is english. The software can be localized for any language.

Contact technical support at the [coppermountaintech.com](http://coppermountaintech.com) website if the software does not have the desired language available.



To select the interface language, use the following softkeys:

**System > Misc Setup > Language**

Then choose the language.

---

## Create Localize Language File

To localize, do the following:

- Find the lang\_template.txt file in the VNA application home directory in the \System\Lang folder;
- Rename this file to the lang\_xx.txt, where xx is the two-letter language code, for example, lang\_ch.txt.
- Open lang\_xx.txt file.
- Find the "Name" field in the file. Enter the name of the language into which all text will be translated in field to the right of the equal sign. For example, 中文 (chinese).
- In the same manner, enter the translation of the other fields. For example:

Frequency = 頻率

Rules for inputting translation text:

- To have space before or after a term, add the desired space around the term using quotes. For example:

Frequency = " 頻率 "

- To enter empty text, use two quotation marks. For example:

Frequency = ""

- To leave a word unchanged, leave a blank. For example:

Frequency =

---

### **WARNING**

Do not change the field to the left of the equal sign. This can cause the software to malfunction.

---

The language code will appear on the softkey in the Language menu after renaming the file and restarting the application. To apply localization, press the softkey with the language code. The application will restart, and the inscriptions on the interface elements will change to the localized inscriptions.



To select the interface language, use the following softkeys:

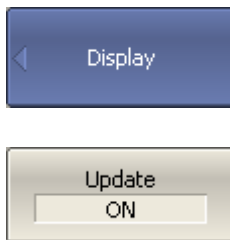
**System > Misc Setup > Language**

---

## Screen Update Setting

Screen updating can be disabled to reduce the sweep time. This function can be useful when remotely controlling the Analyzer via COM/DCOM interfaces.

A single trace update is possible when screen update is disabled. Click on the trace with the mouse, or hover the mouse over the graticule labels.



To disable the screen updating, use the following softkeys:

**Display > Update [ ON|OFF]**

---

[DISP:ENAB](#)

Turns the display update ON/OFF.

---

---

### NOTE

If screen updating is off, the message **Update Off** appears in the instrument status bar.

---

## User Interface Setting

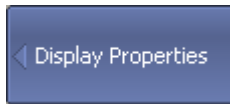
The software allows to adjust the following user interface settings:

- Toggle between full screen and window display (See [Full Screen](#)).
- The font size of all displayed items (See [Font Size](#)).
- Style and width of data traces, memory traces, graph grid (See [Trace and grid styles](#)).
- Set color of data and memory traces, markers, background, grid (See [Color](#)).
- Invert color of diagram (See [Invert Color of Diagram](#)).
- Hide/show menu bar (See [Hide/Show Menu Bar](#)).
- Change vertical or horizontal graticule ([Hide/Show Horizontal Graticule](#) or [Set Vertical Graticule Label](#)).
- Hide/show sweep mark (See [Hide/Show Sweep Mark](#)).
- Hide/show date and time on status bar (See [Hide/Show Date and Time](#)).
- Hide/show cycle time (See [Hide/Show Cycle Time](#)).
- Interface presetting (See [Interface Presetting](#)).
- Save/load display setting (See [Save/Load Display Setting](#)).

The user interface settings are automatically saved and will be restored the next time the analyzer is turned on. No particular saving procedure is required. If necessary, user interface settings can be reset to default factory settings (See [Interface Presetting](#)).

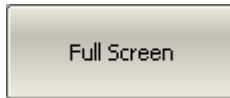
## Full Screen

The control program on the PC screen is displayed as a window. If necessary, use full screen mode.



To toggle between full screen and window display, use the following softkeys:

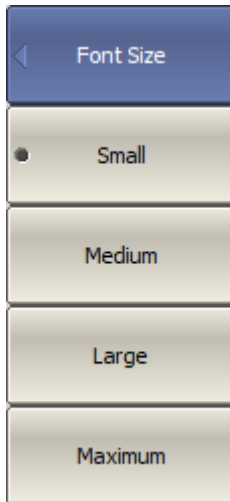
**Display > Properties > Full Screen**



## Font Size

The default font size setting for all items is 14.

The font size of all displayed items can be changed. The font size can be changed to any size between 10 to 22.

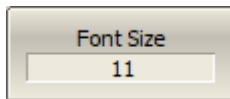


To change the font size, use the following softkeys:

**Display > Properties > Font**

Choose from four standard font sizes:

- **Small**
- **Medium**
- **Large**
- **Maximum**

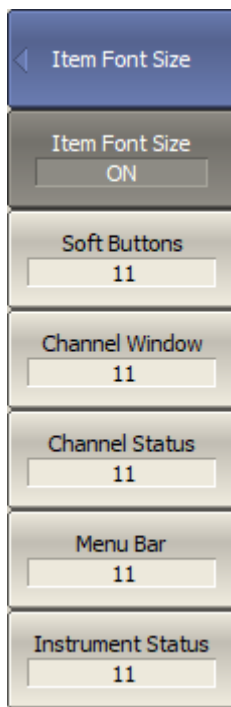


The **Font Size** softkey allows to select a custom font size from 10 to 22 for all displayed items.

---

[DISP:FONT:SIZE](#)

Sets/gets one font size for all displayed elements of the application.



To change the font size by categories of displayed items, use the following softkeys:

**Display > Properties > Font > Item Font Size**

To turn on the font size selection by category, use the following softkeys:

**Display > Properties > Font > Item Font Size > Item Font Size [ ON | OFF ]**

Select displayed items to customize:

- **Soft Buttons**
- **Channel Window**
- **Channel Status**
- **Menu Bar**
- **Instrument status**

Then select the font size from 10 to 22.

---

[DISP:PART:FONT:SIZE:STAT](#)

Specifies whether different elements of the application window have individual font sizes or the same font size.

---

[DISP:PART:FONT:SIZE](#)

Sets/gets the font size of the item specified by the <char> parameter.

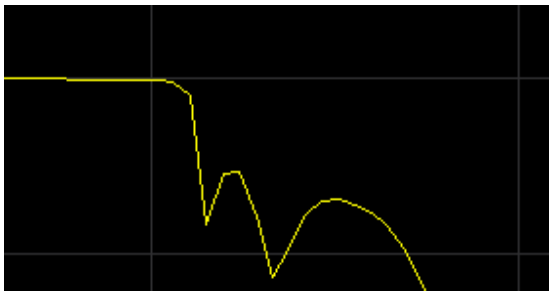
---

# Trace and Grid Styles

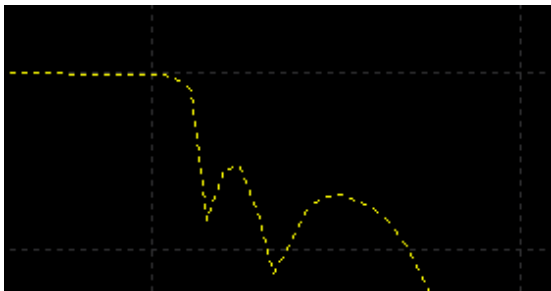
The style and width of data and memory traces and graph grid can be changed.

The width of a data and memory traces ranges from 1 to 3 pixels.

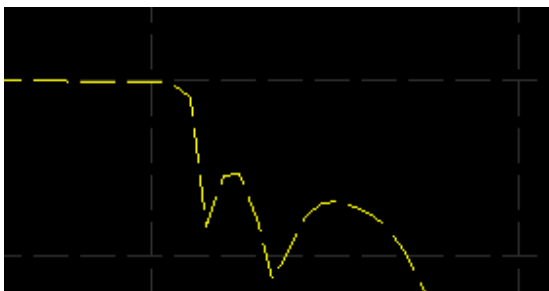
The line style of the trace and grid can also be customized: choose between solid, dash, dot, and dash-dot (See the figure below).



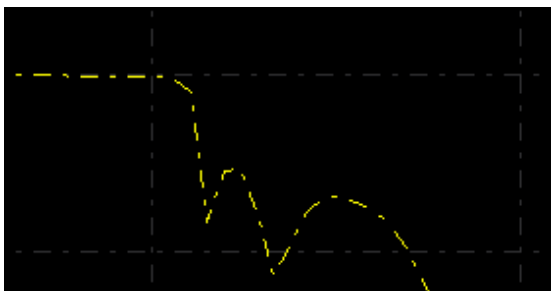
Solid



Dash

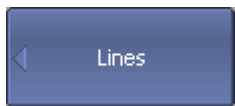


Dot

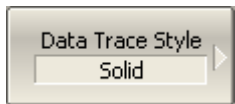


Dash-dot

Trace and grid styles

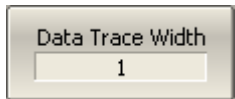


To change the style and width of a data trace, use the following softkeys:



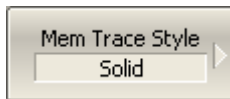
**Display > Properties > Lines > Data Trace Style > [Solid | Dash | Dot | Dash-dot]**

Width ranges from 1 to 3 pixels.

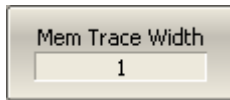


**Display > Properties > Lines > Data Trace Width**





To change the style and width of a memory trace, use the following softkeys:



**Display > Properties > Lines > Mem Trace Style > [Solid | Dash | Dot | Dash-dot]**

**Display > Properties > Lines > Mem Trace Width**



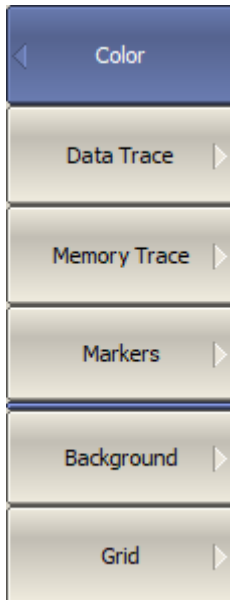
To change the grid style, the following softkeys:

**Display > Properties > Lines > Grid Style > [Solid | Dash | Dot | Dash-dot]**

---

## Color

The color of data and memory traces, markers, the background, and the grid can be changed if necessary.

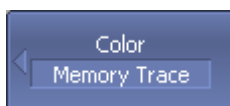


To change the color by categories of displayed items, use the following softkeys:

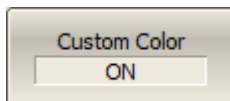
### **Display > Properties > Colors**

Select displayed items to customize:

- **Data Trace**
- **Memory Trace**
- **Markers**
- **Background**
- **Grid**



The color setting for the different displayed items is identical. For example, consider changing the color of an active memory trace.



To change the color of the active memory trace, use the following softkeys:

### **Display > Properties > Colors > Memory Trace**

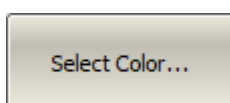
To turn on Custom Color, use the following softkeys:



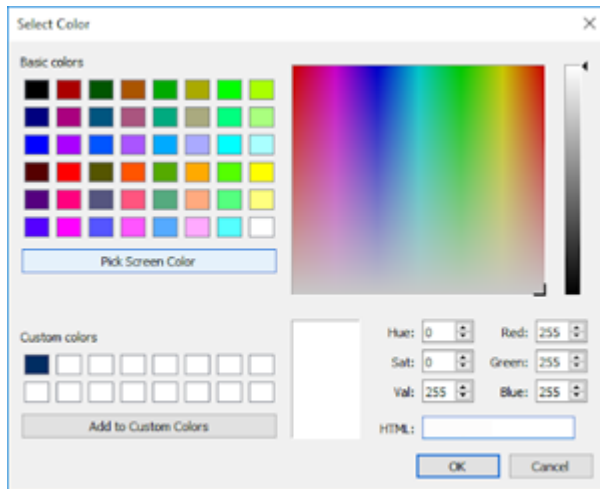
### **Display > Properties > Colors > Memory Trace > Custom Color [ON | OFF]**

Then select the rate (from 0 to 255) of color components, use the following softkeys: **Red**, **Green**, **Blue**.

The changes made to the color of the active memory trace will affect all the traces with the same number in other channels.



Alternatively, press the **Select Color...** softkey and go to the Microsoft Windows color palette. Select color and click **OK**.



---

[DISP:COL:TRAC:DATA](#)

Sets or reads out the data trace color.

---

[DISP:COL:TRAC:MEM](#)

Sets or reads out the data trace color.

---

[DISP:COL:BACK](#)

Sets or reads out the background color for trace display.

---

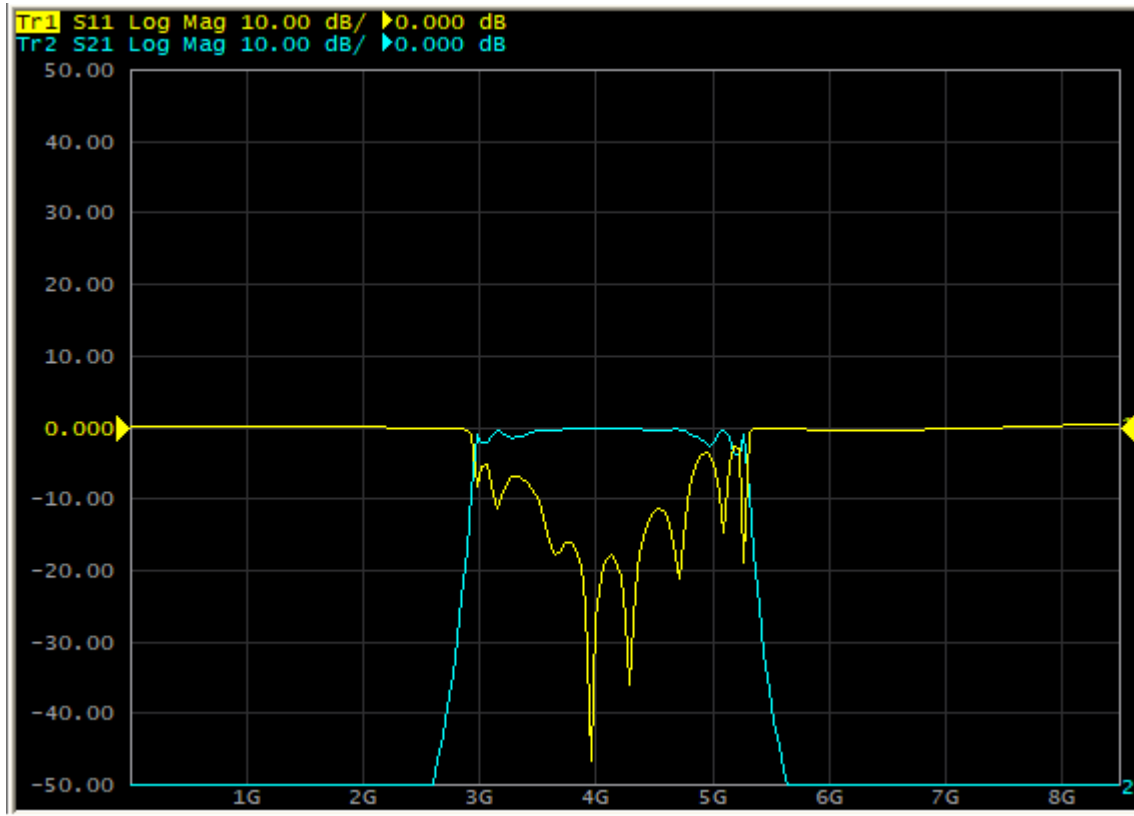
[DISP:COL:GRAT](#)

Sets or reads out the grid and the graticule label color for trace display.

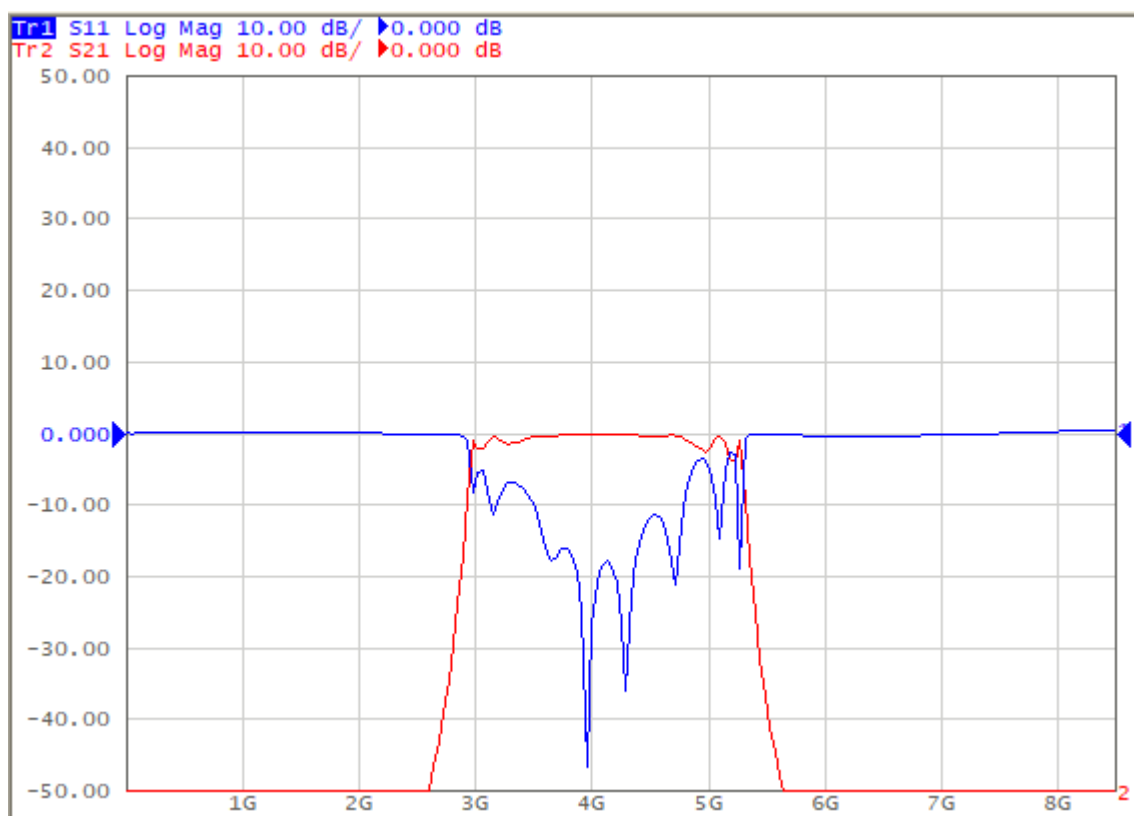
---

## Invert Color of Diagram

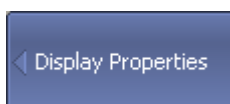
By default, the diagram is in dark color mode. The color mode can be switched to light mode.



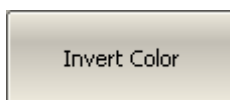
Dark Mode of Diagram (by default)



Light Color Mode of Diagram



To change the color mode of diagram, use the following softkeys:



**Display > Properties > Invert Color**

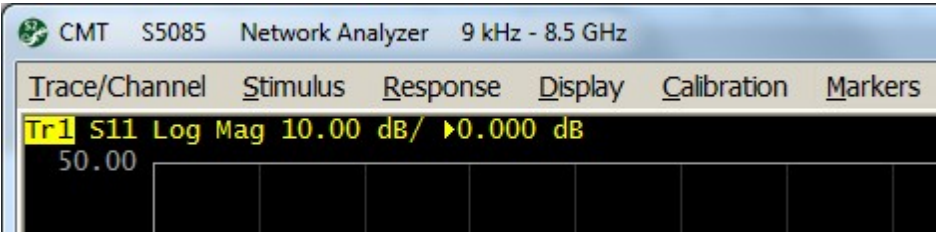
[DISP:IMAG](#)

Turns the inversion of display colors of the trace area ON/OFF.

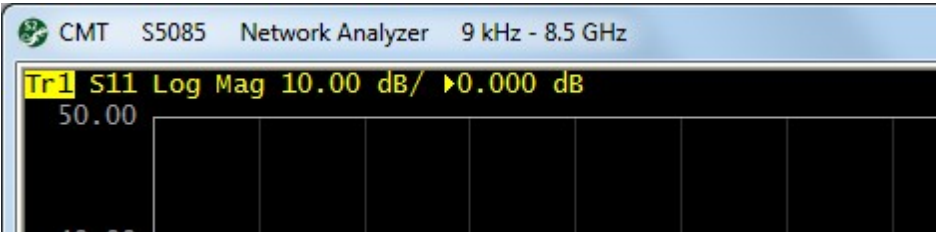
# Hide/Show Menu Bar

By default, the [menu bar](#) is located at the top of the screen (See figure below).

The menu bar can be optionally hidden to gain more screen space for the channel window, and is controlled by mouse.



Menu Bar ON



Menu Bar OFF



To hide/show the menu bar, use the following softkeys:

**Display > Properties > Menu Bar [ON | OFF]**

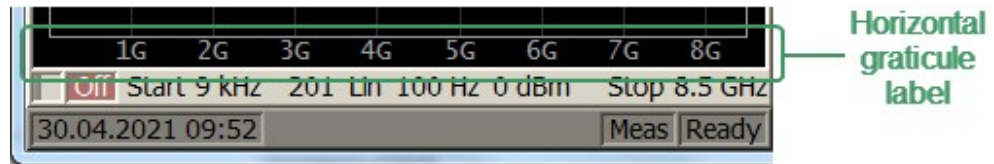


[DISP:PART:VIS](#)

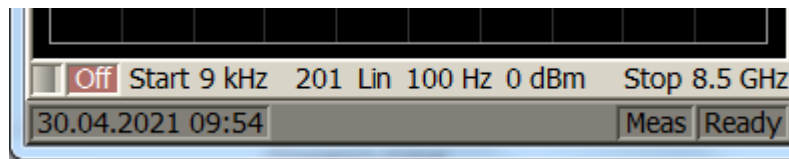
Shows or hides the display partition specified by the <char> parameter.

## Hide/Show Horizontal Graticule Label

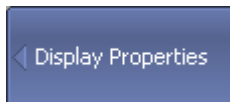
Horizontal graticule label is located at the down of the screen (See figure below). The horizontal graticule label can be hidden to gain more screen space for the trace display.



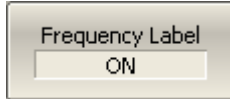
Horizontal graticule label ON



Horizontal graticule label OFF



To hide/show horizontal graticule label, use the following softkeys:



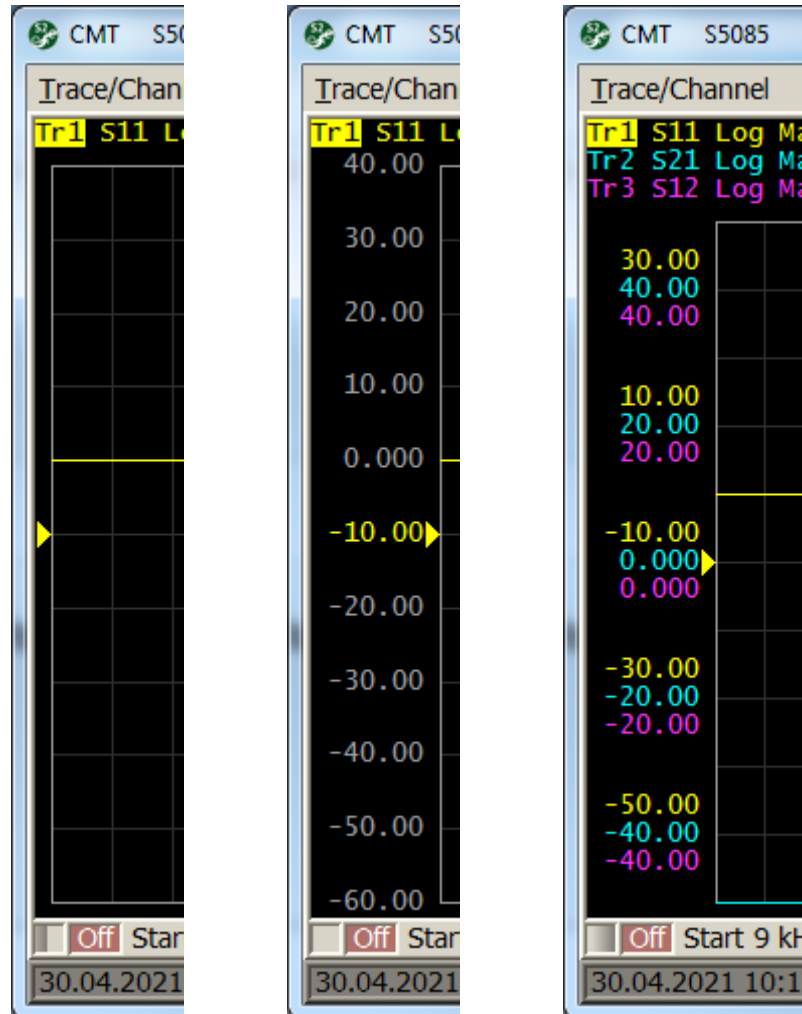
**Display > Properties > Frequency Label [ ON|OFF ]**

[DISP:PART:VIS](#)

Shows or hides the display partition specified by the <char> parameter.

## Set Vertical Graticule Label

Vertical graticule label is located at the left of the screen (See figure below). By default the scale of the active chart trace is displayed. If necessary, the display of the scales of all traces can be enabled or disabled to gain more screen space for the trace display (See figure below).

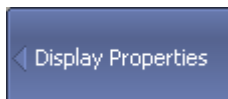


OFF

Active Trace

All Trace

Vertical Graticule Label



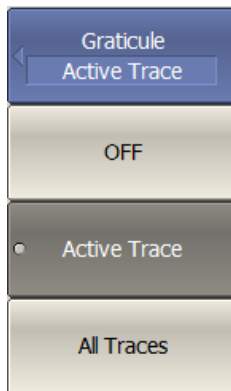
To open vertical graticule label submenu, use the following softkeys:

**Display > Properties > Graticule Label [ ON|OFF ]**

Then select display type of vertical graticule label:

- **OFF**





- **Active Trace**

- **All Traces**

---

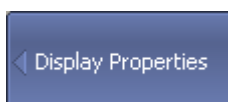
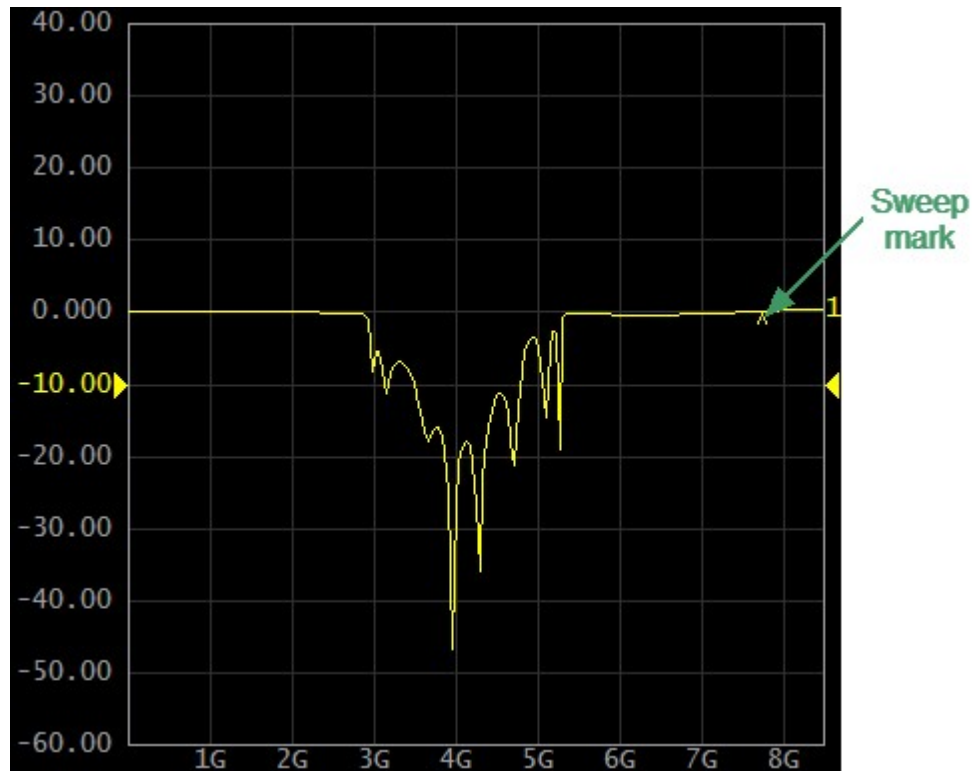
[DISP:GLAB](#)

Sets/gets the Graticule Label state.

---

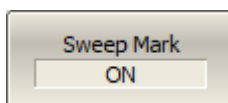
## Hide/Show Sweep Mark

The sweep mark is visible during measurement if the measurement cycle time is long, due to a narrow IF filter bandwidth or a large number of measurement points (See figure below). If necessary, the sweep mark can be hidden.



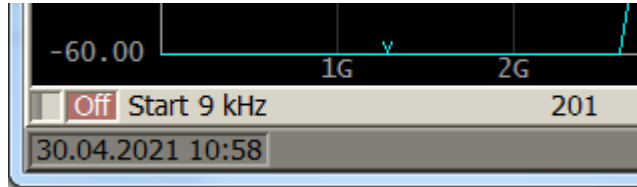
To hide/show sweep mark, use the following softkeys:

**Display > Properties > Sweep Mark [ON | OFF]**

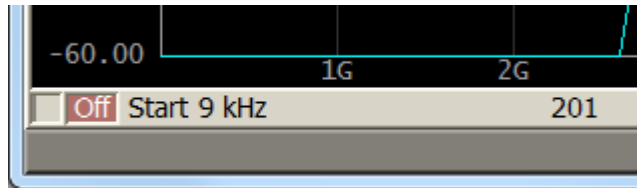


## Hide/Show Date and Time

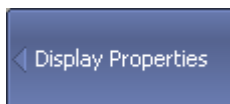
System date and time display is located on the analyzer status bar. If necessary, date and/or time can be turned off.



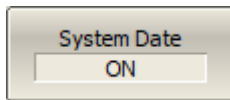
System date and time show in the analyzer status bar



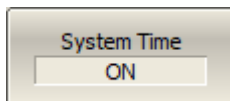
System date and time hide in the analyzer status bar



To enable / disable the current date in the analyzer status bar, use the following softkeys:



**Display > Properties > System Date [ ON|OFF ]**

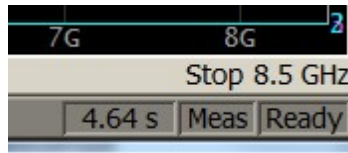


To enable / disable the current time in the analyzer status bar, use the following softkeys:

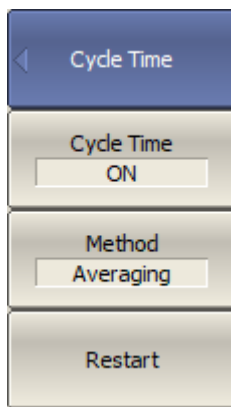
**Display > Properties > System Time [ ON|OFF ]**

## Hide/Show Cycle Time

By default, the sweep time isn't displayed in the Analyzer status bar (See [Sweep time](#)). It can be enabled if necessary. Depending on the selected method, cycle time can be defined as an average value or as a maximum hold.



Cycle Time ON



To enable/disable the display of the scan cycle time in the analyzer status bar, use the following softkeys:

**Display > Properties > Cycle Time > Cycle Time [ ON|OFF ]**

Use the following softkeys to select the method for determining the scan cycle time:

**Display > Properties > Cycle Time > Method [ Averaging | Max Hold ]**

The relation between the cycle time function and the sweep time function are shown in this [table](#).

The **Restart** softkey is used to restart the Cycle Time definition and reset the previous values.

---

[SYST:CYCL:TIME:MEAS?](#)

Reads out the measured cycle time.

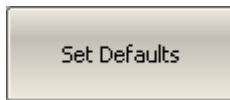
---

## Interface Presetting

All set user interface settings can be reset.



To restore the interface settings to the default factory settings, use the following softkeys:



**Display > Properties > Set Defaults**

---

[DISP:COL:RES](#)

Restores the display settings to the default values.

---

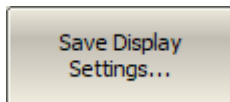
## Save/Load Display Setting

The user interface settings can be recorded in a \*.CFG file.

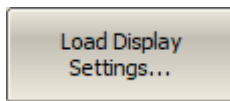


To save the display settings in file, use the following softkeys:

**Display > Properties > Save Default Display Settings**



Enter the file name in the dialog that appears.



To load the display settings from file, use the following softkeys:

**Display > Properties > Load Display Setting**

Select the name of the display settings file in the window that opens.

---

## Demo Mode

Demo mode is designed to simulate DUT measurement. The measurement results of the DUT are pre-recorded in the program memory. Any analyzer model can be selected from the list of supported devices in demo mode (See [Analyzer Model](#)).

---

### NOTE

The simulation of the Analyzer in demo mode may differ from the real measurements of the analyzer. For example, the accuracy of the sweep time dependence on the IF filter setting is not guaranteed.

---

---

### WARNING

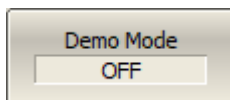
The software restarts automatically when the demo mode state changes.

---



To enable/disable Demo Mode, use the following softkeys:

**System > Misc Setup > Demo Mode [ ON|OFF ]**



# Plugins

A plugin is an executable file that performs the user defined function using COM automation or SCPI commands of the VNA application. Create own plugin or download the plugin from the [coppermountaintech.com](http://coppermountaintech.com) website. Place the plugin in the VNA application home directory in the "plugins" folder.



The **Plugin** softkey will become active after placing the plugin in the specified folder.

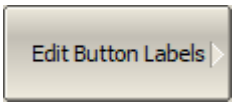
To launch the plugin, enter the "plugins" menu using the following softkeys:

## System > Plugin



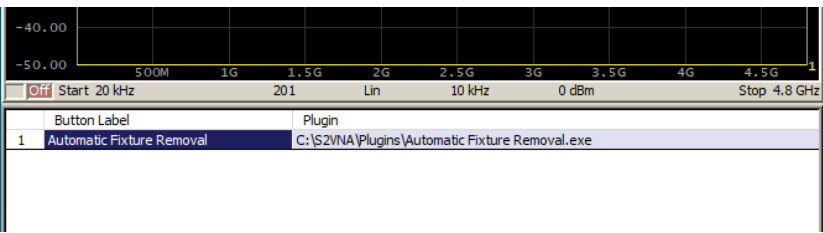
The name of the executable file will appear on the button in the "plugins" menu.

To launch the plugin, click the softkey with its name.



The label on the plugin button is edited in the plugins table. To open the plugin table, use the following softkeys:

## System > Plugin > Edit Button Labels



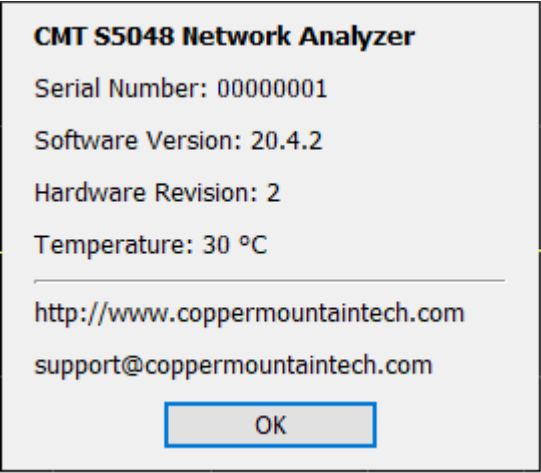
Button Label		Plugin
1	Automatic Fixture Removal	C:\S2VNA\Plugins\Automatic Fixture Removal.exe

Plugin table

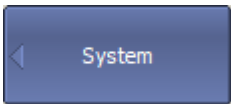


# About

The analyzer model name, serial number, software and hardware versions, the temperature can be found in the System menu.

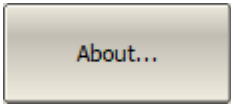


About



To request information, use the following softkeys:

**System > About...**



## Programming

This section describes programming of the Analyzer using SCPI commands or COM/DCOM technology. SCPI commands are primarily described in this manual. A corresponding COM command description is provided at the end of each SCPI command description.

**SCPI** (Standard Commands for Programmable Instruments) defines a standard for syntax and commands to use in controlling programmable instruments. SCPI commands are ASCII textual strings that are sent to the analyzer program over the LAN physical layer using the HiSLIP or TCP/IP Socket network protocol. These protocols can also be used within a single PC when using the IP address 127.0.0.1 or *localhost*.

**HiSLIP** (High-Speed LAN Instrument Protocol) is a network protocol intended for remote control of measuring and testing equipment, and is based on the TCP/IP network protocol. HiSLIP is developed by the consortium IVI Foundation as the successor to GPIB and VXI-11 protocols. The user program relies on the implementation of the HiSLIP protocol in the VISA library.

**TCP/IP Socket** is a general-purpose network protocol. The user program can connect to the Analyzer using the TCP/IP Socket protocol both directly and through the VISA library.

**VISA** (Virtual Instrument Software Architecture) library is a widely used software input-output interface for measuring and testing equipment. It is a library of functions for C/C++, C#, Visual Basic, MATLAB, LabVIEW and others. The VISA library unifies access to all measuring instruments, regardless of the protocol and physical layer used. The VISA library is available on the websites of many companies for free download. There are versions of VISA library for Windows, Linux, Mac OS.

**COM/DCOM** (Component Object Model/Distributed Component Object Model) is a program technology developed by Microsoft. The COM/DCOM technology establishes a program interface between the analyzer program and the user program. The analyzer program acts as a COM server. The user program acts as a COM client. COM is used within a single PC. DCOM is used over a LAN.

## References

Standard Commands for Programmable Instruments (SCPI)  
<http://www.ivifoundation.org/specifications>

High-Speed LAN Instrument Protocol (HiSLIP),  
<http://www.ivifoundation.org/specifications>

VISA specifications, <http://www.ivifoundation.org/specifications>

## Connection Setup

To enable remote control of the Analyzer, turn on the HiSLIP server and/or Socket server in the settings of the analyzer's program. The default TCP/IP port number of each protocol can be changed optionally.

HiSLIP is a TCP/IP-based protocol specially designed for measuring and test equipment. TCP/IP Socket is a general-purpose protocol.

Typically, the user program (client) uses VISA library to establish the connection. When using the VISA library, the client selects the protocol by specifying it in the VISA address of the Analyzer.

The VISA library hides the details of protocol implementation from the client and provides a uniform I/O interface. Nevertheless, there are some minor differences in programming methods when using the HiSLIP and TCP/IP Socket protocols, which are described later in [Differences in Use of HiSLIP and Socket Protocols](#).

After a connection has been established by the client, the latter can send SCPI commands and read the results of the measurements. The command set is the same for both protocols and is described in [Command Reference](#).

The client must specify the Analyzer's PC IP address or network name in the VISA address string. The analyzer and user programs can be run on the same PC. In this case, the client specifies the IP address as 127.0.0.1 or *localhost*.

Multiple analyzer programs can be executed on the same PC (when several USB blocks are connected). In this case, the user must specify a unique TCP/IP port number in the settings of each analyzer program.

One analyzer program does not limit the number of simultaneously connected clients. Clients themselves are responsible for the absence of conflicts in the remote control of the Analyzer. The HiSLIP protocol supports the exclusive or shared lock of the Analyzer by the client. For more details about locks, see the [VISA manual](#).

## Analyzer Setting

For remote access to the Analyzer, make the following settings in its program:

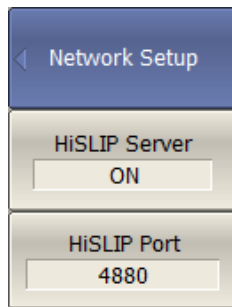
- Enable HiSLIP server and/or Socket server.
- Configure the TCP/IP port number (optional).

---

### NOTE

Configuring the TCP/IP port number is necessary only where several analyzer programs are simultaneously executed on the same PC, and these programs require remote remote control. In other cases, leave the default TCP/IP port number: for the HiSLIP server — 4880, and for the Socket server — 5025.

---



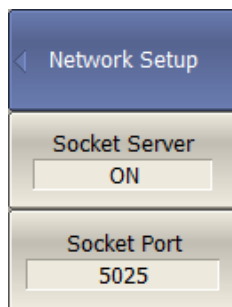
To enable remote control of the Analyzer using the HiSLIP protocol, press the following softkeys:

**System > Misc Setup > Network Setup > HiSLIP Server {ON/OFF}.**

To change the TCP/IP port number of the HiSLIP server, use the following softkeys:

**System > Misc Setup > Network Setup > HiSLIP Port.**

---



To enable remote control of the Analyzer using the Socket protocol, press the following softkeys:

**System > Misc Setup > Network Setup > Socket Server {ON/OFF}.**

To change the TCP/IP port number of the Socket server, use the following softkeys:

**System > Misc Setup > Network Setup > Socket Port.**

---

## Client Setting

Typically, the client uses the VISA library to establish connection to analyzer software. The easiest way to configure the network connection with the Analyzer is using a special utility from VISA package (for example, NI-MAX, Keysight Connection Expert).

Following the manual for the above utilities, add a new network device — specifying the network name or IP address of the Analyzer's PC — and the protocol. Once successfully connected to the Analyzer, the VISA address of the Analyzer will be automatically generated and displayed. Use this VISA address in the client program in order to open the connection.

### The format of the VISA address for the HiSLIP and Socket protocols

HiSLIP	TCPIP[board]:: <i>host address</i> [:, <i>HiSLIP device name</i> [, <i>HiSLIP port</i> ]] [:,INSTR]
Socket	TCPIP[board]:: <i>host address</i> :: <i>port</i> ::SOCKET

### Examples of VISA address for HiSLIP and Socket protocols

HiSLIP	TCPIP0::192.168.0.1::hislip0::INSTR  TCPIP0::localhost::hislip0::INSTR
Socket	TCPIP0::192.168.0.1::5025::SOCKET  TCPIP0::localhost::5025::SOCKET

If the client is a user program that does not use the VISA library, then the only available protocol is the TCP/IP Socket protocol. In this case, the user program establishes a connection using the IP address of the Analyzer's Socket server.

### The format of the IP address of the analyzer's Socket server

Socket	<i>host address:port</i>
--------	--------------------------

### Examples of the IP address of the analyzer's Socket serve

Socket	192.168.0.1:5025  localhost:5025
--------	--

## **VISA Library**

Using the VISA (Virtual Instrument Software Architecture) library is the most common approach. The VISA library is a widely used software input-output interface in the field of testing and measurement for controlling devices from a PC. It is a library of functions for C/C ++, C #, Visual Basic, MATLAB, LabVIEW and others.

The VISA Library unifies access to all measuring instruments, regardless of the protocol and equipment used.

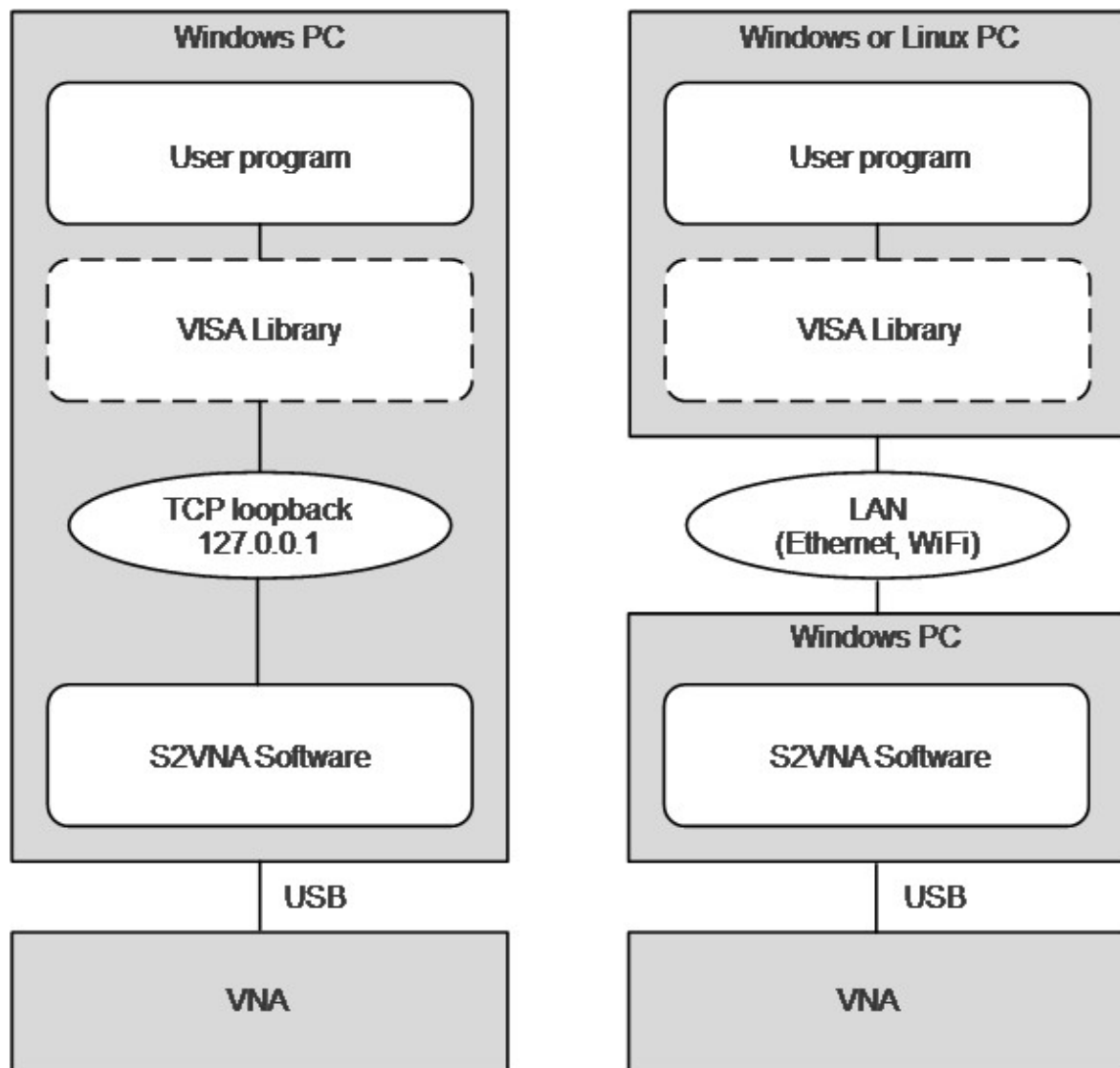
The VISA library is installed on the client side, on the PC where the user program is executed. The VISA library is available on the websites of many companies for free download. There are versions for Linux, Mac OS, Windows.

## Network and Local Configuration

A network configuration involves executing a user program and the analyzer program on different PCs connected by a local area network.

The local configuration involves executing the user program and the analyzer program on the single PC.

The figure below shows the local configuration on the left and the network configuration on the right.



Network and Local Configuration

Local configuration is possible due to the standard TCP/IP stack function — TCP loopback. The TCP loopback function allows network applications to communicate in a standard way within a single PC. The most widely used IP address in the TCP

loopback mechanism is 127.0.0.1. It is also possible to use the symbolic name *localhost* instead of the numeric address 127.0.0.1.

---

**NOTE**

The network configuration does not restrict the client in choice of OS. The local configuration limits the client in choice of OS — only Windows.

---



## Connecting Multiple Analyzers to Single Computer

The section describes in detail how to configure remote control of multiple analyzer programs executed simultaneously on a single PC (provided several USB analyzer hardware units connected to the single PC).

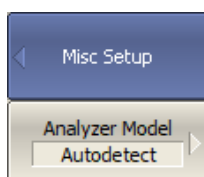
- It is recommended to create a separate folder for each Analyzer with the software. This allows to save individual settings for each Analyzer.
- It is recommended that each copy of the software be linked to a specific hardware unit by its serial number or model (See at the end of this section).
- Assign a unique TCP/IP port number for each copy of the software for the HiSLIP or Socket protocol used. For example, if HiSLIP is used, assign port 4880 to the first analyzer, 4881 to the second, and so on. When assigning a port number, the user must ensure that the port number is not in use by other programs.
- Use the Analyzer's address in the user program with the mandatory indication of the TCP/IP port number assigned to the Analyzer, as in the examples given.

### Examples of the VISA address for the HiSLIP and Socket protocols with the indication of the TCP/IP port

HiSLIP	TCPIP0::192.168.0.1::hislip0,4880::INSTR
	TCPIP0::192.168.0.1::hislip0,4881::INSTR
Socket	TCPIP0::192.168.0.1::5025::SOCKET
	TCPIP0::192.168.0.1::5026::SOCKET

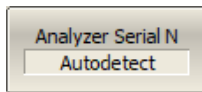
### Examples of the TCP/IP address of the analyzer's Socket server with the indication of the port

Socket	192.168.0.1:5025
	192.168.0.1:5026



To link the analyzer program to the analyzer model, press the following softkeys:

**System > Misc Setup > Analyzer Model**



To link the analyzer program to the analyzer serial number, press the following softkeys:

**System > Misc Setup > Analyzer Serial N**

---

## Differences in Use of HiSLIP and Socket Protocols

The section describes the differences in the methods of writing user programs due to the use of different HiSLIP and TCP/IP Socket protocols. It is assumed that the user program works through the VISA library.

The brief list of differences is given below:

1. The terminal character <newline> in the commands sent to the Analyzer.
2. The terminal character <newline> in the analyzer's responses.
3. Determine the interrupted violation of the message exchange protocol of IEEE488.2.
4. Support for the IEEE488.2 *Status Reporting System*.
5. Support the transfer of binary data.

Then, a detailed description of each item is given.

## Terminal Character in Messages to Analyzer

The user program sends variable-length text messages to the analyzer. The end of the message, according to IEEE488.2, is terminated either by protocol means (not by a symbol), or by the symbol *<newline>* ('*\n*', 0x0A, 10), or both methods together.

The HiSLIP has a mechanism for transmitting the end of the message by protocol means, while the Socket protocol does not. This creates the following requirements for programs sending commands to analyzer:

- Programs using the Socket protocol **shall** send a *<newline>* character at the end of the message.
- Programs using the HiSLIP protocol **may** send the *<newline>* symbol at the end of the message.

---

### NOTE

*For the graphical language LabVIEW when using the Socket protocol to be able to enter the symbol *<newline>* at the end of the message, right-click on the string constant and enable '**\ Codes Display**'. The *<newline>* character is entered as '*\n*'.*

---

---

### NOTE

*For the textual languages it is recommended to use to the symbol *<newline>* at the end of the message regardless of the protocol used.*

---

# Terminal Character in Analyzer Responses

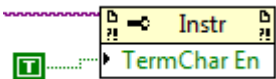
When using the HiSLIP protocol, the analyzer terminates messages with the symbol <newline> + the protocol defined end of message (not symbolic).

When using the Socket protocol, the analyzer terminates messages only with the <newline> symbol, since the Socket protocol does not have the protocol defined end of message.

Depending on the protocol used, the following settings for the VISA library should be made so that it correctly determines the end of the message from the analyzer:

- When using the HiSLIP protocol — no settings are required, the VISA library functions normally with default settings.
- When using the Socket protocol, the user program must set the attribute VI\_ATTR\_TERMCHAR\_EN to TRUE (completion of the read operation when the <newline> character is received).

## Examples of setting up the VISA library using the Socket protocol

C/C++	<code>viSetAttribute(instr, VI_ATTR_TERMCHAR_EN, VI_TRUE);</code>
LabVIEW	

## **Interrupted Error**

The HiSLIP protocol meets the requirements of the IEEE Std 488.2 message exchange protocol to detect an interrupted error. The interrupted error indicates that the Analyzer received an incoming message (command or query) before the client accepted a response from the previous request. In other words, the client is required to read the result of each query before sending the next query or command. If the client fails to do so, the protocol generates an error message and the response from the previous query is cleared by the protocol.

The Socket protocol does not detect the interrupted error. Multiple queries can be sent to the analyzer without a read operation between them. Answers from queries will be returned in the order in which they were sent. The client determines from which request a specific answer has been received.

## **IEEE488.2 Status Reporting System**

The HiSLIP protocol fully supports the analyzer's IEEE488.2 Status Reporting System described in the appendix, while the Socket protocol supports it only partially. The Socket protocol does not support the following functions:

- The MAV (message available) bit in the Status Byte.
- SRQ (service request) generation — request from the Analyzer, implemented by callback functions in the VISA library.
- Read the Status Byte using the dedicated function — `viReadSTB`.

## Transfer of Binary Data

By default, data from the Analyzer is sent in text form. To increase the speed of the data exchange, the user has the option to enable binary data transfer. The transfer of binary data is enabled by the FORMat:DATA command and is effective for commands that transfer large data amounts. A list of such commands is given in the description of the FORMat:DATA command.

The HiSLIP protocol supports VISA formatted input of the binary data, since it provides the protocol defined end of message (not symbolic).

The Socket protocol does not always support VISA formatted input of the binary data (depends on VISA realization), since it uses the *<newline>* byte as the end of the message, which can occur in binary data.



## SCPI Overview

The Analyzer implements a set of commands based on the standard SCPI-1999 (Standard Commands for Programmable Instruments). This is a set of instructions for the exchange of textual messages.

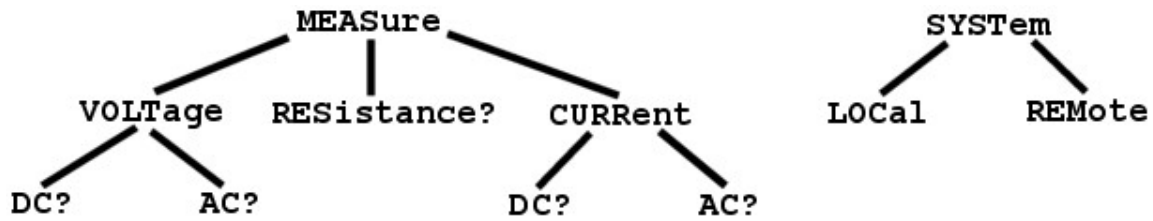
**SCPI** was developed by the SCPI Consortium (currently supported by the IVI Foundation). The main details of the SCPI standard are described further on. More information about the SCPI standard can be downloaded from the [IVI Foundation website](#).

## **Messages**

The SCPI is a text message oriented protocol. The commands are sent as character messages. One message can contain one or several commands. The answer from the instrument is read out as a text message by default. Optionally, an instrument can be programmed to output binary data.

## Command Tree

The SCPI commands are organized in a tree structure. For example:



Each tree structure forms a functional system. The base of the tree is called the root, e.g. MEASure and SYSTem. Each functional system can have subsystems of lower level. The final nodes are called leaves. The entire sequence from root to leaf makes up the command. For example, part of the SOURCe functional system looks as follows:

```
:SOURce
    :POWer
        :CENTer
            :START
            :SPAN
            :STOP
            [:LEVel]
                :SLOPe
                    [:DATA]
                        :STATe
```

This SOURce branch has several levels, where CENTer, START, SPAN, STOP, DATA, STATe are the leaves, which represent the following six commands:

```
:SOURce:POWer:CENTer
:SOURce:POWer:START
:SOURce:POWer:SPAN
:SOURce:POWer:STOP
:SOURce:POWer[:LEVel]:SLOPe[:DATA]
:SOURce:POWer[:LEVel]:SLOPe:STATe
```

The tree can contain subsystems and leaves with the same names if they belong to different branches, e.g. CENTer leaf is on the tips of different branches:

:SOURce	:SENSe
:POWer	:FREQuency
:CENTer	:CENTer

## Subsystems

A colon (':') separates the subsystems. The subsystems which follow the colon are on a lower level. For example, in command:

```
:SOURce:POWer:STARt
```

the start power STARt is a part of the POWer subsystem, which is a part of the SOURce subsystem. The stop power is also a part of the :SOURce:POWer subsystem. It is specified by:

```
:SOURce:POWer:STOP
```

The first colon in the line can be omitted, for example:

```
SOURce:POWer:STOP
```

## Optional Subsystems

Some subsystems can be specified as optional, if omission of such a subsystem will not lead to ambiguity. This means that the subsystem can be omitted in the command line. The optional subsystems are bracketed ("[]"). For example, if the full command specification is written as:

SOURce:POWer[:LEVel]:SLOPe[:DATA]

subsystems LEVel and DATA are optional. Therefore, both commands are valid:

SOURce:POWer:LEVel:SLOPe:DATA

SOURce:POWer:SLOPe

## Long and Short Formats

Each keyword in a command specification has a long format and a short format. The short format of a command is indicated by capital letters. For example, a command specification:

SENSe:FREQuency:CENTer

can be written as:

SENS:FREQ:CENT

SENS:FREQ:CENTer

Only one form can be used at a time, as combining forms will be incorrect. For example, the following specification is incorrect:

:SENS:FREQuen:CEN

## **Case Sensitivity**

The commands are not case sensitive. Upper case and lower case letters are only used to indicate the long and short formats of a command specification. For example, the following commands are equivalent:

`SENS:FREQ:STAR`

`sens:freq:star`



## Parameters

The commands can have parameters. The parameters are separated from the command by a space. If a command has several parameters, they are separated by commas (',').

## Numeric Values

The numeric values are integers or real numbers. These parameters can have measurement units. For example:

SENS:FREQ 1000000000

SENS:FREQ 1000 MHz

SENS:FREQ 1 GHz

SENS:FREQ 1E9

## Multiplier Prefixes

The SCPI standard allows specification of the numeric values with multiplier prefix to the measurement units.

Prefix	Multiplier
A	1e-18
F	1e-15
P	1e-12
N	1e-9
U	1e-6
M	1e-3
K	1e3
MA	1e6
G	1e9
T	1e12
PE	1e15
EX	1e18

There are two exceptions to the above designation: prefix M in combination with HZ or OHM means 1e6 (Mega), and not 1e-3 (milli), i.e. MHZ means Megahertz, same as MAHZ.

## Notations

The SCPI standard allows numeric value specification in different notations. Decimal notation is used by default. To use other notations, specify the numeric values in the following way:

Notation	Prefix	Example
Binary	<b>#B</b>	#B11001010 = 202 <sub>10</sub>
Octal	<b>#Q</b>	#Q107 = 71 <sub>10</sub>
Hexadecimal	<b>#H</b>	#H10FF = 4351 <sub>10</sub>

## Booleans

The booleans can assume two values: logical yes and logical no (ON and OFF), and are specified in command as:

ON or 1 — logical yes

OFF or 0 — logical no

For example:

DISPlay:ENABle OFF

DISPlay:ENABle 0

## Character Data

The SCPI standard allows specification of parameters as character data, as in the following command:

```
TRIGger:SOURce {BUS|IMMediate|EXTernal}
```

where "BUS", "IMMediate", "EXTernal" is the possible values of the character data.

The character data has a long and short format, and the formats are specified in accordance with the same rules as described in [Long and Short Formats](#).

Apart from that, the character data can be combined with numerical parameters. For example:

```
SENSe:FREQuency:STARt {MINimum|MAXimum|<value>}
```

The following specifications are acceptable:

```
SENSe:FREQuency:STARt MIN
```

```
SENSe:FREQuency:STARt maximum
```

```
SENSe:FREQuency:STARt 1000000
```

## String Parameters

In some cases, the Analyzer can accept parameters made of character strings. Such strings are enclosed with single quotes (') or double quotes ("). For example, the file name in the state saving command:

```
MMEMory:STORe "state01.sta"
```

## Numeric Lists

The numeric lists (<numeric list>) are used to specify a variable number of numerical parameters, for example:

CALC:LIMit:DATA 2,1,1E9,3E9,0,0,2,1E9,3E9,-3,-3



## Query Commands

The query commands read out the parameter values from the Analyzer. After a query command has been sent, the response should return via remote control interface.

The query commands has a question mark ('?') at the end of the command. Many of the commands have two forms. The form with a question mark writes the parameter, the form without a question mark reads out the parameter. For example:

SENSe:FREQuency:STARt 1MHz

SENSe:FREQuency:STARt?

## Numeric Suffixes

The Analyzer contains several items of the same type, such as 16 channels, each of which in turn contains 16 traces, etc. A numeric suffix is used to denote the item number in a command. The suffix is added to the keyword of the item (channel, trace, etc). For example, in the following specification the channel number <Ch> and trace number <Tr> indicate the channel and trace, to which this command is addressed:

CALCulate<Ch>:PARameter<Tr>:DEFine

According to this specification, the command referred to the trace 2 of the channel 1 will be written as follows:

CALC1:PAR2:DEF

The numeric suffix can be omitted. In this case, it is 1 by default. For example, the following commands are equivalent:

CALC:PAR:DEF

CALC1:PAR1:DEF

## Compound Commands

It is possible to enter more than one command in the same command line. The commands in the line are separated by a semicolon (;). The specification of the first command is valid for the following command, except for the last leaf before the semicolon. For example:

```
SENS:FREQ:STAR 1 MHZ;STOP 2MHZ
```

To start the next command from the highest level of the structure, begin the command using a colon (':'):

```
SENS:FREQ:STAR 1 MHZ;CALC:PAR:DEF S21
```

## IEEE488.2 Common Commands Overview

A SCPI compatible Analyzer must support a set of common commands of the IEEE488.2 standard. These commands start with an asterisk (\*). The list of such commands can be seen below:

[\\*CLS](#)

[\\*ESE](#)

[\\*ESE?](#)

[\\*ESR?](#)

[\\*IDN?](#)

[\\*OPC](#)

[\\*OPC?](#)

[\\*RST](#)

[\\*SRE](#)

[\\*SRE?](#)

[\\*STB?](#)

[\\*TRG](#)

[\\*WAI](#)

These commands are used for resetting, state queries, etc.

For additional information of functions see [IEEE488.2 Common Commands](#).

## **COM/DOM 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.

## 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 that allows 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.

## Registering COM Server

To register the COM server of the analyzer, run the executable module from the command prompt with the */regserver* keyword. To unregister the COM server of the analyzer, run the executable module from the command prompt with the */unregserver* keyword. Administrative rights are required to register/unregister COM server. The user also has the ability to register the COM server during the software installation procedure.

Example of the COM server registration command:

```
S2VNA.exe /regserver
```

## Automation Controllers

Automation controllers are client programs, which use internal functionality of the 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](http://www.coppermountaintech.com); source code of examples are also located in the Programming Examples\COM\ folder of the application installation folder.

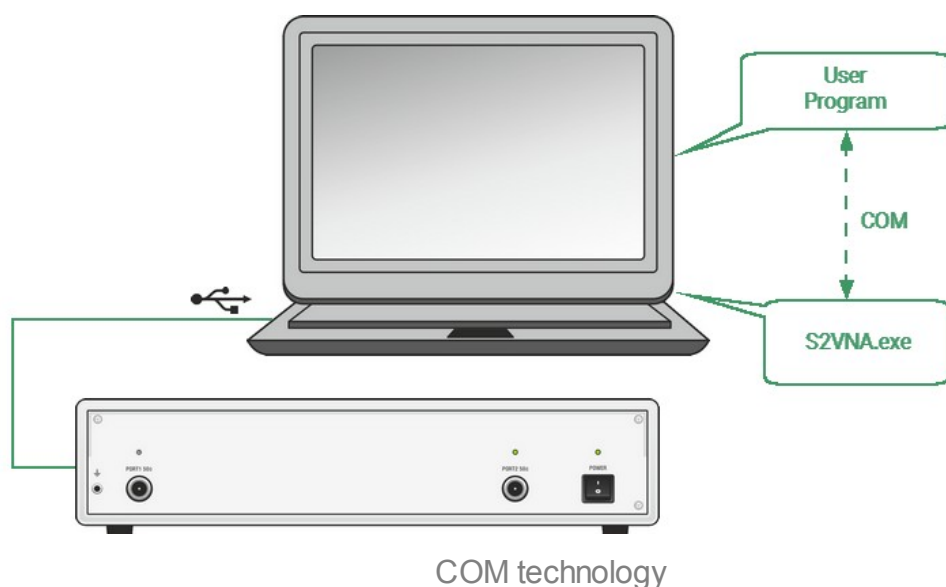
A Labview Driver is also included in the Labview subfolder of the Programming Examples\COM\ folder, and can be downloaded separately from [www.coppermountaintech.com](http://www.coppermountaintech.com). The Labview Driver contains examples of its use.



## Local and Remote Server

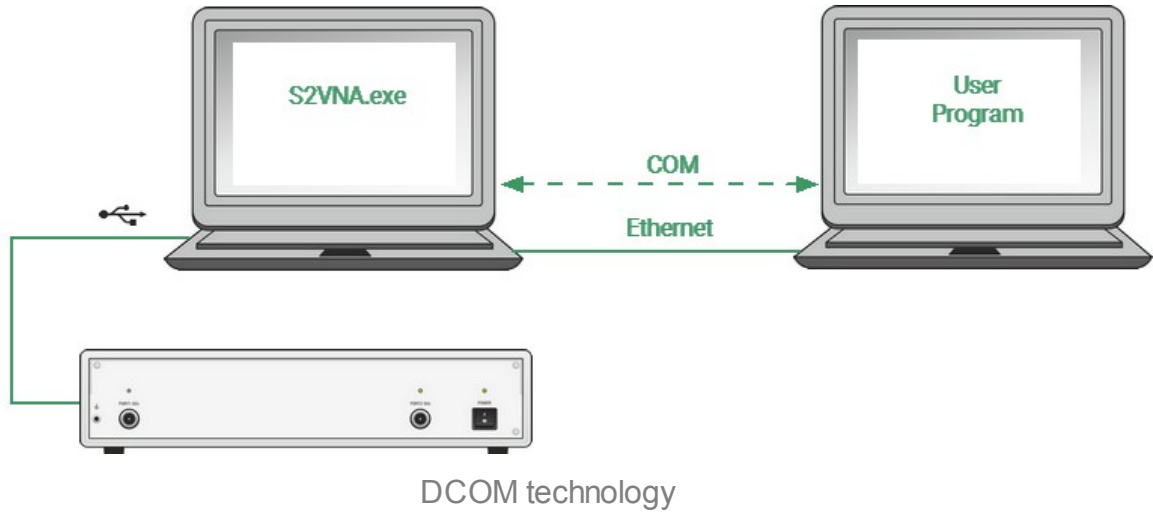
The network analyzer executable module can function either as a local server or as a remote server of COM automation.

The **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 (See figure below).



The **Remote server** and the automation controller run on different PCs connected by LAN. DCOM (Distributed COM) technology is used in this case (See figure below). When using DCOM, 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.



## DCOM Setup

The next section describes the settings for controlling the Analyzer via a network from a remote PC using DCOM technology.

## Instrument Setup

A PC with a connected USB Analyzer must be connected to the local network and configured as a member of a domain or a member of a working group for managing DCOM technology. The network administrator must join the analyzer or control computer to a domain in the first case. An administrator or user assigns a workgroup name and adds user accounts in the second case.

The user category "everyone" has access to DCOM objects of the device. For the working group, the "everyone" user category includes those users with local accounts in the device. In the domain, the "everyone" user category includes users with local accounts, as well as all domain users, even if they do not have local accounts.

The device is configured in one of two ways:

- Join the device to a domain, which makes network connections of domain users to the device easier.
- When using a workgroup, start by creating local accounts on the device for each user who will have access to DCOM objects. The local user account in the device must match the local account on its remote computer (login, password).

## Remote Computer Setup

A **remote computer** is a user's computer from which the analyzer is remotely controlled via a local network.

Copy the S2VNA.exe file to the remote computer from the analyzer with a built-in computer or from the computer controlling the USB Analyzer. Run this file once with the /regserver keyword, and the COM server will register on the remote computer. After that, the file can be deleted.

To replace COM technology with DCOM technology, use one of two methods:

- Make changes to the source code of the programs.
- Change the DCOM settings on the remote computer using the dcomcnfg.exe utility.

The first method requires modifying the CreateObject statement. This operator requires an explicit inclusion of the network name of the device or its IP address, for example:

```
Set app = CreateObject("S2VNA.Application", "o304-000123")  
Set app = CreateObject("S2VNA.Application", "192.168.1.149")
```

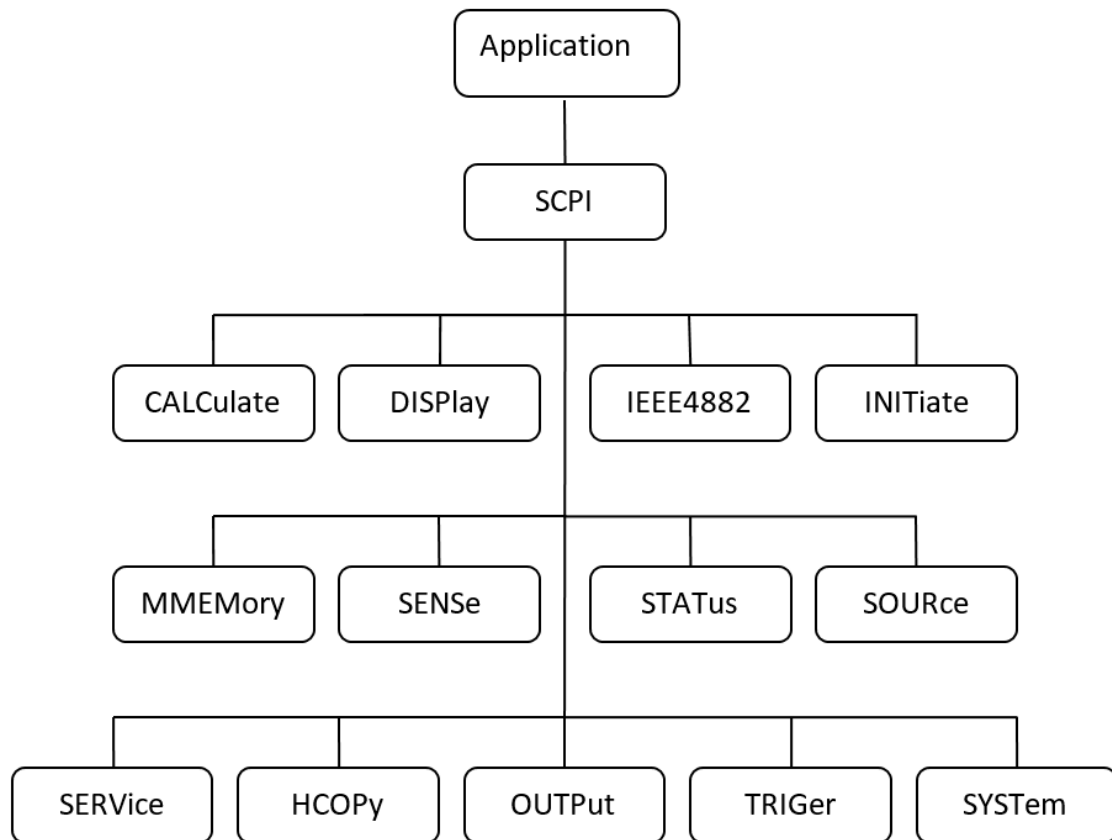
The network name of the device can be found in the system properties (Start> Control Panel> System> Computer Name).

The second method is to indicate the location of the COM server S2VNA.exe using the dcomcnfg.exe utility. Run the specified utility on the remote computer, which is usually located in the C:\WINDOWS\SYSTEM32 folder:

- Go to Component "Services > Computers > My Computer > DCOM Setup".
- Find the "S2VNA Object" in the list and open the "Properties" dialog.
- Click the "Location" tab, deselect the "Run application on this computer" check box, and select the "Run application on the following computer" check box.
- Then, enter the network name of the device.

## Structure of COM Objects

The COM server contains several objects, which provide different functionality of the server. The COM objects of the Analyzer executable module are organized in a hierarchical structure. The figure below 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, as well as allowing access to the objects of the lower levels.



The structure of COM objects

The Object Application is at 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"`

---

# 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:

- Declare a variable as an object.
- Create a COM Object and assign it to this variable.

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

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

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

<i>ObjectName</i>	Object name is always equal to "S2VNA.Application"
<i>HostName</i>	Network name of the PC hosting the COM server. This parameter is not specified in the case of a local server.

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

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

---

**NOTE**

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

---

To allow access to the objects of a lower level on 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.

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

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

For backwards 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.Applction")  
Set app = CreateObject("S5048.Applction")  
Set app = CreateObject("S7530.Applction")  
Set app = CreateObject("Planar304.Applction")  
Set app = CreateObject("Planar804.Applction")
```

## 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 sets the Analyzer to the preset condition:

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

## 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 shows the setting of the *POINTS* property of the *SWEep* object, i.e. the number of sweep points:

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

---

### NOTE

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

---

## Error Handling

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

- Check the value of the `Err.Number` variable after execution of the 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. The `On Error Resume Next` directive instructs VB not to interrupt the program execution when the error is detected, but to pass control to the next operator in natural order.

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

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

```
Dim app
Public Sub HandleError2()
Set app = CreateObject("S2VNA.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
```

## COM Automation Data Types

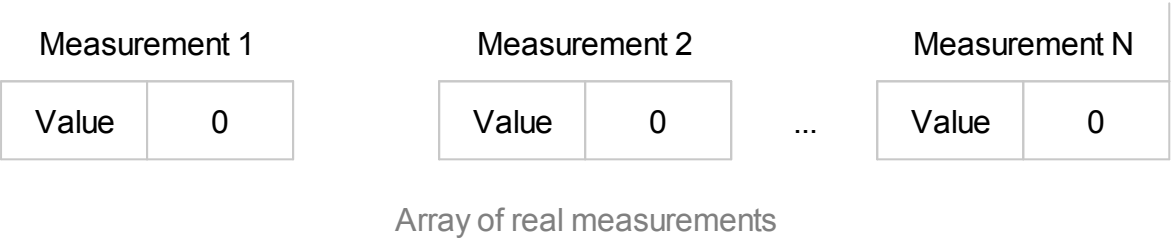
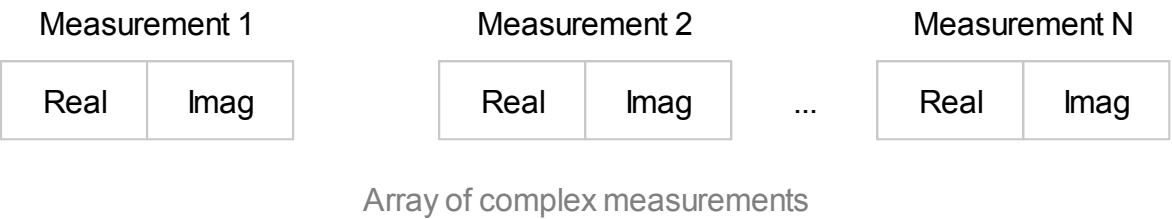
In COM automation contains the following data types, which can be used for client-to-server communication:

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

# 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 complex in polar format.

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



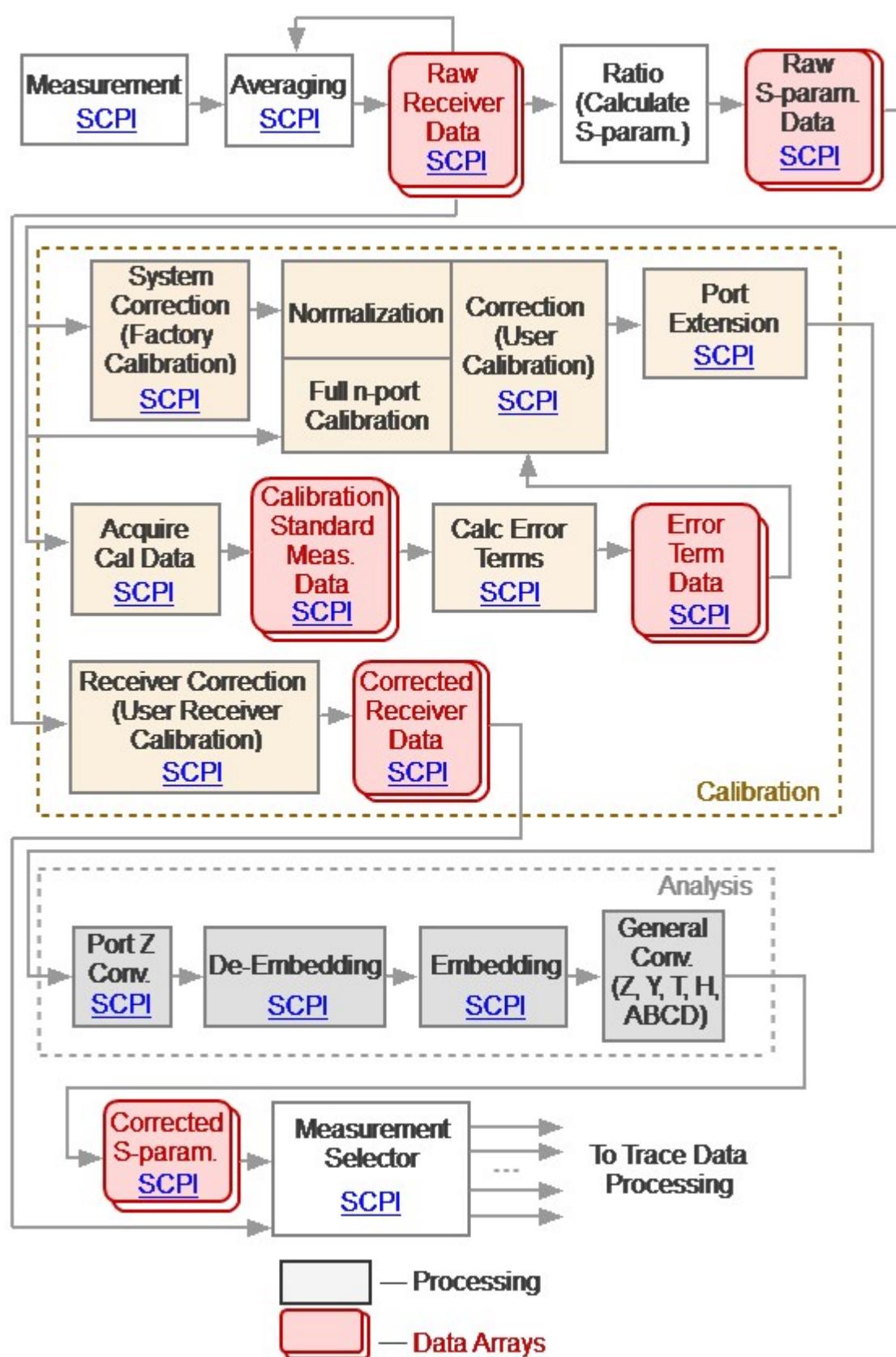
## Internal Data Arrays

This section describes the internal data arrays, access to them, as well as their position in the internal data flow of the Analyzer (See figure below). For a description of internal data processing, see [Internal Data Processing](#). To search for SCPI commands related to arrays and processes, click "[SCPI](#)" in the figures below.

### Channel Data Processing

All internal arrays of channel data processing (See figure below) contain the number of elements equal to twice the number of stimulus points. Each measurement point is represented in the array by a pair of adjacent elements. The odd elements of the array contain the real part of the data, the even ones contain the imaginary part of the data.





Channel Data Processing

Channel data processing of the Analyzer consists of the following arrays:

- **Raw Receivers Data Arrays** are obtained as a result of analog-to-digital conversion and digital filtering of analog signals received by the receivers. If averaging is enabled, then the array elements are averaged pointwise over N sweep cycles. Array data is available for reading using the SCPI command [SENS:DATA:RAWD?](#).
- **Raw S-param. Data Arrays** are obtained by calculating the ratio of the signals two receivers. Array data is available for reading using the SCPI command [SENS:DATA:RAWD?](#).
- **Calibration Standard Meas. Data Arrays** are temporary arrays that contain the results of the performed measurements of the calibration standards. Upon completion of the calibration process, after calculating the error terms, the arrays are cleared. Array data is available for reading or writing using the SCPI commands [SENS:CORR:COLL:DATA:XXXX](#).
- **Error Term Data Arrays** are obtained as a result of processing measurements of calibration standards. Arrays are used in the correction when error terms are applied to the measured S-parameters. Array data is available for reading or writing using the SCPI command [SENS:CORR:COEF](#).

---

**NOTE**

Error terms will be interpolated if, for example, the number of measurement points or stimulus settings for measurements and during calibration differ. In this case, the [SENS:CORR:COEF](#) command will read the interpolated data from the array.

---

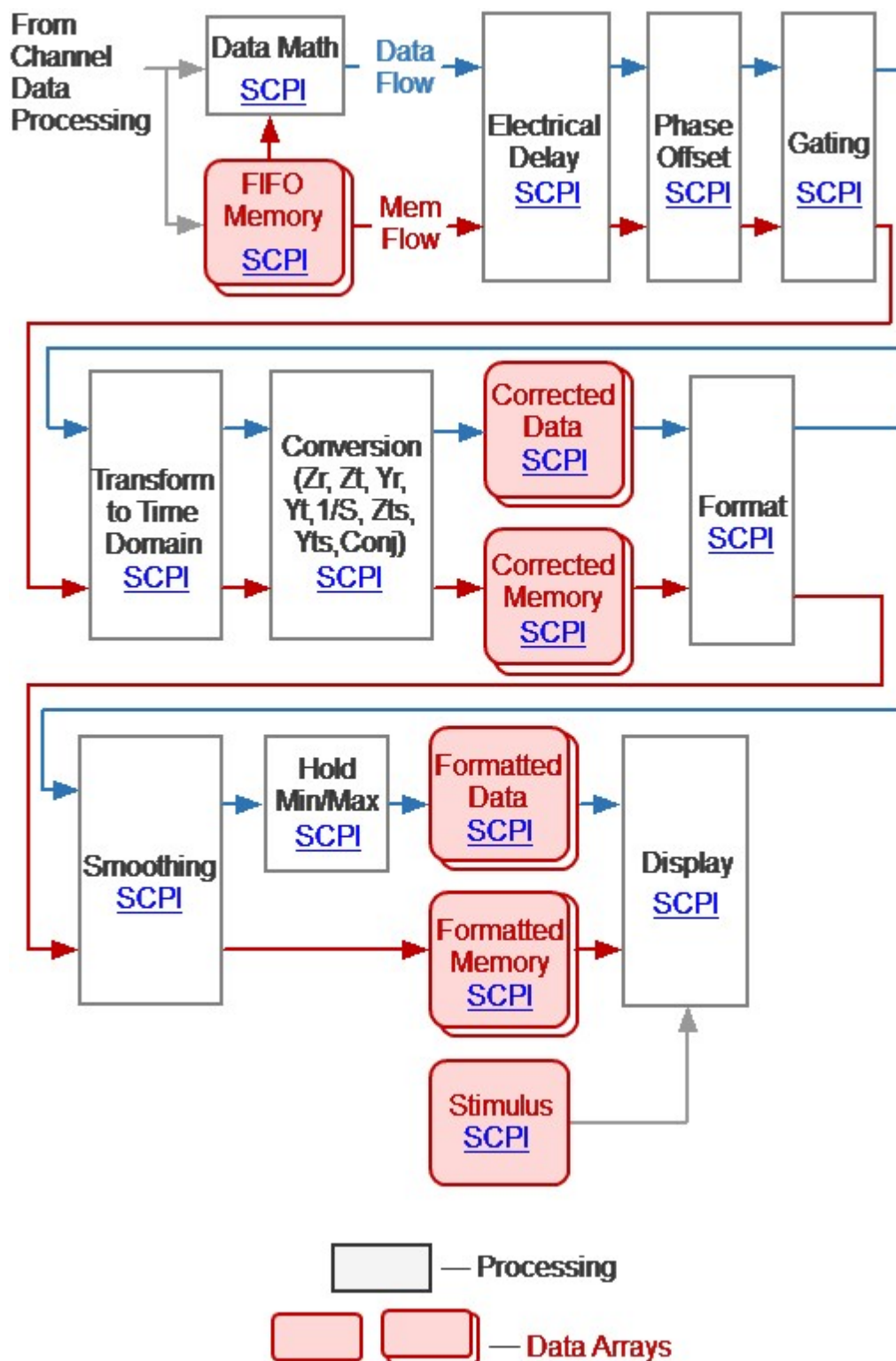
- **Corrected Receivers Data Arrays** are obtained as a result of the correction of the raw receiver data if the receivers are calibrated. This data is displayed on the screen if absolute measurements are selected. Array data is available for reading using the SCPI command [SENS:DATA:CORR?](#)
- **Corrected S-param. Data Arrays** are obtained from raw S-parameter arrays by performing the following operations: Correction, Port Extension, Port Z conversion, Embedding/De-embedding, General Conversion. Array data is readable using SCPI command [SENS:DATA:CORR?](#)

### Trace Data Processing

The following data arrays: FIFO memory, Corrected Data, and Corrected Memory, Formatted Data, and Formatted Memory (See figure below) contain the number of elements equal to twice the number of stimulus points. Each measurement point is represented in the array by a pair of adjacent elements. The stimulus data array has the number of elements equal to the number of stimulus points.

In the following data arrays: FIFO memory, Corrected Data, and Corrected Memory, the odd array elements contain the real part of the data, the even ones contain the imaginary part of the data.

The arrays of Formatted Data and Formatted Memory, depending on the selected data format, contain data of various types (See [table](#)).



Trace Data Processing

- **FIFO Memory** is the queue of memory arrays type "first-in-first-out" basis. The next array is saved in FIFO as the result of activating the "Data-> Memory" function. The

measurement (S-parameter or receiver data) of the associated trace is copied to the array. By default, the FIFO depth (size) is one, which means each trace has one associated memory array. When the FIFO function is enabled, the queue depth increases to eight. The memory can be used both for display and for math operations in conjunction with data. Active memory is selected for math operation with data if the FIFO depth is greater than one. SCPI commands for accessing this array are absent.

---

**NOTE**

Math memory operations are performed between the complex data of the current measurements and the memory, not between their formatted values (memory traces and data traces).

---

---

**NOTE**

The memory arrays are processed in parallel with the measurement data array in subsequent processing stages. For example, the formatting has the same effect on the data trace as it does on the memory trace. In subsequent stages of processing, the number of memory arrays equal to the FIFO depth is used.

---

- **Corrected Data Array** is obtained from the corrected S-parameter arrays or the corrected receiver data arrays as a result of performing the following operations: Trace Math, Electrical Delay, Phase Offset, Gating, Transform to Time Domain, and Conversion S-parameters. Arrays contain data that has been processed, with the exception of formatting. Array data is available for reading or writing using the SCPI command [CALC:DATA:SDAT](#).
- **Corrected Memory Arrays** is obtained from the Memory FIFO arrays as a result of performing the following operations: Electrical Delay, Phase Offset, Gating, Transform to Time Domain, and Conversion S-parameters. Arrays contain data that has been processed, with the exception of formatting. Array data is available for reading or writing using the SCPI command [CALC:DATA:SMEM](#).
- **Formatted Data Array** is obtained by formatting the corrected data array and applying smoothing and hold operations to it. Arrays contain data that is ready to be displayed as a trace. Depending on the data format, the arrays contain two values for each measuring point (See [table](#)). Array data is available for reading or writing using the SCPI command [CALC:DATA:FDAT](#).
- **Formatted Memory Data Arrays** are obtained by formatting corrected memory arrays and applying the smoothing operation to them. Arrays contain data that is ready to be displayed as a trace. Depending on the data format, the arrays contain two values for each measuring point (See [table](#)). Array data is available for reading or writing using the SCPI command [CALC:DATA:FMEM](#).

- **Stimulus Data Array** contains the channel stimulus values for all measurement points. The data is available for reading using the SCPI command [SENS:FREQ:DATA?](#).

## Command Reference

### Conventions

The following conventions are used throughout this section.

### Syntax

The following symbols are used in command syntax:

<>	Identifiers enclosed in angular brackets indicate that a particular type of data must be specified.
[]	Parts enclosed in square brackets can be omitted.
{ }	Parts enclosed in curly brackets indicate that you must select one of the items in this part. Individual items are separated by a vertical bar " ".
Space	Space separates commands from parameters.
,	Comma separates adjacent parameters.
...	Ellipses indicate that parameters in that part are omitted.

### Identifiers

Identifier	Parameter	Description
<numeric>	Number	{<integer> <real>}
<frequency>	Frequency	<numeric>{[HZ] KHZ MHZ GHZ}
<power>	Power	<numeric>{[DBM] DBMW DBW KW W MW UW NW}
<time>	Time	<numeric>{[S] MS US NS PS FS}
<phase>	Phase	<numeric>{[DEG] MADEG KDEG MDEG UDEG}
<stimulus>	Stimulus	{<frequency> <power> <time>}

Identifier	Parameter	Description
<numeric list>	Numeric List	<numeric 1>,<numeric 2>,...<numeric N>
<bool>	Boolean parameter	{0 1 ON OFF}
<char>	Character parameter	Predefined set of character strings without quotes
<port>	Port Number	<integer>
<string>	String parameter	Quoted string

### Equivalent COM Command

The Analyzer command system description is based on the SCPI command system because this system is used primarily in this manual. In addition, the structure of COM objects and their names are borrowed from the SCPI command system. In this manual, COM commands are presented as equivalent to SCPI commands. The description of COM commands shows differences in their function from SCPI commands. If the SCPI command does not have a COM equivalent, this is noted in its description.



## SCPI Command Tree

<a href="#"><u>ABORt</u></a>	Aborts all sweeps.
<a href="#"><u>CALCulate</u></a>	Data processing (conversion, electrical delay, phase offset, gating, fixture simulation, trace hold, smoothing, time domain), trace analysis, limit tests, markers, trace memory, math, statistic, trace data transfer.
<a href="#"><u>DISPlay</u></a>	Display settings.
<a href="#"><u>FORMat</u></a>	Trace format.
<a href="#"><u>HCOPy</u></a>	Hardcopy printing.
<a href="#"><u>IEEE488.2</u></a>	IEEE488.2 Common commands.
<a href="#"><u>INITiate</u></a>	Channel initiation mode.
<a href="#"><u>MMEMory</u></a>	File operations.
<a href="#"><u>OUTP</u></a>	RF power ON/OFF.
<a href="#"><u>SENSe</u></a>	Averaging, calibration, calibration kit management, oort extension, IFBW setting, frequency settings, sweep settings, frequency offset, channel data transfer.
<a href="#"><u>SERVICE</u></a>	Read active channel/trace/marker number, Analyzer capabilities.
<a href="#"><u>SOURce</u></a>	Power settings, power calibration.
<a href="#"><u>STATus</u></a>	Status reporting system.
<a href="#"><u>SYSTem</u></a>	System settings and preset.
<a href="#"><u>TRIGger</u></a>	Trigger system.

## IEEE488.2 Common Commands

The set of common commands of IEEE488.2 standard. These commands start with an asterix (\*").

Command		Description
<a href="#">*CLS</a>	Status System	Clear status
<a href="#">*ESE</a>		Event status enable
<a href="#">*ESR?</a>		Event status enable register
<a href="#">*IDN?</a>		Identify
<a href="#">*OPC</a>		Operation complete command
<a href="#">*OPC?</a>		Operation complete query
<a href="#">*RST</a>		Reset
<a href="#">*SRE</a>		Service request enable
<a href="#">*STB?</a>		Status byte query
<a href="#">*TRG</a>		Trigger signal
<a href="#">*TST?</a>		Result of self-test query
<a href="#">*WAI</a>		Wait

## **\*CLS**

### **SCPI Command**

\*CLS

### **Description**

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.

no query

### **Target**

Status Reporting System

### **Equivalent Softkeys**

None

---

### **Equivalent COM Command**

SCPI.IEEE4882.CLS

### **Syntax**

app.SCPI.IEEE4882.CLS

**Type**

Method

---

Back to [IEEE488.2 Common Commands](#)

## **\*ESE**

### **SCPI Command**

\*ESE <numeric>

\*ESE?

### **Description**

Sets or reads out the value of the Standard Event Status Enable Register.

command/query

### **Target**

Status Reporting System

### **Parameter**

<numeric> 0 to 255

### **Query Response**

<numeric>

### **Preset Value**

0

### **Equivalent Softkeys**

None

---

### **Equivalent COM Command**

None

---

Back to [IEEE488.2 Common Commands](#)

## **\*ESR?**

### **SCPI Command**

\*ESR?

### **Description**

Reads out the value of the Standard Event Status Register. Executing this command clears the register value.

query only

### **Target**

Status Reporting System

### **Query Response**

<numeric>

### **Equivalent Softkeys**

None

---

### **Equivalent COM Command**

None

---

Back to [IEEE488.2 Common Commands](#)

## **\*IDN?**

### **SCPI Command**

\*IDN?

### **Description**

Reads out the Analyzer identification string.

query only

### **Target**

Analyzer

### **Query Response**

The identification string in format: <manufacturer>, <model>, <serial number>, <software version>/<hardware version>.

For example: CMT, C1209, 08080188, 16.2/01

### **Equivalent Softkeys**

None

---

### **Equivalent COM Command**

SCPI.IEEE4882.IDN

NAME

### **Syntax**

StrName = app.NAME

### **Type**

String (read only)

---

Back to [IEEE488.2 Common Commands](#)

## **\*OPC**

### **SCPI Command**

\*OPC

### **Description**

Sets the OPC bit (bit 0) of the Standard Event Status Register at the completion of all pending operations.

The pending operation caused by the command [TRIG:SING](#) only.

no query

### **Target**

Status Reporting System

### **Equivalent Softkeys**

None

---

### **Equivalent COM Command**

SCPI.IEEE4882.OPC

### **Syntax**

Value = app.SCPI.IEEE4882.OPC

app.SCPI.IEEE4882.OPC = Dummy

### **Type**

Long (read/write)

---

Back to [IEEE488.2 Common Commands](#)



## **\*OPC?**

### **SCPI Command**

\*OPC?

### **Description**

Reads out the "1" at the completion of all pending operations. The query blocks the execution of the user program until execution of all previous instructions.

The query \*OPC? can be used for waiting for the end of a sweep initiated by the command [TRIG:SING](#).

query only

### **Target**

Analyzer

### **Query Response**

1

### **Related Commands**

[TRIG:SING](#)

### **Equivalent Softkeys**

None

---

### **Equivalent COM Command**

SCPI.IEEE4882.OPC

### **Syntax**

Value = app.SCPI.IEEE4882.OPC

app.SCPI.IEEE4882.OPC = Dummy

**Type**

Long (read/write)

---

Back to [IEEE488.2 Common Commands](#)

## **\*RST**

### **SCPI Command**

\*RST

### **Description**

Restores the default settings of the Analyzer.

There is difference from presetting the Analyzer with [SYST:PRES](#) command – in this case all channels are set to Hold.

no query

### **Target**

Analyzer

### **Related Commands**

[SYST:PRES](#)

### **Equivalent Softkeys**

None

---

### **Equivalent COM Command**

SCPI.IEEE4882.RST

### **Syntax**

app.SCPI.IEEE4882.RST

### **Type**

Method

---

Back to [IEEE488.2 Common Commands](#)

## **\*SRE**

### **SCPI Command**

\*SRE <numeric>

\*SRE?

### **Description**

Sets or reads out the value of the Service Request Enable Register.

command/query

### **Target**

Status Reporting System

### **Parameter**

<numeric> 0 to 255

### **Query Response**

<numeric>

### **Preset Value**

0

### **Equivalent Softkeys**

None

---

### **Equivalent COM Command**

None

---

Back to [IEEE488.2 Common Commands](#)

## **\*STB?**

### **SCPI Command**

\*STB?

### **Description**

Reads out the value of the Status Byte Register.

query only

### **Target**

Status Reporting System

### **Query Response**

<numeric>

### **Equivalent Softkeys**

None

---

### **Equivalent COM Command**

None

---

Back to [IEEE488.2 Common Commands](#)

## **\*TRG**

### **SCPI Command**

\*TRG

### **Description**

Generates a trigger signal and initiates a sweep under the following conditions.

1. Trigger source is set to the BUS (set by the command [TRIG:SOUR](#) BUS), otherwise an error occurs and the command is ignored.
2. Analyzer must be in the trigger waiting state, otherwise (the analyzer is in the measurement state or hold state) an error occurs and the command is ignored.

The command is completed immediately after the generation of the trigger signal.

no query

### **Target**

Analyzer

### **Related Commands**

[TRIG:SOUR](#)

[INIT](#)

[INIT:CONT](#)

### **Equivalent Softkeys**

None

---

### **Equivalent COM Command**

SCPI.IEEE4882.TRG

**Syntax**

app.SCPI.IEEE4882.TRG

**Type**

Method

---

Back to [IEEE488.2 Common Commands](#)

## **\*TST?**

### **SCPI Command**

\*TST?

### **Description**

Reads out the analyzer self-test result. 0 indicates no failures found. A non-zero value indicates one or more of failure conditions exist. The [SYST:TEST?](#) query returns a textual description of the failures.

**Note:** the query returns a non-zero value when it is issued until the instrument is ready.

query only

### **Target**

Instrument

### **Query Response**

<numeric>

### **Related Commands**

[SYST:TEST?](#)

[SYST:READY?](#)

### **Equivalent Softkeys**

None

---

### **Equivalent COM Command**

None

---

Back to [IEEE488.2 Common Commands](#)



## **\*WAI**

### **SCPI Command**

\*WAI

### **Description**

Waits till the completion of all pending commands. The only command that can be pending is the [TRIG:SING](#) command.

In absence of a pending command [TRIG:SING](#) the command [\\*WAI](#) is equivalent to an empty operation.

A query that follows the command \*WAI blocks the execution of the user program till the completion of the command [TRIG:SING](#), similarly to the query [\\*OPC?](#).

no query

### **Target**

Analyzer

### **Related Commands**

[TRIG:SING](#)

### **Equivalent Softkeys**

None

---

### **Equivalent COM Command**

SCPI.IEEE4882.WAI

### **Syntax**

app.SCPI.IEEE4882.WAI

### **Type**

Method

---

**NOTE**

Since COM server executes commands sequentially and any operation is complete before COM server returns control the WAI command doesn't wait anything.

---

Back to [IEEE488.2 Common Commands](#)

# ABOR

## SCPI Command

ABORt

## Description

Aborts the sweep. The channels in the Single trigger initiation mode transfer to the Hold state. The channels in the Continuous trigger initiation mode transfer to the trigger waiting state. If the trigger source is set to Internal, the channel immediately starts a new sweep.

no query

## Related Commands

[INIT:CONT](#)

## Equivalent Softkeys

Stimulus > Trigger > Restart

---

## Equivalent COM Command

SCPI.ABORt

## Syntax

app.SCPI.ABORt

## Type

Method

---

## CALCulate

Command		Description
<a href="#">CALC:CONV</a>	S-parameter Conversion	Conversion ON/OFF
<a href="#">CALC:CONV:FUNC</a>		Conversion type
<a href="#">CALC:CORR:EDEL:DIST</a>	Electrical Delay	Equivalent distance
<a href="#">CALC:CORR:EDEL:DIST:UNIT</a>		Distance units
<a href="#">CALC:CORR:EDEL:MED</a>		Media
<a href="#">CALC:CORR:EDEL:RVEL</a>		Velocity factor
<a href="#">CALC:CORR:EDEL:TIME</a>		Electrical delay
<a href="#">CALC:CORR:EDEL:WAV:CUT</a>		Waveguide cutoff frequency
<a href="#">CALC:CORR:OFFS:PHAS</a>	Phase Offset	Value of the phase offset
<a href="#">CALC:CORR:STAT?</a>	Misc Calibration Commands	Interpolation/extrapolation status of the error correction
<a href="#">CALC:DATA:FDAT</a>	Data Transfer	Formatted data array
<a href="#">CALC:DATA:FMEM</a>		Formatted memory array

Command		Description
<a href="#">CALC:DATA:SDAT</a>		Corrected data array
<a href="#">CALC:DATA:SMEM</a>		Corrected memory array
<a href="#">CALC:DATA:XAX?</a>		X-axis values array
<a href="#">CALC:FILT:TIME</a>	Gating	Gate type
<a href="#">CALC:FILT:TIME:CENT</a>		Gate center
<a href="#">CALC:FILT:TIME:SHAP</a>		Gate shape
<a href="#">CALC:FILT:TIME:SPAN</a>		Gate span
<a href="#">CALC:FILT:TIME:STAR</a>		Gate start
<a href="#">CALC:FILT:TIME:STAT</a>		Gating function ON/OFF
<a href="#">CALC:FILT:TIME:STOP</a>		Gate stop
<a href="#">CALC:FORM</a>	Channel and Trace Settings	Trace format
<a href="#">CALC:PAR:COUN</a>		Number of traces in the channel
<a href="#">CALC:PAR:SEL</a>		Active trace number (write)

Command		Description
<a href="#">CALC:FSIM:SEND:DEEM:STAT</a>	Two-port Network De-embedding	De-embedding ON/OFF
<a href="#">CALC:FSIM:SEND:DEEM:PORT:STAT</a>		De-embedding for specified port ON/OFF
<a href="#">CALC:FSIM:SEND:DEEM:PORT:USER:FIL</a>		Name of *.S2P touchstone file of the de-embedded circuit
<a href="#">CALC:FSIM:SEND:PMC:STAT</a>	Two-port Network Embedding	Embedding ON/OFF
<a href="#">CALC:FSIM:SEND:PMC:PORT:STAT</a>		Embedding for specified port ON/OFF
<a href="#">CALC:FSIM:SEND:PMC:PORT:USER:FIL</a>		Name of *.S2P Touchstone file of the embedded circuit
<a href="#">CALC:FSIM:SEND:ZCON:PORT:Z0</a>	Port Impedance Conversion	Z0 Real part, Imaginary part is "0"
<a href="#">CALC:FSIM:SEND:ZCON:PORT:Z0:REAL</a>		Z0 Real part
<a href="#">CALC:FSIM:SEND:ZCON:PORT:Z0:IMAG</a>		Z0 Imaginary part
<a href="#">CALC:FSIM:SEND:ZCON:STAT</a>		Port Z conversion ON/OFF
<a href="#">CALC:FSIM:SEND:ZCON:THE</a>		Theory of Port Z Conversion

Command		Description
<a href="#">CALC:FSIM:STAT</a>	Fixture Simulation Function	Fixture simulation ON/OFF
<a href="#">CALC:FUNC:DATA?</a>	Trace Analysis	Analysis result data array
<a href="#">CALC:FUNC:DOM</a>		Arbitrary sweep range ON/OFF
<a href="#">CALC:FUNC:DOM:COUP</a>		Coupling range ON/OFF
<a href="#">CALC:FUNC:DOM:STAR</a>		Analysis range start
<a href="#">CALC:FUNC:DOM:STOP</a>		Analysis range stop
<a href="#">CALC:FUNC:EXEC</a>		Execute analysis
<a href="#">CALC:FUNC:PEXC</a>		Lower limit for the peak excursion value
<a href="#">CALC:FUNC:POIN?</a>		Number of points (data pairs)
<a href="#">CALC:FUNC:PPOL</a>		Peak polarity
<a href="#">CALC:FUNC:TARG</a>		Target level
<a href="#">CALC:FUNC:TTR</a>		Transition type

Command		Description
<a href="#">CALC:FUNC:TYPE</a>		Analysis type
<a href="#">CALC:HOLD:TYPE</a>	Trace Hold	Trace hold type
<a href="#">CALC:HOLD:CLE</a>		Trace hold restart
<a href="#">CALC:LIM</a>	Limit Test	Limit test ON/OFF
<a href="#">CALC:LIM:DATA</a>		Limit line table
<a href="#">CALC:LIM:DISP</a>		Limits display ON/OFF
<a href="#">CALC:LIM:FAIL?</a>		Limit test result
<a href="#">CALC:LIM:OFFS:AMPL</a>		Limit line Y-offset
<a href="#">CALC:LIM:OFFS:MARK</a>		Limit line Y-offset to active marker value
<a href="#">CALC:LIM:OFFS:STIM</a>		Limit line X-offset
<a href="#">CALC:LIM:REP:ALL?</a>		Limit test result report
<a href="#">CALC:LIM:REP:POIN?</a>		Failed points
<a href="#">CALC:LIM:REP?</a>		Stimulus values of failed points



Command		Description
<a href="#">CALC:MARK</a>	Marker Properties	Marker ON/OFF
<a href="#">CALC:MARK:ACT</a>		Sets active marker
<a href="#">CALC:MARK:COUN</a>		Number of markers
<a href="#">CALC:MARK:COUP</a>		Coupling of markers ON/OFF
<a href="#">CALC:MARK:DATA?</a>		Response and stimulus values of all marker
<a href="#">CALC:MARK:DISC</a>		Marker discrete mode ON/OFF
<a href="#">CALC:MARK:REF</a>		Reference marker ON/OFF
<a href="#">CALC:MARK:X</a>		Stimulus value of marker
<a href="#">CALC:MARK:Y?</a>		Response value of marker
<a href="#">CALC:MARK:BWID</a>	Bandwidth Search	Bandwidth search ON/OFF
<a href="#">CALC:MARK:BWID:DATA?</a>		Bandwidth search result
<a href="#">CALC:MARK:BWID:REF</a>		Reference of search
<a href="#">CALC:MARK:BWID:THR</a>		Bandwidth threshold value

Command		Description
<a href="#">CALC:MARK:BWID:TYPE</a>		Type of search
<a href="#">CALC:MARK:FUNC:DOM</a>	Marker Search	Arbitrary search range ON/OFF
<a href="#">CALC:MARK:FUNC:DOM:COUP</a>		Coupling of marker search ranges ON/OFF
<a href="#">CALC:MARK:FUNC:DOM:STAR</a>		Start of the marker search range
<a href="#">CALC:MARK:FUNC:DOM:STOP</a>		Stop of the marker search range
<a href="#">CALC:MARK:FUNC:EXEC</a>		Executes search
<a href="#">CALC:MARK:FUNC:PEXC</a>		Peak excursion value
<a href="#">CALC:MARK:FUNC:PPOL</a>		Peak polarity
<a href="#">CALC:MARK:FUNC:TARG</a>		Target value
<a href="#">CALC:MARK:FUNC:TRAC</a>		Marker search tracking ON/OFF
<a href="#">CALC:MARK:FUNC:TTR</a>		Type of target transition
<a href="#">CALC:MARK:FUNC:TYPE</a>		Search type
<a href="#">CALC:MARK:MATH:FLAT:DATA?</a>	Flatness	Flatness function data

Command		Description
<a href="#">CALC:MARK:MATH:FLAT:STAT</a>		Marker flatness ON/OFF
<a href="#">CALC:MARK:MATH:FLAT:DOM:STAR</a>		Marker specifying start of frequency range
<a href="#">CALC:MARK:MATH:FLAT:DOM:STOP</a>		Marker specifying stop of frequency range
<a href="#">CALC:MARK:SET</a>	Marker Functions	Sets item value according to the position of the marker
<a href="#">CALC:MATH:FUNC</a>	Memory Trace Function	Math operation
<a href="#">CALC:MATH:MEM</a>		Data => Memory
<a href="#">CALC:MST</a>	Statistic	Math statistics ON/OFF
<a href="#">CALC:MST:DATA?</a>		Math statistics data
<a href="#">CALC:MST:DOM</a>		Partial frequency range ON/OFF
<a href="#">CALC:MST:DOM:STAR</a>		Marker specifying start of frequency range
<a href="#">CALC:MST:DOM:STOP</a>		Marker specifying stop of frequency range
<a href="#">CALC:PAR:DEF</a>	Measurement Setting	Measurement parameter of a trace

Command		Description
<a href="#">CALC:PAR:SPOR</a>		Number of the stimulus port for an absolute measurements or DC Voltage measurements
<a href="#">CALC:RLIM</a>	Ripple Limit Test	Ripple limit test ON/OFF
<a href="#">CALC:RLIM:DATA</a>		Ripple limit line table
<a href="#">CALC:RLIM:DISP:LINE</a>		Ripple Limit line display ON/OFF
<a href="#">CALC:RLIM:DISP:SEL</a>		Number of band for ripple value display
<a href="#">CALC:RLIM:DISP:VAL</a>		Display type of ripple value
<a href="#">CALC:RLIM:FAIL?</a>		Ripple limit test result
<a href="#">CALC:RLIM:REP?</a>		Ripple limit test result report
<a href="#">CALC:SMO</a>	Smoothing	Trace smoothing ON/OFF
<a href="#">CALC:SMO:APER</a>		Smoothing aperture
<a href="#">CALC:TRAN:TIME</a>	Setting Time Domain Parameters	Selects Band-pass/Low-pass type
<a href="#">CALC:TRAN:TIME:CEN</a>		Time domain center

Command		Description
<a href="#">CALC:TRAN:TIME:DC:VAL</a>		DC value
<a href="#">CALC:TRAN:TIME:EXTR:DC</a>		DC extrapolation ON/OFF
<a href="#">CALC:TRAN:TIME:IMP:WIDT</a>		Impulse Width
<a href="#">CALC:TRAN:TIME:KBES</a>		Kaiser-Bessel $\beta$
<a href="#">CALC:TRAN:TIME:LPFR</a>		Sets requency Low-Pass
<a href="#">CALC:TRAN:TIME:REFL:TYPE</a>		Selects One way/Round trip
<a href="#">CALC:TRAN:TIME:SPAN</a>		Time domain Span
<a href="#">CALC:TRAN:TIME:STAR</a>		Time domain Start
<a href="#">CALC:TRAN:TIME:STOP</a>		Time domain Stop
<a href="#">CALC:TRAN:TIME:STAT</a>		Time domain transformation ON/OFF
<a href="#">CALC:TRAN:TIME:STEP:RTIM</a>		Step rise time
<a href="#">CALC:TRAN:TIME:STIM</a>		Selects Impulse/Step type
<a href="#">CALC:TRAN:TIME:UNIT</a>		Time domain Unit

## CALC:CONV

### SCPI Command

CALCulate<Ch>[:SElected]:CONVersion[:STATe] {OFF|ON|0|1}

CALCulate<Ch>[:SElected]:CONVersion[:STATe]?

Or

CALCulate<Ch>:TRACe<Tr>:CONVersion[:STATe] {OFF|ON|0|1}

CALCulate<Ch>:TRACe<Tr>:CONVersion[:STATe]?

### Description

Turns the S-parameter conversion function ON/OFF.

command/query

### Target

CALCulate<Ch>[:SElected] — active trace of channel <Ch>,

Or

CALCulate<Ch>:TRACe<Tr> — trace <Tr> of channel <Ch>,

<Ch> = {[1]|2|...16}

<Tr> = {[1]|2|...16}

### Parameter

{ON|1}      ON

{OFF|0}      OFF

### Query Response

{0|1}

## Preset Value

0

## Related Commands

[CALC:CONV:FUNC](#)

## Equivalent Softkeys

Analysis > Conversion > Conversion

---

## Equivalent COM Command

SCPI.CALCulate(Ch).SELEcted.CONVersion.STATe

---

### NOTE

This command is similar to  
CALCulate<Ch>[:SELEcted]:CONVersion[:STATe] only.

---

## Syntax

Status = app.SCPI.CALCulate(Ch).SELEcted.CONVersion.STATe

app.SCPI.CALCulate(Ch).SELEcted.CONVersion.STATe = true

## Type

Boolean (read/write)

---

Back to [CALCulate](#)

## CALC:CONV:FUNC

### SCPI Command

CALCulate<Ch>[:SElected]:CONVersion:FUNCtion <char>

CALCulate<Ch>[:SElected]:CONVersion:FUNCtion?

Or

CALCulate<Ch>:TRACe<Tr>:CONVersion:FUNCtion <char>

CALCulate<Ch>:TRACe<Tr>:CONVersion:FUNCtion?

### Description

Sets or reads out the S-parameter conversion function type.

command/query

### Target

CALCulate<Ch>[:SElected] — active trace of channel <Ch>,

Or

CALCulate<Ch>:TRACe<Tr> — trace <Tr> of channel <Ch>,

<Ch> = {[1]|2|...16}

<Tr> = {[1]|2|...16}

### Parameter

<char> Specifies parameter:

<b>ZREFlection</b>	Reflection equivalent impedance
<b>ZTRansmit</b>	Transmission equivalent impedance
<b>YREFlection</b>	Reflection equivalent admittance
<b>YTRansmit</b>	Transmission equivalent admittance



<b>INVersion</b>	Inverse S-parameter
<b>ZTSHunt</b>	Shunt equivalent impedance
<b>YTSHunt</b>	Shunt equivalent admittance
<b>CONJugation</b>	S-parameter conjugate

## Query Response

{ZREF|ZTR|YREF|YTR|INV|ZTSH|YTSH|CONJ}

## Preset Value

ZREF

## Equivalent Softkeys

Analysis > Conversion > {Zr | Zt | Yr | Yt | 1/S | Z Trans–Shunt | Y Trans–Shunt | Conjugation}

## Equivalent COM Command

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

### NOTE

This command is similar to CALCulate<Ch>[:SELEcted]:CONVersion:FUNction only.

## Syntax

Param = app.SCPI.CALCulate(Ch).SELEcted.CONVersion.FUNction

app.SCPI.CALCulate(Ch).SELEcted.CONVersion.FUNction = "ZTR"

## Type

String (read/write)

Back to [CALCulate](#)

## CALC:CORR:EDEL:DIST

### SCPI Command

CALCulate<Ch>[:SElected]:CORRection:EDELay:DISTance <numeric>

CALCulate<Ch>[:SElected]:CORRection:EDELay:DISTance?

or

CALCulate<Ch>:TRACe<Tr>:CORRection:EDELay:DISTance <numeric>

CALCulate<Ch>:TRACe<Tr>:CORRection:EDELay:DISTance?

### Description

Sets or reads out the value of the equivalent distance in the electrical delay function.

command/query

### Description

CALCulate<Ch>[:SElected] — active trace of channel <Ch>,

Or

CALCulate<Ch>:TRACe<Tr> — trace <Tr> of channel <Ch>,

<Ch> = {[1]|2|...16}

<Tr> = {[1]|2|...16}

### Parameter

<numeric> the distance value.

### Unit

**meter, feet or inches** depending on the [CALC:CORR:EDEL:DIST:UNIT](#) command.

### Out of Range

Sets the value of the limit, which is closer to the specified value.

**Query Response**

<numeric>

**Preset Value**

0

**Equivalent Softkeys**

**Scale > Electrical Delay > Distance**

---

**Equivalent COM Command**

None

---

Back to [CALCulate](#)

## CALC:CORR:EDEL:DIST:UNIT

### SCPI Command

CALCulate<Ch>[:SElected]:CORRection:EDELay:DISTanCe:UNITs <char>

CALCulate<Ch>[:SElected]:CORRection:EDELay:DISTanCe:UNITs?

or

CALCulate<Ch>:TRACe<Tr>:CORRection:EDELay:DISTanCe:UNITs <char>

CALCulate<Ch>:TRACe<Tr>:CORRection:EDELay:DISTanCe:UNITs?

### Description

Sets or reads out the distance units in the electrical delay function.

command/query

### Description

CALCulate<Ch>[:SElected] — active trace of channel <Ch>,

Or

CALCulate<Ch>:TRACe<Tr> — trace <Tr> of channel <Ch>,

<Ch> = {[1]|2|...16}

<Tr> = {[1]|2|...16}

### Parameter

<char> Specifies parameter:

**METer**       Meters

**FEET**       Feet

**INCHes**     Inches

**Query Response**

{MET|FEET|INCH}

**Preset Value**

METer

**Equivalent Softkeys**

Scale > Electrical Delay > Distance Units > {Meter | Feet | Inches}

---

**Equivalent COM Command**

None

---

Back to [CALCulate](#)

## CALC:CORR:EDEL:MED

### SCPI Command

CALCulate<Ch>[:SElected]:CORRection:EDELay:MEDia <char>

CALCulate<Ch>[:SElected]:CORRection:EDELay:MEDia?

or

CALCulate<Ch>:TRACe<Tr>:CORRection:EDELay:MEDia <char>

CALCulate<Ch>:TRACe<Tr>:CORRection:EDELay:MEDia?

### Description

Sets or reads out the type of media in the electrical delay function.

command/query

### Target

CALCulate<Ch>[:SElected] — active trace of channel <Ch>,

Or

CALCulate<Ch>:TRACe<Tr> — trace <Tr> of channel <Ch>,

<Ch> = {[1]2}...16}

<Tr> = {[1]2}...16}

### Parameter

<char> Specifies parameter:

**COAXial**          Coaxial type of media

**WAVEguide**      Waveguide type of media

**Query Response**

{COAX|WAV}

**Preset Value**

COAX

**Equivalent Softkeys**

Scale > Electrical Delay > Media > {Coax | Waveguide}

---

**Equivalent COM Command**

None

---

Back to [CALCulate](#)

## CALC:CORR:EDEL:RVEL

### SCPI Command

CALCulate<Ch>[:SElected]:CORRection:EDELay:RVELocity <numeric>

CALCulate<Ch>[:SElected]:CORRection:EDELay:RVELocity?

Or

CALCulate<Ch>:TRACe<Tr>:CORRection:EDELay:RVELocity <numeric>

CALCulate<Ch>:TRACe<Tr>:CORRection:EDELay:RVELocity?

### Description

Sets or reads out the value of the velocity factor used to calculate between delay and distance in the electrical delay function.

command/query

### Target

CALCulate<Ch>[:SElected] — active trace of channel <Ch>,

Or

CALCulate<Ch>:TRACe<Tr> — trace <Tr> of channel <Ch>,

<Ch> = {[1]|2|...16}

<Tr> = {[1]|2|...16}

### Parameter

<numeric> the velocity factor value from 0 to 1.

### Out of Range

Sets the value of the limit, which is closer to the specified value.



**Query Response**

<numeric>

**Preset Value**

0

**Equivalent Softkeys**

**Scale > Electrical Delay > Velocity Factor**

---

**Equivalent COM Command**

None

---

Back to [CALCulate](#)

## CALC:CORR:EDEL:TIME

### SCPI Command

CALCulate<Ch>[:SElected]:CORRection:EDELay:TIME <time>

CALCulate<Ch>[:SElected]:CORRection:EDELay:TIME?

Or

CALCulate<Ch>:TRACe<Tr>:CORRection:EDELay:TIME <time>

CALCulate<Ch>:TRACe<Tr>:CORRection:EDELay:TIME?

### Description

Sets or reads out the value of the electrical delay.

command/query

### Target

CALCulate<Ch>[:SElected] — active trace of channel <Ch>,

Or

CALCulate<Ch>:TRACe<Tr> — trace <Tr> of channel <Ch>,

<Ch> = {[1]|2|...16}

<Tr> = {[1]|2|...16}

### Parameter

<time> the electrical delay value from –10 to 10

### Unit

sec (second)

### Out of Range

Sets the value of the limit, which is closer to the specified value.

## Query Response

<numeric>

## Preset Value

0

## Equivalent Softkeys

Scale > Electrical Delay> Electrical Delay

---

## Equivalent COM Command

SCPI.CALCulate(Ch).SELEcted.CORRection.EDELaY.TIME

---

### NOTE

This command is similar to  
CALCulate<Ch>[:SELEcted]:CORRection:EDELaY:TIME  
only.

---

## Syntax

Value = app.SCPI.CALCulate(Ch).SELEcted.CORRection.EDELaY.TIME

app.SCPI.CALCulate(Ch).SELEcted.CORRection.EDELaY.TIME = 1e-9

## Type

Double (read/write)

---

Back to [CALCulate](#)

## CALC:CORR:EDEL:WAV:CUT

### SCPI Command

CALCulate<Ch>[:SElected]:CORRection:EDELay:WAVeguide:CUToff <numeric>

CALCulate<Ch>[:SElected]:CORRection:EDELay:WAVeguide:CUToff?

Or

CALCulate<Ch>:TRACe<Tr>:CORRection:EDELay:WAVeguide:CUToff <numeric>

CALCulate<Ch>:TRACe<Tr>:CORRection:EDELay:WAVeguide:CUToff?

### Description

Sets or reads out the value of the waveguide cutoff frequency in the electrical delay function if the type of media set to the "WAVeguide" by the command [CALC:CORR:EDEL:MED](#).

command/query

### Target

CALCulate<Ch>[:SElected] — active trace of channel <Ch>,

Or

CALCulate<Ch>:TRACe<Tr> — trace <Tr> of channel <Ch>,

<Ch> = {[1]|2|...16}

<Tr> = {[1]|2|...16}

### Parameter

<numeric> the cutoff frequency 0 to 999e9.

### Unit

Hz (hertz)

### Out of Range

Sets the value of the limit, which is closer to the specified value.

**Query Response**

<numeric>

**Preset Value**

0

**Equivalent Softkeys**

**Scale > Electrical Delay > Waveguide Cutoff**

---

**Equivalent COM Command**

None

---

Back to [CALCulate](#)

## CALC:CORR:OFFS:PHAS

### SCPI Command

CALCulate<Ch>[:SElected]:CORRection:OFFSet:PHASe <phase>

CALCulate<Ch>[:SElected]:CORRection:OFFSet:PHASe?

Or

CALCulate<Ch>:TRACe<Tr>:CORRection:OFFSet:PHASe <phase>

CALCulate<Ch>:TRACe<Tr>:CORRection:OFFSet:PHASe?

### Description

Sets or reads out the value of the phase offset.

command/query

### Target

CALCulate<Ch>[:SElected] — active trace of channel <Ch>,

Or

CALCulate<Ch>:TRACe<Tr> — trace <Tr> of channel <Ch>,

<Ch> = {[1]|2|...16}

<Tr> = {[1]|2|...16}

### Parameter

<phase> the phase offset value from –360 to 360

### Unit

° (degree)

### Out of Range

Sets the value of the limit, which is closer to the specified value.

## Query Response

<numeric>

## Preset Value

0

## Equivalent Softkeys

## Scale > Phase Offset

---

## Equivalent COM Command

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

---

### NOTE

This command is similar to  
CALCulate<Ch>[:SElected]:CORRection:OFFSet:PHASE  
only.

---

## Syntax

Value = app.SCPI.CALCulate(Ch).SElected.CORRection.OFFSet.PHASE

app.SCPI.CALCulate(Ch).SElected.CORRection.OFFSet.PHASE = 360

## Type

Double (read/write)

---

Back to [CALCulate](#)

## CALC:CORR:STAT?

### SCPI Command

CALCulate<Ch>[:SElected]:CORRection:STATus?

Or

CALCulate<Ch>:TRACe<Tr>:CORRection:STATus?

### Description

Reads out the interpolation/extrapolation status of the error correction.

query only

### Target

CALCulate<Ch>[:SElected] — active trace of channel <Ch>,

Or

CALCulate<Ch>:TRACe<Tr> — trace <Tr> of channel <Ch>,

<Ch> = {[1]|2|...16}

<Tr> = {[1]|2|...16}

### Query Response

Trace represents S-parameter:

<b>NONE</b>	Correction not applied
<b>COR</b>	Correction applied exactly
<b>C?</b>	Correction interpolated
<b>C!</b>	Correction extrapolated

Trace represents absolute parameter:

<b>NONE</b>	Correction not applied
-------------	------------------------



<b>RC</b>	Correction applied exactly
<b>RC?</b>	Correction interpolated
<b>RC!</b>	Correction extrapolated

### Equivalent Softkeys

None

---

### Equivalent COM Command

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

---

#### NOTE

This command is similar to  
CALCulate<Ch>[:SELEcted]:CORRection:STATus? only.

---

### Syntax

Param = app.SCPI.CALCulate(Ch).SELEcted.CORRection.STATus

### Type

String (read only)>

---

Back to [CALCulate](#)

## CALC:DATA:FDAT

### SCPI Command

CALCulate<Ch>[:SElected]:DATA:FDATa <numeric list>

CALCulate<Ch>[:SElected]:DATA:FDATa?

Or

CALCulate<Ch>:TRACe<Tr>:DATA:FDATa <numeric list>

CALCulate<Ch>:TRACe<Tr>:DATA:FDATa?

### Description

Reads out or writes the formatted data array.

The formatted data array is the data, whose processing is completed including the formatting as the last step. Such data represent the data trace values as they are shown on the screen.

The array size is 2N, where N is the number of measurement points.

For the n–th point, where n from 1 to N:

<numeric 2n–1> Value 1 depends on the trace format (see table below);

<numeric 2n> Value 2 depends on the trace format (see table below). Reads out or writes the formatted data array.

Trace Format	Value 1	Value 2
Log Mag	Logarithmic magnitude, dB	0
SWR	Voltage standing wave ratio	0
Phase	Phase, deg	0
Expand Phase	Expanded phase, deg	0

Trace Format	Value 1	Value 2
Group Delay	Group delay, sec	0
Lin Mag	Linear magnitude	0
Real	Real part	0
Imag	Imaginary part	0
Smith (Log/Phase)	Logarithmic magnitude, dB	Phase, deg
Smith (Lin/Phase)	Linear magnitude	Phase, deg
Smith (Real/Imag)	Real part	Imaginary part
Smith (R + jX)	Impedance (real part), Ohm	Impedance (imaginary part), Ohm
Smith (G + jB)	Admittance (real part), S	Admittance (imaginary part), S
Polar (Log/Phase)	Logarithmic magnitude, dB	Phase, deg
Polar (Lin/Phase)	Linear magnitude	Phase, deg
Polar (Real/Imag)	Real part	Imaginary part

**Note:** When data is being written it is recommended to hold the sweep before and update the screen after write.

command/query

## Target

CALCulate<Ch>[:SElected] — active trace of channel <Ch>,

Or

CALCulate<Ch>:TRACe<Tr> — trace <Tr> of channel <Ch>,

<Ch> = {[1]|2|...16}

<Tr> = {[1]|2|...16}

### Query Response

<numeric 1>, <numeric 2>, ...<numeric 2N>

### Related Commands

[CALC:FORM](#)

### Equivalent Softkeys

None

---

### Equivalent COM Command

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

SCPI.CALCulate(Ch).TRACe(Tr).DATA.FDATa

### Syntax

Data = app.SCPI.CALCulate(Ch).SElected.DATA.FDATa

app.SCPI.CALCulate(Ch).SElected.DATA.FDATa = Data

Data = app.SCPI.CALCulate(Ch).Trace(Tr).DATA.FDATa

app.SCPI.CALCulate(Ch).Trace(Tr).DATA.FDATa = Data

### Type

Variant (array of Double) (read/write)

---

Back to [CALCulate](#)

## CALC:DATA:FMEM

### SCPI Command

CALCulate<Ch>[:SElected]:DATA:FMEMory <numeric list>

CALCulate<Ch>[:SElected]:DATA:FMEMory?

Or

CALCulate<Ch>:TRACe<Tr>:DATA:FMEMory <numeric list>

CALCulate<Ch>:TRACe<Tr>:DATA:FMEMory?

### Description

Reads out or writes the formatted memory array.

The formatted memory array is the data, whose processing is completed including the formatting as the last step. Such data represent the memory trace values as they are shown on the screen.

The array size is  $2N$ , where  $N$  is the number of measurement points.

For the  $n$ -th point, where  $n$  from 1 to  $N$ :

<numeric  $2n-1$ > Value 1 depends on the trace format (see table below);

<numeric  $2n$ > Value 2 depends on the trace format (see table below).

Trace Format	Value 1	Value 2
Log Mag	Logarithmic magnitude, dB	0
SWR	Voltage standing wave ratio	0
Phase	Phase, deg	0
Expand Phase	Expanded phase, deg	0

Trace Format	Value 1	Value 2
Group Delay	Group delay, sec	0
Lin Mag	Linear magnitude	0
Real	Real part	0
Imag	Imaginary part	0
Smith (Log/Phase)	Logarithmic magnitude, dB	Phase, deg
Smith (Lin/Phase)	Linear magnitude	Phase, deg
Smith (Real/Imag)	Real part	Imaginary part
Smith (R + jX)	Impedance (real part), Ohm	Impedance (imaginary part), Ohm
Smith (G + jB)	Admittance (real part), S	Admittance (imaginary part), S
Polar (Log/Phase)	Logarithmic magnitude, dB	Phase, deg
Polar (Lin/Phase)	Linear magnitude	Phase, deg
Polar (Real/Imag)	Real part	Imaginary part

**Note:** When data is being written it is recommended to hold the sweep before and update the screen after write.

command/query

## Target

CALCulate<Ch>[:SElected] — active trace of channel <Ch>,

Or

CALCulate<Ch>:TRACe<Tr> — trace <Tr> of channel <Ch>,

<Ch> = {[1]|2|...16}

<Tr> = {[1]|2|...16}

### Query Response

<numeric 1>, <numeric 2>, ...<numeric 2N>

### Related Commands

[CALC:MATH:MEM](#)

[CALC:FORM](#)

### Equivalent Softkeys

None

---

### Equivalent COM Command

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

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

### Syntax

Data = app.SCPI.CALCulate(Ch).SElected.DATA.FMEMory

app.SCPI.CALCulate(Ch).SElected.DATA.FMEMory = Data

Data = app.SCPI.CALCulate(Ch).Trace(Tr).DATA.FMEMory

app.SCPI.CALCulate(Ch).Trace(Tr).DATA.FMEMory = Data

### Type

Variant (array of Double) (read/write)

---

Back to [CALCulate](#)

## CALC:DATA:SDAT

### SCPI Command

CALCulate<Ch>[:SElected]:DATA:SDATa <numeric list>

CALCulate<Ch>[:SElected]:DATA:SDATa?

Or

CALCulate<Ch>:TRACe<Tr>:DATA:SDATa <numeric list>

CALCulate<Ch>:TRACe<Tr>:DATA:SDATa?

### Description

Reads out or writes the corrected data array.

The corrected data array is the data, whose processing is completed excluding the formatting as the last step. Such data represent S-parameter complex values.

The array size is 2N, where N is the number of measurement points.

For the n-th point, where n from 1 to N:

<numeric 2n-1> the real part of corrected measurement;

<numeric 2n> the imaginary part of corrected measurement.

**Note:** When data is being written it is recommended to hold the sweep before and update the screen after write.

command/query

### Target

CALCulate<Ch>[:SElected] — active trace of channel <Ch>,

Or

CALCulate<Ch>:TRACe<Tr> — trace <Tr> of channel <Ch>,

<Ch> = {[1]|2|...16}

<Tr> = {[1]|2|...16}



### Query Response

<numeric 1>, <numeric 2>, ...<numeric 2N>

### Equivalent Softkeys

None

---

### Equivalent COM Command

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

SCPI.CALCulate(Ch).TRACe(Tr).DATA.SDATa

### Syntax

Data = app.SCPI.CALCulate(Ch).SElected.DATA.SDATa

app.SCPI.CALCulate(Ch).SElected.DATA.SDATa = Data

Data = app.SCPI.CALCulate(Ch).Trace(Tr).DATA.SDATa

app.SCPI.CALCulate(Ch).Trace(Tr).DATA.SDATa = Data

### Type

Variant (array of Double) (read/write)

---

Back to [CALCulate](#)

## CALC:DATA:SMEM

SCPI Command

CALCulate<Ch>[:SElected]:DATA:SMEMory <numeric list>

CALCulate<Ch>[:SElected]:DATA:SMEMory?

Or

CALCulate<Ch>:TRACe<Tr>:DATA:SMEMory <numeric list>

CALCulate<Ch>:TRACe<Tr>:DATA:SMEMory?

### Description

Reads out or writes the corrected memory array.

The corrected memory array is the data, whose processing is completed excluding the formatting as the last step. Such data represent S-parameter complex values.

The array size is  $2N$ , where  $N$  is the number of measurement points.

For the  $n$ -th point, where  $n$  from 1 to  $N$ :

<numeric  $2n-1$ > the real part of corrected measurement memory;

<numeric  $2n$ > the imaginary part of corrected measurement memory.

command/query

### Target

CALCulate<Ch>[:SElected] — active trace of channel <Ch>,

Or

CALCulate<Ch>:TRACe<Tr> — trace <Tr> of channel <Ch>,

<Ch> = {[1]|2|...16}

<Tr> = {[1]|2|...16}

### Query Response

<numeric 1>, <numeric 2>, ...<numeric 2N>

### Equivalent Softkeys

None

---

### Equivalent COM Command

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

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

### Syntax

Data = app.SCPI.CALCulate(Ch).SElected.DATA.SMEMory

app.SCPI.CALCulate(Ch).SElected.DATA.SMEMory = Data

Data = app.SCPI.CALCulate(Ch).Trace(Tr).DATA.SMEMory

app.SCPI.CALCulate(Ch).Trace(Tr).DATA.SMEMory = Data

### Type

Variant (array of Double) (read/write)

---

Back to [CALCulate](#)

## CALC:DATA:XAX?

### SCPI Command

CALCulate<Ch>[:SElected]:DATA:XAXis?

Or

CALCulate<Ch>:TRACe<Tr>:DATA:XAXis?

### Description

Reads out the X-axis values array.

The X-axis values array is the frequency, power or time values array depending on the trace setup. The array contains real values.

The array size is N, where N is the number of measurement points.

For the n–th point, where n from 1 to N:

<numeric n> the X-axis value.

query only

### Target

CALCulate<Ch>[:SElected] — active trace of channel <Ch>,

Or

CALCulate<Ch>:TRACe<Tr> — trace <Tr> of channel <Ch>,

<Ch> = {[1]|2|...16}

<Tr> = {[1]|2|...16}

### Query Response

<numeric 1>, <numeric 2>, ...<numeric N>

## Related Commands

[SENS:SWE:TYPE](#)

[CALC:TRAN:TIME:STAT](#)

## Equivalent Softkeys

None

---

## Equivalent COM Command

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

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

## Syntax

Data = app.SCPI.CALCulate(Ch).SElected.DATA.XAXis

Data = app.SCPI.CALCulate(Ch).Trace(Tr).DATA.XAXis

## Type

Variant (array of Double) (read/write)

---

Back to [CALCulate](#)

## CALC:FILT:TIME

### SCPI Command

CALCulate<Ch>[:SElected]:FILTer[:GATE]:TIME[:TYPE] <char>

CALCulate<Ch>[:SElected]:FILTer[:GATE]:TIME[:TYPE]?

Or

CALCulate<Ch>:TRACe<Tr>:FILTer[:GATE]:TIME[:TYPE] <char>

CALCulate<Ch>:TRACe<Tr>:FILTer[:GATE]:TIME[:TYPE]?

### Description

Sets or reads out the gate type of the gating function.

command/query

### Target

CALCulate<Ch>[:SElected] — active trace of channel <Ch>,

Or

CALCulate<Ch>:TRACe<Tr> — trace <Tr> of channel <Ch>,

<Ch> = {[1]|2|...16}

<Tr> = {[1]|2|...16}

### Parameter

<char> Specifies the gate type:

**BPASs**      Bandpass type

**NOTCh**      Notch type

### Query Response

{BPAS|NOTC}

## Preset Value

BPAS

## Equivalent Softkeys

Analysis > Gating > Type

---

## Equivalent COM Command

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

---

### NOTE

This command is similar to  
CALCulate<Ch>[:SElected]:FILTer[:GATE]:TIME[:TYPE]  
only.

---

## Syntax

Param = app.SCPI.CALCulate(Ch).SElected.FILTer.GATE.TIME.TYPE

app.SCPI.CALCulate(Ch).SElected.FILTer.GATE.TIME.TYPE = "bpas"

## Type

String (read/write)

---

Back to [CALCulate](#)

## CALC:FILT:TIME:CEN

### SCPI Command

CALCulate<Ch>[:SElected]:FILTer[:GATE]:TIME:CENTer <time>

CALCulate<Ch>[:SElected]:FILTer[:GATE]:TIME:CENTer?

Or

CALCulate<Ch>:TRACe<Tr>:FILTer[:GATE]:TIME:CENTer <time>

CALCulate<Ch>:TRACe<Tr>:FILTer[:GATE]:TIME:CENTer?

### Description

Sets or reads out the gate center value of the gating function.

command/query

### Target

CALCulate<Ch>[:SElected] — active trace of channel <Ch>,

Or

CALCulate<Ch>:TRACe<Tr> — trace <Tr> of channel <Ch>,

<Ch> = {[1]|2|...16}

<Tr> = {[1]|2|...16}

### Parameter

<time> the center value of the gate, the range varies depending on the frequency span and the number of points

### Unit

sec (second)

### Out of Range

Sets the value of the limit, which is closer to the specified value.



## Query Response

<numeric>

## Preset Value

0

## Equivalent Softkeys

**Analysis > Gating > Center**

---

## Equivalent COM Command

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

---

### NOTE

This command is similar to CALCulate<Ch>[:SELEcted]:FILTer[:GATE]:TIME:CENTer only.

---

## Syntax

Value = app.SCPI.CALCulate(Ch).SELEcted.FILTer.GATE.TIME.CENTer

app.SCPI.CALCulate(Ch).SELEcted.FILTer.GATE.TIME.CENTer = 1e-8

## Type

Double (read/write)

---

Back to [CALCulate](#)

## CALC:FILT:TIME:SHAP

### SCPI Command

CALCulate<Ch>[:SElected]:FILTer[:GATE]:TIME:SHAPE <char>

CALCulate<Ch>[:SElected]:FILTer[:GATE]:TIME:SHAPE?

Or

CALCulate<Ch>:TRACe<Tr>:FILTer[:GATE]:TIME:SHAPE <char>

CALCulate<Ch>:TRACe<Tr>:FILTer[:GATE]:TIME:SHAPE?

### Description

Sets or reads out the gate shape of the gating function.

command/query

### Target

CALCulate<Ch>[:SElected] — active trace of channel <Ch>,

Or

CALCulate<Ch>:TRACe<Tr> — trace <Tr> of channel <Ch>,

<Ch> = {[1]|2|...16}

<Tr> = {[1]|2|...16}

### Parameter

<char> Specifies the gate shape:

<b>MAXimum</b>	Maximum shape
<b>WIDE</b>	Wide shape
<b>NORMal</b>	Normal shape
<b>MINimum</b>	Minimum shape

## Query Response

{MAX|WIDE|NORM|MIN}

## Preset Value

NORM

## Equivalent Softkeys

Analysis > Gating > Shape > {Maximum | Wide | Normal | Minimum}

---

## Equivalent COM Command

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

---

### NOTE

This command is similar to CALCulate<Ch>[:SELEcted]:FILTer[:GATE]:TIME:SHAPe only.

---

## Syntax

Param = app.SCPI.CALCulate(Ch).SELEcted.FILTer.GATE.TIME.SHAPe

app.SCPI.CALCulate(Ch).SELEcted.FILTer.GATE.TIME.SHAPe = "MAX"

## Type

String (read/write)

---

Back to [CALCulate](#)

## CALC:FILT:TIME:SPAN

### SCPI Command

CALCulate<Ch>[:SElected]:FILTer[:GATE]:TIME:SPAN <time>

CALCulate<Ch>[:SElected]:FILTer[:GATE]:TIME:SPAN?

Or

CALCulate<Ch>:TRACe<Tr>:FILTer[:GATE]:TIME:SPAN <time>

CALCulate<Ch>:TRACe<Tr>:FILTer[:GATE]:TIME:SPAN?

### Description

Sets or reads out the gate span value of the gating function.

command/query

### Target

CALCulate<Ch>[:SElected] — active trace of channel <Ch>,

Or

CALCulate<Ch>:TRACe<Tr> — trace <Tr> of channel <Ch>,

<Ch> = {[1]|2|...16}

<Tr> = {[1]|2|...16}

### Parameter

<time> the span value of the gate, the range varies depending on the frequency span and the number of points

### Unit

sec (second)

### Out of Range

Sets the value of the limit, which is closer to the specified value.

## Query Response

<numeric>

## Preset Value

2e−8

## Equivalent Softkeys

Analysis > Gating > Span

---

## Equivalent COM Command

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

---

### NOTE

This command is similar to  
CALCulate<Ch>[:SELEcted]:FILTer[:GATE]:TIME:SPAN  
only.

---

## Syntax

Value = app.SCPI.CALCulate(Ch).SELEcted.FILTer.GATE.TIME.SPAN

app.SCPI.CALCulate(Ch).SELEcted.FILTer.GATE.TIME.SPAN = 1e−8

## Type

Double (read/write)

---

Back to [CALCulate](#)

## CALC:FILT:TIME:STAR

### SCPI Command

CALCulate<Ch>[:SElected]:FILTer[:GATE]:TIME:STARt <time>

CALCulate<Ch>[:SElected]:FILTer[:GATE]:TIME:STARt?

Or

CALCulate<Ch>:TRACe<Tr>:FILTer[:GATE]:TIME:STARt <time>

CALCulate<Ch>:TRACe<Tr>:FILTer[:GATE]:TIME:STARt?

### Description

Sets or reads out the gate start value of the gating function.

command/query

### Target

CALCulate<Ch>[:SElected] — active trace of channel <Ch>,

Or

CALCulate<Ch>:TRACe<Tr> — trace <Tr> of channel <Ch>,

<Ch> = {[1]|2|...16}

<Tr> = {[1]|2|...16}

### Parameter

<time> the start value of the gate, the range varies depending on the frequency span and the number of points

### Unit

sec (second)

### Out of Range

Sets the value of the limit, which is closer to the specified value.

## Query Response

<numeric>

## Preset Value

-1e-8

## Equivalent Softkeys

Analysis > Gating > Start

---

## Equivalent COM Command

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

---

### NOTE

This command is similar to  
CALCulate<Ch>[:SElected]:FILTer[:GATE]:TIME:START  
only.

---

## Syntax

Value = app.SCPI.CALCulate(Ch).SElected.FILTer.GATE.TIME.STARTt

app.SCPI.CALCulate(Ch).SElected.FILTer.GATE.TIME.STARTt = 1e-8

## Type

Double (read/write)

---

Back to [CALCulate](#)

## CALC:FILT:TIME:STAT

### SCPI Command

CALCulate<Ch>[:SElected]:FILTer[:GATE]:TIME:STATe {OFF|ON|0|1}

CALCulate<Ch>[:SElected]:FILTer[:GATE]:TIME:STATe?

Or

CALCulate<Ch>:TRACe<Tr>:FILTer[:GATE]:TIME:STATe {OFF|ON|0|1}

CALCulate<Ch>:TRACe<Tr>:FILTer[:GATE]:TIME:STATe?

### Description

Turns the gating function ON/OFF.

command/query

### Target

CALCulate<Ch>[:SElected] — active trace of channel <Ch>,

Or

CALCulate<Ch>:TRACe<Tr> — trace <Tr> of channel <Ch>,

<Ch> = {[1]|2|...16}

<Tr> = {[1]|2|...16}

### Parameter

{ON|1}      ON

{OFF|0}     OFF

### Query Response

{0|1}



## Preset Value

0

## Equivalent Softkeys

Analysis > Gating > Gating

---

## Equivalent COM Command

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

---

### NOTE

This command is similar to  
CALCulate<Ch>[:SElected]:FILTer[:GATE]:TIME:STATe  
only.

---

## Syntax

Status = app.SCPI.CALCulate(Ch).SElected.FILTer.GATE.TIME.STATe

app.SCPI.CALCulate(Ch).SElected.FILTer.GATE.TIME.STATe = Status

## Type

Boolean (read/write)

---

Back to [CALCulate](#)

## CALC:FILT:TIME:STOP

### SCPI Command

CALCulate<Ch>[:SElected]:FILTer[:GATE]:TIME:STOP <time>

CALCulate<Ch>[:SElected]:FILTer[:GATE]:TIME:STOP?

Or

CALCulate<Ch>:TRACe<Tr>:FILTer[:GATE]:TIME:STOP <time>

CALCulate<Ch>:TRACe<Tr>:FILTer[:GATE]:TIME:STOP?

### Description

Sets or reads out the gate stop value of the gating function.

command/query

### Target

CALCulate<Ch>[:SElected] — active trace of channel <Ch>,

Or

CALCulate<Ch>:TRACe<Tr> — trace <Tr> of channel <Ch>,

<Ch> = {[1]2|...16}

<Tr> = {[1]2|...16}

### Parameter

<time> the stop value of the gate, the range varies depending on the frequency span and the number of points

### Unit

sec (second)

### Out of Range

Sets the value of the limit, which is closer to the specified value.

## Query Response

<numeric>

## Preset Value

+1e−8

## Equivalent Softkeys

**Analysis > Gating > Stop**

---

## Equivalent COM Command

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

---

### NOTE

This command is similar to  
CALCulate<Ch>[:SELEcted]:FILTer[:GATE]:TIME:STOP  
only

---

## Syntax

Value = app.SCPI.CALCulate(Ch).SELEcted.FILTer.GATE.TIME.STOP

app.SCPI.CALCulate(Ch).SELEcted.FILTer.GATE.TIME.STOP = 1e−7

## Type

Double (read/write)

---

Back to [CALCulate](#)

## CALC:FORM

### SCPI Command

CALCulate<Ch>[:SElected]:FORMat <char>

CALCulate<Ch>[:SElected]:FORMat?

Or

CALCulate<Ch>:TRACe<Tr>:FORMat <char>

CALCulate<Ch>:TRACe<Tr>:FORMat?

### Description

Sets or reads out the trace format.

command/query

### Target

CALCulate<Ch>[:SElected] — active trace of channel <Ch>,

Or

CALCulate<Ch>:TRACe<Tr> — trace <Tr> of channel <Ch>,

<Ch> = {[1]|2|...16}

<Tr> = {[1]|2|...16}

### Parameter

<char> Specifies the trace format:

<b>MLOGarithmic</b>	Logarithmic magnitude
<b>PHASe</b>	Phase
<b>GDELay</b>	Group delay time
<b>SLINear</b>	Smith chart format (Lin)

<b>SLOGarithmic</b>	Smith chart format (Log)
<b>SCOMplex</b>	Smith chart format (Real/Imag)
<b>SMITh</b>	Smith chart format ( $R + jX$ )
<b>SADMittance</b>	Smith chart format ( $G + jB$ )
<b>PLINear</b>	Polar format (Lin)
<b>PLOGarithmic</b>	Polar format (Log)
<b>POLar</b>	Polar format (Real/Imag)
<b>MLINear</b>	Linear magnitude
<b>SWR</b>	Voltage standing wave ratio
<b>REAL</b>	Real part
<b>IMAGinary</b>	Imaginary part
<b>UPHase</b>	Expanded phase

### Query Response

{MLOG|PHAS|GDEL|SLIN|SLOG|SCOM|SMIT|SADM|PLIN|PLOG|POL|MLIN|SWR|  
REAL|IMAG|UPH}

### Preset Value

MLOG

## Equivalent Softkeys

Format > {Log Mag | Phase | Group Delay | Lin Mag | SWR | Real | Imag | Phase > 180}

Format > Smith > {Log/Phase | Lin/Phase | Real/Imag | R+jX | G+jB}

Format > Polar > {Log/Phase | Ling/Phase | Real/Imag}

---

## Equivalent COM Command

SCPI.CALCulate(Ch).SElected.FORMat

---

### NOTE

This command is similar to  
CALCulate<Ch>[:SElected]:FORMat only.

---

## Syntax

Param = app.SCPI.CALCulate(Ch).SElected.FORMat

app.SCPI.CALCulate(Ch).SElected.FORMat = "PHAS"

## Type

String (read/write)

---

Back to [CALCulate](#)

## CALC:FSIM:SEND:DEEM:STAT

### SCPI Command

CALCulate<Ch>:FSIMulator:SENDEd:DEEMbed:STATe {OFF|ON|0|1}

CALCulate<Ch>:FSIMulator:SENDEd:DEEMbed:STATe?

### Description

Turns the 2-port network de-embedding function ON/OFF.

command/query

### Target

The channel <Ch>={1|2|...16}

### Parameter

{ON|1}      ON

{OFF|0}     OFF

### Query Response

{0|1}

### Preset Value

0

### Equivalent Softkeys

Analysis > Fixture Simulator > De-Embedding > De-Embedding

---

### Equivalent COM Command

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

### Syntax

Status = app.SCPI.CALCulate(Ch).FSIMulator.SENDEd.DEEMbed.STATe

app.SCPI.CALCulate(Ch).FSIMulator.SENDEd.DEEMbed.STATe = True

**Type**

Boolean (read/write)

---

Back to [CALCulate](#)



## CALC:FSIM:SEND:DEEM:PORT:STAT

### SCPI Command

CALCulate<Ch>:FSIMulator:SENDEd:DEEMbed:PORT<Pt>:STATe {OFF|ON|0|1}

CALCulate<Ch>:FSIMulator:SENDEd:DEEMbed:PORT<Pt>:STATe?

### Description

Turns the 2-port network de-embedding function for specified port ON/OFF.

command/query

### Target

Port <Pt> of channel <Ch>,

<Ch>={[1]|2|...16}

<Pt>={[1]|2}

### Parameter

{ON|1}      ON

{OFF|0}     OFF

### Query Response

{0|1}

### Preset Value

0

### Equivalent Softkeys

Analysis > Fixture Simulator > De-Embedding > Port n {ON/OFF}

---

### Equivalent COM Command

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

## Syntax

Status = app.SCPI.CALCulate(Ch).FSIMulator.SENDEd.DEEMbed.STATe

app.SCPI.CALCulate(Ch).FSIMulator.SENDEd.DEEMbed.STATe = True

## Type

Boolean (read/write)

---

Back to [CALCulate](#)

## CALC:FSIM:SEND:DEEM:PORT:USER:FILE

### SCPI Command

CALCulate<Ch>:FSIMulator:SENDEd:DEEMbed:PORT<Pt>:USER:FILEname  
<string>

CALCulate<Ch>:FSIMulator:SENDEd:DEEMbed:PORT<Pt>:USER:FILEname?

### Description

Sets or reads out the name of the \*.S2P file of the de-embedded circuit of the 2-port network de-embedding function. The file contains the circuit S-parameters in Touchstone format.

**Note:** If the full path of the file is not specified, the \FixtureSim subdirectory of the application directory will be searched for the file.

command/query

### Target

Port <Pt> of channel <Ch>,

<Ch>={[1]|2|...16}

<Pt>={[1]|2}

### Parameter

<string>, up to 256 characters

### Equivalent Softkeys

**Analysis > Fixture Simulator > De-Embedding > S-parameters File**

---

## Equivalent COM Command

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

## Syntax

File =  
app.SCPI.CALCulate(Ch).FSIMulator.SENDEd.DEEMbed.PORT(Pt).USER.FILename

app.SCPI.CALCulate(Ch).FSIMulator.SENDEd.DEEMbed.PORT(Pt).USER.FILename = "network.S2P"

## Type

String (read/write)

---

Back to [CALCulate](#)

## CALC:FSIM:SEND:PMC:STAT

### SCPI Command

CALCulate<Ch>:FSIMulator:SENDEd:PMCircuit:STATe {OFF|ON|0|1}

CALCulate<Ch>:FSIMulator:SENDEd:PMCircuit:STATe?

### Description

Turns the 2-port network embedding function ON/OFF.

command/query

### Target

The channel <Ch>={1|2|...16}

### Parameter

{ON|1}      ON

{OFF|0}     OFF

### Query Response

{0|1}

### Preset Value

0

### Equivalent Softkeys

Analysis > Fixture Simulator > Embedding > Embedding {ON/OFF}

---

### Equivalent COM Command

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

## Syntax

Status =  
app.SCPI.CALCulate(Ch).FSIMulator.SENDEd.DEEMbed.PORT(Pt).STATE

app.SCPI.CALCulate(Ch).FSIMulator.SENDEd.DEEMbed.PORT(Pt).STATE = True

## Type

Boolean (read/write)

---

Back to [CALCulate](#)

## CALC:FSIM:SEND:PMC:PORT:STAT

### SCPI Command

CALCulate<Ch>:FSIMulator:SENDEd:PMCircuit:PORT<Pt>:STATe {OFF|ON|0|1}

CALCulate<Ch>:FSIMulator:SENDEd:PMCircuit:PORT<Pt>:STATe?

### Description

Turns the 2-port network embedding function for each port ON/OFF.

command/query

### Target

Port <Pt> of channel <Ch>,

<Ch>={{1}|2|...16}

<Pt>={{1}|2}

### Parameter

{ON|1}      ON

{OFF|0}     OFF

### Query Response

{0|1}

### Preset Value

0

### Equivalent Softkeys

Analysis > Fixture Simulator > Embedding > Port n {ON/OFF}

---

### Equivalent COM Command

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

## Syntax

Status =  
app.SCPI.CALCulate(Ch).FSIMulator.SENDEd.DEEMbed.PORT(Pt).STATE

app.SCPI.CALCulate(Ch).FSIMulator.SENDEd.DEEMbed.PORT(Pt).STATE = True

## Type

Boolean (read/write)

---

Back to [CALCulate](#)



## CALC:FSIM:SEND:PMC:PORT:USER:FILE

### SCPI Command

CALCulate<Ch>:FSIMulator:SENDEd:PMCircuit:PORT<Pt>:USER:FILEname  
<string>

CALCulate<Ch>:FSIMulator:SENDEd:PMCircuit:PORT<Pt>:USER:FILEname?

### Description

Sets or reads out the name of the \*.S2P file of the embedded circuit of the 2-port network embedding function. The file contains the circuit S-parameters in Touchstone format.

**Note:** If the full path of the file is not specified, the \FixtureSim subdirectory of the application directory will be searched for the file.

command/query

### Target

Port <Pt> of channel <Ch>,

<Ch>={{1}|2|...16}

<Pt>={{1}|2}

### Parameter

<string>, up to 256 characters

### Equivalent Softkeys

**Analysis > Fixture Simulator > Embedding > S-parameters File**

---

## Equivalent COM Command

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

## Syntax

File =  
app.SCPI.CALCulate(Ch).FSIMulator.SENDEd.PMCircuit.PORT(Pt).USER.FILenam  
e

app.SCPI.CALCulate(Ch).FSIMulator.SENDEd.PMCircuit.PORT(Pt).USER.FILenam  
e = "network.S2P"

## Type

String (read/write)

---

Back to [CALCulate](#)

## CALC:FSIM:SEND:ZCON:PORT:Z0

### SCPI Command

CALCulate<Ch>:FSIMulator:SENDEd:ZCONversion:PORT<Pt>:Z0[:R] <numeric>

CALCulate<Ch>:FSIMulator:SENDEd:ZCONversion:PORT<Pt>:Z0[:R]?

### Description

Sets or reads out the value of the impedance of the port impedance conversion function. The function sets the real part and zeros the imaginary part of the port impedance.

command/query

### Target

Port <Pt> of channel <Ch>,

<Ch>={[1]|2|...16}

<Pt>={[1]|2}

### Parameter

<numeric> the impedance value from 1e–6 to 1e10

### Unit

Ω (Ohm)

### Out of Range

Sets the value of the limit, which is closer to the specified value.

### Query Response

<numeric>

### Preset Value

50 Ω

## Equivalent Softkeys

**Analysis > Fixture Simulator > Port Z Conversion > Port n Z0 Real**

**Analysis > Fixture Simulator > Port Z Conversion > Port n Z0 Imag**

---

## Equivalent COM Command

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

## Syntax

Value =  
app.SCPI.CALCulate(Ch).FSIMulator.SENDEd.ZCONversion.PORT(Pt).Z0.R

app.SCPI.CALCulate(Ch).FSIMulator.SENDEd.ZCONversion.PORT(Pt).Z0.R = 50

## Type

Double (read/write)

---

Back to [CALCulate](#)

## **CALC:FSIM:SEND:ZCON:PORT:Z0:REAL**

### **SCPI Command**

CALCulate<Ch>:FSIMulator:SENDEd:ZCONversion:PORT<Pt>:Z0:REAL <numeric>

CALCulate<Ch>:FSIMulator:SENDEd:ZCONversion:PORT<Pt>:Z0:REAL?

### **Description**

Sets or reads out the real part of the impedance of the port impedance conversion function.

command/query

### **Target**

Port <Pt> of channel <Ch>,

<Ch>={[1]|2|...16}

<Pt>={[1]|2}

### **Parameter**

<numeric> the impedance value from 1e-6 to 1e10

### **Unit**

$\Omega$  (Ohm)

### **Out of Range**

Sets the value of the limit, which is closer to the specified value.

### **Query Response**

<numeric>

### **Preset Value**

50  $\Omega$

## Equivalent Softkeys

Analysis > Fixture Simulator > Port Z Conversion > Port n Z0 Real

---

## Equivalent COM Command

None

---

Back to [CALCulate](#)

## CALC:FSIM:SEND:ZCON:PORT:Z0:IMAG

### SCPI Command

CALCulate<Ch>:FSIMulator:SENDEd:ZCONversion:PORT<Pt>:Z0:IMAGinary  
<numeric>

CALCulate<Ch>:FSIMulator:SENDEd:ZCONversion:PORT<Pt>:Z0:IMAGinary?

### Description

Sets or reads out the imaginary part of the impedance of the port impedance conversion function.

command/query

### Target

Port <Pt> of channel <Ch>,

<Ch>={[1]|2|...16}

<Pt>={[1]|2}

### Parameter

<numeric> the impedance value from 1e-6 to 1e10

### Unit

$\Omega$  (Ohm)

### Out of Range

Sets the value of the limit, which is closer to the specified value.

### Query Response

<numeric>

### Preset Value

0

## Equivalent Softkeys

Analysis > Fixture Simulator > Port Z Conversion > Port n Z0 Imag

---

## Equivalent COM Command

None

---

Back to [CALCulate](#)



## CALC:FSIM:SEND:ZCON:STAT

### SCPI Command

CALCulate<Ch>:FSIMulator:SENDEd:ZCONversion:STATe {OFF|ON|0|1}

CALCulate<Ch>:FSIMulator:SENDEd:ZCONversion:STATe?

### Description

Turns the port impedance conversion function ON/OFF.

command/query

### Target

The channel <Ch>={1|2|...16}

### Parameter

{ON|1}      ON

{OFF|0}     OFF

### Query Response

{0|1}

### Preset Value

0

### Equivalent Softkeys

Analysis > Fixture Simulator > Port Z Conversion > Port Z Conversion  
{ON/OFF}

---

### Equivalent COM Command

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

## Syntax

Status = app.SCPI.CALCulate(Ch).FSIMulator.SENDEd.ZCONversion.STATe

app.SCPI.CALCulate(Ch).FSIMulator.SENDEd.ZCONversion.STATe = True

## Type

Boolean (read/write)

---

Back to [CALCulate](#)

# CALC:FSIM:SEND:ZCON:THE

## SCPI Command

CALCulate<Ch>:FSIMulator:SENDEd:ZCONversion:THEory {TRAVelling | POWer}

CALCulate<Ch>:FSIMulator:SENDEd:ZCONversion:THEory?

## Description

Selects the theory of the S-parameters Re-normalization (Port Z Conversion).

command/query

## Target

The channel <Ch>={1|2|...16}

## Parameter

<char> Specifies Theory:

<b>TRAVelling</b>	The travelling waves theory
<b>POWer</b>	The power waves theory

## Query Response

{TRAV|POV}

## Preset Value

TRAV

## Equivalent Softkeys

Analysis > Fixture Simulator >Port ZConversion > TTheory > {Travelling Waves  
| Power Waves}

---

### Equivalent COM Command

None

---

Back to [CALCulate](#)

## CALC:FSIM:STAT

### SCPI Command

CALCulate<Ch>:FSIMulator:STATe {OFF|ON|0|1}

CALCulate<Ch>:FSIMulator:STATe?

### Description

Turns the fixture simulation function ON/OFF.

command/query

### Target

The channel <Ch>={1|2|...16}

### Parameter

{ON|1}      ON

{OFF|0}      OFF

### Query Response

{0|1}

### Preset Value

0

### Equivalent Softkeys

Analysis > Fixture Simulator > Fixture Simulator {ON/OFF}

---

### Equivalent COM Command

SCPI.CALCulate(Ch).FSIMulator.STATe

### Syntax

Status = app.SCPI.CALCulate(Ch).FSIMulator.STATe

app.SCPI.CALCulate(Ch).FSIMulator.STATe = True

**Type**

Boolean (read/write)

---

Back to [CALCulate](#)

## CALC:FUNC:DATA?

### SCPI Command

CALCulate<Ch>[:SElected]:FUNCtion:DATA?

Or

CALCulate<Ch>:TRACe<Tr>:FUNCtion:DATA?

### Description

Reads out the data array, which is the [CALC:FUNC:EXEC](#) command analysis result.

The array size is 2N, where N is the number of points.

For the n-th point, where n from 1 to N:

<numeric 2n-1> the response value in n-th measurement point;

<numeric 2n> the stimulus value in n-th measurement point. Always set to 0 for the analysis of mean value, standard deviation, and peak-to-peak value.

query only

### Target

CALCulate<Ch>[:SElected] — active trace of channel <Ch>,

Or

CALCulate<Ch>:TRACe<Tr> — trace <Tr> of channel <Ch>,

<Ch> = {[1]2|...16}

<Tr> = {[1]2|...16}

### Query Response

<numeric 1>, <numeric 2>, ...<numeric 2N>

## Related Commands

[CALC:FUNC:EXEC](#)

[CALC:FUNC:POIN?](#)

## Equivalent Softkeys

None

---

## Equivalent COM Command

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

## Syntax

Data = app.SCPI.CALCulate(Ch).SElected.FUNCtion.DATA

## Type

Variant (array of Double) (read only)

---

Back to [CALCulate](#)



## CALC:FUNC:DOM

### SCPI Command

CALCulate<Ch>[:SElected]:FUNCtion:DOMain[:STATe] {OFF|ON|0|1}

CALCulate<Ch>[:SElected]:FUNCtion:DOMain[:STATe]?

Or

CALCulate<Ch>:TRACe<Tr>:FUNCtion:DOMain[:STATe] {OFF|ON|0|1}

CALCulate<Ch>:TRACe<Tr>:FUNCtion:DOMain[:STATe]?

### Description

Specifies whether an arbitrary range or the entire sweep range is used when the [CALC:FUNC:EXEC](#) command is executed.

command/query

### Target

All traces of channel <Ch> (if the coupling is set to ON by the [CALC:FUNC:DOM:COUP](#) command),

Or

CALCulate<Ch>[:SElected] — active trace of channel <Ch>,

Or

CALCulate<Ch>:TRACe<Tr> — trace <Tr> of channel <Ch>,

<Ch> = {[1]|2|...16}

<Tr> = {[1]|2|...16}

## Parameter

Select the following:

- {ON|1}**      Arbitrary range
- {OFF|0}**    Entire sweep range

## Query Response

{0|1}

## Preset Value

0

## Related Commands

[CALC:FUNC:EXEC](#)

[CALC:FUNC:DOM:COUP](#)

## Equivalent Softkeys

None

---

## Equivalent COM Command

SCPI.CALCulate(Ch).SELEcted.FUNCtion.DOMain.STATe

---

### NOTE

This command is similar to CALCulate<Ch>[:SELEcted]:FUNCtion:DOMain[:STATe] only.

---

## Syntax

Status = app.SCPI.CALCulate(Ch).SELEcted.FUNCtion.DOMain.STATe

app.SCPI.CALCulate(Ch).SELEcted.FUNCtion.DOMain.STATe = true

## Type

Boolean (read/write)

---

Back to [CALCulate](#)

## CALC:FUNC:DOM:COUP

### SCPI Command

CALCulate<Ch>[:SElected]:FUNCtion:DOMain:COUPle {OFF|ON|0|1}

CALCulate<Ch>[:SElected]:FUNCtion:DOMain:COUPle?

Or

CALCulate<Ch>:TRACe<Tr>:FUNCtion:DOMain:COUPle {OFF|ON|0|1}

CALCulate<Ch>:TRACe<Tr>:FUNCtion:DOMain:COUPle?

### Description

If the arbitrary range is turned ON by the [CALC:FUNC:DOM](#) command, specifies whether all traces of the channel use the same range (coupling) or if each trace uses an individual range when the [CALC:FUNC:EXEC](#) command is executed.

command/query

### Target

All traces of channel <Ch>,

<Ch> = {[1]|2|...16}

### Parameter

{ON|1}      ON

{OFF|0}      OFF

### Query Response

{0|1}

### Preset Value

0

## Related Commands

[CALC:FUNC:EXEC](#)

## Equivalent Softkeys

None

---

## Equivalent COM Command

SCPI.CALCulate(Ch).SELEcted.FUNCtion.DOMain.COUPle

---

### NOTE

This command is similar to CALCulate<Ch>[:SELEcted]:FUNCtion:DOMain:COUPle only.

---

## Syntax

Status = app.SCPI.CALCulate(Ch).SELEcted.FUNCtion.DOMain.COUPle

app.SCPI.CALCulate(Ch).SELEcted.FUNCtion.DOMain.COUPle = Status

## Type

Boolean (read/write)

---

Back to [CALCulate](#)

## CALC:FUNC:DOM:STAR

### SCPI Command

CALCulate<Ch>[:SElected]:FUNCtion:DOMain:STARt <stimulus>

CALCulate<Ch>[:SElected]:FUNCtion:DOMain:STARt?

Or

CALCulate<Ch>:TRACe<Tr>:FUNCtion:DOMain:STARt <stimulus>

CALCulate<Ch>:TRACe<Tr>:FUNCtion:DOMain:STARt?

### Description

Sets the start value of the analysis range of the [CALC:FUNC:EXEC](#) command.

command/query

### Target

All traces of channel <Ch> (if the coupling is set to ON by the CALC:FUNC:DOM:COUP command),

Or

CALCulate<Ch>[:SElected] — active trace of channel <Ch>,

Or

CALCulate<Ch>:TRACe<Tr> — trace <Tr> of channel <Ch>,

<Ch> = {[1]|2|...16}

<Tr> = {[1]|2|...16}

### Parameter

<stimulus> the start value of analysis range

### Unit

Hz |s |dBm

## Query Response

<numeric>

## Preset Value

0

## Related Commands

[CALC:FUNC:DOM](#)

## Equivalent Softkeys

None

---

## Equivalent COM Command

SCPI.CALCulate(Ch).SElected.FUNCtion.DOMain.STARTt

---

### NOTE

This command is similar to  
CALCulate<Ch>[:SElected]:FUNCtion:DOMAIN:START  
only.

---

## Syntax

Value = app.SCPI.CALCulate(Ch).SElected.FUNCtion.DOMain.STARTt

app.SCPI.CALCulate(Ch).SElected.FUNCtion.DOMain.STARTt = 1e9

## Type

Double (read/write)

---

Back to [CALCulate](#)

## CALC:FUNC:DOM:STOP

### SCPI Command

CALCulate<Ch>[:SElected]:FUNCtion:DOMain:STOP <stimulus>

CALCulate<Ch>[:SElected]:FUNCtion:DOMain:STOP?

Or

CALCulate<Ch>:TRACe<Tr>:FUNCtion:DOMain:STOP <stimulus>

CALCulate<Ch>:TRACe<Tr>:FUNCtion:DOMain:STOP?

### Description

Sets the stop value of the analysis range of the [CALC:FUNC:EXEC](#) command.

command/query

### Target

All traces of channel <Ch> (if the coupling is set to ON by the CALC:FUNC:DOM:COUP command),

Or

CALCulate<Ch>[:SElected] — active trace of channel <Ch>,

Or

CALCulate<Ch>:TRACe<Tr> — trace <Tr> of channel <Ch>,

<Ch> = {[1]2|...16}

<Tr> = {[1]2|...16}

### Parameter

<stimulus> the start value of analysis range

### Unit

Hz |s |dBm



## Query Response

<numeric>

## Preset Value

0

## Related Commands

[CALC:FUNC:DOM](#)

## Equivalent Softkeys

None

---

## Equivalent COM Command

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

---

### NOTE

This command is similar to  
CALCulate<Ch>[:SElected]:FUNCtion:DOMAIN:STOP  
only.

---

## Syntax

Value = app.SCPI.CALCulate(Ch).SElected.FUNCtion.DOMain.STOP

app.SCPI.CALCulate(Ch).SElected.FUNCtion.DOMain.STOP = 2e9

## Type

Double (read/write)

---

Back to [CALCulate](#)

## CALC:FUNC:EXEC

### SCPI Command

CALCulate<Ch>[:SElected]:FUNCtion:EXECute

Or

CALCulate<Ch>:TRACe<Tr>:FUNCtion:EXECute

### Description

Executes the analysis specified by the [CALC:FUNC:TYPE](#) command.

The analysis result can be read out by the [CALC:FUNC:DATA?](#) command.

no query

### Target

CALCulate<Ch>[:SElected] — active trace of channel <Ch>,

Or

CALCulate<Ch>:TRACe<Tr> — trace <Tr> of channel <Ch>,

<Ch> = {[1]|2|...16}

<Tr> = {[1]|2|...16}

### Related Commands

[CALC:FUNC:TYPE](#)

[CALC:FUNC:DATA?](#)

### Equivalent Softkeys

None

---

## Equivalent COM Command

SCPI.CALCulate(Ch).SELEcted.FUNCtion.EXECute

---

### NOTE

This command is similar to  
CALCulate<Ch>[:SELEcted]:FUNCtion:EXECute only.

---

## Syntax

app.SCPI.CALCulate(Ch).SELEcted.FUNCtion.EXECute

## Type

Method

---

Back to [CALCulate](#)

## CALC:FUNC:PEXC

### SCPI Command

CALCulate<Ch>[:SElected]:FUNCtion:PEXCursion <numeric>

CALCulate<Ch>[:SElected]:FUNCtion:PEXCursion?

Or

CALCulate<Ch>:TRACe<Tr>:FUNCtion:PEXCursion <numeric>

CALCulate<Ch>:TRACe<Tr>:FUNCtion:PEXCursion?

### Description

Sets the lower limit for the peak excursion value when executing the peak search with the [CALC:FUNC:EXEC](#) command.

command/query

### Target

CALCulate<Ch>[:SElected] — active trace of channel <Ch>,

Or

CALCulate<Ch>:TRACe<Tr> — trace <Tr> of channel <Ch>,

<Ch> = {[1]|2|...16}

<Tr> = {[1]|2|...16}

### Parameter

<numeric> the lower limit of the peak excursion value, varies depending on the data format

### Unit

dB |° |s

### Query Response

<numeric>

**Preset Value**

3.0

**Related Commands**

[CALC:FUNC:EXEC](#)

**Equivalent Softkeys**

None

---

**Equivalent COM Command**

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

---

**NOTE**

This command is similar to  
CALCulate<Ch>[:SELEcted]:FUNCtion:PEXCursion only.

---

**Syntax**

Value = app.SCPI.CALCulate(Ch).SELEcted.FUNCtion.PEXCursion

app.SCPI.CALCulate(Ch).SELEcted.FUNCtion.PEXCursion = 1.5

**Type**

Double (read/write)

---

Back to [CALCulate](#)

## CALC:FUNC:POIN?

### SCPI Command

CALCulate<Ch>[:SElected]:FUNCtion:POINts?

Or

CALCulate<Ch>:TRACe<Tr>:FUNCtion:POINts?

### Description

Reads out the number of points (data pairs) of the analysis result by the [CALC:FUNC:EXEC](#) command.

Always reads out 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 peak or all targets.

query only

### Target

CALCulate<Ch>[:SElected] — active trace of channel <Ch>,

Or

CALCulate<Ch>:TRACe<Tr> — trace <Tr> of channel <Ch>,

<Ch> = {[1]|2|...16}

<Tr> = {[1]|2|...16}

### Query Response

<numeric>

### Related Commands

[CALC:FUNC:EXEC](#)

### Equivalent Softkeys

None

---

## Equivalent COM Command

SCPI.CALCulate(Ch).SELEcted.FUNCtion.POINts

---

### NOTE

This command is similar to  
CALCulate<Ch>[:SELEcted]:FUNCtion:POINts? only.

---

## Syntax

Value = app.SCPI.CALCulate(Ch).SELEcted.FUNCtion.POINts

## Type

Long (read only)

---

Back to [CALCulate](#)

## CALC:FUNC:PPOL

### SCPI Command

CALCulate<Ch>[:SElected]:FUNCtion:PPOLarity <char>

CALCulate<Ch>[:SElected]:FUNCtion:PPOLarity?

Or

CALCulate<Ch>:TRACe<Tr>:FUNCtion:PPOLarity <char>

CALCulate<Ch>:TRACe<Tr>:FUNCtion:PPOLarity?

### Description

Selects the polarity when performing the peak search with the [CALC:FUNC:EXEC](#) command.

command/query

### Target

CALCulate<Ch>[:SElected] — active trace of channel <Ch>,

Or

CALCulate<Ch>:TRACe<Tr> — trace <Tr> of channel <Ch>,

<Ch> = {[1]2|...16}

<Tr> = {[1]2|...16}

### Parameter

<char> Specifies the polarity:

<b>POSitive</b>	Positive peaks
<b>NEGative</b>	Negative peaks
<b>BOTH</b>	Both positive peaks and negative peaks



## Query Response

{POS|NEG|BOTH}

## Preset Value

POS

## Related Commands

[CALC:FUNC:EXEC](#)

## Equivalent Softkeys

None

---

## Equivalent COM Command

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

---

### NOTE

This command is similar to  
CALCulate<Ch>[:SElected]:FUNCtion:PPOLarity only.

---

## Syntax

Param = app.SCPI.CALCulate(Ch).SElected.FUNCtion.PPOLarity

app.SCPI.CALCulate(Ch).SElected.FUNCtion.PPOLarity = "NEG"

## Type

String (read/write)

---

Back to [CALCulate](#)

## CALC:FUNC:TARG

### SCPI Command

CALCulate<Ch>[:SElected]:FUNCtion:TARGet <numeric>

CALCulate<Ch>[:SElected]:FUNCtion:TARGet?

Or

CALCulate<Ch>:TRACe<Tr>:FUNCtion:TARGet <numeric>

CALCulate<Ch>:TRACe<Tr>:FUNCtion:TARGet?

### Description

Selects the target level when performing the search for the trace and the target level crosspoints with the [CALC:FUNC:EXEC](#) command.

command/query

### Target

CALCulate<Ch>[:SElected] — active trace of channel <Ch>,

Or

CALCulate<Ch>:TRACe<Tr> — trace <Tr> of channel <Ch>,

<Ch> = {[1]|2|...16}

<Tr> = {[1]|2|...16}

### Parameter

<numeric> the target value, varies depending on the data format

### Unit

dB |° |s

### Query Response

<numeric>

### Preset Value

0

### Related Commands

[CALC:FUNC:EXEC](#)

### Equivalent Softkeys

None

---

### Equivalent COM Command

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

---

#### NOTE

This command is similar to  
CALCulate<Ch>[:SELEcted]:FUNCtion:TARGet only.

---

### Syntax

Value = app.SCPI.CALCulate(Ch).SELEcted.FUNCtion.TARGet

app.SCPI.CALCulate(Ch).SELEcted.FUNCtion.TARGet = -10

### Type

Double (read/write)

---

Back to [CALCulate](#)

## CALC:FUNC:TTR

### SCPI Command

CALCulate<Ch>[:SElected]:FUNCtion:TTRansition <char>

CALCulate<Ch>[:SElected]:FUNCtion:TTRansition?

Or

CALCulate<Ch>:TRACe<Tr>:FUNCtion:TTRansition <char>

CALCulate<Ch>:TRACe<Tr>:FUNCtion:TTRansition?

### Description

Selects the transition type when performing the search for the trace and the target level crosspoints with the [CALC:FUNC:EXEC](#) command.

command/query

### Target

CALCulate<Ch>[:SElected] — active trace of channel <Ch>,

Or

CALCulate<Ch>:TRACe<Tr> — trace <Tr> of channel <Ch>,

<Ch> = {[1]}2|...16}

<Tr> = {[1]}2|...16}

### Parameter

<char> Specifies the transition:

<b>POSitive</b>	Positive peaks
<b>NEGative</b>	Negative peaks
<b>BOTH</b>	Both positive peaks and negative peaks

### Query Response

{POS|NEG|BOTH}

### Preset Value

POS

### Related Commands

[CALC:FUNC:EXEC](#)

### Equivalent Softkeys

None

---

### Equivalent COM Command

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

---

#### NOTE

This command is similar to  
CALCulate<Ch>[:SElected]:FUNCtion.TTRansition only.

---

### Syntax

Param = app.SCPI.CALCulate(Ch).SElected.FUNCtion.TTRansition

app.SCPI.CALCulate(Ch).SElected.FUNCtion.TTRansition = "both"

### Type

String (read/write)

---

Back to [CALCulate](#)

## CALC:FUNC:TYPE

### SCPI Command

CALCulate<Ch>[:SElected]:FUNCtion:TYPE <char>

CALCulate<Ch>[:SElected]:FUNCtion:TYPE?

Or

CALCulate<Ch>:TRACe<Tr>:FUNCtion:TYPE <char>

CALCulate<Ch>:TRACe<Tr>:FUNCtion:TYPE?

### Description

Selects the type of analysis executed with the [CALC:FUNC:EXEC](#) command.

command/query

### Target

CALCulate<Ch>[:SElected] — active trace of channel <Ch>,

Or

CALCulate<Ch>:TRACe<Tr> — trace <Tr> of channel <Ch>,

<Ch> = {[1]|2|...16}

<Tr> = {[1]|2|...16}

### Parameter

<char> Specifies the transition:

<b>PTPeak</b>	Peak-to-peak (difference between the maximum value and the minimum value)
<b>STDEV</b>	Standard deviation
<b>MEAN</b>	Mean value

<b>MAXimum</b>	Maximum value
<b>MINimum</b>	Minimum value
<b>PEAK</b>	Search for peak
<b>APEak</b>	Search for all the peaks
<b>ATARget</b>	Search for all targets

**Query Response**

{PTP|STDEV|MEAN|MAX|MIN|PEAK|APE|ATAR}

**Preset Value**

PTP

**Related Commands**

[CALC:FUNC:EXEC](#)

**Equivalent Softkeys**

None

---

**Equivalent COM Command**

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

---

<b>NOTE</b>	This command is similar to CALCulate<Ch>[:SElected]:FUNCtion:TYPE only.
-------------	--

---

**Syntax**

Param = app.SCPI.CALCulate(Ch).SElected.FUNCtion.TYPE

app.SCPI.CALCulate(Ch).SElected.FUNCtion.TYPE = "STDEV"

## Type

String (read/write)

---

Back to [CALCulate](#)



## CALC:HOLD:TYPE

### SCPI Command

CALCulate<Ch>[:SElected]:HOLD:TYPE <char>

CALCulate<Ch>[:SElected]:HOLD:TYPE?

Or

CALCulate<Ch>:TRACe<Tr>:HOLD:TYPE <char>

CALCulate<Ch>:TRACe<Tr>:HOLD:TYPE?

### Description

Sets the type of the trace hold function. The function holds the trace at the maximum or minimum point.

command/query

### Target

CALCulate<Ch>[:SElected] — active trace of channel <Ch>,

Or

CALCulate<Ch>:TRACe<Tr> — trace <Tr> of channel <Ch>,

<Ch> = {[1]2|...16}

<Tr> = {[1]2|...16}

### Parameter

<char> Specifies the type of the trace hold function:

<b>OFF</b>	Turns off the trace hold function
<b>MAXimum</b>	Maximum hold
<b>MINimum</b>	Minimum hold

**Query Response**

{OFF|MAX|MIN}

**Preset Value**

OFF

**Related Commands**

[CALC:HOLD:CLEAr](#)

**Equivalent Softkeys**

Display > Trace Hold > Hold Type {OFF | Maximum | Minimum}

---

**Equivalent COM Command**

None

---

Back to [CALCulate](#)

## CALC:HOLD:CLE

### SCPI Command

CALCulate<Ch>[:SElected]:HOLD:CLEar

Or

CALCulate<Ch>:TRACe<Tr>:HOLD:CLEar

### Description

This command resets the trace hold function.

command

### Target

CALCulate<Ch>[:SElected] — active trace of channel <Ch>,

Or

CALCulate<Ch>:TRACe<Tr> — trace <Tr> of channel <Ch>,

<Ch> = {[1]|2|...16}

<Tr> = {[1]|2|...16}

### Related Commands

[CALC:HOLD:TYPE](#)

### Equivalent Softkeys

Display > Trace Hold > Restart

---

### Equivalent COM Command

None

---

Back to [CALCulate](#)

## CALC:LIM

### SCPI Command

CALCulate<Ch>[:SElected]:LIMit[:STATe] {OFF|ON|0|1}

CALCulate<Ch>[:SElected]:LIMit[:STATe]?

Or

CALCulate<Ch>:TRACe<Tr>:LIMit[:STATe] {OFF|ON|0|1}

CALCulate<Ch>:TRACe<Tr>:LIMit[:STATe]?

### Description

Turns the limit test ON/OFF.

command/query

### Target

CALCulate<Ch>[:SElected] — active trace of channel <Ch>,

Or

CALCulate<Ch>:TRACe<Tr> — trace <Tr> of channel <Ch>,

<Ch> = {[1]|2|...16}

<Tr> = {[1]|2|...16}

### Parameter

{ON|1}      ON

{OFF|0}     OFF

### Query Response

{0|1}

## Preset Value

0

## Equivalent Softkeys

**Analysis > Limit Test > Limit Test**

---

## Equivalent COM Command

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

---

### NOTE

This command is similar to  
CALCulate<Ch>[:SElected]:LIMit[:STATe] only.

---

## Syntax

Status = app.SCPI.CALCulate(Ch).SElected.LIMit.STATe

app.SCPI.CALCulate(Ch).SElected.LIMit.STATe = true

## Type

Boolean (read/write)

---

Back to [CALCulate](#)

## **CALC:LIM:DATA**

### **SCPI Command**

CALCulate<Ch>[:SElected]:LIMit:DATA <numeric list>

CALCulate<Ch>[:SElected]:LIMit:DATA?

Or

CALCulate<Ch>:TRACe<Tr>:LIMit:DATA <numeric list>

CALCulate<Ch>:TRACe<Tr>:LIMit:DATA?

## Description

Sets 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 limit line segments.

For the  $n$ -th point, where  $n$  from 1 to  $N$ :

<numeric 1> the number of limit line segments  $N$  is from 0 to 100. Setting 0 clears the limit line.

<numeric  $5n-3$ > type of the  $n$ -th limit line segment:

0: Off.

1: Upper limit

2: Lower limit

3: Single Point limit

<numeric  $5n-2$ > the stimulus value in the start point of the  $n$ -th segment

<numeric  $5n-1$ > the stimulus value in the end point of the  $n$ -th segment

<numeric  $5n-0$ > the response value in the start point of the  $n$ -th segment

<numeric  $5n+1$ > the response value in the end point of the  $n$ -th segment

**Note:** If the array size is not  $1 + 5N$ , where  $N$  is <numeric 1>, an error occurs. If <numeric  $5n-3$ > is less than 0 or more than 2, an error occurs. When <numeric  $5n-2$ >, <numeric  $5n-1$ >, <numeric  $5n-0$ >, and <numeric  $5n+1$ > elements are out of allowable range, the value is set to the limit, which is closer to the specified value.

command/query

## Target

CALCulate<Ch>[:SElected] — active trace of channel <Ch>,

Or

CALCulate<Ch>:TRACe<Tr> — trace <Tr> of channel <Ch>,

<Ch> = {[1]2|...16}

<Tr> = {[1]2|...16}

## Query Response

<numeric 1>, <numeric 2>, ...<numeric 5N+1>

## Equivalent Softkeys

**Analysis > Limit Test > Edit Limit Line**

---

## Equivalent COM Command

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

---

### NOTE

This command is similar to  
CALCulate<Ch>[:SElected]:LIMit:DATA only.

---

## Syntax

Data = app.SCPI.CALCulate(Ch).SElected.LIMit.DATA

app.SCPI.CALCulate(Ch).SElected.LIMit.DATA = Array(1,2,800,900,-10,-10)

## Type

Variant (array of Double) (read/write)

---

Back to [CALCulate](#)



## CALC:LIM:DISP

### SCPI Command

CALCulate<Ch>[:SElected]:LIMit:DISPlay[:STATe] {OFF|ON|0|1}

CALCulate<Ch>[:SElected]:LIMit:DISPlay[:STATe]?

Or

CALCulate<Ch>:TRACe<Tr>:LIMit:DISPlay[:STATe] {OFF|ON|0|1}

CALCulate<Ch>:TRACe<Tr>:LIMit:DISPlay[:STATe]?

### Description

Turns the limit line display of the limit test function ON/OFF.

command/query

### Target

CALCulate<Ch>[:SElected] — active trace of channel <Ch>,

Or

CALCulate<Ch>:TRACe<Tr> — trace <Tr> of channel <Ch>,

<Ch> = {[1]|2|...16}

<Tr> = {[1]|2|...16}

### Parameter

{ON|1}      ON

{OFF|0}      OFF

### Query Response

{0|1}

## Preset Value

0

## Equivalent Softkeys

Analysis > Limit Test > Limit Line

---

## Equivalent COM Command

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

---

### NOTE

This command is similar to CALCulate<Ch>[:SElected]:LIMit:DISPlay[:STATe] only.

---

## Syntax

Status = app.SCPI.CALCulate(Ch).SElected.LIMit.DISPlay.STATe

app.SCPI.CALCulate(Ch).SElected.LIMit.DISPlay.STATe = true

## Type

Boolean (read/write)

---

Back to [CALCulate](#)

## CALC:LIM:FAIL?

### SCPI Command

CALCulate<Ch>[:SElected]:LIMit:FAIL?

Or

CALCulate<Ch>:TRACe<Tr>:LIMit:FAIL?

### Description

Reads out the limit test result.

query only

### Target

CALCulate<Ch>[:SElected] — active trace of channel <Ch>,

Or

CALCulate<Ch>:TRACe<Tr> — trace <Tr> of channel <Ch>,

<Ch> = {[1]|2|...16}

<Tr> = {[1]|2|...16}

### Parameter

**1**      Fail

**0**      Pass

### Equivalent Softkeys

None

---

### Equivalent COM Command

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

---

**NOTE**

This command is similar to  
CALCulate<Ch>[:SElected]:LIMit:FAIL? only.

---

**Syntax**

Status = app.SCPI.CALCulate(Ch).SElected.LIMit.FAIL

**Type**

Boolean (read/write)

---

Back to [CALCulate](#)

## CALC:LIM:OFFS:AMPL

### SCPI Command

CALCulate<Ch>[:SElected]:LIMit:OFFSet:AMPLitude <numeric>

CALCulate<Ch>[:SElected]:LIMit:OFFSet:AMPLitude?

Or

CALCulate<Ch>:TRACe<Tr>:LIMit:OFFSet:AMPLitude <numeric>

CALCulate<Ch>:TRACe<Tr>:LIMit:OFFSet:AMPLitude?

### Description

Sets and reads out the value of the limit line offset along the Y-axis.

command/query

### Target

CALCulate<Ch>[:SElected] — active trace of channel <Ch>,

Or

CALCulate<Ch>:TRACe<Tr> — trace <Tr> of channel <Ch>,

<Ch> = {[1]|2|...16}

<Tr> = {[1]|2|...16}

### Parameter

<numeric> the value of the limit line offset along Y-axis, varies depending on the data format.

### Unit

dB |° |s

### Query Response

<numeric>

## Preset Value

0

## Equivalent Softkeys

**Analysis > Limit Test > Limit Line Offsets > Response Offset**

---

## Equivalent COM Command

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

---

### NOTE

This command is similar to CALCulate<Ch>[:SElected]:LIMit:OFFSet:AMPLitude only.

---

## Syntax

Value = app.SCPI.CALCulate(Ch).SElected.LIMit.OFFSet.AMPLitude

app.SCPI.CALCulate(Ch).SElected.LIMit.OFFSet.AMPLitude = -10

## Type

Double (read/write)

---

Back to [CALCulate](#)

## CALC:LIM:OFFS:MARK

### SCPI Command

CALCulate<Ch>[:SElected]:LIMit:OFFSet:MARKer

Or

CALCulate<Ch>:TRACe<Tr>:LIMit:OFFSet:MARKer

### Description

Sets the value of the limit line offset along the Y-axis to the active marker value.

no query

### Target

CALCulate<Ch>[:SElected] — active trace of channel <Ch>,

Or

CALCulate<Ch>:TRACe<Tr> — trace <Tr> of channel <Ch>,

<Ch> = {[1]2|...16}

<Tr> = {[1]2|...16}

### Equivalent Softkeys

Analysis > Limit Test > Limit Line Offsets > Marker > Response Ofs

### Equivalent COM Command

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

---

#### NOTE

This command is similar to  
CALCulate<Ch>[:SElected]:LIMit:OFFSet:MARKer only.

---

## Syntax

app.SCPI.CALCulate(Ch).SELEcted.LIMit.OFFSet.MARKer

## Type

Method

---

Back to [CALCulate](#)



## CALC:LIM:OFFS:STIM

### SCPI Command

CALCulate<Ch>[:SElected]:LIMit:OFFSet:STIMulus <stimulus>

CALCulate<Ch>[:SElected]:LIMit:OFFSet:STIMulus?

Or

CALCulate<Ch>:TRACe<Tr>:LIMit:OFFSet:STIMulus <stimulus>

CALCulate<Ch>:TRACe<Tr>:LIMit:OFFSet:STIMulus?

### Description

Sets and reads out the value of the limit line offset along the X-axis.

command/query

### Target

CALCulate<Ch>[:SElected] — active trace of channel <Ch>,

Or

CALCulate<Ch>:TRACe<Tr> — trace <Tr> of channel <Ch>,

<Ch> = {[1]|2|...16}

<Tr> = {[1]|2|...16}

### Parameter

<stimulus> the value of the limit line offset along X-axis

### Unit

Hz | s | dBm

### Query Response

<numeric>

## Preset Value

0

## Equivalent Softkeys

**Analysis > Limit Test > Limit Lines Offsets > Stimulus Offset**

---

## Equivalent COM Command

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

---

### NOTE

This command is similar to CALCulate<Ch>[:SELEcted]:LIMit:OFFSet:STIMulus only.

---

## Syntax

Value = app.SCPI.CALCulate(Ch).SELEcted.LIMit.OFFSet.STIMulus

app.SCPI.CALCulate(Ch).SELEcted.LIMit.OFFSet.STIMulus = 1e6

## Type

Double (read/write)

---

Back to [CALCulate](#)

## CALC:LIM:REP:ALL?

### SCPI Command

CALCulate<Ch>[:SElected]:LIMit:REPort:ALL?

Or

CALCulate<Ch>:TRACe<Tr>:LIMit:REPort:ALL?

### Description

Reads out the limit test result report.

The array size is  $4N$ , where  $N$  is the number of measurement points.

For the  $n$ -th point, where  $n$  from 1 to  $N$ :

<numeric  $4n-3$ > the stimulus value in the  $n$ -th point;

<numeric  $4n-2$ > the limit test result in the  $n$ -th point;

–1: No limit

0: Fail

1: Pass

<numeric  $4n-1$ > the upper limit value in the  $n$ -th point (0 — if there is no limit)

<numeric  $4n-0$ > the lower limit value in the  $n$ -th point (0 — if there is no limit)

query only

### Target

CALCulate<Ch>[:SElected] — active trace of channel <Ch>,

Or

CALCulate<Ch>:TRACe<Tr> — trace <Tr> of channel <Ch>,

<Ch> = {[1]2|...16}

<Tr> = {[1]2|...16}

### Query Response

<numeric 1>, <numeric 2>, ...<numeric  $4N$ >

## Related Commands

[FORM:DATA](#)

## Equivalent Softkeys

None

---

## Equivalent COM Command

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

---

### NOTE

This command is similar to  
CALCulate<Ch>[:SElected]:LIMit:REPort:ALL only.

---

## Syntax

Data = app.SCPI.CALCulate(Ch).SElected.LIMit.REPort.ALL

## Type

Variant (array of Double) (read only)

---

Back to [CALCulate](#)

## CALC:LIM:REP:POIN?

### SCPI Command

CALCulate<Ch>[:SElected]:LIMit:REPort:POINts?

Or

CALCulate<Ch>:TRACe<Tr>:LIMit:REPort:POINts?

### Description

Reads out the number of the measurement points that failed the limit test.

The stimulus data array of these points can be read out by the [CALC:LIM:REP?](#) command.

query only

### Target

CALCulate<Ch>[:SElected] — active trace of channel <Ch>,

Or

CALCulate<Ch>:TRACe<Tr> — trace <Tr> of channel <Ch>,

<Ch> = {[1]|2|...16}

<Tr> = {[1]|2|...16}

### Query Response

<numeric>

### Related Commands

[CALC:LIM:REP?](#)

### Equivalent Softkeys

None

---

## Equivalent COM Command

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

---

### NOTE

This command is similar to  
CALCulate<Ch>[:SELEcted]:LIMit:REPort:POINts? only.

---

## Syntax

Cnt = app.SCPI.CALCulate(Ch).SELEcted.LIMit.REPort.POINts

## Type

Long (read only)

---

Back to [CALCulate](#)

## CALC:LIM:REP?

### SCPI Command

CALCulate<Ch>[:SElected]:LIMit:REPort[:DATA]?

Or

CALCulate<Ch>:TRACe<Tr>:LIMit:REPort[:DATA]?

### Description

Reads out the data array, which is the stimulus values of the measurement points that failed the limit test.

The array size is set by the [CALC:LIM:REP:POIN?](#) command.

query only

### Target

CALCulate<Ch>[:SElected] — active trace of channel <Ch>,

Or

CALCulate<Ch>:TRACe<Tr> — trace <Tr> of channel <Ch>,

<Ch> = {[1]|2|...16}

<Tr> = {[1]|2|...16}

### Query Response

<numeric 1>, <numeric 2>, ... <numeric N>

### Related Commands

[CALC:LIM:REP:POIN?](#)

### Equivalent Softkeys

None

---

## Equivalent COM Command

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

---

### NOTE

This command is similar to  
CALCulate<Ch>[:SElected]:LIMit:REPort[:DATA]? only.

---

### Syntax

Data = app.SCPI.CALCulate(Ch).SElected.LIMit.REPort.DATA

### Type

Variant (array of Double) (read only)

---

Back to [CALCulate](#)



## CALC:MARK

### SCPI Command

CALCulate<Ch>[:SElected]:MARKer<Mk>[:STATe] {OFF|ON|0|1}

CALCulate<Ch>[:SElected]:MARKer<Mk>[:STATe]?

Or

CALCulate<Ch>:TRACe<Tr>:MARKer<Mk>[:STATe] {OFF|ON|0|1}

CALCulate<Ch>:TRACe<Tr>:MARKer<Mk>[:STATe]?

### Description

Turns the marker ON/OFF.

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 of the reference marker with number 16). 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 between relative and absolute measurement mode.

command/query

### Target

Marker <Mk> of the active trace of channel <Ch>,

Or

Marker <Mk> of the trace <Tr> of channel <Ch>,

<Ch> = {[1]|2|...16}

<Tr> = {[1]|2|...16}

<Mk> = {[1]|2|...16}

## Parameter

{ON|1}      ON

{OFF|0}      OFF

## Query Response

{0|1}

## Preset Value

0

## Equivalent Softkeys

**Markers > Add Marker | Remove Marker**

**Markers > Reference Marker**

---

## Equivalent COM Command

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

---

### NOTE

This command is similar to  
CALCulate<Ch>[:SElected]:MARKer<Mk>[:STATe] only.

---

## Syntax

Status = app.SCPI.CALCulate(Ch).SElected.MARKer(Mk).STATe

app.SCPI.CALCulate(Ch).SElected.MARKer(Mk).STATe = true

## Type

Boolean (read/write)

---

Back to [CALCulate](#)

## CALC:MARK:ACT

### SCPI Command

CALCulate<Ch>[:SElected]:MARKer<Mk>:ACTivate

Or

CALCulate<Ch>:TRACe<Tr>MARKer<Mk>:ACTivate

### Description

Sets the active marker.

If the marker is not ON, this function will turn the marker 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.

no query

### Target

Marker <Mk> of the active trace of channel <Ch>,

Or

Marker <Mk> of the trace <Tr> of channel <Ch>,

<Ch> = {[1]|2|...16}

<Tr> = {[1]|2|...16}

<Mk> = {[1]|2|...16}

### Related Commands

[SERV:CHAN:TRAC:MARK:ACT?](#)

### Equivalent Softkeys

**Markers > Select > Marker n**

**Markers > Reference Marker**

---

## Equivalent COM Command

SCPI.CALCulate(Ch).SELEcted.MARKer(Mk).ACTivate

---

### NOTE

This command is similar to  
CALCulate<Ch>[:SELEcted]:MARKer<Mk>:ACTivate only.

---

## Syntax

app.SCPI.CALCulate(Ch).SELEcted.MARKer(Mk).ACTivate

## Type

Method

---

Back to [CALCulate](#)

## CALC:MARK:BWID

### SCPI Command

CALCulate<Ch>[:SElected]:MARKer:BWIDth[:STATe] {OFF|ON|0|1}

CALCulate<Ch>[:SElected]:MARKer:BWIDth[:STATe]?

Or

CALCulate<Ch>:TRACe<Tr>:MARKer:BWIDth[:STATe] {OFF|ON|0|1}

CALCulate<Ch>:TRACe<Tr>:MARKer:BWIDth[:STATe]?

### Description

Turns the bandwidth search function ON/OFF.

command/query

### Target

CALCulate<Ch>[:SElected] — active trace of channel <Ch>,

Or

CALCulate<Ch>:TRACe<Tr> — trace <Tr> of channel <Ch>,

<Ch> = {[1]|2|...16}

<Tr> = {[1]|2|...16}

### Parameter

{ON|1}      ON

{OFF|0}     OFF

### Query Response

{0|1}

## Preset Value

0

## Equivalent Softkeys

**Markers > Marker Math > Bandwidth Search > Bandwidth Search**

---

## Equivalent COM Command

SCPI.CALCulate(Ch).SElected.MARKer(1).BWIDth.STATe

---

### NOTE

This command is similar to CALCulate<Ch>[:SElected]:MARKer:BWIDth[:STATe] only.

---

## Syntax

Status = app.SCPI.CALCulate(Ch).SElected.MARKer.BWIDth.STATe

app.SCPI.CALCulate(Ch).SElected.MARKer.BWIDth.STATe = true

## Type

Boolean (read/write)

---

### WARNING

Object MARKer has an index of 1, which can be omitted in Visual Basic but it cannot be omitted in other programming languages.

---

---

Back to [CALCulate](#)

## CALC:MARK:BWID:DATA?

### SCPI Command

CALCulate<Ch>[:SElected]:MARKer<Mk>:BWIDth:DATA?

Or

CALCulate<Ch>:TRACe<Tr>:MARKer<Mk>:BWIDth:DATA?

### Description

Reads out the bandwidth search result.

The bandwidth search can be performed relatively to the marker <Mk>, or relatively to the absolute maximum value of the trace (in this case the number of the marker is ignored), what is set by the CALC:MARK:BWID:REF command.

The data include 4 elements:

- <numeric 1> Bandwidth;
- <numeric 2> Center frequency;
- <numeric 3> Q value;
- <numeric 4> Loss.

**Note:** 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.

query only

### Target

Marker <Mk> of the active trace of channel <Ch>,

Or

Marker <Mk> of the trace <Tr> of channel <Ch>,

<Ch> = {[1]2|...16}

<Tr> = {[1]2|...16}

<Mk> = {[1]2|...16}

## Query Response

<numeric 1>, <numeric 2>, ...<numeric 4>

## Related Commands

[CALC:MARK:BWID:REF](#)

## Equivalent Softkeys

None

---

## Equivalent COM Command

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

---

### NOTE

This command is similar to  
CALCulate<Ch>[:SElected]:MARKer<Mk>:BWIDth:DATA  
? only.

---

## Syntax

Data = app.SCPI.CALCulate(Ch).SElected.MARKer(Mk).BWIDth.DATA

## Type

Variant (array of Double) (read only)

---

Back to [CALCulate](#)



## CALC:MARK:BWID:REF

### SCPI Command

CALCulate<Ch>[:SElected]:MARKer:BWIDth:REFerence <char>

CALCulate<Ch>[:SElected]:MARKer:BWIDth:REFerence?

Or

CALCulate<Ch>:TRACe<Tr>:MARKer:BWIDth:REFerence <char>

CALCulate<Ch>:TRACe<Tr>:MARKer:BWIDth:REFerence?

### Description

Selects the reference for the bandwidth search function: reference marker or absolute maximum value of the trace.

command/query

### Target

CALCulate<Ch>[:SElected] — active trace of channel <Ch>,

Or

CALCulate<Ch>:TRACe<Tr> — trace <Tr> of channel <Ch>,

<Ch> = {[1]2|...16}

<Tr> = {[1]2|...16}

### Parameter

<char> Choose from:

<b>MARKer</b>	Bandwidth search relative to the reference marker
<b>MAXimum</b>	Bandwidth search relative to the absolute maximum of the trace
<b>MINimum</b>	Bandwidth search relative to the absolute minimum of the trace

## Query Response

{MAX|MARK|MIN}

## Preset Value

MAX

## Equivalent Softkeys

**Markers > Marker Math > Bandwidth Search > Search Ref To**

---

## Equivalent COM Command

SCPI.CALCulate(Ch).SELEcted.MARKer(1).BWIDth.REFerence

---

### NOTE

This command is similar to CALCulate<Ch>[:SELEcted]:MARKer:BWIDth:REFerence only.

---

## Syntax

Param = app.SCPI.CALCulate(Ch).SELEcted.MARKer.BWIDth.REFerence

app.SCPI.CALCulate(Ch).SELEcted.MARKer.BWIDth.REFerence = "marker"

## Type

String (read/write)

---

### WARNING

Object MARKer has an index of 1, which can be omitted in Visual Basic but it cannot be omitted in other programming languages.

---

Back to [CALCulate](#)

## CALC:MARK:BWID:THR

### SCPI Command

CALCulate<Ch>[:SElected]:MARKer<Mk>:BWIDth:THReshold <numeric>

CALCulate<Ch>[:SElected]:MARKer<Mk>:BWIDth:THReshold?

Or

CALCulate<Ch>:TRACe<Tr>:MARKer<Mk>:BWIDth:THReshold <numeric>

CALCulate<Ch>:TRACe<Tr>:MARKer<Mk>:BWIDth:THReshold?

### Description

Sets the bandwidth search threshold value.

command/query

### Target

Marker <Mk> of the active trace of channel <Ch>,

Or

Marker <Mk> of the trace <Tr> of channel <Ch>,

<Ch> = {[1]|2|...16}

<Tr> = {[1]|2|...16}

<Mk> = {[1]|2|...16}

### Parameter

<numeric> the bandwidth definition value, the range varies depending on the data format.

### Unit

dB |° |s

## Query Response

<numeric>

## Preset Value

-3.0

## Equivalent Softkeys

**Markers > Marker Math > Bandwidth Search > Bandwidth Value**

---

## Equivalent COM Command

SCPI.CALCulate(Ch).SElected.MARKer(1).BWIDth.THReshold

---

### NOTE

This command is similar to CALCulate<Ch>[:SElected]:MARKer<Mk>:BWIDth:THReshold only.

---

## Syntax

Value = app.SCPI.CALCulate(Ch).SElected.MARKer(Mk).BWIDth. THReshold

app.SCPI.CALCulate(Ch).SElected.MARKer(Mk).BWIDth.THReshold = -6.0

## Type

Double (read/write)

---

### WARNING

Object MARKer has an index of 1, which can be omitted in Visual Basic but it cannot be omitted in other programming languages.

---

Back to [CALCulate](#)

## CALC:MARK:BWID:TYPE

### SCPI Command

CALCulate<Ch>[:SElected]:MARKer:BWIDth:TYPE <char>

CALCulate<Ch>[:SElected]:MARKer:BWIDth:TYPE?

Or

CALCulate<Ch>:TRACe<Tr>:MARKer:BWIDth:TYPE <char>

CALCulate<Ch>:TRACe<Tr>:MARKer:BWIDth:TYPE?

### Description

Sets the type of the bandwidth search function.

command/query

### Target

CALCulate<Ch>[:SElected] — active trace of channel <Ch>,

Or

CALCulate<Ch>:TRACe<Tr> — trace <Tr> of channel <Ch>,

<Ch> = {[1]|2|...16}

<Tr> = {[1]|2|...16}

### Parameter

<char> Specifies the type of the bandwidth:

**BPASs**      Bandpass

**NOTCh**      Notch

### Query Response

{BPAS|NOTC}

## Preset Value

BPAS

## Equivalent Softkeys

**Markers > Marker Math > Bandwidth Search > Type**

---

## Equivalent COM Command

SCPI.CALCulate(Ch).SELEcted.MARKer(1).BWIDth.TYPE

---

### NOTE

This command is similar to CALCulate<Ch>[:SELEcted]:MARKer:BWIDth:TYPE only.

---

## Syntax

Param = app.SCPI.CALCulate(Ch).SELEcted.MARKer.BWIDth.TYPE

app.SCPI.CALCulate(Ch).SELEcted.MARKer.BWIDth.TYPE = "notc"

## Type

String (read/write)

---

### WARNING

Object MARKer has an index of 1, which can be omitted in Visual Basic but it cannot be omitted in other programming languages.

---

Back to [CALCulate](#)

## CALC:MARK:COUN

### SCPI Command

CALCulate<Ch>[:SElected]:MARKer:COUNt <numeric>

CALCulate<Ch>[:SElected]:MARKer:COUNt?

Or

CALCulate<Ch>:TRACe<Tr>:MARKer:COUNt <numeric>

CALCulate<Ch>:TRACe<Tr>:MARKer:COUNt?

### Description

Sets the number of turned ON markers.

**Note:** Choosing 16 turns on the reference marker and sets the markers 1 to 15 to the relative values.

command/query

### Target

CALCulate<Ch>[:SElected] — active trace of channel <Ch>,

Or

CALCulate<Ch>:TRACe<Tr> — trace <Tr> of channel <Ch>,

<Ch> = {[1]|2|...16}

<Tr> = {[1]|2|...16}

### Parameter

<numeric>, range from 0 to 16

### Out of Range

Sets the value of the limit, which is closer to the specified value.

## Query Response

<numeric>

## Preset Value

0

## Equivalent Softkeys

None

---

## Equivalent COM Command

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

---

### NOTE

This command is similar to only.

---

## Syntax

MarkerCnt = app.SCPI.CALCulate(Ch).SElected.MARKer.COUNT

app.SCPI.CALCulate(Ch).SElected.MARKer.COUNT = 5

## Type

Long (read/write)

---

### WARNING

Object MARKer has an index of 1, which can be omitted in Visual Basic but it cannot be omitted in other programming languages.

---

---

Back to [CALCulate](#)



## CALC:MARK:COUP

### SCPI Command

CALCulate<Ch>[:SElected]:MARKer:COUPle {OFF|ON|0|1}

CALCulate<Ch>[:SElected]:MARKer:COUPle?

Or

CALCulate<Ch>:TRACe<Tr>:MARKer:COUPle {OFF|ON|0|1}

CALCulate<Ch>:TRACe<Tr>:MARKer:COUPle?

### Description

Turns the marker coupling between traces ON/OFF. When coupled, the markers of different traces but with the same number track the X-axis position.

command/query

### Target

All the traces of channel <Ch>,

<Ch>={[1]|2|...16}

### Parameter

{ON|1}      ON

{OFF|0}      OFF

### Query Response

{0|1}

### Preset Value

1

## Equivalent Softkeys

Marker > Properties > Marker Couple

---

## Equivalent COM Command

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

---

### NOTE

This command is similar to  
CALCulate<Ch>[:SElected]:MARKer:COUPle only.

---

## Syntax

Status = app.SCPI.CALCulate(Ch).SElected.MARKer.COUPle

app.SCPI.CALCulate(Ch).SElected.MARKer.COUPle = false

## Type

Boolean (read/write)

---

### WARNING

Object MARKer has an index of 1, which can be omitted in Visual Basic but it cannot be omitted in other programming languages.

---

---

Back to [CALCulate](#)

## CALC:MARK:DATA?

### SCPI Command

CALCulate<Ch>[:SElected]:MARKer:DATA?

Or

CALCulate<Ch>:TRACe<Tr>:MARKer:DATA?

### Description

Reads out 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 the 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  $N$ :

<numeric 1> the number of turned ON markers including the reference marker ( $N$ );

<numeric  $3n-1$ > the stimulus value of the  $n$ -th marker;

<numeric  $3n$ > the real data in rectangular format, real part in polar and Smith chart formats of the  $n$ -th marker;

<numeric  $3n+1$ > 0 in rectangular format, imaginary part in polar and Smith chart formats of the  $n$ -th marker.

query only

### Target

CALCulate<Ch>[:SElected] — All markers of the active trace of channel <Ch>,

Or

CALCulate<Ch>:TRACe<Tr> — All markers of the trace <Tr> of channel <Ch>,

<Ch> = {[1]2|...16}

<Tr> = {[1]2|...16}

All markers of the active trace of channel <Ch>,

<Ch>={[1]|2|...16}

### Query Response

<numeric 1>, <numeric 2>, ...<numeric 3N+1>

### Related Commands

[CALC:MARK:COUN](#)

[FORM:DATA](#)

### Equivalent Softkeys

None

---

### Equivalent COM Command

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

---

#### NOTE

This command is similar to  
CALCulate<Ch>[:SElected]:MARKer:DATA? only.

---

### Syntax

Data = app.SCPI.CALCulate(Ch).SElected.MARKer(Mk).DATA

### Type

Variant (array of Double) (read only)

---

Back to [CALCulate](#)

## CALC:MARK:DISC

### SCPI Command

CALCulate<Ch>[:SElected]:MARKer:DISCcrete {OFF|ON|0|1}

CALCulate<Ch>[:SElected]:MARKer:DISCcrete?

Or

CALCulate<Ch>:TRACe<Tr>:MARKer:DISCcrete {OFF|ON|0|1}

CALCulate<Ch>:TRACe<Tr>:MARKer:DISCcrete?

### Description

Turns the marker discrete mode ON/OFF.

command/query

### Target

All traces of channel <Ch> (if the marker coupling is set to ON by the CALC:MARK:COUP command),

Or

CALCulate<Ch>[:SElected] — active trace of channel <Ch>,

Or

CALCulate<Ch>:TRACe<Tr> — trace <Tr> of channel <Ch>,

<Ch> = {[1]|2|...16}

<Tr> = {[1]|2|...16}

### Parameter

{ON|1}      ON

{OFF|0}     OFF

## Query Response

{0|1}

## Preset Value

1

## Equivalent Softkeys

**Marker > Properties > Discrete {ON/OFF}**

---

## Equivalent COM Command

SCPI.CALCulate(Ch).SELEcted.MARKer(1).DISCrete

---

### NOTE

This command is similar to  
CALCulate<Ch>[:SELEcted]:MARKer:DISCrete only.

---

## Syntax

Status = app.SCPI.CALCulate(Ch).SELEcted.MARKer.DISCrete

app.SCPI.CALCulate(Ch).SELEcted.MARKer.DISCrete = false

## Type

Boolean (read/write)

---

### WARNING

Object MARKer has an index of 1, which can be omitted in Visual Basic but it cannot be omitted in other programming languages.

---

Back to [CALCulate](#)

## CALC:MARK:FUNC:DOM

### SCPI Command

CALCulate<Ch>[:SElected]:MARKer:FUNCtion:DOMain[:STATe] {OFF|ON|0|1}

CALCulate<Ch>[:SElected]:MARKer:FUNCtion:DOMain[:STATe]?

Or

CALCulate<Ch>:TRACe<Tr>:MARKer:FUNCtion:DOMain[:STATe] {OFF|ON|0|1}

CALCulate<Ch>:TRACe<Tr>:MARKer:FUNCtion:DOMain[:STATe]?

### Description

Turns the state of the arbitrary range when executing the marker search ON/OFF. If the state of an arbitrary range is ON, marker search is performed in the range specified by the [CALC:MARK:FUNC:DOM:STAR](#), [CALC:MARK:FUNC:DOM:STOP](#) commands. Otherwise, the search is performed over the entire sweep range.

command/query

### Target

All traces of channel <Ch> (if the marker search range coupling is set to ON by the CALC:MARK:FUNC:DOM:COUP command),

Or

CALCulate<Ch>[:SElected] — active trace of channel <Ch>,

Or

CALCulate<Ch>:TRACe<Tr> — trace <Tr> of channel <Ch>,

<Ch> = {[1]|2|...16}

<Tr> = {[1]|2|...16}

## Parameter

**{ON|1}**      Arbitrary range  
**{OFF|0}**      Entire sweep range

## Query Response

{0|1}

## Preset Value

0

## Equivalent Softkeys

**Markers > Marker Search > Search Range**

---

## Equivalent COM Command

SCPI.CALCulate(Ch).SElected.MARKer(1).FUNCtion.DOMain.STATe

---

### NOTE

This command is similar to CALCulate<Ch>[:SElected]:MARKer:FUNCtion:DOMAIN[:STATe] only.

---

## Syntax

Status = app.SCPI.CALCulate(Ch).SElected.MARKer.FUNCtion.DOMain.STATe

app.SCPI.CALCulate(Ch).SElected.MARKer.FUNCtion.DOMain.STATe = true



## Type

Boolean (read/write)

---

### **WARNING**

Object MARKer has an index of 1, which can be omitted in Visual Basic but it cannot be omitted in other programming languages.

---

---

Back to [CALCulate](#)

## CALC:MARK:FUNC:DOM:COUP

### SCPI Command

CALCulate<Ch>[:SElected]:MARKer:FUNCtion:DOMain:COUPle {OFF|ON|0|1}

CALCulate<Ch>[:SElected]:MARKer:FUNCtion:DOMain:COUPle?

Or

CALCulate<Ch>:TRACe<Tr>:MARKer:FUNCtion:DOMain:COUPle {OFF|ON|0|1}

CALCulate<Ch>:TRACe<Tr>:MARKer:FUNCtion:DOMain:COUPle?

### Description

Turns the state of the marker search range coupling for different traces ON/OFF. If the arbitrary search range turned ON by the [CALC:MARK:FUNC:DOM](#) command, specifies whether (coupling) or each trace uses individual range when the marker search is performed.

command/query

### Target

All the traces of channel <Ch>,

<Ch>={[1]2|...16}

### Parameter

Specifies the state of the marker search range coupling:

**{ON|1}**      All traces of channel use the same range

**{OFF|0}**      Each trace uses individual range

### Query Response

{0|1}

### Preset Value

1

## Equivalent Softkeys

Markers > Marker Search > Couple

---

## Equivalent COM Command

SCPI.CALCulate(Ch).SElected.MARKer(1).FUNctio.n.DOMain.COUPle

---

### NOTE

This command is similar to CALCulate<Ch>[:SElected]:MARKer:FUNCTio.n:DOMAIN:COUPle only.

---

## Syntax

Status = app.SCPI.CALCulate(Ch).SElected.MARKer.FUNCTio.n.DOMain.COUPle

app.SCPI.CALCulate(Ch).SElected.MARKer.FUNCTio.n.DOMain.COUPle = false

## Type

Boolean (read/write)

---

### WARNING

Object MARKer has an index of 1, which can be omitted in Visual Basic but it cannot be omitted in other programming languages.

---

---

Back to [CALCulate](#)

## CALC:MARK:FUNC:DOM:STAR

### SCPI Command

CALCulate<Ch>[:SElected]:MARKer:FUNCtion:DOMain:STARt <stimulus>

CALCulate<Ch>[:SElected]:MARKer:FUNCtion:DOMain:STARt?

Or

CALCulate<Ch>:TRACe<Tr>:MARKer:FUNCtion:DOMain:STARt <stimulus>

CALCulate<Ch>:TRACe<Tr>:MARKer:FUNCtion:DOMain:STARt?

### Description

Sets or reads out the start value of the marker search range.

command/query

### Target

All traces of channel <Ch> (if the marker search range coupling is set to ON by the CALC:MARK:FUNC:DOM:COUP command),

Or

CALCulate<Ch>[:SElected] — active trace of channel <Ch>,

Or

CALCulate<Ch>:TRACe<Tr> — trace <Tr> of channel <Ch>,

<Ch> = {[1]|2|...16}

<Tr> = {[1]|2|...16}

### Parameter

<stimulus> the start value of the marker search

### Unit

Hz | s | dBm

## Query Response

<numeric>

## Preset Value

Lower limit of the analyzer frequency range

## Equivalent Softkeys

**Markers > Marker Search > Search Start**

---

## Equivalent COM Command

SCPI.CALCulate(Ch).SElected.MARKer(1).FUNCtion.DOMain.START

---

### NOTE

This command is similar to CALCulate<Ch>[:SElected]:MARKer:FUNCtion:DOMAIN:START only.

---

## Syntax

Value = app.SCPI.CALCulate(Ch).SElected.MARKer.FUNCtion.DOMain.START

app.SCPI.CALCulate(Ch).SElected.MARKer.FUNCtion.DOMain.START = 1e6

## Type

Double (read/write)

---

### WARNING

Object MARKer has an index of 1, which can be omitted in Visual Basic but it cannot be omitted in other programming languages.

---

---

Back to [CALCulate](#)

## CALC:MARK:FUNC:DOM:STOP

### SCPI Command

CALCulate<Ch>[:SElected]:MARKer:FUNCtion:DOMain:STOP <stimulus>

CALCulate<Ch>[:SElected]:MARKer:FUNCtion:DOMain:STOP?

Or

CALCulate<Ch>:TRACe<Tr>:MARKer:FUNCtion:DOMain:STOP <stimulus>

CALCulate<Ch>:TRACe<Tr>:MARKer:FUNCtion:DOMain:STOP?

### Description

Sets or reads out the stop value of the marker search range.

command/query

### Target

All traces of channel <Ch> (if the marker search range coupling is set to ON by the [CALC:MARK:FUNC:DOM:COUP](#) command),

Or

CALCulate<Ch>[:SElected] — active trace of channel <Ch>,

Or

CALCulate<Ch>:TRACe<Tr> — trace <Tr> of channel <Ch>,

<Ch> = {[1]2|...16}

<Tr> = {[1]2|...16}

### Parameter

<stimulus> the stop value of the marker search

### Unit

Hz | s | dBm

## Query Response

<numeric>

## Preset Value

Upper limit of the analyzer frequency range

## Equivalent Softkeys

**Markers > Marker Search > Search Stop**

---

## Equivalent COM Command

SCPI.CALCulate(Ch).SELEcted.MARKer(1).FUNCtion.DOMain.STOP

---

### NOTE

This command is similar to CALCulate<Ch>[:SELEcted]:MARKer:FUNCtion:DOMAIN:STOP only.

---

## Syntax

Value = app.SCPI.CALCulate(Ch).SELEcted.MARKer.FUNCtion.DOMain.STOP

app.SCPI.CALCulate(Ch).SELEcted.MARKer.FUNCtion.DOMain.STOP = 1e6

## Type

Double (read/write)

---

### WARNING

Object MARKer has an index of 1, which can be omitted in Visual Basic but it cannot be omitted in other programming languages.

---

---

Back to [CALCulate](#)

## CALC:MARK:FUNC:EXEC

### SCPI Command

CALCulate<Ch>[:SElected]:MARKer<Mk>:FUNCtion:EXECute

Or

CALCulate<Ch>:TRACe<Tr>:MARKer<Mk>:FUNCtion:EXECute

### Description

Executes the marker search according to the specified criterion. The type of the marker search is set by the [CALC:MARK:FUNC:TYPE](#) command.

no query

### Target

Marker <Mk> of the active trace of channel <Ch>,

Or

Marker <Mk> of the trace <Tr> of channel <Ch>,

<Ch> = {[1]|2|...16}

<Tr> = {[1]|2|...16}

<Mk> = {[1]|2|...16}

### Related Commands

[CALC:MARK:FUNC:TYPE](#)

[CALC:MARK:FUNC:DOM](#)

### Equivalent Softkeys

**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}**

---

### **Equivalent COM Command**

SCPI.CALCulate(Ch).SELEcted.MARKer(Mk).FUNCtion.EXECute

---

#### **NOTE**

This command is similar to CALCulate<Ch>[:SELEcted]:MARKer<Mk>:FUNCtion:EXECute only.

---

### **Syntax**

Data = app.SCPI.CALCulate(Ch).SELEcted.MARKer(Mk).FUNCtion.EXECute

### **Type**

Method

---

Back to [CALCulate](#)

## CALC:MARK:FUNC:PEXC

### SCPI Command

CALCulate<Ch>[:SElected]:MARKer<Mk>:FUNCtion:PEXCursion <numeric>

CALCulate<Ch>[:SElected]:MARKer<Mk>:FUNCtion:PEXCursion?

Or

CALCulate<Ch>:TRACe<Tr>:MARKer<Mk>:FUNCtion:PEXCursion <numeric>

CALCulate<Ch>:TRACe<Tr>:MARKer<Mk>:FUNCtion:PEXCursion?

### Description

Sets or reads out the peak excursion value when the marker peak search is performed by the [CALC:MARK:FUNC:EXEC](#) command.

command/query

### Target

Marker <Mk> of the active trace of channel <Ch>,

Or

Marker <Mk> of the trace <Tr> of channel <Ch>,

<Ch> = {[1]|2|...16}

<Tr> = {[1]|2|...16}

<Mk> = {[1]|2|...16}

### Parameter

<numeric> the peak excursion value, the range varies depending on the data format

### Unit

dB | ° | s

## Query Response

<numeric>

## Preset Value

1

## Equivalent Softkeys

**Markers > Marker Search > Peak > Peak Excursion**

---

## Equivalent COM Command

SCPI.CALCulate(Ch).SELEcted.MARKer(Mk).FUNCtion.PEXCursion

---

### NOTE

This command is similar to CALCulate<Ch>[:SELEcted]:MARKer<Mk>:FUNCtion:PEXCursion only.

---

## Syntax

Value = app.SCPI.CALCulate(Ch).SELEcted.MARKer(Mk).FUNCtion.PEXCursion

app.SCPI.CALCulate(Ch).SELEcted.MARKer(Mk).FUNCtion.PEXCursion = 3.0

## Type

Double (read/write)

---

Back to [CALCulate](#)

## CALC:MARK:FUNC:PPOL

### SCPI Command

CALCulate<Ch>[:SElected]:MARKer<Mk>:FUNCtion:PPOLarity <char>

CALCulate<Ch>[:SElected]:MARKer<Mk>:FUNCtion:PPOLarity?

Or

CALCulate<Ch>:TRACe<Tr>:MARKer<Mk>:FUNCtion:PPOLarity <char>

CALCulate<Ch>:TRACe<Tr>:MARKer<Mk>:FUNCtion:PPOLarity?

### Description

Selects the peak polarity when the marker peak search is performed by the [CALC:MARK:FUNC:EXEC](#) command.

command/query

### Target

Marker <Mk> of the active trace of channel <Ch>,

Or

Marker <Mk> of the trace <Tr> of channel <Ch>,

<Ch> = {[1]2|...16}

<Tr> = {[1]2|...16}

<Mk> = {[1]2|...16}

## Parameter

<char> Specifies the peak polarity:

<b>POSitive</b>	Positive polarity
<b>NEGative</b>	Negative polarity
<b>BOTH</b>	Both positive polarity and negative polarity

## Query Response

{POS|NEG|BOTH}

## Preset Value

POS

## Related Commands

[CALC:MARK:FUNC:EXEC](#)

## Equivalent Softkeys

**Markers > Marker Search > Peak > Peak Polarity > {Positive | Negative | Both}**

---

## Equivalent COM Command

SCPI.CALCulate(Ch).SELEcted.MARKer(Mk).FUNCtion.PPOLarity

---

### NOTE

This command is similar to CALCulate<Ch>[:SELEcted]:MARKer<Mk>:FUNCtion:PPOLarity only.

---

## Syntax

Param = app.SCPI.CALCulate(Ch).SELEcted.MARKer(Mk).FUNCtion.PPOLarity

app.SCPI.CALCulate(Ch).SELEcted.MARKer(Mk).FUNCtion.PPOLarity = "neg"

**Type**

String (read/write)

---

Back to [CALCulate](#)

## CALC:MARK:FUNC:TARG

### SCPI Command

CALCulate<Ch>[:SElected]:MARKer<Mk>:FUNCtion:TARGet <numeric>

CALCulate<Ch>[:SElected]:MARKer<Mk>:FUNCtion:TARGet?

Or

CALCulate<Ch>:TRACe<Tr>:MARKer<Mk>:FUNCtion:TARGet <numeric>

CALCulate<Ch>:TRACe<Tr>:MARKer<Mk>:FUNCtion:TARGet?

### Description

Sets or reads out the target value when the marker target search is performed by the [CALC:MARK:FUNC:EXEC](#) command.

command/query

### Target

Marker <Mk> of the active trace of channel <Ch>,

Or

Marker <Mk> of the trace <Tr> of channel <Ch>,

<Ch> = {[1]|2|...16}

<Tr> = {[1]|2|...16}

<Mk> = {[1]|2|...16}

### Parameter

<numeric> the peak excursion value, the range varies depending on the data format

### Unit

dB | ° | s

## Query Response

<numeric>

## Preset Value

0

## Equivalent Softkeys

**Markers > Marker Search > Target > Target Value**

---

## Equivalent COM Command

SCPI.CALCulate(Ch).SELEcted.MARKer(Mk).FUNCtion.TARGet

---

### NOTE

This command is similar to CALCulate<Ch>[:SELEcted]:MARKer<Mk>:FUNCtion:TARGet only.

---

## Syntax

Value = app.SCPI.CALCulate(Ch).SELEcted.MARKer(Mk).FUNCtion.TARGet

app.SCPI.CALCulate(Ch).SELEcted.MARKer(Mk).FUNCtion.TARGet = -10

## Type

Double (read/write)

---

Back to [CALCulate](#)



## CALC:MARK:FUNC:TRAC

### SCPI Command

CALCulate<Ch>[:SElected]:MARKer<Mk>:FUNCtion:TRACking {OFF|ON|0|1}

CALCulate<Ch>[:SElected]:MARKer<Mk>:FUNCtion:TRACking?

Or

CALCulate<Ch>:TRACe<Tr>:MARKer<Mk>:FUNCtion:TRACking {OFF|ON|0|1}

CALCulate<Ch>:TRACe<Tr>:MARKer<Mk>:FUNCtion:TRACking?

### Description

Turns the marker search tracking ON/OFF.

command/query

### Target

Marker <Mk> of the active trace of channel <Ch>,

Or

Marker <Mk> of the trace <Tr> of channel <Ch>,

<Ch> = {[1]|2|...16}

<Tr> = {[1]|2|...16}

<Mk> = {[1]|2|...16}

### Parameter

{ON|1}      ON

{OFF|0}      OFF

### Query Response

{0|1}

## Preset Value

0

## Equivalent Softkeys

**Markers > Marker Search > Tracking**

---

## Equivalent COM Command

SCPI.CALCulate(Ch).SELEcted.MARKer(Mk).FUNction.TRACking

---

### NOTE

This command is similar to CALCulate<Ch>[:SELEcted]:MARKer<Mk>:FUNction:TRACking only.

---

## Syntax

Status = app.SCPI.CALCulate(Ch).SELEcted.MARKer(Mk).FUNction.TRACking

app.SCPI.CALCulate(Ch).SELEcted.MARKer(Mk).FUNction.TRACking = true

## Type

Boolean (read/write)

---

Back to [CALCulate](#)

## CALC:MARK:FUNC:TTR

### SCPI Command

CALCulate<Ch>[:SElected]:MARKer<Mk>:FUNCtion:TTRansition <char>

CALCulate<Ch>[:SElected]:MARKer<Mk>:FUNCtion:TTRansition?

Or

CALCulate<Ch>:TRACe<Tr>:MARKer<Mk>:FUNCtion:TTRansition <char>

CALCulate<Ch>:TRACe<Tr>:MARKer<Mk>:FUNCtion:TTRansition?

### Description

Selects the type of the target transition when the marker transition search is performed by the [CALC:MARK:FUNC:EXEC](#) command.

command/query

### Target

Marker <Mk> of the active trace of channel <Ch>,

Or

Marker <Mk> of the trace <Tr> of channel <Ch>,

<Ch> = {[1]|2|...16}

<Tr> = {[1]|2|...16}

<Mk> = {[1]|2|...16}

### Parameter

<char> Specifies the type of the target transition:

**POSitive**            Positive target transition

**NEGative**            Negative target transition

**BOTH**            Both positive target transition and negative target transition

### Query Response

{POS|NEG|BOTH}

### Preset Value

POS

### Related Commands

[CALC:MARK:FUNC:EXEC](#)

### Equivalent Softkeys

**Marker > Marker Search > Target > Target Transition > {Positive | Negative | Both}**

---

### Equivalent COM Command

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

---

#### NOTE

This command is similar to CALCulate<Ch>[:SElected]:MARKer<Mk>:FUNCtion:TTRansition only.

---

### Syntax

Param = app.SCPI.CALCulate(Ch).SElected.MARKer(Mk).FUNCtion.TTRansition

app.SCPI.CALCulate(Ch).SElected.MARKer(Mk).FUNCtion.TTRansition = "neg"

### Type

String (read/write)

---

Back to [CALCulate](#)

## CALC:MARK:FUNC:TYPE

### SCPI Command

CALCulate<Ch>[:SElected]:MARKer<Mk>:FUNCtion:TYPE <char>

CALCulate<Ch>[:SElected]:MARKer<Mk>:FUNCtion:TYPE?

Or

CALCulate<Ch>:TRACe<Tr>:MARKer<Mk>:FUNCtion:TYPE <char>

CALCulate<Ch>:TRACe<Tr>:MARKer<Mk>:FUNCtion:TYPE?

### Description

Selects the type of the marker search, which is performed by the [CALC:MARK:FUNC:EXEC](#) command.

command/query

### Target

Marker <Mk> of the active trace of channel <Ch>,

Or

Marker <Mk> of the trace <Tr> of channel <Ch>,

<Ch> = {[1]|2|...16}

<Tr> = {[1]|2|...16}

<Mk> = {[1]|2|...16}

### Parameter

<char> Specifies the type of the marker search:

**MAXimum**      Maximum value search

**MINimum**      Minimum value search

<b>PEAK</b>	Peak search
<b>LPEak</b>	Peak search to the left from the marker
<b>RPEak</b>	Peak search to the right from the marker
<b>TARGet</b>	Target search
<b>LTARget</b>	Target search to the left from the marker
<b>RTARget</b>	Target search to the right from the marker

## Query Response

{MAX|MIN|PEAK|LPE|RPE|TARG|LTAR|RTAR}

## Preset Value

MAX

## Related Commands

[CALC:MARK:FUNC:EXEC](#)

## Equivalent Softkeys

**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}**

## Equivalent COM Command

SCPI.CALCulate(Ch).SELEcted.MARKer(Mk).FUNCtion.TYPE

### NOTE

This command is similar to CALCulate<Ch>[:SELEcted]:MARKer<Mk>:FUNCtion:TYPE only.

## Syntax

Param = app.SCPI.CALCulate(Ch).SElected.MARKer(Mk).FUNction.TYPE

app.SCPI.CALCulate(Ch).SElected.MARKer(Mk).FUNction.TYPE = "MIN"

## Type

String (read/write)

---

Back to [CALCulate](#)

## CALC:MARK:MATH:FLAT:DATA?

### SCPI Command

CALCulate<Ch>[:SElected]:MARKer:MATH:FLATness:DATA?

Or

CALCulate<Ch>:TRACe<Tr>:MARKer:MATH:FLATness:DATA?

### Description

Reads out the FLATNESS function data array. The FLATNESS function is applied within the range determined by two markers.

The array includes 4 elements:

- <numeric 1> Span;
- <numeric 2> Gain;
- <numeric 3> Slope;
- <numeric 4> Flatness.

query only

### Target

CALCulate<Ch>[:SElected] — active trace of channel <Ch>,

Or

CALCulate<Ch>:TRACe<Tr> — trace <Tr> of channel <Ch>,

<Ch> = {[1]2|...16}

<Tr> = {[1]2|...16}

### Query Response

<numeric 1>, <numeric 2>, ... <numeric 4>



## Related Commands

[CALC:MARK:MATH:FLAT:DOM:STAR](#)

[CALC:MARK:MATH:FLAT:DOM:STOP](#)

## Equivalent Softkeys

None

---

## Equivalent COM Command

SCPI.CALCulate(Ch).SELEcted.MARKer(1).MATH.FLATness.DATA

---

### NOTE

This command is similar to CALCulate<Ch>[:SELEcted]:MARKer:MATH:FLATness:DATA? only.

---

## Syntax

Data = app.SCPI.CALCulate(Ch).SELEcted.MARKer.MATH.FLATness.DATA

## Type

Variant (array of Double) (read only)

---

### WARNING

Object MARKer has an index of 1, which can be omitted in Visual Basic but it cannot be omitted in other programming languages.

---

Back to [CALCulate](#)

## CALC:MARK:MATH:FLAT:STAT

### SCPI Command

CALCulate<Ch>[:SElected]:MARKer:MATH:FLATness:STATe {OFF|ON|0|1}

CALCulate<Ch>[:SElected]:MARKer:MATH:FLATness:STATe?

Or

CALCulate<Ch>:TRACe<Tr>:MARKer:MATH:FLATness:STATe {OFF|ON|0|1}

CALCulate<Ch>:TRACe<Tr>:MARKer:MATH:FLATness:STATe?

### Description

Turns the marker flatness function ON/OFF.

command/query

### Target

CALCulate<Ch>[:SElected] — active trace of channel <Ch>,

Or

CALCulate<Ch>:TRACe<Tr> — trace <Tr> of channel <Ch>,

<Ch> = {[1]|2|...16}

<Tr> = {[1]|2|...16}

### Parameter

{ON|1}      ON

{OFF|0}      OFF

### Query Response

{0|1}

## Preset Value

0

## Equivalent Softkeys

**Markers > Marker Math > Flatness > Flatness**

---

## Equivalent COM Command

SCPI.CALCulate(Ch).SELEcted.MARKer(1).MATH.FLATness.STATe

---

### NOTE

This command is similar to CALCulate<Ch>[:SELEcted]:MARKer:MATH:FLATness:STATe only.

---

## Syntax

Status = app.SCPI.CALCulate(Ch).SELEcted.MARKer.MATH.FLATness.STATe

app.SCPI.CALCulate(Ch).SELEcted.MARKer.MATH.FLATness.STATe = true

## Type

Boolean (read/write)

---

### WARNING

Object MARKer has an index of 1, which can be omitted in Visual Basic but it cannot be omitted in other programming languages.

---

---

Back to [CALCulate](#)

## CALC:MARK:MATH:FLAT:DOM:STAR

### SCPI Command

CALCulate<Ch>[:SElected]:MARKer:MATH:FLATness:STARt <numeric>

CALCulate<Ch>[:SElected]:MARKer:MATH:FLATness:STARt?

Or

CALCulate<Ch>:TRACe<Tr>:MARKer:MATH:FLATness:STARt <numeric>

CALCulate<Ch>:TRACe<Tr>:MARKer:MATH:FLATness:STARt?

### Description

Sets or reads out the number of the marker, which specifies the start frequency of the flatness function domain.

command/query

### Target

CALCulate<Ch>[:SElected] — active trace of channel <Ch>,

Or

CALCulate<Ch>:TRACe<Tr> — trace <Tr> of channel <Ch>,

<Ch> = {[1]|2|...16}

<Tr> = {[1]|2|...16}

### Parameter

<numeric> marker number from 1 to 16

### Query Response

<numeric>

### Preset Value

1

## Equivalent Softkeys

**Markers > Marker Math > Flatness > Flatness Start**

---

## Equivalent COM Command

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

---

### NOTE

This command is similar to CALCulate<Ch>[:SELEcted]:MARKer:MATH:FLATness:START only.

---

## Syntax

MkrNum =  
app.SCPI.CALCulate(Ch).SELEcted.MARKer.MATH.FLATness.DOMain.START

app.SCPI.CALCulate(Ch).SELEcted.MARKer.MATH.FLATness.DOMain.START = 1

## Type

Long (read/write)

---

### WARNING

Object MARKer has an index of 1, which can be omitted in Visual Basic but it cannot be omitted in other programming languages.

---

---

Back to [CALCulate](#)

## CALC:MARK:MATH:FLAT:DOM:STOP

### SCPI Command

CALCulate<Ch>[:SElected]:MARKer:MATH:FLATness:STOP <numeric>

CALCulate<Ch>[:SElected]:MARKer:MATH:FLATness:STOP?

Or

CALCulate<Ch>:TRACe<Tr>:MARKer:MATH:FLATness:STOP <numeric>

CALCulate<Ch>:TRACe<Tr>:MARKer:MATH:FLATness:STOP?

### Description

Sets or reads out the number of the marker, which specifies the stop frequency of the flatness function domain.

command/query

### Target

CALCulate<Ch>[:SElected] — active trace of channel <Ch>,

Or

CALCulate<Ch>:TRACe<Tr> — trace <Tr> of channel <Ch>,

<Ch> = {[1]|2|...16}

<Tr> = {[1]|2|...16}

### Parameter

<numeric> marker number from 1 to 16

### Query Response

<numeric>

### Preset Value

2

## Equivalent Softkeys

Markers > Marker Math > Flatness > Flatness Stop

---

## Equivalent COM Command

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

---

### NOTE

This command is similar to CALCulate<Ch>[:SELEcted]:MARKer:MATH:FLATness:STOP only.

---

## Syntax

MkrNum =  
app.SCPI.CALCulate(Ch).SELEcted.MARKer.MATH.FLATness.DOMain.STOP

app.SCPI.CALCulate(Ch).SELEcted.MARKer.MATH.FLATness.DOMain.STOP = 1

## Type

Long (read/write)

---

### WARNING

Object MARKer has an index of 1, which can be omitted in Visual Basic but it cannot be omitted in other programming languages.

---

---

Back to [CALCulate](#)

## CALC:MARK:REF

### SCPI Command

CALCulate<Ch>[:SElected]:MARKer:REFerence[:STATe] {OFF|ON|0|1}

CALCulate<Ch>[:SElected]:MARKer:REFerence[:STATe]?

Or

CALCulate<Ch>:TRACe<Tr>:MARKer:REFerence[:STATe] {OFF|ON|0|1}

CALCulate<Ch>:TRACe<Tr>:MARKer:REFerence[:STATe]?

### Description

Turns the reference marker ON/OFF. When the reference marker is turned ON, all the values of the other markers turn to relative values.

command/query

### Target

CALCulate<Ch>[:SElected] — active trace of channel <Ch>,

Or

CALCulate<Ch>:TRACe<Tr> — trace <Tr> of channel <Ch>,

<Ch> = {[1]|2|...16}

<Tr> = {[1]|2|...16}

### Parameter

{ON|1}      Reference marker ON

{OFF|0}      Reference marker OFF

### Query Response

{0|1}



## Preset Value

0

## Equivalent Softkeys

Markers > Reference Marker

---

## Equivalent COM Command

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

---

### NOTE

This command is similar to CALCulate<Ch>[:SElected]:MARKer:REFerence[:STATe] only.

---

## Syntax

Status = app.SCPI.CALCulate(Ch).SElected.MARKer.REFerence.STATe

app.SCPI.CALCulate(Ch).SElected.MARKer.REFerence.STATe = true

## Type

Boolean (read/write)

---

### WARNING

Object MARKer has an index of 1, which can be omitted in Visual Basic but it cannot be omitted in other programming languages.

---

---

Back to [CALCulate](#)

## CALC:MARK:SET

### SCPI Command

CALCulate<Ch>[:SElected]:MARKer<Mk>:SET <char>

Or

CALCulate<Ch>:TRACe<Tr>:MARKer<Mk>:SET <char>

### Description

Sets the value of the specified item to the value of the position of the marker.

no query

### Target

Marker <Mk> of the active trace of channel <Ch>,

Or

Marker <Mk> of the trace <Tr> of channel <Ch>,

<Ch> = {[1]2|...16}

<Tr> = {[1]2|...16}

<Mk> = {[1]2|...16}

### Parameter

<char> Specifies the type of the marker search:

<b>START</b>	Sweep start value set to the stimulus value of the marker position.
<b>STOP</b>	Sweep stop value set to the stimulus value of the marker position.
<b>CENTer</b>	Sweep center value set to the stimulus value of the marker position.
<b>RLEVel</b>	Reference value set to the response value of the marker position.
<b>DELay</b>	Delay value set to the response value of the marker position

## Equivalent Softkeys

**Markers > Marker Functions > {Marker→Start | Marker→Stop | Marker –>Center | Marker→Ref Value | Marker→Delay}**

---

## Equivalent COM Command

SCPI.CALCulate(Ch).SELEcted.MARKer(Mk).SET

---

### NOTE

This command is similar to  
CALCulate<Ch>[:SELEcted]:MARKer<Mk>:SET <char>  
only.

---

## Syntax

app.SCPI.CALCulate(Ch).SELEcted.MARKer(Mk).POSition = "STOP"

app.SCPI.CALCulate(Ch).SELEcted.MARKer(Mk).SET = "STOP"

## Type

String (read/write)

---

Back to [CALCulate](#)

## CALC:MARK:X

### SCPI Command

CALCulate<Ch>[:SElected]:MARKer<Mk>:X <stimulus>

CALCulate<Ch>[:SElected]:MARKer<Mk>:X?

Or

CALCulate<Ch>:TRACe<Tr>:MARKer<Mk>:X <stimulus>

CALCulate<Ch>:TRACe<Tr>:MARKer<Mk>:X?

### Description

Sets or reads out the stimulus value of the marker.

command/query

### Target

Marker <Mk> of the active trace of channel <Ch>,

Or

Marker <Mk> of the trace <Tr> of channel <Ch>,

<Ch> = {[1]|2|...16}

<Tr> = {[1]|2|...16}

<Mk> = {[1]|2|...16}

### Parameter

<stimulus> the stimulus value of the marker, the range is from the stimulus start value to the stimulus stop value currently set

### Unit

Hz | s | dBm

## Out of Range

Sets the value of the limit, which is closer to the specified value.

## Query Response

<numeric>

## Preset Value

Stimulus center value

## Equivalent Softkeys

## Markers > Edit Stimulus

---

### Equivalent COM Command

SCPI.CALCulate(Ch).SELEcted.MARKer(Mk).X

---

#### NOTE

This command is similar to  
CALCulate<Ch>[:SELEcted]:MARKer<Mk>:X only.

---

## Syntax

Value = app.SCPI.CALCulate(Ch).SELEcted.MARKer(Mk).X

app.SCPI.CALCulate(Ch).SELEcted.MARKer(Mk).X = 1e9

## Type

Double (read/write)

---

Back to [CALCulate](#)

## CALC:MARK:Y?

### SCPI Command

CALCulate<Ch>[:SElected]:MARKer<Mk>:Y?

Or

CALCulate<Ch>:TRACe<Tr>:MARKer<Mk>:Y?

### Description

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 data include 2 elements:

<numeric 1> real number in rectangular format, real part in polar and Smith chart formats;

<numeric 2> 0 in rectangular format, imaginary part in polar and Smith chart formats.

query only

### Target

Marker <Mk> of the active trace of channel <Ch>,

Or

Marker <Mk> of the trace <Tr> of channel <Ch>,

<Ch> = {[1]|2|...16}

<Tr> = {[1]|2|...16}

<Mk> = {[1]|2|...16}

### Query Response

<numeric 1>, <numeric 2>

## Related Commands

[CALC:MARK:REF](#)

## Equivalent Softkeys

None

---

## Equivalent COM Command

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

---

### NOTE

This command is similar to  
CALCulate<Ch>[:SElected]:MARKer<Mk>:Y? only.

---

## Syntax

Data = app.SCPI.CALCulate(Ch).SElected.MARKer(Mk).Y

## Type

Variant (array of Double) (read only)

---

Back to [CALCulate](#)

## CALC:MATH:FUNC

### SCPI Command

CALCulate<Ch>[:SElected]:MATH:FUNCtion <char>

CALCulate<Ch>[:SElected]:MATH:FUNCtion?

Or

CALCulate<Ch>:TRACe<Tr>:MATH:FUNCtion <char>

CALCulate<Ch>:TRACe<Tr>:MATH:FUNCtion?

### Description

Selects the math operation between the data trace and the memory trace. The math result replaces the data trace. If the memory trace does not exist, the command is ignored.

command/query

### Target

CALCulate<Ch>[:SElected] — active trace of channel <Ch>,

Or

CALCulate<Ch>:TRACe<Tr> — trace <Tr> of channel <Ch>,

<Ch> = {[1]|2|...16}

<Tr> = {[1]|2|...16}

### Parameter

<char> Specifies the math operation:

<b>DIVide</b>	Division Data / Mem
<b>MULTiply</b>	Multiplication Data x Mem
<b>ADD</b>	Addition Data + Mem



<b>SUBTract</b>	Subtraction Data – Mem
<b>OFF</b>	No math

**Query Response**

{OFF|DIV|MULT|SUBT|ADD}

**Preset Value**

OFF

**Related Commands**

[CALC:MATH:MEM](#)

**Equivalent Softkeys**

**Display > Data Math > {Data/Mem | Data\*Mem | Data+Mem | Data–Mem | OFF}**

**Equivalent COM Command**

SCPI.CALCulate(Ch).SELEcted.MATH.FUNCtion

<b>NOTE</b>	This command is similar to CALCulate<Ch>[:SELEcted]:MATH:FUNCtion only.
-------------	--

**Syntax**

Param = app.SCPI.CALCulate(Ch).SELEcted.MATH.FUNCtion

app.SCPI.CALCulate(Ch).SELEcted.MATH.FUNCtion = "DIV"

**Type**

String (read/write)

Back to [CALCulate](#)

## CALC:MATH:MEM

### SCPI Command

CALCulate<Ch>[:SElected]:MATH:MEMorize

Or

CALCulate<Ch>:TRACe<Tr>:MATH:MEMorize

### Description

Copies the measurement data to the memory trace. Automatically turns on the display the memory trace.

no query

### Target

CALCulate<Ch>[:SElected] — active trace of channel <Ch>,

Or

CALCulate<Ch>:TRACe<Tr> — trace <Tr> of channel <Ch>,

<Ch> = {[1]|2|...16}

<Tr> = {[1]|2|...16}

### Equivalent Softkeys

Display > Memory > Data→Memory

### Equivalent COM Command

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

#### NOTE

This command is similar to  
CALCulate<Ch>[:SElected]:MATH:MEMorize only.

### Syntax

app.SCPI.CALCulate(Ch).SElected.MATH.MEMorize

**Type**

Method

---

Back to [CALCulate](#)

## CALC:MST

### SCPI Command

CALCulate<Ch>[:SElected]:MSTatistics[:STATe] {OFF|ON|0|1}

CALCulate<Ch>[:SElected]:MSTatistics[:STATe]?

Or

CALCulate<Ch>:TRACe<Tr>:MSTatistics[:STATe] {OFF|ON|0|1}

CALCulate<Ch>:TRACe<Tr>:MSTatistics[:STATe]?

### Description

Turns the math statistics display ON/OFF.

command/query

### Target

CALCulate<Ch>[:SElected] — active trace of channel <Ch>,

Or

CALCulate<Ch>:TRACe<Tr> — trace <Tr> of channel <Ch>,

<Ch> = {[1]|2|...16}

<Tr> = {[1]|2|...16}

### Parameter

{ON|1}      Reference marker ON

{OFF|0}      Reference marker OFF

### Query Response

{0|1}

## Preset Value

0

## Equivalent Softkeys

**Markers > Marker Math > Statistics > Statistics**

---

## Equivalent COM Command

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

---

### NOTE

This command is similar to  
CALCulate<Ch>[:SElected]:MStatisticks[:STATe] only.

---

## Syntax

Status = app.SCPI.CALCulate(Ch).SElected.MStatisticks.STATe

app.SCPI.CALCulate(Ch).SElected.MStatisticks.STATe = true

## Type

Boolean (read/write)

---

Back to [CALCulate](#)

## CALC:MST:DATA?

### SCPI Command

CALCulate<Ch>[:SElected]:MSTatistics:DATA?

Or

CALCulate<Ch>:TRACe<Tr>:MSTatistics:DATA?

### Description

Reads out the math statistics values.

The statistics function is applied either over the whole range, or within the range specified by the [CALC:MST:DOM](#) command (the range limits are determined by two markers).

The data include 3 elements:

<numeric 1> Mean value;

<numeric 2> Standard deviation;

<numeric 3> Peak-to-peak (difference between the maximum value and the minimum value).

query only

### Target

CALCulate<Ch>[:SElected] — active trace of channel <Ch>,

Or

CALCulate<Ch>:TRACe<Tr> — trace <Tr> of channel <Ch>,

<Ch> = {[1]|2|...16}

<Tr> = {[1]|2|...16}

### Query Response

<numeric 1>, <numeric 2>, numeric 3>

## Related Commands

[CALC:MST](#)

## Equivalent Softkeys

None

---

## Equivalent COM Command

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

---

### NOTE

This command is similar to  
CALCulate<Ch>[:SElected]:MStatisticks:DATA? only.

---

## Syntax

Data = app.SCPI.CALCulate(Ch).SElected.MStatisticks.DATA

## Type

Variant (array of Double) (read only)

---

Back to [CALCulate](#)

## CALC:MST:DOM

### SCPI Command

CALCulate<Ch>[:SElected]:MSTatistics:DOMain[:STATe] {OFF|ON|0|1}

CALCulate<Ch>[:SElected]:MSTatistics:DOMain[:STATe]?

Or

CALCulate<Ch>:TRACe<Tr>:MSTatistics:DOMain[:STATe] {OFF|ON|0|1}

CALCulate<Ch>:TRACe<Tr>:MSTatistics:DOMain[:STATe]?

### Description

Selects either the partial frequency range or the entire frequency range to be used for math statistic calculation. The partial frequency range is limited by two markers.

command/query

### Target

CALCulate<Ch>[:SElected] — active trace of channel <Ch>,

Or

CALCulate<Ch>:TRACe<Tr> — trace <Tr> of channel <Ch>,

<Ch> = {[1]|2|...16}

<Tr> = {[1]|2|...16}

### Parameter

Choose from:

**{ON|1}**      Partial frequency range

**{OFF|0}**      Entire frequency range



## Query Response

{0|1}

## Preset Value

0

## Related Commands

[CALC:MST:DOM:STAR](#)

[CALC:MST:DOM:STOP](#)

## Equivalent Softkeys

**Markers > Marker Math > Statistics > Statistics Range**

---

## Equivalent COM Command

SCPI.CALCulate(Ch).SELEcted.MSTatistics.STATe

---

### NOTE

This command is similar to CALCulate<Ch>[:SELEcted]:MSTatistics:DOMain[:STATe] only.

---

## Syntax

Status = app.SCPI.CALCulate(Ch).SELEcted.MSTatistics.STATe

app.SCPI.CALCulate(Ch).SELEcted.MSTatistics.STATe = true

## Type

Boolean (read/write)

---

Back to [CALCulate](#)

## CALC:MST:DOM:STAR

### SCPI Command

CALCulate<Ch>[:SElected]:MSTatistics:DOMain[:MARKer]:STARt <numeric>

CALCulate<Ch>[:SElected]:MSTatistics:DOMain[:MARKer]:STARt?

Or

CALCulate<Ch>:TRACe<Tr>:MSTatistics:DOMain[:MARKer]:STARt <numeric>

CALCulate<Ch>:TRACe<Tr>:MSTatistics:DOMain[:MARKer]:STARt?

### Description

Sets or reads out the number of the marker, which specifies the start frequency of the math statistics range.

command/query

### Target

CALCulate<Ch>[:SElected] — active trace of channel <Ch>,

Or

CALCulate<Ch>:TRACe<Tr> — trace <Tr> of channel <Ch>,

<Ch> = {[1]|2|...16}

<Tr> = {[1]|2|...16}

### Parameter

<numeric> marker number from 1 to 16

### Out of Range

Sets the value of the limit, which is closer to the specified value.

### Query Response

<numeric>

## Preset Value

1

## Equivalent Softkeys

**Markers > Marker Math > Statistics > Statistics Start**

---

## Equivalent COM Command

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

---

### NOTE

This command is similar to CALCulate<Ch>[:SELEcted]:MSTatistics:DOMain[:MARKer]:START only.

---

## Syntax

MkrNum =

app.SCPI.CALCulate(Ch).SELEcted.MSTatistics.DOMain.MARKer.START

app.SCPI.CALCulate(Ch).SELEcted.MSTatistics.DOMain.MARKer.START = 3

## Type

Long (read/write)

---

Back to [CALCulate](#)

## CALC:MST:DOM:STOP

### SCPI Command

CALCulate<Ch>[:SElected]:MSTatistics:DOMain[:MARKer]:STOP <numeric>

CALCulate<Ch>[:SElected]:MSTatistics:DOMain[:MARKer]:STOP?

Or

CALCulate<Ch>:TRACe<Tr>:MSTatistics:DOMain[:MARKer]:STOP <numeric>

CALCulate<Ch>:TRACe<Tr>:MSTatistics:DOMain[:MARKer]:STOP?

### Description

Sets or reads out the number of the marker, which specifies the stop frequency of the math statistics range.

command/query

### Target

CALCulate<Ch>[:SElected] — active trace of channel <Ch>,

Or

CALCulate<Ch>:TRACe<Tr> — trace <Tr> of channel <Ch>,

<Ch> = {[1]|2|...16}

<Tr> = {[1]|2|...16}

### Parameter

<numeric> marker number from 1 to 16

### Out of Range

Sets the value of the limit, which is closer to the specified value.

### Query Response

<numeric>

## Preset Value

2

## Equivalent Softkeys

**Markers > Marker Math > Statistics > Statistics Stop**

---

## Equivalent COM Command

SCPI.CALCulate(Ch).SELEcted.MStatisticks.DOMain.MARKer.STOP

---

### NOTE

This command is similar to CALCulate<Ch>[:SELEcted]:MStatisticks:DOMAIN[:MARKer]:STOP only.

---

## Syntax

MarkerNum =  
app.SCPI.CALCulate(Ch).SELEcted.MStatisticks.DOMain.MARKer.STOP

app.SCPI.CALCulate(Ch).SELEcted.MStatisticks.DOMain.MARKer.STOP = 4

## Type

Long (read/write)

---

Back to [CALCulate](#)

## CALC:PAR:COUN

### SCPI Command

CALCulate<Ch>:PARAmeter:COUNt <numeric>

CALCulate<Ch>:PARAmeter:COUNt?

### Description

Sets or reads out the number of traces in the channel.

command/query

### Target

The channel <Ch>,

<Ch>={{[1]|2|...16}}

### Parameter

<numeric> The number of the traces in the channel from 1 to 16

### Out of Range

Sets the value of the limit, which is closer to the specified value.

### Query Response

<numeric>

### Preset Value

1

### Equivalent Softkeys

Display > Num of Traces

---

### Equivalent COM Command

SCPI.CALCulate(Ch).PARAmeter(1).COUNt

## Syntax

TraceNum = app.SCPI.CALCulate(Ch).PARAmeter.COUNT

app.SCPI.CALCulate(Ch).PARAmeter.COUNT = 2

## Type

Long (read/write)

---

### **WARNING**

Object PARAmeter has an index of 1, which can be omitted in Visual Basic but it cannot be omitted in other programming languages.

---

---

Back to [CALCulate](#)

# CALC:PAR:DEF

## SCPI Command

CALCulate<Ch>:PARAmeter<Tr>:DEFine <char>

CALCulate<Ch>:PARAmeter<Tr>:DEFine?

## Description

Selects the measurement parameter of the trace.
command/query

## Target

Trace <Tr> of channel <Ch>,

<Tr>={{1}|2|...16}

<Ch>={{1}|2|...16}

## Parameter

<char> Specifies parameter:

<b>S11, S21, S12, S22</b>	S-parameter
<b>A, B or</b>	Test receiver
<b>T1, T2</b>	
<b>R1, R2</b>	Reference receiver
<b>AUX1, AUX2 or</b>	DC Voltage
<b>V1, V2</b>	

## Query Response

{S11|S21|S12|S22|R1(n)|R2(n)|A(n)|B(n)|V1(n)|V2(n)},

Where n is the stimulus port number



## **Preset Value**

Depends on the trace number.

## **Equivalent Softkeys**

**Measurement > S11 | S21 | S12 | S22 ...**

**Measurement > Absolute > {A(1) | B(1) | R1(1) | A(2) | B(2) | R2(2)}**

---

## **Equivalent COM Command**

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

## **Syntax**

StrMeas = app.SCPI.CALCulate(Ch).PARAmeter(Tr).DEFine

app.SCPI.CALCulate(Ch).PARAmeter(Tr).DEFine = "S11"

## **Type**

String (read/write)

---

Back to [CALCulate](#)

## CALC:PAR:SEL

### SCPI Command

CALCulate<Ch>:PARAmeter<Tr>:SElect

### Description

Selects the active trace in the channel.

**Note:** If the trace number is greater than the number of the traces displayed in the channel, an error occurs and the command is ignored.

no query

### Target

Trace <Tr> of channel <Ch>,

<Tr>={[1]|2|...16}

<Ch>={[1]|2|...16}

### Related Commands

[CALC:PAR:COUN](#)

[SERV:CHAN:TRAC:ACT?](#)

### Equivalent Softkeys

Display > Active Trace/Channel > Active Channel

Display > Active Trace/Channel > Active Trace

---

### Equivalent COM Command

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

### Syntax

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

**Type**

Method

---

Back to [CALCulate](#)

## CALC:PAR:SPOR

### SCPI Command

CALCulate<Ch>:PARAmeter<Tr>:SPORt <port>

CALCulate<Ch>:PARAmeter<Tr>:SPORt?

### Description

Sets or reads out the number of the stimulus port when performing absolute measurements or DC Voltage measurements.

command/query

### Target

Trace <Tr> of channel <Ch>,

<Tr>={[1]|2|...16}

<Ch>={[1]|2|...16}

### Parameter

<port> the number of the stimulus port

### Out of Range

Error occurs. The command is ignored.

### Query Response

<port>

### Preset Value

1

### Equivalent Softkeys

Measurement > Absolute > {A(1) | B(1) | R1(1) | A(2) | B(2) | R2(2)}

---

### Equivalent COM Command

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

### **Syntax**

StimPort = app.SCPI.CALCulate(Ch).PARAmeter(Tr).SPORT

app.SCPI.CALCulate(Ch).PARAmeter(Tr).SPORT = 1

### **Type**

Long (read/write)

---

Back to [CALCulate](#)

## CALC:RLIM

### SCPI Command

CALCulate<Ch>[:SElected]:RLIMit[:STATe] {OFF|ON|0|1}

CALCulate<Ch>[:SElected]:RLIMit[:STATe]?

Or

CALCulate<Ch>:TRACe<Tr>:RLIMit[:STATe] {OFF|ON|0|1}

CALCulate<Ch>:TRACe<Tr>:RLIMit[:STATe]?

### Description

Turns the ripple limit test ON/OFF.

command/query

### Target

CALCulate<Ch>[:SElected] — active trace of channel <Ch>,

Or

CALCulate<Ch>:TRACe<Tr> — trace <Tr> of channel <Ch>,

<Ch> = {[1]|2|...16}

<Tr> = {[1]|2|...16}

### Parameter

Choose from:

{ON|1}      ON

{OFF|0}      OFF

### Query Response

{0|1}

## Preset Value

0

## Equivalent Softkeys

## Analysis > Ripple Limit > Ripple Test

---

## Equivalent COM Command

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

---

### NOTE

This command is similar to  
CALCulate<Ch>[:SElected]:RLIMit[:STATe] only.

---

## Syntax

Status = app.SCPI.CALCulate(Ch).SElected.RLIMit.STATe

app.SCPI.CALCulate(Ch).SElected.RLIMit.STATe = true

## Type

Boolean (read/write)

---

Back to [CALCulate](#)

## CALC:RLIM:DATA

### SCPI Command

CALCulate<Ch>[:SElected]:RLIMit:DATA <numeric list>

CALCulate<Ch>[:SElected]:RLIMit:DATA?

Or

CALCulate<Ch>:TRACe<Tr>:RLIMit:DATA <numeric list>

CALCulate<Ch>:TRACe<Tr>:RLIMit:DATA?

### Description

Sets 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  $N$ :

<numeric 1> the number of limit line segments  $N$  is the integer from 0 to 12. Setting 0 clears the limit line.

<numeric  $4n-2$ > type of the  $n$ -th limit line segment

0: Off.

1: On

<numeric  $4n-1$ > the stimulus value in the beginning point of the  $n$ -th segment

<numeric  $4n-0$ > the stimulus value in the end point of the  $n$ -th segment

<numeric  $4n+1$ > the ripple limit value of the  $n$ -th segment.

**Note:** If the array size is not  $1 + 4N$ , where  $N$  is <numeric 1>, an error occurs. If <numeric  $4n-2$ > is less than 0 or more than 1, an error occurs. When <numeric  $4n-1$ >, <numeric  $4n-0$ >, and <numeric  $4n+1$ > elements are out of allowable range, the value is set to the limit, which is closer to the specified value.

command/query



## Target

CALCulate<Ch>[:SElected] — active trace of channel <Ch>,

Or

CALCulate<Ch>:TRACe<Tr> — trace <Tr> of channel <Ch>,

<Ch> = {[1]|2|...16}

<Tr> = {[1]|2|...16}

## Query Response

<numeric 1>, <numeric 2>, ...<numeric 4N+1>

## Equivalent Softkeys

**Analysis > Ripple Limit > Edit Ripple Limit**

---

## Equivalent COM Command

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

---

### NOTE

This command is similar to  
CALCulate<Ch>[:SElected]:RLIMit:DATA only.

---

## Syntax

Data = app.SCPI.CALCulate(Ch).SElected.RLIMit.DATA

app.SCPI.CALCulate(Ch).SElected.RLIMit.DATA = Array(1,1,800,900,10)

## Type

Variant (array of Double) (read only)

---

Back to [CALCulate](#)

## CALC:RLIM:DISP:LINE

### SCPI Command

CALCulate<Ch>[:SElected]:RLIMit:DISPlay:LINE {OFF|ON|0|1}

CALCulate<Ch>[:SElected]:RLIMit:DISPlay:LINE?

Or

CALCulate<Ch>:TRACe<Tr>:RLIMit:DISPlay:LINE {OFF|ON|0|1}

CALCulate<Ch>:TRACe<Tr>:RLIMit:DISPlay:LINE?

### Description

Turns the ripple limit line display ON/OFF.

command/query

### Target

CALCulate<Ch>[:SElected] — active trace of channel <Ch>,

Or

CALCulate<Ch>:TRACe<Tr> — trace <Tr> of channel <Ch>,

<Ch> = {[1]|2|...16}

<Tr> = {[1]|2|...16}

### Parameter

Choose from:

{ON|1}      ON

{OFF|0}      OFF

### Query Response

{0|1}

## Preset Value

0

## Equivalent Softkeys

Analysis > Ripple Limit > Ripple Limit

---

## Equivalent COM Command

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

---

### NOTE

This command is similar to  
CALCulate<Ch>[:SElected]:RLIMit:DISPlay:LINE only.

---

## Syntax

Status = app.SCPI.CALCulate(Ch).SElected.RLIMit.DISPlay.LINE

app.SCPI.CALCulate(Ch).SElected.RLIMit.DISPlay.LINE = true

## Type

Boolean (read/write)

---

Back to [CALCulate](#)

## CALC:RLIM:DISP:SEL

### SCPI Command

CALCulate<Ch>[:SElected]:RLIMit:DISPlay:SElect <numeric>

CALCulate<Ch>[:SElected]:RLIMit:DISPlay:SElect?

Or

CALCulate<Ch>:TRACe<Tr>:RLIMit:DISPlay:SElect <numeric>

CALCulate<Ch>:TRACe<Tr>:RLIMit:DISPlay:SElect?

### Description

Sets or reads out the number of the ripple limit test band selected for the ripple value display.

command/query

### Target

CALCulate<Ch>[:SElected] — active trace of channel <Ch>,

Or

CALCulate<Ch>:TRACe<Tr> — trace <Tr> of channel <Ch>,

<Ch> = {[1]2|...16}

<Tr> = {[1]2|...16}

### Parameter

<numeric>, range from 1 to 12

### Out of Range

Sets the value of the limit, which is closer to the specified value.

### Query Response

<numeric>

## Preset Value

1

## Equivalent Softkeys

Analysis > Ripple Limit > Ripple Value Band

---

## Equivalent COM Command

SCPI.CALCulate(Ch).SELEcted.RLIMit.DISPlay.SELEct

---

### NOTE

This command is similar to  
CALCulate<Ch>[:SELEcted]:RLIMit:DISPlay:SELEct only.

---

## Syntax

Value = app.SCPI.CALCulate(Ch).SELEcted.RLIMit.DISPlay.SELEct

app.SCPI.CALCulate(Ch).SELEcted.RLIMit.DISPlay.SELEct = 2

## Type

Long (read/write)

---

Back to [CALCulate](#)

## CALC:RLIM:DISP:VAL

### SCPI Command

CALCulate<Ch>[:SElected]:RLIMit:DISPlay:VALue <char>

CALCulate<Ch>[:SElected]:RLIMit:DISPlay:VALue?

Or

CALCulate<Ch>:TRACe<Tr>:RLIMit:DISPlay:VALue <char>

CALCulate<Ch>:TRACe<Tr>:RLIMit:DISPlay:VALue?

### Description

Selects the display type of the ripple value in the specified band.

command/query

### Target

CALCulate<Ch>[:SElected] — active trace of channel <Ch>,

Or

CALCulate<Ch>:TRACe<Tr> — trace <Tr> of channel <Ch>,

<Ch> = {[1]|2|...16}

<Tr> = {[1]|2|...16}

### Parameter

<char> Specifies the math operation:

<b>OFF</b>	Ripple value display OFF
<b>ABSolute</b>	Absolute value
<b>MARgin</b>	Margin (difference between the ripple limit and the absolute value)

## Query Response

{OFF|ABS|MAR}

## Preset Value

OFF

## Equivalent Softkeys

## Analysis > Ripple Limit > Ripple Value

---

## Equivalent COM Command

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

---

### NOTE

This command is similar to  
CALCulate<Ch>[:SElected]:RLIMit:DISPlay:VALue only.

---

## Syntax

Param = app.SCPI.CALCulate(Ch).SElected.RLIMit.DISPlay.VALue

app.SCPI.CALCulate(Ch).SElected.RLIMit.DISPlay.VALue = "ABS"

## Type

String (read/write)

---

Back to [CALCulate](#)

## CALC:RLIM:FAIL?

### SCPI Command

CALCulate<Ch>[:SElected]:RLIMit:FAIL?

Or

CALCulate<Ch>:TRACe<Tr>:RLIMit:FAIL?

### Description

Reads out the ripple limit test result.

query only

### Target

CALCulate<Ch>[:SElected] — active trace of channel <Ch>,

Or

CALCulate<Ch>:TRACe<Tr> — trace <Tr> of channel <Ch>,

<Ch> = {[1]|2|...16}

<Tr> = {[1]|2|...16}

### Parameter

**1**      Fail

**0**      Pass

### Equivalent Softkeys

None



---

## Equivalent COM Command

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

---

### NOTE

This command is similar to  
CALCulate<Ch>[:SElected]:RLIMit:FAIL? only.

---

## Syntax

Status = app.SCPI.CALCulate(Ch).SElected.RLIMit.FAIL

## Type

Boolean (read/write)

---

Back to [CALCulate](#)

## CALC:RLIM:REP?

### SCPI Command

CALCulate<Ch>[:SElected]:RLIMit:REPort[:DATA]?

Or

CALCulate<Ch>:TRACe<Tr>:RLIMit:REPort[:DATA]?

### Description

Reads out the data array, which is the ripple limit test result.

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:

<numeric 1> N total number of the bands

<numeric 3n–1> n number of the band

<numeric 3n–0> Ripple value in the n–th band

<numeric 3n+1> Ripple limit test result in the n–th band:

0 — Pass

1 — Fail

query only

### Target

CALCulate<Ch>[:SElected] — active trace of channel <Ch>,

Or

CALCulate<Ch>:TRACe<Tr> — trace <Tr> of channel <Ch>,

<Ch> = {[1]2|...16}

<Tr> = {[1]2|...16}

## Query Response

<numeric 1>, <numeric 2>, ...<numeric 3N+1>

## Equivalent Softkeys

None

---

## Equivalent COM Command

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

---

### NOTE

This command is similar to  
CALCulate<Ch>[:SELEcted]:RLIMit:REPort[:DATA]? only.

---

## Syntax

Data = app.SCPI.CALCulate(Ch).SELEcted.RLIMit.REPort.DATA

## Type

Variant (array of Double) (read only)

---

Back to [CALCulate](#)

## CALC:SMO

### SCPI Command

CALCulate<Ch>[:SElected]:SMOothing[:STATe] {OFF|ON|0|1}

CALCulate<Ch>[:SElected]:SMOothing[:STATe]?

Or

CALCulate<Ch>:TRACe<Tr>:SMOothing[:STATe] {OFF|ON|0|1}

CALCulate<Ch>:TRACe<Tr>:SMOothing[:STATe]?

### Description

Turns the trace smoothing ON/OFF.

command/query

### Target

CALCulate<Ch>[:SElected] — active trace of channel <Ch>,

Or

CALCulate<Ch>:TRACe<Tr> — trace <Tr> of channel <Ch>,

<Ch> = {[1]|2|...16}

<Tr> = {[1]|2|...16}

### Parameter

Choose from:

{ON|1}      ON

{OFF|0}      OFF

### Query Response

{0|1}

## Preset Value

0

## Equivalent Softkeys

## Average > Smoothing

---

## Equivalent COM Command

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

---

### NOTE

This command is similar to  
CALCulate<Ch>[:SElected]:SMOothing[:STATe] only.

---

## Syntax

Status = app.SCPI.CALCulate(Ch).SElected.SMOothing.STATe

app.SCPI.CALCulate(Ch).SElected.SMOothing.STATe = true

## Type

Boolean (read/write)

---

Back to [CALCulate](#)

## CALC:SMO:APER

### SCPI Command

CALCulate<Ch>[:SElected]:SMOothing:APERture <numeric>

CALCulate<Ch>[:SElected]:SMOothing:APERture?

Or

CALCulate<Ch>:TRACe<Tr>:SMOothing:APERture <numeric>

CALCulate<Ch>:TRACe<Tr>:SMOothing:APERture?

### Description

Sets or reads out the smoothing aperture when performing smoothing function.

command/query

### Target

CALCulate<Ch>[:SElected] — active trace of channel <Ch>,

Or

CALCulate<Ch>:TRACe<Tr> — trace <Tr> of channel <Ch>,

<Ch> = {[1]|2|...16}

<Tr> = {[1]|2|...16}

### Parameter

<numeric> the smoothing aperture from 0.01 to 20

### Unit

% (percent)

### Out of Range

Sets the value of the limit, which is closer to the specified value.

## Query Response

<numeric>

## Preset Value

1

## Equivalent Softkeys

Average > Smo Aperture

---

## Equivalent COM Command

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

---

### NOTE

This command is similar to  
CALCulate<Ch>[:SElected]:SMOothing:APERture only.

---

## Syntax

Value = app.SCPI.CALCulate(Ch).SElected.SMOothing.APERture

app.SCPI.CALCulate(Ch).SElected.SMOothing.APERture = 1.5

## Type

Double (read/write)

---

Back to [CALCulate](#)

## CALC:TRAN:TIME

### SCPI Command

CALCulate<Ch>[:SElected]:TRANSform:TIME[:TYPE] <char>

CALCulate<Ch>[:SElected]:TRANSform:TIME[:TYPE]?

Or

CALCulate<Ch>:TRACe<Tr>:TRANSform:TIME[:TYPE] <char>

CALCulate<Ch>:TRACe<Tr>:TRANSform:TIME[:TYPE]?

### Description

Selects the transformation type for the time domain transformation function: band-pass or low-pass.

command/query

### Target

CALCulate<Ch>[:SElected] — active trace of channel <Ch>,

Or

CALCulate<Ch>:TRACe<Tr> — trace <Tr> of channel <Ch>,

<Ch> = {[1]2|...16}

<Tr> = {[1]2|...16}

### Parameter

<char> Specifies the transformation type:

**BPASs**      Ban-dpass

**LPASs**      Low-pass

### Query Response

{BPAS|LPAS}



## Preset Value

BPAS

## Equivalent Softkeys

**Analysis > Time Domain > Type > {Bandpass | Lowpass Step | Lowpass Impulse}**

---

## Equivalent COM Command

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

---

### NOTE

This command is similar to  
CALCulate<Ch>[:SElected]:TRANSform:TIME[:TYPE]  
only.

---

## Syntax

Param = app.SCPI.CALCulate(Ch).SElected.TRANSform.TIME.TYPE

app.SCPI.CALCulate(Ch).SElected.TRANSform.TIME.TYPE = "STEP"

## Type

String (read/write)

---

Back to [CALCulate](#)

## CALC:TRAN:TIME:CEN

### SCPI Command

CALCulate<Ch>[:SElected]:TRANSform:TIME:CENTer <time>

CALCulate<Ch>[:SElected]:TRANSform:TIME:CENTer?

Or

CALCulate<Ch>:TRACe<Tr>:TRANSform:TIME:CENTer <time>

CALCulate<Ch>:TRACe<Tr>:TRANSform:TIME:CENTer?

### Description

Sets or reads out the time domain center value when the time domain transformation function is turned ON.

command/query

### Target

CALCulate<Ch>[:SElected] — active trace of channel <Ch>,

Or

CALCulate<Ch>:TRACe<Tr> — trace <Tr> of channel <Ch>,

<Ch> = {[1]2|...16}

<Tr> = {[1]2|...16}

### Parameter

<time> the time domain center value, the range varies depending on the specified frequency range and the number of points

### Unit

sec (second)

### Out of Range

Sets the value of the limit, which is closer to the specified value.

## Query Response

<numeric>

## Preset Value

1

## Related Commands

[CALC:TRAN:TIME:UNIT](#)

## Equivalent Softkeys

Analysis > Time Domain > Center

---

## Equivalent COM Command

SCPI.CALCulate(Ch).SELEcted.TRANSform.TIME.CENTer

---

### NOTE

This command is similar to  
CALCulate<Ch>[:SELEcted]:TRANSform.TIME.CENTer  
only.

---

## Syntax

Value = app.SCPI.CALCulate(Ch).SELEcted.TRANSform.TIME.CENTer

app.SCPI.CALCulate(Ch).SELEcted.TRANSform.TIME.CENTer = 1e-8

## Type

Double (read/write)

---

Back to [CALCulate](#)

## CALC:TRAN:TIME:DC:VAL

### SCPI Command

CALCulate<Ch>[:SElected]:TRANSform:TIME:DC:VALue <numeric>

CALCulate<Ch>[:SElected]:TRANSform:TIME:DC:VALue?

Or

CALCulate<Ch>:TRACe<Tr>:TRANSform:TIME:DC:VALue <numeric>

CALCulate<Ch>:TRACe<Tr>:TRANSform:TIME:DC:VALue?

### Description

Sets or reads out the DC value used in the low-pass type of the time domain transformation, when the DC extrapolation is OFF.

command/query

### Target

CALCulate<Ch>[:SElected] — active trace of channel <Ch>,

Or

CALCulate<Ch>:TRACe<Tr> — trace <Tr> of channel <Ch>,

<Ch> = {[1]|2|...16}

<Tr> = {[1]|2|...16}

### Parameter

<numeric> the DC value, from -1.0 to 1.0

### Out of Range

Sets the value of the limit, which is closer to the specified value.

### Query Response

<numeric>

**Preset Value**

1.0

**Related Commands**

[CALC:TRAN:TIME:EXTR:DC](#)

**Equivalent Softkeys**

Analysis > Time Domain > DC Value

---

**Equivalent COM Command**

None

---

Back to [CALCulate](#)

## CALC:TRAN:TIME:EXTR:DC

### SCPI Command

CALCulate<Ch>[:SElected]:TRANSform:TIME:EXTRapolate:DC[:STATe] {OFF|ON|0|1}

CALCulate<Ch>[:SElected]:TRANSform:TIME:EXTRapolate:DC[:STATe]?

Or

CALCulate<Ch>:TRACe<Tr>:TRANSform:TIME:EXTRapolate:DC[:STATe] {OFF|ON|0|1}

CALCulate<Ch>:TRACe<Tr>:TRANSform:TIME:EXTRapolate:DC[:STATe]?

### Description

Turns ON/OFF the DC extrapolation, when the time domain transformation function is turned ON. The DC value is used in the low-pass type of transformation. When the DC extrapolation is OFF the DC value set by the [CALC:TRAN:TIME:DC:VAL](#) command is used.

command/query

### Target

CALCulate<Ch>[:SElected] — active trace of channel <Ch>,

Or

CALCulate<Ch>:TRACe<Tr> — trace <Tr> of channel <Ch>,

<Ch> = {[1]|2|...16}

<Tr> = {[1]|2|...16}

### Parameter

{ON|1}            ON

{OFF|0}           OFF

**Query Response**

{0|1}

**Preset Value**

1

**Equivalent Softkeys**

Analysis > Time Domain > Extrapolate DC

---

**Equivalent COM Command**

None

---

Back to [CALCulate](#)

## CALC:TRAN:TIME:IMP:WIDT

### SCPI Command

CALCulate<Ch>[:SElected]:TRANSform:TIME:IMPulse:WIDTh <time>

CALCulate<Ch>[:SElected]:TRANSform:TIME:IMPulse:WIDTh?

Or

CALCulate<Ch>:TRACe<Tr>:TRANSform:TIME:IMPulse:WIDTh <time>

CALCulate<Ch>:TRACe<Tr>:TRANSform:TIME:IMPulse:WIDTh?

### Description

Sets or reads out the impulse width (time domain transformation resolution), coupled with the Kaiser–Bessel window shape  $\beta$  parameter. The impulse width setting changes the  $\beta$  parameter, and setting of  $\beta$  parameter changes the impulse width.

command/query

### Target

CALCulate<Ch>[:SElected] — active trace of channel <Ch>,

Or

CALCulate<Ch>:TRACe<Tr> — trace <Tr> of channel <Ch>,

<Ch> = {[1]|2|...16}

<Tr> = {[1]|2|...16}

### Parameter

<time> the impulse width, the range varies depending on the specified frequency range and the number of points

### Unit

sec (second)



## Out of Range

Sets the value of the limit, which is closer to the specified value.

## Query Response

<numeric>

## Equivalent Softkeys

### Analysis > Time Domain > Window > Impulse Width

(when the transformation type is set to Bandpass or Lowpass Impulse)

---

## Equivalent COM Command

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

---

### NOTE

This command is similar to CALCulate<Ch>[:SElected]:TRANSform:TIME:IMPulse:WIDTh only.

---

## Syntax

Value = app.SCPI.CALCulate(Ch).SElected.TRANSform.TIME.IMPulse.WIDTh

app.SCPI.CALCulate(Ch).SElected.TRANSform.TIME.IMPulse.WIDTh = 1e-8

## Type

Double (read/write)

---

Back to [CALCulate](#)

## CALC:TRAN:TIME:KBES

### SCPI Command

CALCulate<Ch>[:SElected]:TRANSform:TIME:KBESsel <numeric>

CALCulate<Ch>[:SElected]:TRANSform:TIME:KBESsel?

Or

CALCulate<Ch>:TRACe<Tr>:TRANSform:TIME:KBESsel <numeric>

CALCulate<Ch>:TRACe<Tr>:TRANSform:TIME:KBESsel?

### Description

Sets or reads out the  $\beta$  parameter, which controls the Kaiser–Bessel window shape when performing the time domain transformation.

command/query

### Target

CALCulate<Ch>[:SElected] — active trace of channel <Ch>,

Or

CALCulate<Ch>:TRACe<Tr> — trace <Tr> of channel <Ch>,

<Ch> = {[1]|2|...16}

<Tr> = {[1]|2|...16}

### Parameter

<numeric>  $\beta$  parameter from 0 to 13

### Out of Range

Sets the value of the limit, which is closer to the specified value.

### Query Response

<numeric>

## Preset Value

6

## Equivalent Softkeys

Analysis > Time Domain > Window > Kaiser Beta

---

## Equivalent COM Command

SCPI.CALCulate(Ch).SELEcted.TRANSform.TIME.KBESsel

---

### NOTE

This command is similar to  
CALCulate<Ch>[:SELEcted]:TRANSform:TIME:KBESsel  
only.

---

## Syntax

Value = app.SCPI.CALCulate(Ch).SELEcted.TRANSform.TIME.KBESsel

app.SCPI.CALCulate(Ch).SELEcted.TRANSform.TIME.KBESsel = 13

## Type

Double (read/write)

---

Back to [CALCulate](#)

## CALC:TRAN:TIME:LPFR

### SCPI Command

CALCulate<Ch>[:SElected]:TRANSform:TIME:LPFRequency

Or

CALCulate<Ch>:TRACe<Tr>:TRANSform:TIME:LPFRequency

### Description

Changes the frequency range to the harmonic grid in order to match with the low-pass type of the time domain transformation function.

no query

### Target

CALCulate<Ch>[:SElected] — active trace of channel <Ch>,

Or

CALCulate<Ch>:TRACe<Tr> — trace <Tr> of channel <Ch>,

<Ch> = {[1]|2|...16}

<Tr> = {[1]|2|...16}

### Equivalent Softkeys

**Analysis > Time Domain > Set Frequency Low Pass**

### Equivalent COM Command

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

#### NOTE

This command is similar to CALCulate<Ch>[:SElected]:TRANSform:TIME:LPFRequency only.

## Syntax

app.SCPI.CALCulate(Ch).SELEcted.TRANSform.TIME.LPFRequency

## Type

Method

---

Back to [CALCulate](#)

## CALC:TRAN:TIME:REFL:TYPE

### SCPI Command

CALCulate<Ch>[:SElected]:TRANSform:TIME:REFLection:TYPE <char>

CALCulate<Ch>[:SElected]:TRANSform:TIME:REFLection:TYPE?

Or

CALCulate<Ch>:TRACe<Tr>:TRANSform:TIME:REFLection:TYPE <char>

CALCulate<Ch>:TRACe<Tr>:TRANSform:TIME:REFLection:TYPE?

### Description

Selects the reflection distance either one way or round trip for the time domain transformation function.

command/query

### Target

CALCulate<Ch>[:SElected] — active trace of channel <Ch>,

Or

CALCulate<Ch>:TRACe<Tr> — trace <Tr> of channel <Ch>,

<Ch> = {[1]|2|...16}

<Tr> = {[1]|2|...16}

### Parameter

<char> Choose from:

**RTRip**      Round Trip

**OWAY**      One Way

### Query Response

{RTR|OWAY}

## Preset Value

RTR

## Equivalent Softkeys

**Analysis > Time Domain > Reflection Type > {Round Trip | One Way}**

---

## Equivalent COM Command

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

---

### NOTE

This command is similar to CALCulate<Ch>[:SElected]:TRANSform:TIME:REFLection:TYPE only.

---

## Syntax

Param = app.SCPI.CALCulate(Ch).SElected.TRANSform.TIME.REFLection.TYPE

app.SCPI.CALCulate(Ch).SElected.TRANSform.TIME.REFLection.TYPE = "RTR"

## Type

String (read/write)

---

Back to [CALCulate](#)

## CALC:TRAN:TIME:SPAN

### SCPI Command

CALCulate<Ch>[:SElected]:TRANSform:TIME:SPAN <time>

CALCulate<Ch>[:SElected]:TRANSform:TIME:SPAN?

Or

CALCulate<Ch>:TRACe<Tr>:TRANSform:TIME:SPAN <time>

CALCulate<Ch>:TRACe<Tr>:TRANSform:TIME:SPAN?

### Description

Sets or reads out the time domain span value when the time domain transformation function is turned ON.

command/query

### Target

CALCulate<Ch>[:SElected] — active trace of channel <Ch>,

Or

CALCulate<Ch>:TRACe<Tr> — trace <Tr> of channel <Ch>,

<Ch> = {[1]2|...16}

<Tr> = {[1]2|...16}

### Parameter

<time> the time domain span value, the range varies depending on the specified frequency range and the number of points

### Unit

sec (second)

### Out of Range

Sets the value of the limit, which is closer to the specified value.



## Query Response

<numeric>

## Preset Value

2e-8

## Related Commands

[CALC:TRAN:TIME:UNIT](#)

## Equivalent Softkeys

Analysis > Time Domain > Span

---

## Equivalent COM Command

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

---

### NOTE

This command is similar to  
CALCulate<Ch>[:SELEcted]:TRANSform:TIME:SPAN only.

---

## Syntax

Value = app.SCPI.CALCulate(Ch).SELEcted.TRANSform.TIME.SPAN

app. SCPI.CALCulate(Ch).SELEcted.TRANSform.TIME.SPAN = 1e-8

## Type

Double (read/write)

---

Back to [CALCulate](#)

## CALC:TRAN:TIME:STAR

### SCPI Command

CALCulate<Ch>[:SElected]:TRANSform:TIME:STARt <time>

CALCulate<Ch>[:SElected]:TRANSform:TIME:STARt?

Or

CALCulate<Ch>:TRACe<Tr>:TRANSform:TIME:STARt <time>

CALCulate<Ch>:TRACe<Tr>:TRANSform:TIME:STARt?

### Description

Sets or reads out the time domain start value when the time domain transformation function is turned ON.

command/query

### Target

CALCulate<Ch>[:SElected] — active trace of channel <Ch>,

Or

CALCulate<Ch>:TRACe<Tr> — trace <Tr> of channel <Ch>,

<Ch> = {[1]|2|...16}

<Tr> = {[1]|2|...16}

### Parameter

<time> the time domain start value, the range varies depending on the specified frequency range and the number of points

### Unit

sec (second)

### Out of Range

Sets the value of the limit, which is closer to the specified value.

## Query Response

<numeric>

## Preset Value

-1e-8

## Related Commands

[CALC:TRAN:TIME:UNIT](#)

## Equivalent Softkeys

Analysis > Time Domain > Start

---

## Equivalent COM Command

SCPI.CALCulate(Ch).SELEcted.TRANSform.TIME.STARt

---

### NOTE

This command is similar to  
CALCulate<Ch>[:SELEcted]:TRANSform:TIME:STARt  
only.

---

## Syntax

Value = app.SCPI.CALCulate(Ch).SELEcted.TRANSform.TIME.STARt

app.SCPI.CALCulate(Ch).SELEcted.TRANSform.TIME.STARt = 1e-8

## Type

Double (read/write)

---

Back to [CALCulate](#)

## CALC:TRAN:TIME:STOP

### SCPI Command

CALCulate<Ch>[:SElected]:TRANSform:TIME:STOP <time>

CALCulate<Ch>[:SElected]:TRANSform:TIME:STOP?

Or

CALCulate<Ch>:TRACe<Tr>:TRANSform:TIME:STOP <time>

CALCulate<Ch>:TRACe<Tr>:TRANSform:TIME:STOP?

### Description

Sets or reads out the time domain stop value when the time domain transformation function is turned ON.

command/query

### Target

CALCulate<Ch>[:SElected] — active trace of channel <Ch>,

Or

CALCulate<Ch>:TRACe<Tr> — trace <Tr> of channel <Ch>,

<Ch> = {[1]|2|...16}

<Tr> = {[1]|2|...16}

### Parameter

<time> the time domain stop value, the range varies depending on the specified frequency range and the number of points

### Unit

sec (second)

### Out of Range

Sets the value of the limit, which is closer to the specified value.

## Query Response

<numeric>

## Preset Value

+1e-8

## Related Commands

[CALC:TRAN:TIME:UNIT](#)

## Equivalent Softkeys

Analysis > Time Domain > Stop

---

## Equivalent COM Command

SCPI.CALCulate(Ch).SELEcted.TRANSform.TIME.STOP

---

### NOTE

This command is similar to  
CALCulate<Ch>[:SELEcted]:TRANSform:TIME:STOP only.

---

## Syntax

Value = app.SCPI.CALCulate(Ch).SELEcted.TRANSform.TIME.STOP

app.SCPI.CALCulate(Ch).SELEcted.TRANSform.TIME.STOP = 2e-8

## Type

Double (read/write)

---

Back to [CALCulate](#)

## CALC:TRAN:TIME:STAT

### SCPI Command

CALCulate<Ch>[:SElected]:TRANSform:TIME:STATe {OFF|ON|0|1}

CALCulate<Ch>[:SElected]:TRANSform:TIME:STATe?

Or

CALCulate<Ch>:TRACe<Tr>:TRANSform:TIME:STATe {OFF|ON|0|1}

CALCulate<Ch>:TRACe<Tr>:TRANSform:TIME:STATe?

### Description

Turns the time domain transformation function ON/OFF.

command/query

### Target

CALCulate<Ch>[:SElected] — active trace of channel <Ch>,

Or

CALCulate<Ch>:TRACe<Tr> — trace <Tr> of channel <Ch>,

<Ch> = {[1]|2|...16}

<Tr> = {[1]|2|...16}

### Parameter

{ON|1}      ON

{OFF|0}     OFF

### Query Response

{0|1}

## Preset Value

0

## Equivalent Softkeys

Analysis > Time Domain > Time Domain

---

## Equivalent COM Command

SCPI.CALCulate(Ch).SELEcted.TRANSform.TIME.STATe

---

### NOTE

This command is similar to  
CALCulate<Ch>[:SELEcted]:TRANSform:TIME:STATe  
only.

---

## Syntax

Status = app.SCPI.CALCulate(Ch).SELEcted.TRANSform.TIME.STATe

app.SCPI.CALCulate(Ch).SELEcted.TRANSform.TIME.STATe = true

## Type

Boolean (read/write)

---

Back to [CALCulate](#)

## CALC:TRAN:TIME:STEP:RTIM

### SCPI Command

CALCulate<Ch>[:SElected]:TRANSform:TIME:STEP:RTIME <time>

CALCulate<Ch>[:SElected]:TRANSform:TIME:STEP:RTIME?

Or

CALCulate<Ch>:TRACe<Tr>:TRANSform:TIME:STEP:RTIME <time>

CALCulate<Ch>:TRACe<Tr>:TRANSform:TIME:STEP:RTIME?

### Description

Sets or reads out the rise time of the step signal (time domain transformation resolution), coupled with the Kaiser–Bessel window shape  $\beta$  parameter. The impulse width setting changes the  $\beta$  parameter, and setting of  $\beta$  parameter changes the impulse width.

command/query

### Target

CALCulate<Ch>[:SElected] — active trace of channel <Ch>,

Or

CALCulate<Ch>:TRACe<Tr> — trace <Tr> of channel <Ch>,

<Ch> = {[1]|2|...16}

<Tr> = {[1]|2|...16}

### Parameter

<time> the impulse width, the range varies depending on the specified frequency range and the number of points

### Unit

sec (second)



## Out of Range

Sets the value of the limit, which is closer to the specified value.

## Query Response

<numeric>

## Equivalent Softkeys

### Analysis > Time Domain > Window > Impulse Width

(when the transformation type is set to Lowpass Step)

---

## Equivalent COM Command

SCPI.CALCulate(Ch).SELEcted.TRANSform.TIME.STARt

---

### NOTE

This command is similar to  
CALCulate<Ch>[:SELEcted]:TRANSform:TIME:STEP:RTI  
Me only.

---

## Syntax

Value = app.SCPI.CALCulate(Ch).SELEcted.TRANSform.TIME.STARt

app.SCPI.CALCulate(Ch).SELEcted.TRANSform.TIME.STARt = 1e-8

## Type

Double (read/write)

---

Back to [CALCulate](#)

## CALC:TRAN:TIME:STIM

### SCPI Command

CALCulate<Ch>[:SElected]:TRANSform:TIME:STIMulus <char>

CALCulate<Ch>[:SElected]:TRANSform:TIME:STIMulus?

Or

CALCulate<Ch>:TRACe<Tr>:TRANSform:TIME:STIMulus <char>

CALCulate<Ch>:TRACe<Tr>:TRANSform:TIME:STIMulus?

### Description

Selects the stimulus type for the time domain transformation function: impulse or step. The stimulus type is valid for the low-pass devices. For the band-pass devices the impulse type is always used.

command/query

### Target

CALCulate<Ch>[:SElected] — active trace of channel <Ch>,

Or

CALCulate<Ch>:TRACe<Tr> — trace <Tr> of channel <Ch>,

<Ch> = {[1]|2|...16}

<Tr> = {[1]|2|...16}

### Parameter

<char> Specifies the stimulus type:

**IMPulse**      Impulse

**STEP**          Step

## Query Response

{IMP|STEP}

## Preset Value

IMP

## Equivalent Softkeys

Analysis > Time Domain > Type > {Bandpass | Lowpass Step | Lowpass Impulse}

---

## Equivalent COM Command

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

---

### NOTE

This command is similar to CALCulate<Ch>[:SElected]:TRANSform:TIME:STIMulus only.

---

## Syntax

Param = app.SCPI.CALCulate(Ch).SElected.TRANSform.TIME.STIMulus

app.SCPI.CALCulate(Ch).SElected.TRANSform.TIME.STIMulus = "STEP"

## Type

String (read/write)

---

Back to [CALCulate](#)

## CALC:TRAN:TIME:UNIT

### SCPI Command

CALCulate<Ch>[:SElected]:TRANSform:TIME:UNIT <char>

CALCulate<Ch>[:SElected]:TRANSform:TIME:UNIT?

Or

CALCulate<Ch>:TRACe<Tr>:TRANSform:TIME:UNIT <char>

CALCulate<Ch>:TRACe<Tr>:TRANSform:TIME:UNIT?

### Description

Selects the transformation unit for the time domain transformation function: seconds, meters, feet.

command/query

### Target

CALCulate<Ch>[:SElected] — active trace of channel <Ch>,

Or

CALCulate<Ch>:TRACe<Tr> — trace <Tr> of channel <Ch>,

<Ch> = {[1]|2|...16}

<Tr> = {[1]|2|...16}

### Parameter

<char> Choose from:

**SEConds**    Seconds

**METers**     Meters

**FEET**       Feet

## Query Response

{SEC|MET|FEET}

## Preset Value

SEC

## Equivalent Softkeys

Analysis > Time Domain > Unit > {Seconds | Meters | Feet}

---

## Equivalent COM Command

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

---

### NOTE

This command is similar to  
CALCulate<Ch>[:SElected]:TRANSform:TIME:UNIT only.

---

## Syntax

Param = app.SCPI.CALCulate(Ch).SElected.TRANSform.TIME.UNIT

app.SCPI.CALCulate(Ch).SElected.TRANSform.TIME.UNIT = "MET"

## Type

String (read/write)

---

Back to [CALCulate](#)

## DISPlay

Command		Description
<a href="#">DISP:COL:BACK</a>	Color Settings	Background color
<a href="#">DISP:COL:GRAT</a>		Grid and graticule label color
<a href="#">DISP:COL:TRAC:DATA</a>		Data trace color
<a href="#">DISP:COL:TRAC:MEM</a>		Memory trace color
<a href="#">DISP:IMAG</a>		Colors inversion
<a href="#">DISP:COL:RES</a>	Interface Settings	Resets display settings to default
<a href="#">DISP:ENAB</a>		Display update ON/OFF
<a href="#">DISP:GLAB</a>		Graticule Label state
<a href="#">DISP:HIDE</a>		Hides the Analyzer window
<a href="#">DISP:MAX</a>		Maximizes the channel window ON/OFF
<a href="#">DISP:PART:VIS</a>		Display elements ON/OFF
<a href="#">DISP:POS</a>		Position and size of Analyzer window

Command		Description
<a href="#">DISP:SHOW</a>		Shows the Analyzer window
<a href="#">DISP:UPD</a>		One-time display update
<a href="#">DISP:WIND:MAX</a>		Maximizes the trace in channel ON/OFF
<a href="#">DISP:WIND:TITL</a>		Channel title display ON/OFF
<a href="#">DISP:WIND:TITL:DATA</a>		Channel title label
<a href="#">DISP:WIND:X:SPAC</a>		X-axis type for segment sweep
<a href="#">DISP:FONT:SIZE</a>	Font Size Settings	Font size for all elements
<a href="#">DISP:PART:FONT:SIZE</a>		Font size of specified element
<a href="#">DISP:PART:FONT:SIZE:STAT</a>		Individual font sizes for elements ON/OFF
<a href="#">DISP:FSIG</a>	Limit Test, Ripple Limit Test	"Fail" sign display ON/OFF
<a href="#">DISP:MARK:TABL</a>	Marker Properties	Marker table ON/OFF
<a href="#">DISP:WIND:ANN:MARK:ALIG</a>		Marker annotation alignment

Command		Description
<a href="#">DISP:WIND:ANN:MARK:SING</a>		Active marker only ON/OFF
<a href="#">DISP:WIND:TRAC:ANN:MARK:POS:X</a>		X-position of marker annotation
<a href="#">DISP:WIND:TRAC:ANN:MARK:POS:Y</a>		Y-position of marker annotation
<a href="#">DISP:SPL</a>	Channel and Trace Settings	Number and Layout of channels
<a href="#">DISP:WIND:ACT</a>		Active channel number (write)
<a href="#">DISP:WIND:SPL</a>		Allocation of traces in the channel window
<a href="#">DISP:WIND:TRAC:MEM</a>	Memory Trace Function	Memory trace display ON/OFF
<a href="#">DISP:WIND:TRAC:STAT</a>		Data trace display ON/OFF
<a href="#">DISP:WIND:TRAC:Y:AUTO</a>	Scale	Auto scale
<a href="#">DISP:WIND:TRAC:Y:PDIV</a>		Scale per division
<a href="#">DISP:WIND:TRAC:Y:RLEV</a>		Reference line value
<a href="#">DISP:WIND:TRAC:Y:RLEV:AUTO</a>		Auto Reference Level
<a href="#">DISP:WIND:TRAC:Y:RPOS</a>		Reference line position



Command		Description
<a href="#">DISP:WIND:Y:DIV</a>		Number of the scale divisions

## DISP:COL:BACK

### SCPI Command

DISPlay:COLor:BACK <numeric 1>,<numeric 2>,<numeric 3>

DISPlay:COLor:BACK?

### Description

Sets or reads out the background color for trace display.

command/query

### Parameter

<numeric 1>	Red value R from 0 to 255
<numeric 2>	Green value G from 0 to 255
<numeric 3>	Blue value B from 0 to 255

### Out of Range

Sets the value of the limit, which is closer to the specified value.

### Query Response

<numeric 1>, <numeric 2>, <numeric 3>

### Preset Value

0,0,0

### Equivalent Softkeys

Display > Properties > Color > Background > {Red | Green | Blue}

---

### Equivalent COM Command

SCPI.DISPlay.COLor.BACK

**Syntax**

Data = app.SCPI.DISPlay.COLOr.BACK

app.SCPI.DISPlay.COLOr.BACK = Array(255, 255, 255)

**Type**

Variant (array of long) (read/write)

---

Back to [DISPlay](#)

## DISP:COL:GRAT

### SCPI Command

DISPlay:COLor:GRATicule <numeric 1>,<numeric 2>,<numeric 3>

DISPlay:COLor:GRATicule?

### Description

Sets or reads out the grid and the graticule label color for trace display.

command/query

### Parameter

<numeric 1>	Red value R from 0 to 255
<numeric 2>	Green value G from 0 to 255
<numeric 3>	Blue value B from 0 to 255

### Out of Range

Sets the value of the limit, which is closer to the specified value.

### Query Response

<numeric 1>, <numeric 2>, <numeric 3>

### Preset Value

160,160,164

### Equivalent Softkeys

Display > Properties > Color > Grid > {Red | Green | Blue}

---

### Equivalent COM Command

SCPI.DISPlay.COLor.GRATicule

**Syntax**

Data = app.SCPI.DISPlay.COLOr.GRATicule

app.SCPI.DISPlay.COLOr.GRATicule = Array(128, 128, 128)

**Type**

Variant (array of long) (read/write)

---

Back to [DISPlay](#)

## DISP:COL:RES

### SCPI Command

DISPlay:COLor:RESet

### Description

Restores the display settings to the default values.

no query

### Equivalent Softkeys

Display > Properties > Set Defaults

---

### Equivalent COM Command

SCPI.DISPlay.COLor.RESet

### Syntax

app.SCPI.DISPlay.COLor.RESet

### Type

Method

---

Back to [DISPlay](#)

# DISP:COL:TRAC:DATA

## SCPI Command

DISPlay:COLor:TRACe<Tr>:DATA <numeric 1>,<numeric 2>,<numeric 3>

DISPlay:COLor:TRACe<Tr>:DATA?

## Description

Sets or reads out the data trace color.

command/query

## Target

Trace <Tr>,

<Tr>={[1]|2|...16}

## Parameter

<numeric 1>	Red value R from 0 to 255
<numeric 2>	Green value G from 0 to 255
<numeric 3>	Blue value B from 0 to 255

## Out of Range

Sets the value of the limit, which is closer to the specified value.

## Query Response

<numeric 1>, <numeric 2>, <numeric 3>

## Preset Value

Varies depending on the trace number.

## Equivalent Softkeys

Display > Properties > Color > Data Trace > {Red | Green | Blue}

---

## Equivalent COM Command

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

## Syntax

Data = app.SCPI.DISPlay.COLor.TRACe(Tr).DATA

app.SCPI.DISPlay.COLor.TRACe(Tr).DATA = Array(255, 255, 0)

## Type

Variant (array of long) (read/write)

---

Back to [DISPlay](#)



## DISP:COL:TRAC:MEM

### SCPI Command

DISPlay:COLor:TRACe<Tr>:MEMory <numeric 1>,<numeric 2>,<numeric 3>

DISPlay:COLor:TRACe<Tr>:MEMory?

### Description

Sets or reads out the data trace color.

command/query

### Target

Trace <Tr>,

<Tr>={{1}|2|...16}

### Parameter

<numeric 1>	Red value R from 0 to 255
<numeric 2>	Green value G from 0 to 255
<numeric 3>	Blue value B from 0 to 255

### Out of Range

Sets the value of the limit, which is closer to the specified value.

### Query Response

<numeric 1>, <numeric 2>, <numeric 3>

### Preset Value

Varies depending on the trace number.

### Equivalent Softkeys

Display > Properties > Color > Memory Trace > {Red | Green | Blue}

---

## Equivalent COM Command

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

## Syntax

Data = app.SCPI.DISPlay.COLor.TRACe(Tr).MEMory

app.SCPI.DISPlay.COLor.TRACe(Tr).MEMory = Array(255, 255, 0)

## Type

Variant (array of long) (read/write)

---

Back to [DISPlay](#)

## DISP:ENAB

### SCPI Command

DISPlay:ENABle {OFF|ON|0|1}

DISPlay:ENABle?

### Description

Turns the display update ON/OFF.

command/query

### Parameter

{ON|1}      ON

{OFF|0}      OFF

### Query Response

{0|1}

### Preset Value

1

### Equivalent Softkeys

Display > Update

---

### Equivalent COM Command

SCPI.DISPlay.ENABLE

### Syntax

Status = app.SCPI.DISPlay.ENABLE

app.SCPI.DISPlay.ENABLE = true

## Type

Boolean (read/write)

---

Back to [DISPlay](#)

## DISP:FONT:SIZE

### SCPI Command

DISPlay:FONT:SIZE <numeric>

DISPlay:FONT:SIZE?

### Description

Sets/gets one font size for all displayed elements of the application.

command/query

### Parameter

<numeric> Specifies the font size from 10 to 22.

### Query Response

<numeric>

### Preset Value

11

### Equivalent Softkeys

Display > Properties > Font Size

---

### Equivalent COM Command

None

---

Back to [DISPlay](#)

## DISP:FSIG

### SCPI Command

DISPlay:FSIGn {OFF|ON|0|1}

DISPlay:FSIGn?

### Description

Turns the "Fail" sign display ON/OFF when performing limit test or ripple limit test.

command/query

### Parameter

{ON|1}      ON

{OFF|0}      OFF

### Query Response

{0|1}

### Preset Value

0

### Equivalent Softkeys

Analysis > Limit Test > Fail Sign

Analysis > Ripple Limit > Fail Sign

---

### Equivalent COM Command

SCPI.DISPlay.FSIGn

### Syntax

Status = app.SCPI.DISPlay.FSIGn

app.SCPI.DISPlay.FSIGn = true

## Type

Boolean (read/write)

---

Back to [DISPlay](#)

## DISP:GLAB

### SCPI Command

DISPlay:GLABel <char>

DISPlay:GLABel?

### Description

Sets/gets the Graticule Label state.

command/query

### Parameter

Specifies the Graticule Label state:

- OFF**            Graticule label is OFF
- ACTive**        Only active trace has graticule label
- ALL**            All traces have graticule label

### Query Response

<char>

### Preset Value

ACTive

### Equivalent Softkeys

Display > Properties > Graticule Label

---

### Equivalent COM Command

SCPI.DISPlay.GLABel



**Syntax**

Param = app.SCPI.DISPlay.GLABel

app.SCPI.DISPlay.GLABel = "OFF"

**Type**

String (read/write)

---

Back to [DISPlay](#)

## DISP:IMAG

### SCPI Command

DISPlay:IMAGe <char>

DISPlay:IMAGe?

### Description

Turns the inversion of display colors of the trace area ON/OFF.

command/query

### Parameter

<char> Choose from:

**NORMal**      Normal display

**INVert**      Inverted color display

### Query Response

{NORM|INV}

### Preset Value

NORM

### Equivalent Softkeys

Display > Properties > Invert Color

---

### Equivalent COM Command

SCPI.DISPlay.IMAGe

**Syntax**

Param = app.SCPI.DISPlay.IMAGe

app.SCPI.DISPlay.IMAGe = "INV"

**Type**

String (read/write)

---

Back to [DISPlay](#)

## DISP:HIDE

### SCPI Command

DISPlay:HIDE

### Description

Hides the Analyzer GUI.

no query

### Related Commands

[DISP:SHOW](#)

### Equivalent Softkeys

None

---

### Equivalent COM Command

SCPI.DISPlay.HIDE

### Syntax

app.SCPI.DISPlay.HIDE

### Type

Method

---

Back to [DISPlay](#)

# DISP:MARK:TABL

## SCPI Command

DISPlay:MARKer:TABLE[:STATe] {OFF|ON|0|1}

DISPlay:MARKer:TABLE[:STATe]?

## Description

Turns the marker table ON/OFF.
command/query

## Parameter

- {ON|1}      ON
- {OFF|0}     OFF

## Query Response

{0|1}

## Preset Value

0

## Equivalent Softkeys

Display > Marker > Properties > Marker Table

---

## Equivalent COM Command

None

---

Back to [DISPlay](#)

## DISP:MAX

### SCPI Command

DISPlay:MAXimize {OFF|ON|0|1}

DISPlay:MAXimize?

### Description

Turns the maximization of the active channel window ON/OFF.

command/query

### Target

The active channel set by the command [DISP:WIND:ACT](#).

### Parameter

{ON|1}      ON

{OFF|0}     OFF

### Query Response

{0|1}

### Preset Value

0

### Equivalent Softkeys

Display > Active Trace/Channel > Maximize channel

---

### Equivalent COM Command

SCPI.DISPlay.MAXimize

**Syntax**

Status = app.SCPI.DISPlay.MAXimize

app.SCPI.DISPlay.MAXimize = true

**Type**

Boolean (read/write)

---

Back to [DISPlay](#)

## DISP:PART:FONT:SIZE

### SCPI Command

DISPlay:PARTition:FONT:SIZE <char>, <numeric>

DISPlay:PARTition:FONT:SIZE? <char>

### Description

Sets/gets the font size of the item specified by the <char> parameter.

command/query

### Parameter

<numeric> Specifies the font size from 10 to 22.

<char> Specifies display item:

<b>BUTTON</b>	Soft buttons
<b>MENU</b>	Menu bar
<b>CSTATUS</b>	Channel status
<b>ASTATUS</b>	Analyzer status
<b>CHANNEL</b>	Channel window

### Query Response

<numeric>

### Preset Value

11

### Equivalent Softkeys

Display > Properties > Font Size > Item Font Size



---

## Equivalent COM Command

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

## Syntax

Size = app.SCPI.DISPlay.PARTition.FONT.SIZE("CHAN")

app.SCPI.DISPlay.PARTition.FONT.SIZE("CHAN") = 20

## Type

Long (read/write)

---

Back to [DISPlay](#)

## DISP:PART:FONT:SIZE:STAT

### SCPI Command

DISPlay:PARTition:FONT:SIZE:STATe {OFF|ON|0|1}

DISPlay:PARTition:FONT:SIZE:STATe?

### Description

Specifies whether different elements of the application window have individual font sizes or the same font size.

command/query

### Parameter

<bool> Specifies the following:

- |                |   |
|----------------|---|
| <b>{ON 1}</b>  | Different elements of the application window may have individual font sizes |
| <b>{OFF 0}</b> | Different elements of the application window have the same font size        |

### Query Response

{0|1}

### Related Commands

[DISP:PART:FONT:SIZE](#)

### Equivalent Softkeys

Display > Properties > Font Size > Item Font Size > Item Font Size

---

### Equivalent COM Command

None

---

Back to [DISPlay](#)

## DISP:PART:VIS

### SCPI Command

DISPlay:PARTition:VISible <char>, {OFF|ON|0|1}

DISPlay:PARTition:VISible? <char>

### Description

Shows or hides the display partition specified by the <char> parameter.

command/query

### Parameter

<bool> Specifies the status of the display partition:

**{ON|1}** Different elements of the application window may have individual font sizes

**{OFF|0}** Different elements of the application window have the same font size

<char> Specifies display partition:

**BUTTON** Soft buttons

**MENU** Menu bar

**CSTATUS** Channel status

**ASTATUS** Analyzer status

**TITLE** Main window title

**FLABEL** Frequency label

**MTABLE** Marker table

## Query Response

{0|1}

## Equivalent Softkeys

**Display > Properties > Menu Bar**

**Display > Display Properties > Frequency Label**

**Markers > Properties > Marker Table or None**

---

## Equivalent COM Command

SCPI.DISPlay.PARTition.VISible(Param)

## Syntax

State = app.SCPI.DISPlay.PARTition.VISible("MENU")

app.SCPI.DISPlay.PARTition.VISible("MENU") = true

## Type

Boolean (read/write)

---

Back to [DISPlay](#)

# DISP:POS

## SCPI Command

DISPlay:POSition <numeric 1>, <numeric 2>, <numeric 3>, <numeric 4>

DISPlay:POSition?

## Description

Sets/gets the Analyzer window position on the screen and its size.

command/query

## Parameter

Parameters determine the position of the main window:

- |             |   |
|-------------|---|
| <numeric 1> | Specifies the coordinate of the left side of the window |
| <numeric 2> | Specifies the coordinate of the top of the window       |
| <numeric 3> | Specifies the width of the window                       |
| <numeric 4> | Specifies the height of the window                      |

## Unit

Screen pixel

## Range

From 0 to the screen resolution.

## Out of Range

Sets the value of the limit, which is closer to the specified value.

## Query Response

<numeric>

## Preset Value

<numeric 1> = (screen width – 800) / 2,

<numeric 2> = (screen height – 600) / 2,

<numeric 3> = 800,

<numeric 4> = 600,

Preset: **Display > Properties > Set Defaults**

## Equivalent Softkeys

None

---

## Equivalent COM Command

SCPI.DISPlay.POSition

## Syntax

Pos = app.SCPI.DISPlay.POSition

app.SCPI.DISPlay.POSition = Array(0, 0, 800, 600)

## Type

Variant (array of long) (read/write)

---

Back to [DISPlay](#)

# DISP:SHOW

## SCPI Command

DISPlay:SHOW

## Description

Shows the Analyzer GUI hidden by the [DISP:HIDE](#) command.

no query

## Related Commands

[DISP:HIDE](#)

## Equivalent Softkeys

None

---

## Equivalent COM Command

SCPI.DISPlay.SHOW

## Syntax

app.SCPI.DISPlay.SHOW

## Type

Method

---

Back to [DISPlay](#)

# DISP:SPL

## SCPI Command

DISPlay:SPLit <numeric>

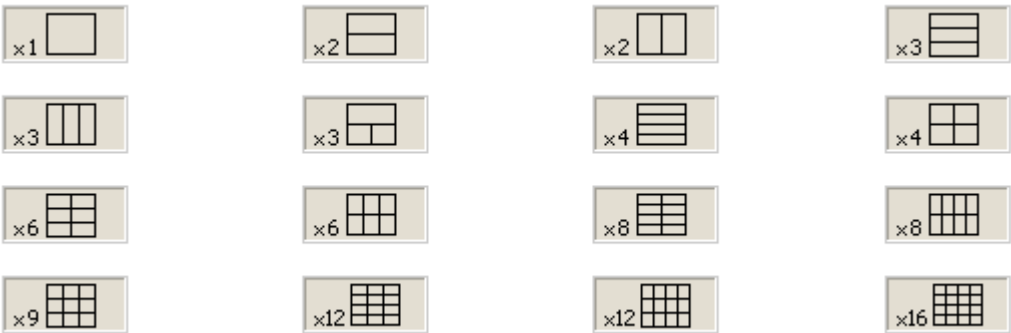
DISPlay:SPLit?

## Description

Sets or reads out the number of channels and channel layout on the screen. The channel layouts on the screen is shown below.

command/query

## Channel window layout on the screen



## Parameter

<numeric> the code of the channel window layout from 1 to 16. Note: the layout code does not correspond to the number of channels.

## Query Response

<numeric>

## Preset Value

1

## Equivalent Softkeys

Display > Allocate channels



---

## Equivalent COM Command

SCPI.DISPlay.SPLit

## Syntax

Value = app.SCPI.DISPlay.SPLit

app.SCPI.DISPlay.SPLit = 2

## Type

Long (read/write)

---

Back to [DISPlay](#)

## DISP:UPD

### SCPI Command

DISPlay:UPDate[:IMMediate]

### Description

Updates the display once when the display update is set to OFF by the [DISP:ENAB](#) command.

no query

### Related Commands

[DISP:ENAB](#)

### Equivalent Softkeys

None

---

### Equivalent COM Command

SCPI.DISPlay.UPDate.IMMediate

### Syntax

app.SCPI.DISPlay.REFResh.IMMediate

app.SCPI.DISPlay.UPDate.IMMediate

### Type

Method

---

Back to [DISPlay](#)

## DISP:WIND:ACT

### SCPI Command

DISPlay:WINDow<Ch>:ACTivate

### Description

Sets the active channel.

**Note:** Trying to set an active channel that is not displayed with the [DISP:SPL](#) command will produce an error.

no query

### Target

Channel <Ch>,

<Ch>={[1]|2|...16}

### Related Commands

[DISP:SPL](#)

[SERV:CHAN:ACT?](#)

### Equivalent Softkeys

Display > Active Trace / Channel > Active Channel

---

### Equivalent COM Command

SCPI.DISPlay.WINDow(Ch).ACTivate

**Syntax**

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

**Type**

Method

---

Back to [DISPlay](#)

## DISP:WIND:ANN:MARK:ALIG

### SCPI Command

DISPlay:WINDow<Ch>:ANNotation:MARKer:ALIGn[:TYPE] <char>

DISPlay:WINDow<Ch>:ANNotation:MARKer:ALIGn[:TYPE]?

### Description

Sets or reads out the alignment of the marker annotation when the active marker only feature is turned OFF by the [DISP:WIND:ANN:MARK:SING](#) command.

command/query

### Target

Channel <Ch>,

<Ch>={[1]|2|...16}

### Parameter

<char> Choose from:

<b>VERTical</b>	Vertical alignment
<b>HORizontal</b>	Horizontal alignment
<b>NONE</b>	No alignment

### Query Response

{NONE|VERT|HOR}

### Preset Value

NONE

### Related Commands

[DISP:WIND:ANN:MARK:SING](#)

## Equivalent Softkeys

Markers > Properties > Align > {Vertical | Horizontal | OFF}

---

## Equivalent COM Command

SCPI.DISPlay.WINDow(Ch).ANNotation.MARKer.ALIGn.TYPE

## Syntax

Param = app.SCPI.DISPlay.WINDow(Ch).ANNotation.MARKer.ALIGn.TYPE

app.SCPI.DISPlay.WINDow(Ch).ANNotation.MARKer.ALIGn.TYPE = "VERT"

## Type

String (read/write)

---

Back to [DISPlay](#)

## DISP:WIND:ANN:MARK:SING

### SCPI Command

DISPlay:WINDow<Ch>:ANNotation:MARKer:SINGle[:STATe] {OFF|ON|0|1}

DISPlay:WINDow<Ch>:ANNotation:MARKer:SINGle[:STATe]?

### Description

Selects display of either the active trace markers or all trace markers.

command/query

### Target

Channel <Ch>,

<Ch>={{1}|2|...16}

### Parameter

<char> Choose from:

{ON|1}      Active trace markers

{OFF|0}      All trace markers

### Query Response

{0|1}

### Preset Value

1

### Equivalent Softkeys

Markers > Properties > Active Only

---

### Equivalent COM Command

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

**Syntax**

Status = app.SCPI SCPI.DISPlay.WINDow(Ch).ANNotation.MARKer.SINGle.STATe

app.SCPI SCPI.DISPlay.WINDow(Ch).ANNotation.MARKer.SINGle.STATe = tru

**Type**

Boolean (read/write)

---

Back to [DISPlay](#)



## DISP:WIND:MAX

### SCPI Command

DISPlay:WINDow<Ch>:MAXimize {OFF|ON|0|1}

DISPlay:WINDow<Ch>:MAXimize?

### Description

Turns the active trace maximization inside the specified channel ON/OFF.

command/query

### Target

Channel <Ch>,

<Ch>={{1}|2|...16}

### Parameter

{ON|1}      ON

{OFF|0}      OFF

### Query Response

{0|1}

### Preset Value

0

### Equivalent Softkeys

Display > Active Trace/Channel > Maximize Trace

---

### Equivalent COM Command

SCPI.DISPlay.WINDow(Ch).MAXimize

**Syntax**

Status = app.SCPI.DISPlay.WINDow(Ch).MAXimize

app.SCPI.DISPlay.WINDow(Ch).MAXimize = true

**Type**

Boolean (read/write)

---

Back to [DISPlay](#)

## DISP:WIND:SPL

### SCPI Command

DISPlay:WINDow<Ch>:SPLit <numeric>

DISPlay:WINDow<Ch>:SPLit?

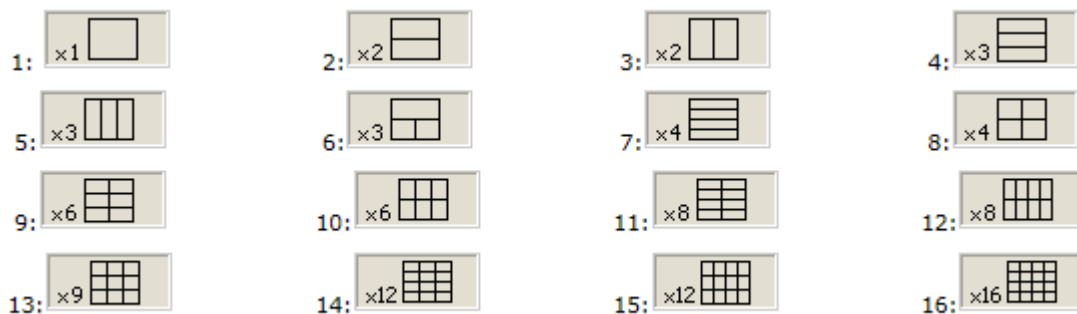
### Description

Sets or reads out the number of the graph layout in the channel window. The graph layout in the channel window is shown below.

**Note:** This function does not determine the number of traces in the channel window; the [CALC:PAR:COUN](#) command sets the number of traces.

command/query

### Graph layout in the channel window



### Target

Channel <Ch>,

<Ch>={[1]2|...16}

### Parameter

<numeric> the number of the graph layout from 1 to 16

### Out of Range

Sets the value of the limit, which is closer to the specified value.

## Query Response

<numeric>

## Preset Value

1

## Equivalent Softkeys

Display > Allocate Traces

---

## Equivalent COM Command

SCPI.DISPlay.SPLit

## Syntax

Value = app.SCPI.DISPlay.SPLit

app.SCPI.DISPlay.SPLit = 2

## Type

Long (read/write)

---

Back to [DISPlay](#)

## DISP:WIND:TITL

### SCPI Command

DISPlay:WINDow<Ch>:TITLe[:STATe] {OFF|ON|0|1}

DISPlay:WINDow<Ch>:TITLe[:STATe]?

### Description

Turns the channel title display ON/OFF.

command/query

### Target

Channel <Ch>,

<Ch>={[1]|2|...16}

### Parameter

{ON|1}      ON

{OFF|0}      OFF

### Query Response

{0|1}

### Preset Value

0

### Equivalent Softkeys

Display > Title Label

---

### Equivalent COM Command

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

**Syntax**

Status = app.SCPI.DISPlay.WINDow(Ch).TITLe.STATe

app.SCPI.DISPlay.WINDow(Ch).TITLe.STATe = true

**Type**

Boolean (read/write)

---

Back to [DISPlay](#)

## DISP:WIND:TITL:DATA

### SCPI Command

DISPlay:WINDow<Ch>:TITLe:DATA <string>

DISPlay:WINDow<Ch>:TITLe:DATA?

### Description

Sets or reads out the channel title label.

command/query

### Target

Channel <Ch>,

<Ch>={{1}|2|...16}

### Parameter

<string>, up to 256 characters

### Query Response

<string>

### Preset Value

Empty string

### Equivalent Softkeys

Display > Edit Title Label

---

### Equivalent COM Command

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

**Syntax**

Text = app.SCPI.DISPlay.WINDow(Ch).TITLe.DATA

app.SCPI.DISPlay.WINDow(Ch).TITLe.DATA = "Network 1"

**Type**

String (read/write)

---

Back to [DISPlay](#)



## DISP:WIND:TRAC:ANN:MARK:POS:X

### SCPI Command

DISPlay:WINDow<Ch>:TRACe<Tr>:ANNotation:MARKer:POSition:X <numeric>

DISPlay:WINDow<Ch>:TRACe<Tr>:ANNotation:MARKer:POSition:X?

### Description

Sets or reads out the display position of the marker annotation on the X-axis by a percentage of the display width.

command/query

### Target

Trace <Tr> of channel <Ch>,

<Tr>={{1}|2|...16}

<Ch>={{1}|2|...16}

### Parameter

<numeric> the display position of the marker value on the X-axis from 0 to 100

### Unit

% (percent)

### Out of Range

Sets the value of the limit, which is closer to the specified value.

### Query Response

<numeric>

### Preset Value

0

## Equivalent Softkeys

Markers > Properties > Data X Position

---

## Equivalent COM Command

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

## Syntax

Value =

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

app.SCPI.DISPlay.WINDow(Ch).TRACe(Tr).ANNotation.MARKer.POSition.X = 50

## Type

Double (read/write)

---

Back to [DISPlay](#)

## DISP:WIND:TRAC:ANN:MARK:POS:Y

### SCPI Command

DISPlay:WINDow<Ch>:TRACe<Tr>:ANNotation:MARKer:POSition:Y <numeric>

DISPlay:WINDow<Ch>:TRACe<Tr>:ANNotation:MARKer:POSition:Y?

### Description

Sets or reads out the display position of the marker annotation on the Y-axis by a percentage of the display height.

command/query

### Target

Trace <Tr> of channel <Ch>,

<Tr>={[1]|2|...16}

<Ch>={[1]|2|...16}

### Parameter

<numeric> the display position of the marker value on the Y-axis from 0 to 100

### Unit

% (percent)

### Out of Range

Sets the value of the limit, which is closer to the specified value.

### Query Response

<numeric>

### Preset Value

0

## Equivalent Softkeys

Markers > Properties > Data Y Position

---

## Equivalent COM Command

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

## Syntax

Value =

app.SCPI.DISPlay.WINDow(Ch).TRACe(Tr).ANNotation.MARKer.POSition.Y

app.SCPI.DISPlay.WINDow(Ch).TRACe(Tr).ANNotation.MARKer.POSition.Y = 50

## Type

Double (read/write)

---

Back to [DISPlay](#)

## DISP:WIND:TRAC:MEM

### SCPI Command

DISPlay:WINDow<Ch>:TRACe<Tr>:MEMory[:STATe] {OFF|ON|0|1}

DISPlay:WINDow<Ch>:TRACe<Tr>:MEMory[:STATe]?

### Description

Turns the memory trace display ON/OFF.

**Note:** If the memory trace does not exist, an error occurs and the command is ignored.

command/query

### Target

Trace <Tr> of channel <Ch>,

<Tr>={[1]|2|...16}

<Ch>={[1]|2|...16}

**Parameter**

{ON|1}      ON

{OFF|0}      OFF

**Query Response**

{0|1}

**Preset Value**

0

### Equivalent Softkeys

Display > Trace Display > {Memory | Data & Memory} (ON)

Display > Trace Display > {Data | OFF} (OFF)

---

### Equivalent COM Command

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

### Syntax

Status = app.SCPI.DISPlay.WINDow(Ch).TRACe(Tr).MEMory.STATe

app.SCPI.DISPlay.WINDow(Ch).TRACe(Tr).MEMory.STATe = true

### Type

Boolean (read/write)

---

Back to [DISPlay](#)

## DISP:WIND:TRAC:STAT

### SCPI Command

DISPlay:WINDow<Ch>:TRACe<Tr>:STATe {OFF|ON|0|1}

DISPlay:WINDow<Ch>:TRACe<Tr>:STATe?Description

Turns the data trace display ON/OFF.

command/query

### Target

Trace <Tr> of channel <Ch>,

<Tr>={{1}|2|...16}

<Ch>={{1}|2|...16}

### Parameter

{ON|1}      ON

{OFF|0}     OFF

### Query Response

{0|1}

### Preset Value

1



### Equivalent Softkeys

Display > Trace Display > {Data | Data & Memory} (ON)

Display > Trace Display > {Memory | OFF} (OFF)

---

### Equivalent COM Command

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

### Syntax

Status = app.SCPI.DISPlay.WINDow(Ch).TRACe(Tr).STATe

app.SCPI.DISPlay.WINDow(Ch).TRACe(Tr).STATe = false

### Type

Boolean (read/write)

---

Back to [DISPlay](#)

## DISP:WIND:TRAC:Y:AUTO

### SCPI Command

DISPlay:WINDow<Ch>:TRACe<Tr>:Y[:SCALe]:AUTO

### Description

Executes the auto scale function for the trace. The function automatically sets both the PDIVision and the RLEVel values.

no query

### Target

Trace <Tr> of channel <Ch>,

<Tr>={[1]|2|...16}

<Ch>={[1]|2|...16}

### Related Commands

[DISP:WIND:TRAC:Y:PDIV](#)

[DISP:WIND:TRAC:Y:RLEV](#)

### Equivalent Softkeys

Scale > Auto Scale

---

### Equivalent COM Command

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

### Syntax

app.SCPI.DISPlay.WINDow(Ch).TRACe(Tr).Y.SCALe.AUTO

**Type**

Method

---

Back to [DISPlay](#)

## DISP:WIND:TRAC:Y:PDIV

### SCPI Command

DISPlay:WINDow<Ch>:TRACe<Tr>:Y[:SCALe]:PDIVision <numeric>

DISPlay:WINDow<Ch>:TRACe<Tr>:Y[:SCALe]:PDIVision?

### Description

Sets or reads out the trace scale. Sets the scale per division when the data format is in the rectangular format. Sets the full scale value when the data format is in the Smith chart format or the polar format.

command/query

### Target

Trace <Tr> of channel <Ch> ,

<Tr>={[1]|2|...16}

<Ch>={[1]|2|...16}

### Parameter

<numeric> the scale value from 10E−18 to 1E18

### Out of Range

Sets the value of the limit, which is closer to the specified value.

### Unit

dB |° |s

### Out of Range

<numeric>

### Query Response

<numeric>

## **Preset Value**

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

## **Equivalent Softkeys**

### **Scale > Scale**

---

## **Equivalent COM Command**

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

## **Syntax**

Value = app.SCPI.DISPlay.WINDow(Ch).TRACe(Tr).Y.SCALe.PDIVision

app.SCPI.DISPlay.WINDow(Ch).TRACe(Tr).Y.SCALe.PDIVision = 20

## **Type**

Double (read/write)

---

Back to [DISPlay](#)

## DISP:WIND:TRAC:Y:RLEV

### SCPI Command

DISPlay:WINDow<Ch>:TRACe<Tr>:Y[:SCALe]:RLEVel <numeric>

DISPlay:WINDow<Ch>:TRACe<Tr>:Y[:SCALe]:RLEVel?

### Description

Sets the value of the reference line (response value on the reference line). For the rectangular format only.

command/query

### Target

Trace <Tr> of channel <Ch>,

<Tr>={{1}|2|...16}

<Ch>={{1}|2|...16}

### Parameter

<numeric> the scale value from 10E–18 to 1E18

### Out of Range

Sets the value of the limit, which is closer to the specified value.

### Unit

dB | ° | s

### Out of Range

Sets the value of the limit, which is closer to the specified value.

### Query Response

<numeric>

**Preset Value**

0 (except for SWR: 1)

**Equivalent Softkeys****Scale > Ref Value**

---

**Equivalent COM Command**

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

**Syntax**

Value = app.SCPI.DISPlay.WINDow(Ch).TRACe(Tr).Y.SCALe.RLEVel

app.SCPI.DISPlay.WINDow(Ch).TRACe(Tr).Y.SCALe.RLEVel = 10

**Type**

Double (read/write)

---

Back to [DISPlay](#)

## DISP:WIND:TRAC:Y:RLEV:AUTO

### SCPI Command

DISPlay:WINDow<Ch>:TRACe<Tr>:Y[:SCALE]:RLEVel:AUTO

### Description

Executes the auto reference function for the trace. The function automatically sets the RLEVel value.

no query

### Target

Trace <Tr> of channel <Ch>,

<Tr>={[1]|2|...16}

<Ch>={[1]|2|...16}

### Related Commands

[DISP:WIND:TRAC:Y:RLEV](#)

### Equivalent Softkeys

Scale > Auto Ref Value

---

### Equivalent COM Command

None

---

Back to [DISPlay](#)



## DISP:WIND:TRAC:Y:RPOS

### SCPI Command

DISPlay:WINDow<Ch>:TRACe<Tr>:Y[:SCALe]:RPOSition <numeric>

DISPlay:WINDow<Ch>:TRACe<Tr>:Y[:SCALe]:RPOSition?

### Description

Sets the position of the reference line. For the rectangular format only.

command/query

### Target

Trace <Tr> of channel <Ch>,

<Tr>={[1]|2|...16}

<Ch>={[1]|2|...16}

### Parameter

<numeric> the reference line position from 0 to the number of the scale divisions (set by the DISP:WIND:Y:DIV command, 10 by default)

### Out of Range

Sets the value of the limit, which is closer to the specified value.

### Query Response

<numeric>

### Preset Value

5 (except for SWR: 0)

### Equivalent Softkeys

### Scale > Ref Position

---

## Equivalent COM Command

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

## Syntax

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

app.SCPI.DISPlay.WINDow(Ch).TRACe(Tr).Y.SCALe.RPOSition = 10

## Type

Long (read/write)

---

Back to [DISPlay](#)

## DISP:WIND:X:SPAC

### SCPI Command

DISPlay:WINDow<Ch>:X:SPACing <char>

DISPlay:WINDow<Ch>:X:SPACing?

### Description

Sets or reads out the display method of the graph horizontal axis for the segment sweep.

command/query

### Target

Channel <Ch>,

<Ch>={[1]|2|...16}

### Parameter

<char> Choose from:

**LINear**      Frequency base (linear frequency axis)

**OBASe**      Order base (linear axis of the point numbers)

### Out of Range

The command is ignored.

### Query Response

{LIN|OBAS}

### Preset Value

LIN

## Related Commands

[SENS:SWE:TYPE](#)

## Equivalent Softkeys

**Stimulus > Segment Table > Segment Display**

---

## Equivalent COM Command

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

## Syntax

Param = app.SCPI.DISPlay.WINDow(Ch).X.SPACing

app.SCPI.DISPlay.WINDow(Ch).X.SPACing = "OBAS"

## Type

String (read/write)

---

Back to [DISPlay](#)

## DISP:WIND:Y:DIV

### SCPI Command

DISPlay:WINDow<Ch>:Y[:SCALe]:DIVisions <numeric>

DISPlay:WINDow<Ch>:Y[:SCALe]:DIVisions?

### Description

Sets the number of the vertical scale divisions. For the rectangular format only.

command/query

### Target

Channel <Ch>,

<Ch>={[1]|2|...16}

### Parameter

<numeric> the number of the vertical scale divisions from 4 to 30

### Out of Range

Sets the value of the limit, which is closer to the specified value.

### Query Response

<numeric>

### Preset Value

10

### Resolution

2

### Equivalent Softkeys

### Scale > Divisions

---

## Equivalent COM Command

SCPI.DISPlay.WINDow(Ch).Y.SCALE.DIVisions

## Syntax

Value = app.SCPI.DISPlay.WINDow(Ch).Y.SCALE.DIVisions

app.SCPI.DISPlay.WINDow(Ch).Y.SCALE.DIVisions = 12

## Type

Long (read/write)

---

Back to [DISPlay](#)

## FORMat

Command		Description
<a href="#">FORM:BORD</a>	Data Transfer	Byte order
<a href="#">FORM:DATA</a>		Text or binary transfer format
<a href="#">FORM:PUSH</a>		Push and set byte order and transfer format
<a href="#">FORM:POP</a>		Pop byte order and transfer format

# FORM:BORD

## SCPI Command

FORMat:BORDER <char>

FORMat:BORDER?

## Description

Sets or reads out the transfer order of each byte in data when the binary data transfer format is set by the [FORM:DATA](#) command.

**Note:** The [x86](#) compatible processors use the little-endian format.

command/query

## Parameter

<char> Choose from:

**NORMal**          Normal (big-endian format)

**SWAPped**        Swapped (little-endian format)

## Query Response

{NORM|SWAP}

## Preset Value

NORM

## Related Commands

[FORM:DATA](#)

## Equivalent Softkeys

None



---

## Equivalent COM Command

None

---

Back to [FORMat](#)

## FORM:DATA

### SCPI Command

FORMat:DATA <char>

FORMat:DATA?

### Description

Sets or reads out the data transfer format when responding to the following queries:

<a href="#">CALC:DATA:FDAT?</a>	<a href="#">SENS:CORR:COLL:DATA:LOAD?</a>
<a href="#">CALC:DATA:FMEM?</a>	<a href="#">SENS:CORR:COLL:DATA:OPEN?</a>
<a href="#">CALC:DATA:SDAT?</a>	<a href="#">SENS:CORR:COLL:DATA:SHOR?</a>
<a href="#">CALC:DATA:SMEM?</a>	<a href="#">SENS:CORR:COLL:DATA:THRU:MAT Ch?</a>
<a href="#">CALC:DATA:XAX?</a>	<a href="#">SENS:CORR:COLL:DATA:THRU:TRAN?</a>
<a href="#">CALC:FUNC:DATA?</a>	<a href="#">SENS:DATA:CORR?</a>
<a href="#">CALC:LIM:DATA?</a>	<a href="#">SENS:DATA:RAW?</a>
<a href="#">CALC:LIM:REP?</a>	<a href="#">SENS:FREQ:DATA?</a>
<a href="#">CALC:LIM:REP:ALL?</a>	<a href="#">SENS:OFFS:SOUR:DATA?</a>
<a href="#">CALC:MARK:DATA?</a>	<a href="#">SENS:OFFS:REC:DATA?</a>
<a href="#">CALC:RLIM:DATA?</a>	<a href="#">SENS:OFFS:PORT:DATA?</a>
<a href="#">CALC:RLIM:REP?</a>	<a href="#">SENS:SEGM:DATA?</a>
<a href="#">SENS:CORR:COEF?</a>	<a href="#">SOUR:POW:PORT:CORR:COLL:TABLE:LOSS:DATA?</a>
<a href="#">SENS:CORR:COLL:DATA:ISOL?</a>	<a href="#">SOUR:POW:PORT:CORR:DATA?</a>

**Note:** The command is applicable with the TCP/IP HiSLIP protocol. The command is NOT applicable with the TCP/IP Socket protocol.

command/query

## Parameter

<char> Choose from:

<b>ASCIi</b>	Character format
<b>REAL</b>	Binary format (IEEE–64 floating point)
<b>REAL32</b>	Binary format (IEEE–32 floating point)

## Query Response

{ASC|REAL|REAL32}

## Preset Value

ASC

## Related Commands

[FORM:BORD](#)

## Equivalent Softkeys

None

---

## Equivalent COM Command

None

---

Back to [FORMat](#)

# FORM:PUSH

## SCPI Command

FORMat:PUSH <format>,<border>

## Description

Saves the current settings and sets new values for the data transfer format and byte order.

**Note:** The [x86](#) compatible processors use the little-endian format.

command only

## Parameter

<char> Choose from:

- |               |  |
|---------------|--|
| <b>ASCIi</b>  | Character format                       |
| <b>REAL</b>   | Binary format (IEEE–64 floating point) |
| <b>REAL32</b> | Binary format (IEEE–32 floating point) |

<border> Choose from:

- |                |                                |
|----------------|--------------------------------|
| <b>NORMal</b>  | Normal (big-endian format)     |
| <b>SWAPped</b> | Swapped (little-endian format) |

## Related Commands

FORM:POP

## Equivalent Softkeys

None

---

## Equivalent COM Command

None

---

Back to [FORMat](#)

# FORM:POP

## SCPI Command

FORMat:POP

## Description

Restores the settings for the data transfer format and byte order saved by the preceeding [FORM:PUSH](#) command.

command/query

## Related Commands

[FORM:PUSH](#)

## Equivalent Softkeys

None

---

## Equivalent COM Command

None

---

Back to [FORMat](#)

## HCOPy

Command		Description
<a href="#">HCOP</a>	Printing	Quick print
<a href="#">HCOP:ABOR</a>		Aborts the printout
<a href="#">HCOP:DATE:STAM</a>		Date and time stamp ON/OFF
<a href="#">HCOP:IMAG</a>		Inverted color of image
<a href="#">HCOP:PAIN</a>		Color chart for image printout

# HCOP

## SCPI Command

HCOPy[:IMMediate]

## Description

Prints out the image displayed on the screen without previewing.

no query

## Equivalent Softkeys

System > Print > Print Embedded

---

## Equivalent COM Command

SCPI.HCOPy.IMMediate

## Syntax

app.SCPI.HCOPy.IMMediate

## Type

Method

---

Back to [HCOPy](#)



## HCOP:ABOR

### SCPI Command

HCOPy:ABORt

### Description

Aborts the printout.

no query

### Equivalent Softkeys

None

---

### Equivalent COM Command

SCPI.HCOPy.ABORt

### Syntax

app.SCPI.HCOPy.ABORt

### Type

Method

---

Back to [HCOPy](#)

## HCOP:DATE:STAM

### SCPI Command

HCOPy:DATE:STAMp {OFF|ON|0|1}

HCOPy:DATE:STAMp?

### Description

Turns the date and time printout in the upper right corner of the image ON/OFF.

no query

### Parameter

{ON|1}      ON

{OFF|0}      OFF

### Query Responder

{0|1}

### Preset Value

1

### Equivalent Softkeys

**System > Print > Print Date & Time**

---

### Equivalent COM Command

SCPI.HCOPy.DATE.STAMp

### Syntax

Status = app.SCPI.HCOPy.DATE.STAMp

app.SCPI.HCOPy.DATE.STAMp = False

## Type

Boolean (read/write)

---

Back to [HCOPy](#)

## HCOP:IMAG

### SCPI Command

HCOPy:IMAGe <char>

HCOPy:IMAGe?

### Description

Sets or reads out the inverted color image printout.

command/query

### Parameter

<char> Choose from:

**NORMal**      Normal printout

**INVert**      Inverted color printout

### Query Response

{NORM|INV}

### Preset Value

NORM

### Equivalent Softkeys

System > Print > Invert Image

---

### Equivalent COM Command

SCPI.HCOPy.IMAGe

### Syntax

Param = app.SCPI.HCOPy.IMAGe

app.SCPI.HCOPy.IMAGe = "INV"

**Type**

String (read/write)

---

Back to [HCOPy](#)

# HCOP:PAIN

## SCPI Command

HCOPy:PAINt <char>

HCOPy:PAINt?

## Description

Sets or reads out the color chart for the image printout.
command/query

## Parameter

<char> Choose from:

- COLor**      Color printout
- GRAY**      Grayscale printout
- BW**          Black&white printout

## Query Responser

{COL|GRAY|BW}

## Preset Value

BW

## Equivalent Softkeys

System > Print > Print Color

---

## Equivalent COM Command

SCPI.HCOPy.PAINt

**Syntax**

Param = app.SCPI.HCOPy.PAINt

app.SCPI.HCOPy.PAINt = "COL "

**Type**

String (read/write)

---

Back to [HCOPy](#)

**INITiate**

Command		Description
<a href="#">INIT</a>	Trigger	Initiates channel once
<a href="#">INIT:CONT</a>		Continuous channel initiation mode ON/OFF
<a href="#">INIT:CONT:ALL</a>		Continuous channel initiation mode for all channels ON/OFF



# INIT

## SCPI Command

INITiate<Ch>[:IMMediate]

## Description

Puts the channel into the Trigger Waiting state for one trigger event. The channel should be in the hold state, otherwise an error occurs and the command is ignored. The channel goes into Hold as a result of the command [INIT:CONT](#) OFF.

If the Internal 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 Waiting for a Single Trigger 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 Hold state.

no query

## Target

Channel <Ch>,

<Ch>={[1]|2|...16}

## Related Commands

[TRIG:SOUR](#)

[INIT:CONT](#)

## Equivalent Softkeys

Stimulus > Trigger > Single

---

## Equivalent COM Command

SCPI.INITiate(Ch).IMMediate

**Syntax**

app.SCPI.INITiate(Ch).IMMediate

**Type**

Method

---

Back to [INITiate](#)

## INIT:CONT

### SCPI Command

INITiate<Ch>:CONTInuous {OFF|ON|0|1}

INITiate<Ch>:CONTInuous?

### Description

Turns the continuous trigger initiation mode ON/OFF.

When the continuous initiation mode is turned ON:

- If the Internal trigger source is selected by the command [TRIG:SOUR INT](#), then the channel continuously sweeps;
- If the trigger source selected is one other than the internal, then the channel goes to the trigger waiting 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 trigger waiting state.

When the continuous trigger initiation mode is turned OFF the channel is in the Hold state, to initiate a sweep use the [INIT](#) command.

command /query

### Target

Channel <Ch>,

<Ch>={[1]|2|...16}

### Parameter

Specifies the continuous trigger initiation mode:

{ON|1}      ON

{OFF|0}      OFF

### Query Responder

{0|1}

## Preset Value

1

## Related Commands

[TRIG:SOUR](#)

[INIT](#)

## Equivalent Softkeys

Stimulus > Trigger > Continuous

Stimulus > Trigger > Hold

---

## Equivalent COM Command

SCPI:INITiate(Ch).CONTInuous

## Syntax

Status = app.SCPI.INITiate(Ch).CONTInuous

app.SCPI.INITiate(Ch).CONTInuous = False

## Type

Boolean (read/write)

---

Back to [INITiate](#)

# INIT:CONT:ALL

## SCPI Command

INITiate:CONTinuous:ALL {OFF|ON|0|1}

## Description

Turns the continuous trigger initiation mode for all channels ON/OFF.

command

## Target

Channel <Ch>,

<Ch>={[1]|2|...16}

## Parameter

Specifies the continuous trigger initiation mode:

{ON|1}      ON

{OFF|0}      OFF

## Preset Value

1

## Related Commands

[INIT:CONT](#)

## Equivalent Softkeys

Stimulus > Trigger > Continuous All Channels

Stimulus > Trigger > Hold All Channels

---

## Equivalent COM Command

None

---

Back to [INTiate](#)



## MMEMory

Command		Description
<a href="#">MMEM:CAT?</a>	Disk Operations	Information about the hard drive
<a href="#">MMEM:COPY</a>		Copies the file
<a href="#">MMEM:DEL</a>		Deletes the file
<a href="#">MMEM:MDIR</a>		Creates a directory
<a href="#">MMEM:TRAN?</a>		Transfers the contents of the file
<a href="#">MMEM:LOAD</a>	Save/Recall Analyzer State, Calibration	Recalls the Analyzer state
<a href="#">MMEM:LOAD:CHAN</a>		Recalls the channel state from memory register
<a href="#">MMEM:LOAD:CHAN:CAL</a>		Recalls the channel calibration
<a href="#">MMEM:STOR</a>		Saves the Analyzer state
<a href="#">MMEM:STOR:CHAN</a>		Saves the channel state in memory register
<a href="#">MMEM:STOR:CHAN:CAL</a>		Saves the channel calibration
<a href="#">MMEM:STOR:CHAN:CLE</a>		Clears memory registers



Command		Description
<a href="#">MMEM:STOR:STYP</a>		Saving type
<a href="#">MMEM:LOAD:CKIT</a>	Calibration Kit Management	Recalls calibration kit definition from the file
<a href="#">MMEM:STOR:CKIT</a>		Save calibration kit definition to the file
<a href="#">MMEM:LOAD:LIM</a>	Limit Test	Recalls limit table from file
<a href="#">MMEM:STOR:LIM</a>		Saves limit table into file
<a href="#">MMEM:LOAD:PLOS</a>	Power Calibration	Recalls the loss compensation file
<a href="#">MMEM:STOR:PLOS</a>		Saves the loss compensation file
<a href="#">MMEM:LOAD:RLIM</a>	Ripple Limit Test	Recalls ripple limit table from file
<a href="#">MMEM:STOR:RLIM</a>		Saves ripple limit table into file
<a href="#">MMEM:LOAD:SEGM</a>	Stimulus Settings	Recalls the segment table file
<a href="#">MMEM:STOR:SEGM</a>		Saves the segment table to a file
<a href="#">MMEM:LOAD:SNP</a>	Save S-parameters to Touchstone File	Loads file to S-parameters
<a href="#">MMEM:LOAD:SNP:TRAC:MEM</a>		Loads file to the memory trace

Command		Description
<a href="#">MME:LOAD:SNP:FREQ</a>		Enables the frequency setting from a Touchstone file when it loaded
<a href="#">MME:STOR:SNP</a>		Saves channel data
<a href="#">MME:STOR:SNP:FORM</a>		Data format
<a href="#">MME:STOR:SNP:SEP</a>		Separator of touchstone file
<a href="#">MME:STOR:SNP:TRAC:TRAN</a>		Including trace transform ON/OFF
<a href="#">MME:STOR:SNP:TYPE?</a>		Save type query
<a href="#">MME:STOR:SNP:TYPE:S1P</a>		Sets 1-port file type and port number
<a href="#">MME:STOR:SNP:TYPE:S2P</a>		Sets 2-port file type and ports number
<a href="#">MME:STOR:FDAT</a>	Save Trace Data to CSV File	Saves CSV file
<a href="#">MME:STOR:FDAT:SCOP</a>		Saving scope
<a href="#">MME:STOR:FDAT:FORM</a>		Data format
<a href="#">MME:STOR:FDAT:COMM</a>		Comment ON/OFF

Command		Description
<a href="#">MME:STOR:FDAT:STIM</a>		Stimulus data ON/OFF
<a href="#">MME:STOR:FDAT:SEP</a>		Decimal and value separators
<a href="#">MME:STOR:IMAG</a>	Saving Display Image	Saves the screen to BMP or PNG file

## MMEM:CAT?

### SCPI Command

MMEMory:CATalog? <string>

### Description

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.

query only

### Parameter

<string> Directory name

### Query Response

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} is space in use of the hard drive (byte),

{B} is available space of the hard drive (byte),

{Name n} is name of the n-th file (directory),

{Size n} is size (byte) of the n-th file (directory). Always 0 for directories.

### Equivalent Softkeys

None

---

### Equivalent COM Command

SCPI.MMEMory.CATalog(Dir)

**Syntax**

```
Cat = app.SCPI.MMEemory.CATalog("\.")
```

**Type**

String (read/write)

---

Back to [MMEemory](#)

## MMEM:COPY

### SCPI Command

MMEMory:COPY <string1>,<string2>

### Description

Copies a file.

no query

### Parameter

<string1> Source file name

<string2> Destination file name

### Equivalent Softkeys

None

---

### Equivalent COM Command

SCPI.MMEMory.COPY(Src, Dst)

### Syntax

app.SCPI.MMEMory.COPY(Src, Dst)

### Type

Method

---

Back to [MMEMory](#)

## MMEM:DEL

### SCPI Command

MMEMory:DELeTe <string>

### Description

Deletes a file.

no query

### Parameter

<string> File name

### Equivalent Softkeys

None

---

### Equivalent COM Command

SCPI.MMEMory.DELeTe(File)

### Syntax

app.SCPI.MMEMory.DELeTe(File)

### Type

Method

---

Back to [MMEMory](#)

## MMEM:LOAD

### SCPI Command

MMEMory:LOAD[:STATe] <string>

### Description

Recalls the specified Analyzer state file. The file must be saved by the [MMEM:STOR](#) command.

**Note:** If the full path of the file is not specified, the \State subdirectory of the application directory will be searched. The Analyzer state file has \*.STA extension by default.

no query

### Parameter

<string> File name

### Equivalent Softkeys

Save/Recall > Recall State > State...

---

### Equivalent COM Command

SCPI.MMEMory.LOAD.STATe

### Syntax

app.SCPI.MMEMory.LOAD.STATe = File

### Type

String (write only)

---

Back to [MMEMory](#)



## MMEM:LOAD:CHAN

### SCPI Command

MMEMory:LOAD:CHANnel[:STATe] <char>

### Description

Recalls the Analyzer state for the active channel from the memory register. The state must be saved in one of the four memory registers using the [MMEM:STOR:CHAN](#) command.

no query

### Target

Active channel set by the [DISP:WIND:ACT](#) command.

### Parameter

<char> Choose from:

- A**      Recall from register A
- B**      Recall from register B
- C**      Recall from register C
- D**      Recall from register D

### Equivalent Softkeys

Save/Recall > Recall Channel > {State A | B | C | D}

---

### Equivalent COM Command

SCPI.MMEMory.LOAD.CHANnel.STATe

### Syntax

app.SCPI.MMEMory.LOAD.CHANnel.STATe = "A"

## Type

String (write only)

---

Back to [MMEMory](#)

## MMEM:LOAD:CHAN:CAL

### SCPI Command

MMEMory:LOAD:CHANnel<ch>:CALibration <string>

### Description

Recalls the calibration for the specified channel from the file. The file must be saved using the [MMEM:STOR:CHAN:CAL](#) command.

**Note:** If the full path of the file is not specified, the \State subdirectory of the application directory will be searched. The Analyzer calibration file has \*.CAL extension by default.

no query

### Target

Channel <Ch>,

<Ch>={1|2|...16}

### Parameter

<string> File name

### Equivalent Softkeys

Save/Recall > Recall Channel Calibration...

---

### Equivalent COM Command

None

---

Back to [MMEMory](#)

## MMEM:LOAD:CKIT

### SCPI Command

MMEMory:LOAD:CKIT<Ck> <string>

### Description

Recalls the definition file for the calibration kit. The file must be saved using the [MMEM:STOR:CKIT](#) command.

**Note:** If the full path of the file is not specified, the \CalKit subdirectory of the application directory will be searched. The limit table file has \*.CKD extension by default.

no query

### Target

Calibration kit <Ck>,

<Ck>={{[1]|2|...50}}

### Parameter

<string> File name

### Equivalent Softkeys

Calibration > Cal Kit > Load From File

---

### Equivalent COM Command

SCPI.MMEMory.LOAD.CKIT(Ck)

### Syntax

app.SCPI.MMEMory.LOAD.CKIT(Ck) = File

## Type

String (write only)

---

Back to [MMEMory](#)

## MMEM:LOAD:LIM

### SCPI Command

MMEMory:LOAD:LIMit <string>

### Description

Recalls the limit table file. The file must be saved using the [MMEM:STOR:LIM](#) command.

**Note:** If the full path of the file is not specified, the \Limit subdirectory of the application directory will be searched. The limit table file has \*.LIM extension by default.

no query

### Target

Active trace of the active channel, set by the CALC:PAR:SEL command.

### Parameter

<string> File name

### Equivalent Softkeys

Analysis > Limit Test > Edit Limit Line > Restore Limit Table

---

### Equivalent COM Command

SCPI.MMEMory.LOAD.LIMit

### Syntax

app.SCPI.MMEMory.LOAD.LIMit = File

### Type

String (write only)

---

Back to [MMEMory](#)

## MMEM:LOAD:PLOS

### SCPI Command

MMEMory:LOAD:PLOSs<Pt> <string>

### Description

Recalls the loss compensation file. The file must be saved using the [MMEM:STOR:PLOS](#) command.

**Note:** If the full path of the file is not specified, the \CalKit subdirectory of the application directory will be searched. The loss compensation file has \*.LCT extension by default.

no query

### Target

Port <Pt> of the active channel, set by the [DISP:WIND:ACT](#) command

<Pt>={{1}|2}

### Parameter

<string> File name

### Equivalent Softkeys

**Calibration > Power Calibration > Loss Compen > Import Loss Table**

---

### Equivalent COM Command

SCPI.MMEMory.LOAD.PLOSs(Pt)

### Syntax

app.SCPI.MMEMory.LOAD.PLOSs(Pt) = File

## Type

String (write only)

---

Back to [MMEMory](#)



## MMEM:LOAD:RLIM

### SCPI Command

MMEMory:LOAD:RLIMit <string>

### Description

Recalls the ripple limit table file. The file must be saved using the [MMEM:STOR:RLIM](#) command.

**Note:** If the full path of the file is not specified, the \Limit subdirectory of the application directory will be searched. The ripple limit file has \*.RLM extension by default.

no query

### Target

Active trace of the active channel, set by the [CALC:PAR:SEL](#) command.

### Parameter

<string> File name

### Equivalent Softkeys

Analysis > Ripple Limit > Edit Ripple Limit > Restore Ripple Limit Table

---

### Equivalent COM Command

SCPI.MMEMory.LOAD.RLIMit

### Syntax

app.SCPI.MMEMory.LOAD.RLIMit = File

### Type

String (write only)

---

Back to [MMEMory](#)

## MMEM:LOAD:SEGM

### SCPI Command

MMEMory:LOAD:SEGMENT <string>

### Description

Recalls the segment table file. The file must be saved using the [MMEM:STOR:SEGM](#) command.

**Note:** If the full path of the file is not specified, the \Segment subdirectory of the application directory will be searched. The segment file has \*.SEG extension by default.

no query

### Target

Active channel, set by the [DISP:WIND:ACT](#) command.

### Parameter

<string> File name

### Equivalent Softkeys

Stimulus > Segment Table > Recall...

---

### Equivalent COM Command

SCPI.MMEMory.LOAD.SEGMENT

### Syntax

app.SCPI.MMEMory.LOAD.SEGMENT = File

### Type

String (write only)

---

Back to [MMEMory](#)

## MMEM:LOAD:SNP

### SCPI Command

MMEMory:LOAD:SNP[:DATA] <string>

### Description

Loads the touchstone file with the specified name to the measured S-parameters of the active channel. The touchstone file types 1, 2 port (file extensions S1P, S2P) are supported. On completion of the command, the channel goes to the hold state.

no query

### Target

The active channel set by the [DISP:WIND:ACT](#) command.

### Parameter

<string> File name

### Equivalent Softkeys

Save/Recall > Load Data From Touchstone File > To S-parameters...

---

### Equivalent COM Command

SCPI.MMEMory.LOAD.SNP.DATA

### Syntax

app.SCPI.MMEMory.LOAD.SNP.DATA = File

### Type

String (write only)

---

Back to [MMEMory](#)

# MMEM:LOAD:SNP:FREQ

## SCPI Command

MMEMory:LOAD:SNP:FREQuency[:STATe] {OFF|ON|0|1}

MMEMory:LOAD:SNP:FREQuency[:STATe]?

## Description

Determines whether frequency is set from touchstone file or not when the file is loaded by the command [MMEM:LOAD:SNP](#). If this setting is OFF then the touchstone file data is interpolated or extrapolated.

command/query

## Parameter

- {ON|1}

ON
- {OFF|0}

OFF

## Query Response

{0|1}

## Preset Value

0

## Equivalent Softkeys

None

---

## Equivalent COM Command

None

---

Back to [MMEMory](#)

Enter topic text here.

## MMEM:LOAD:SNP:TRAC:MEM

### SCPI Command

MMEMory:LOAD:SNP:TRACe<Tr>:MEMory <string>

### Description

Loads the Touchstone file with the specified name to the memory trace. The Touchstone file types 1, 2 port (file extensions S1P, S2P) are supported. The current measured S-parameter of data trace selects the appropriate S-parameter from the Touchstone file. After loading, the display of memory trace is automatically switched on.

no query

### Target

The specified memory trace <Tr> of active channel,

<Tr>={{1}|2|...16}

Active channel set by the [DISP:WIND:ACT](#) command.

### Parameter

<string> File name

### Equivalent Softkeys

**Save/Recall > Load Data From Touchstone File > To Active Trace Memory...**

---

### Equivalent COM Command

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

### Syntax

app.SCPI.MMEMory.LOAD.SNP.TRACe(Tr).MEMory = File

## Type

String (write only)

---

Back to [MMEMory](#)

## MMEM:MDIR

### SCPI Command

MMEMory:MDIRectory <string>

### Description

Creates a new directory.

no query

### Parameter

<string> Directory full name

### Equivalent Softkeys

None

---

### Equivalent COM Command

SCPI.MMEMory.MDIRectory

### Syntax

app.SCPI.MMEMory.MDIRectory = Path

### Type

String (write only)

---

Back to [MMEMory](#)

## MMEM:STOR

### SCPI Command

MMEMory:STORe[:STATe] <string>

### Description

Saves the Analyzer state into a file.

**Note:** If the full path of the file is not specified, the \State subdirectory of the application directory will be searched. The state file has \*.STA extension by default.

no query

### Parameter

<string> File name

### Equivalent Softkeys

Save/Recall > Save State > State...

---

### Equivalent COM Command

SCPI.MMEMory.STORe.STATe

### Syntax

app.SCPI.MMEMory.STORe.STATe = File

### Type

String (write only)

---

Back to [MMEMory](#)



## MMEM:STOR:CHAN

### SCPI Command

MMEMory:STORe:CHANnel[:STATe] <char>

### Description

Saves the Analyzer state of the items set for the active channel into one of the four memory registers.

no query

### Target

Active channel set by the DISP:WIND:ACT command

### Parameter

<char> Choose from:

- A**      Save to register A
- B**      Save to register B
- C**      Save to register C
- D**      Save to register D

### Equivalent Softkeys

Save/Recall > Save Channel > {State A | B | C | D}

---

### Equivalent COM Command

SCPI.MMEMory.STORe.CHANnel.STATe

### Syntax

app.SCPI.MMEMory.STORe.CHANnel.STATe = "A"

## Type

String (write only)

---

Back to [MMEMory](#)

## MMEM:STOR:CHAN:CAL

### SCPI Command

MMEMory:STORe:CHANnel<ch>:CALibration <string>

### Description

Stores the calibration of the specified channel to the file.

**Note:** If the full path of the file is not specified, the \State subdirectory of the application directory will be searched. The Analyzer calibration file has \*.CAL extension by default.

no query

### Target

Channel <Ch>,

<Ch>={[1]|2|...16}

### Parameter

<string> File name

### Equivalent Softkeys

Save/Recall > Save Channel Calibration...

---

### Equivalent COM Command

None

---

Back to [MMEMory](#)

# MMEM:STOR:CHAN:CLE

## SCPI Command

MMEMory:STORe:CHANnel:CLEar

## Description

Clears the memory of the channel state saved using the [MMEM:STOR:CHAN](#) command.

no query

## Equivalent Softkeys

Save/Recall > Save Channel > Clear States

---

## Equivalent COM Command

SCPI.MMEMory.STORe.CHANnel.CLEar

## Syntax

app.SCPI.MMEMory.STORe.CHANnel.CLEar

## Type

Method

---

Back to [MMEMory](#)

## MMEM:STOR:CKIT

### SCPI Command

MMEMory:STORe:CKIT<Ck> <string>

### Description

Saves the definition file for the calibration kit.

**Note:** If the full path of the file is not specified, the \CalKit subdirectory of the application directory will be searched. The calibration kit definition file has \*.CKD extension by default.

no query

### Target

Calibration kit <Ck>,

<Ck>={[1]|2|...50}

### Parameter

<string> File name

### Equivalent Softkeys

Calibration > Cal Kit > Save To File

---

### Equivalent COM Command

SCPI.MMEMory.STORe.CKIT(Ck)

### Syntax

app.SCPI.MMEMory.STORe.CKIT(Ck) = File

### Type

String (write only)

---

Back to [MMEMory](#)

## MMEM:STOR:FDAT

### SCPI Command

MMEMory:STORe:FDATa <string>

### Description

Saves the data of one or several traces to a CSV file. The trace number and the file settings can be configured using the [MMEM:STOR:FDAT:XXXX](#) commands.

**Note:** If the full path of the file is not specified, the \CSV subdirectory of the application directory will be searched. The file has \*.CSV extension by default.

no query

### Target

Depending on the [MMEM:STOR:FDAT:SCOPE](#) setting, the active trace or all traces of the active channel.

### Parameter

<string> File name

### Equivalent Softkeys

Save/Recall > Save Trace Data

---

### Equivalent COM Command

SCPI.MMEMory.STORe.FDATa

### Syntax

app.SCPI.MMEMory.STORe.FDATa = File

### Type

String (write only)

---

Back to [MMEMory](#)

# MMEM:STOR:FDAT:SCOP

## SCPI Command

MMEMory:STORe:FDAT:SCOPE {ACTive|ALL}

MMEMory:STORe:FDAT:SCOPE?

## Description

Sets whether the active trace or all traces of the active channel will be saved using the [MMEM:STOR:FDAT](#) command.

command/query

## Parameter

<char> Choose from:

**ACTive**      Active trace only

**ALL**          All traces of the active channel

## Query Response

{ACT|ALL}

## Preset Value

ACT

## Equivalent Softkeys

Save/Recall > Save Trace Data > Scope

---

## Equivalent COM Command

None

---

Back to [MMEMory](#)

# MMEM:STOR:FDAT:FORM

## SCPI Command

MMEMory:STORe:FDAT:FORMat {DB|RI|DISPplayed}

MMEMory:STORe:FDAT:FORMat?

## Description

Sets the data format when the CSV file is saved using the [MMEM:STOR:FDAT](#) command.

command/query

## Parameter

<char> Choose from:

<b>DB</b>	dB/Angle format
<b>RI</b>	Real/Imag format
<b>DISPplayed</b>	Currently displayed trace format

## Query Response

{DB|RI|DISP}

## Preset Value

DB

## Equivalent Softkeys

Save/Recall > Save Trace Data > Format



---

## Equivalent COM Command

None

---

Back to [MMEMory](#)

# MMEM:STOR:FDAT:COMM

## SCPI Command

MMEMory:STORe:FDAT:COMMe[n]t[:STATe] {OFF|ON|0|1}

MMEMory:STORe:FDAT:COMMe[n]t[:STATe]?

## Description

Turns the comment strings at the beginning of the CSV file saved with the [MMEM:STOR:FDAT](#) command ON/OFF. The comment string starts with the '!' character.

command/query

## Parameter

{ON|1}      ON

{OFF|0}      OFF

## Query Response

{0|1}

## Preset Value

0

## Equivalent Softkeys

Save/Recall > Save Trace Data > Comment

## Equivalent COM Command

None

Back to [MMEMory](#)

## MMEM:STOR:FDAT:STIM

### SCPI Command

MMEMory:STORe:FDAT:STIMulus[:STATe] {OFF|ON|0|1}

MMEMory:STORe:FDAT:STIMulus[:STATe]?

### Description

Turns the column with the stimulus data in the CSV file saved with the [MMEM:STOR:FDAT](#) command ON/OFF. The stimulus column is located at the leftmost position.

command/query

### Parameter

{ON|1}      ON

{OFF|0}      OFF

### Query Response

{0|1}

### Preset Value

0

### Equivalent Softkeys

Save/Recall > Save Trace Data > Stimulus

---

### Equivalent COM Command

None

---

Back to [MMEMory](#)

# MMEM:STOR:FDAT:SEP

## SCPI Command

MMEMory:STORe:FDAT:SEParator {POINt|LOCa}

MMEMory:STORe:FDAT:SEParator?

## Description

Sets the separators used when the CSV file is saved with the [MMEM:STOR:FDAT](#) command.

command/query

## Parameter

<char> Choose from:

**POINt** Uses point ('.') as decimal separator and comma (',') as value separator

**LOCa** Uses separators from the Windows locale

## Query Response

{POIN|LOC}

## Preset Value

POINt

## Equivalent Softkeys

Save/Recall > Save Trace Data > Separator

---

## Equivalent COM Command

None

---

Back to [MMEMory](#)

## MMEM:STOR:IMAG

### SCPI Command

MMEMory:STORe:IMAGe <string>

### Description

Saves the display image in BMP or PNG format into a file.

**Note:** If the full path of the file is not specified, the \Image subdirectory of the application directory will be searched. If the file has \*.PNG extension, the file had PNG format, in all the other cases the file has BMP format.

no query

### Parameter

<string> File name

### Equivalent Softkeys

**System > Print > Print Windows > Save as...**

---

### Equivalent COM Command

SCPI.MMEMory.STORe.IMAGe

### Syntax

app.SCPI.MMEMory.STORe.IMAGe = File

### Type

String (write only)

---

Back to [MMEMory](#)

## MMEM:STOR:LIM

### SCPI Command

MMEMory:STORe:LIMit <string>

### Description

Saves the limit table into a file.

**Note:** If the full path of the file is not specified, the \Limit subdirectory of the application directory will be searched. The file has \*.LIM extension by default.

no query

### Target

Active trace of the active channel, set by the CALC:PAR:SEL command.

### Parameter

<string> File name

### Equivalent Softkeys

**Analysis > Limit Test > Edit Limit Line > Save Limit Table**

---

### Equivalent COM Command

SCPI.MMEMory.STORe.LIMit

### Syntax

app.SCPI.MMEMory.STORe.LIMit = File

### Type

String (write only)

---

Back to [MMEMory](#)

# MMEM:STOR:PLOS

## SCPI Command

MMEMory:STORe:PLOSs<Pt> <string>

## Description

Saves the loss compensation file.

**Note:** If the full path of the file is not specified, the \CalKit subdirectory of the application directory will be searched. The loss compensation file has \*.LCT extension by default.

no query

## Target

Port <Pt> of the active channel, set by the [DISP:WIND:ACT](#) command

<Pt>={[1]|2}

## Parameter

<string> File name

## Equivalent Softkeys

Calibration > Power Calibration > Loss Compn > Export Loss Table

---

## Equivalent COM Command

SCPI.MMEMory.STORe.PLOSs(Pt)

## Syntax

app.SCPI.MMEMory.STORe.PLOSs(Pt) = File

## Type

String (write only)

---

Back to [MMEMory](#)

## MMEM:STOR:RLIM

### SCPI Command

MMEMory:STORe:RLIMit <string>

### Description

Saves the ripple limit table into a file.

**Note:** If the full path of the file is not specified, the \Limit subdirectory of the application directory will be searched. The ripple limit file has \*.RLM extension by default.

no query

### Target

Active trace of the active channel, set by the [CALC:PAR:SEL](#) command

### Parameter

<string> File name

### Equivalent Softkeys

**Analysis > Ripple Limit > Edit Ripple Limit > Save Ripple Limit Table**

---

### Equivalent COM Command

SCPI.MMEMory.STORe.RLIMit

### Syntax

app.SCPI.MMEMory.STORe.RLIMit = File

### Type

String (write only)

---

Back to [MMEMory](#)



## MMEM:STOR:SEGM

### SCPI Command

MMEMory:STORe:SEGMent <string>

### Description

Saves the segment table into a file.

**Note:** If the full path of the file is not specified, the \Segment subdirectory of the application directory will be searched. The segment file has \*.SEG extension by default.

no query

### Target

Active channel, set by the [DISP:WIND:ACT](#) command.

### Parameter

<string> File name

### Equivalent Softkeys

Stimulus > Segment Table > Save...

---

### Equivalent COM Command

SCPI.MMEMory.STORe.SEGMent

### Syntax

app.SCPI.MMEMory.STORe.SEGMent = File

### Type

String (write only)

---

Back to [MMEMory](#)

## MMEM:STOR:SNP

### SCPI Command

MMEMory:STORe:SNP[:DATA] <string>

### Description

Saves the measured S-parameters of the active channel into a Touchstone file. The file type (1-port to 4-port) is set by the following commands: [MMEM:STOR:SNP:TYPE:S1P](#), [MMEM:STOR:SNP:TYPE:S2P](#).

**Note:** If the full path of the file is not specified, the \FixtureSim subdirectory of the application directory will be searched. The file has \*.SNP extension by default.

no query

### Target

Active channel, set by the DISP:WIND:ACT command.

### Parameter

<string> File name

### Equivalent Softkeys

Save/Recall > Save Data to Touchstone File > Save File...

---

### Equivalent COM Command

SCPI.MMEMory.STORe.SNP.DATA

### Syntax

app.SCPI.MMEMory.STORe.SNP.DATA = File

### Type

String (write only)

---

Back to [MMEMory](#)

## MMEM:STOR:SNP:FORM

### SCPI Command

MMEMory:STORe:SNP:FORMat <char>

MMEMory:STORe:SNP:FORMat?

### Description

Sets or reads out the data format for the S-parameter saved using the [MMEM:STOR:SNP](#) command.

command/query

### Parameter

<char> Choose from:

**DB**      Logarithmic Magnitude / Angle format

**MA**      Linear Magnitude / Angle format

**RI**      Real part /Imaginary part format

### Query Response

{RI|DB|MA}

### Preset Value

RI

### Equivalent Softkeys

Save/Recall > Save Data to Touchstone File > Format

---

### Equivalent COM Command

SCPI.MMEMory.STORe.SNP.FORMat

**Syntax**

Param = app.SCPI.MMEMory.STORe.SNP.FORMat

app.SCPI.MMEMory.STORe.SNP.FORMat = "DB"

**Type**

String (write only)

---

Back to [MMEMory](#)

## MMEM:STOR:SNP:SEP

### SCPI Command

MMEMory:STORe:SNP:SEParator <char>

MMEMory:STORe:SNP:SEParator?

### Description

Sets or reads out the Touchstone file separator symbol when the S-parameters are saved using the [MMEM:STOR:SNP](#) command.

command/query

### Parameter

<char> Choose from:

**TAB**            Tab symbol (0x09)

**SPACe**        Space symbol (0x20)

### Query Response

{TAB|SPAC}

### Preset Value

TAB

### Equivalent Softkeys

Save/Recall > Save Data to Touchstone File > Separator

---

### Equivalent COM Command

SCPI.MMEMory.STORe.SNP.SEParator

**Syntax**

Param = app.SCPI.MMEMory.STORe.SNP.SEParator

app.SCPI.MMEMory.STORe.SNP.SEParator = "SPACe"

**Type**

String (write only)

---

Back to [MMEMory](#)

# MMEM:STOR:SNP:TRAC:TRAN

## SCPI Command

MMEMory:STORe:SNP:TRACe:TRANSform[:STATe] {OFF|ON|0|1}

MMEMory:STORe:SNP:TRACe:TRANSform[:STATe]?

## Description

Determines whether the S-parameters include the transformation of the active trace or not when saving the Touchstone file. When this feature is ON, the transformation of the active trace takes effect for all S-parameters (Time domain transform, Gating).
command/query

## Parameter

- {ON|1}      ON
- {OFF|0}     OFF

## Query Response

{0|1}

## Preset Value

0

## Equivalent Softkeys

Save/Recall > Save Data to Touchstone File > Including Trace Transform

## Equivalent COM Command

None

Back to [MMEMory](#)

## MMEM:STOR:SNP:TYPE?

### SCPI Command

MMEMory:STORe:SNP:TYPE?

### Description

Reads out the type of Touchstone file (S1P, S2P) to be used when saving S-parameters with the [MMEM:STOR:SNP](#) command.

query only

### Query Response

<string>

{S1P|S2P}

### Equivalent Softkeys

Save/Recall > Save Data to Touchstone File > Type

---

### Equivalent COM Command

None

---

Back to [MMEMory](#)



## MMEM:STOR:SNP:TYPE:S1P

### SCPI Command

MMEMory:STORe:SNP:TYPE:S1P <port>

MMEMory:STORe:SNP:TYPE:S1P?

### Description

Sets or reads out the 1-port Touchstone file type (\*.S1P) and the port number when saving S-parameters using the [MMEM:STOR:SNP](#) command.

command/query

### Parameter

<port> port number from 1 to 2

### Query Response

<numeric>

### Preset Value

1

### Equivalent Softkeys

Save/Recall > Save Data to Touchstone File > Type > 1-Port (S1P)

Save/Recall > Save Data to Touchstone File > Select Port

---

### Equivalent COM Command

SCPI.MMEMory.STORe.SNP.TYPE.S1P

### Syntax

Value = app.SCPI.MMEMory.STORe.SNP.TYPE.S1P

app.SCPI.MMEMory.STORe.SNP.TYPE.S1P = 2

**Type**

Long (read/write)

---

Back to [MMEMory](#)

## MMEM:STOR:SNP:TYPE:S2P

### SCPI Command

MMEMory:STORe:SNP:TYPE:S2P <port1>,<port2>

MMEMory:STORe:SNP:TYPE:S2P?

### Description

Sets or reads out the 2-port Touchstone file type (\*.S2P) and the port number when saving S-parameters using the MMEM:STOR:SNP command.

command/query

### Parameter

<port1>      First port number

<port2>      Second port number

<port> port number from 1 to 2

### Query Response

<numeric1>, <numeric2>

### Equivalent Softkeys

**Save/Recall > Save Data to Touchstone File > Type > 2-Port (S2P)**

**Save/Recall > Save Data to Touchstone File > Select Port (S2P)**

---

### Equivalent COM Command

SCPI.MMEMory.STORe.SNP.TYPE.S2P

**Syntax**

Value = app.SCPI.MMEemory.STORE.SNP.TYPE.S2P

app.SCPI.MMEemory.STORE.SNP.TYPE.S2P = Array(1, 2)

**Type**

Variant (array of long) (read/write)

---

Back to [MMEemory](#)

## MMEM:STOR:STYP

### SCPI Command

MMEMory:STORe:STYPe <char>

MMEMory:STORe:STYPe?

### Description

Selects the type of the Analyzer or channel state saving using the [MMEM:STOR](#) or [MMEM:STOR:CHAN](#) command.

command/query

### Parameter

<char> Choose from:

<b>STATe</b>	Measurement conditions
<b>CSTate</b>	Measurement conditions and calibration
<b>DSTate</b>	Measurement conditions and data
<b>CDSTate</b>	Measurement conditions, calibration, data and memory
<b>CMSTate</b>	Measurement conditions, calibration and memory

### Query Response

{STAT|CST|DST|CDST|CMST}

### Preset Value

CST

### Equivalent Softkeys

Save/Recall > Save Type

---

## Equivalent COM Command

SCPI.MMEemory.STORe.STYPE

## Syntax

Param = app.SCPI.MMEemory.STORe.STYPE

app.SCPI.MMEemory.STORe.STYPE = "STATe"

## Type

String (write only)

---

Back to [MMEemory](#)

## MMEM:TRAN?

### SCPI Command

MMEMory:TRANsfer? <string>

### Description

Transfers the contents of a specified file from the Analyzer to the external PC.

**Note:** The command is not applicable with the TCP/IP Socket protocol. The file must be 20 Mbytes or less.

command/query

### Parameter

<string> the file name with the full path

### Query Response

Block data transfer format. For example:

#6001000<binary block 1000 bytes>

**#6**            Symbol # introduces the data block. The next number indicates how many of the following digits describe the length of the data block;

**001000**       Length of the data block;

### Equivalent Softkeys

None

---

### Equivalent COM Command

None

---

Back to [MMEMory](#)

# OUTP

## SCPI Command

OUTPut[:STATe] {OFF|ON|0|1}

OUTPut[:STATe]?

## Description

Turns the RF signal output ON/OFF. Measurements cannot be performed when the RF signal output is turned OFF.

command/query

## Parameter

{ON|1}      ON

{OFF|0}      OFF

## Query Response

{0|1}

## Preset Value

1

## Equivalent Softkeys

Stimulus > Power > RF Out

---

## Equivalent COM Command

SCPI.OUTPut.STATe

## Syntax

Status = app.SCPI.OUTPut.STATe

app.SCPI.OUTPut.STATe = False



**Type**

Boolean (read/write)

---

## SENSe

Command		Description
<a href="#">SENS:AVER</a>	Averaging	Averaging ON/OFF
<a href="#">SENS:AVER:CLE</a>		Restart averaging
<a href="#">SENS:AVER:COUN</a>		Averaging factor
<a href="#">SENS:BAND</a>	IFBW	IF bandwidth
<a href="#">SENS:BWID</a>		IF bandwidth
<a href="#">SENS:CORR:CLE</a>	Misc Calibration Commands	Clears the table of calibration factors
<a href="#">SENS:CORR:COLL:CLE</a>		Clears data of calibration standards
<a href="#">SENS:CORR:INF?</a>		Information string of calibration
<a href="#">SENS:CORR:STAT</a>		S-parameter error correction state
<a href="#">SENS:CORR:TRIG:FREE</a>		Calibration trigger source
<a href="#">SENS:CORR:TYPE?</a>		Information about trace (calibration type, number of ports)

Command		Description
<a href="#">SENS:CORR:COEF</a>	Read/Write Calibration Coefficients	Calibration coefficient data
<a href="#">SENS:CORR:COEF:METH:ERES</a>		Selects one-path two-port method
<a href="#">SENS:CORR:COEF:METH:OPEN</a>		Selects Response Open method
<a href="#">SENS:CORR:COEF:METH:SHOR</a>		Selects Response Short method
<a href="#">SENS:CORR:COEF:METH:SOLT1</a>		Selects full one-port method
<a href="#">SENS:CORR:COEF:METH:SOLT2</a>		Selects full two-port method
<a href="#">SENS:CORR:COEF:METH:THRU</a>		Selects Response Thru method
<a href="#">SENS:CORR:COEF:SAVE</a>		Enables calibration coefficients
<a href="#">SENS:CORR:COLL:ADAP:DEL</a>	Adapter Removal/ Insertion	Approximate delay of the adapter
<a href="#">SENS:CORR:COLL:ADAP:LENG</a>		Approximate length of the adapter
<a href="#">SENS:CORR:COLL:ADAP:UNIT</a>		Delay units
<a href="#">SENS:CORR:COLL:ADAP:MED</a>		Adapter media
<a href="#">SENS:CORR:COLL:ADAP:PERM</a>		Permittivity of the adapter media

Command		Description
<a href="#">SENS:CORR:COLL:ADAP:WAV:CUT</a>		Cutoff frequency of the waveguide adapter
<a href="#">SENS:CORR:COLL:METH:ADAP:REM</a>		Adapter Removal/Insertion ON/OFF
<a href="#">SENS:CORR:COLL:CKIT</a>	Calibration Kit Management	calibration kit selection
<a href="#">SENS:CORR:COLL:CKIT:DESC</a>		calibration kit description string
<a href="#">SENS:CORR:COLL:CKIT:LAB</a>		calibration kit label
<a href="#">SENS:CORR:COLL:CKIT:RES</a>		Remove or restore a calibration kit
<a href="#">SENS:CORR:COLL:CKIT:STAN:INS</a>		Insert the standard into a calibration kit
<a href="#">SENS:CORR:COLL:CKIT:STAN:REM</a>		Delete the standard from a calibration kit
<a href="#">SENS:CORR:COLL:CKIT:ORD:LOAD</a>	Assigning Class to Calibration Standard	"Load" Class
<a href="#">SENS:CORR:COLL:CKIT:ORD:OPEN</a>		"Open" Class
<a href="#">SENS:CORR:COLL:CKIT:ORD:SEL</a>		Assignment of subclass
<a href="#">SENS:CORR:COLL:CKIT:ORD:SHOR</a>		"Short" class
<a href="#">SENS:CORR:COLL:CKIT:ORD:THRU</a>		"Thru" class

Command		Description
<a href="#">SENS:CORR:COLL:CKIT:ORD:TRLL</a>		"TRL Line" class
<a href="#">SENS:CORR:COLL:CKIT:ORD:TRLT</a>		"TRL Thru" class
<a href="#">SENS:CORR:COLL:CKIT:ORD:TRLR</a>		"TRL Reflect" class
<a href="#">SENS:CORR:COLL:CKIT:STAN:ARB</a>	Calibration Standard Definition	Arbitrary impedance (Load)
<a href="#">SENS:CORR:COLL:CKIT:STAN:C0</a>		Capacitance C0 (Open)
<a href="#">SENS:CORR:COLL:CKIT:STAN:C1</a>		Capacitance C1 (Open)
<a href="#">SENS:CORR:COLL:CKIT:STAN:C2</a>		Capacitance C2 (Open)
<a href="#">SENS:CORR:COLL:CKIT:STAN:C3</a>		Capacitance C3 (Open)
<a href="#">SENS:CORR:COLL:CKIT:STAN:COUN?</a>		Number of standards in the calibration kit
<a href="#">SENS:CORR:COLL:CKIT:STAN:DATA</a>		S-parameters of the data-based calibration standard
<a href="#">SENS:CORR:COLL:CKIT:STAN:DEL</a>		Offset delay
<a href="#">SENS:CORR:COLL:CKIT:STAN:FMAX</a>		Max frequency

Command		Description
<a href="#">SENS:CORR:COLL:CKIT:STAN:FMIN</a>		Min frequency
<a href="#">SENS:CORR:COLL:CKIT:STAN:L0</a>		Inductance L0 (Short)
<a href="#">SENS:CORR:COLL:CKIT:STAN:L1</a>		Inductance L1 (Short)
<a href="#">SENS:CORR:COLL:CKIT:STAN:L2</a>		Inductance L2 (Short)
<a href="#">SENS:CORR:COLL:CKIT:STAN:L3</a>		Inductance L3 (Short)
<a href="#">SENS:CORR:COLL:CKIT:STAN:LAB</a>		Standard label
<a href="#">SENS:CORR:COLL:CKIT:STAN:LOSS</a>		Offset loss
<a href="#">SENS:CORR:COLL:CKIT:STAN:TYPE</a>		Standard type
<a href="#">SENS:CORR:COLL:CKIT:STAN:Z0</a>		Offset Z0
<a href="#">SENS:CORR:COLL:DATA:ISOL</a>	Read/Write Measurement of Calibration Standards	Measurement array of Isolation
<a href="#">SENS:CORR:COLL:DATA:LOAD</a>		Measurement array of Load
<a href="#">SENS:CORR:COLL:DATA:OPEN</a>		Measurement array of Open
<a href="#">SENS:CORR:COLL:DATA:SHOR</a>		Measurement array of Short

Command		Description
<a href="#">SENS:CORR:COLL:DATA:THRU:MATC</a>		Reflection measurement array of Thru
<a href="#">SENS:CORR:COLL:DATA:THRU:TRAN</a>		Transmission measurement array of Thru
<a href="#">SENS:CORR:COLL:ECAL:CCH</a>	Automatic Calibration Module	Confidence check of calibration coefficients
<a href="#">SENS:CORR:COLL:ECAL:ERES</a>		Procedure of one path two-port calibration
<a href="#">SENS:CORR:COLL:ECAL:INF?</a>		Information about connected module
<a href="#">SENS:CORR:COLL:ECAL:ORI:EXEC</a>		Auto-Orientation procedure
<a href="#">SENS:CORR:COLL:ECAL:ORI:STAT</a>		Auto-Orientation ON/OFF
<a href="#">SENS:CORR:COLL:ECAL:PATH</a>		Manual module orientation
<a href="#">SENS:CORR:COLL:ECAL:SOLT1</a>		Procedure of one-port calibration
<a href="#">SENS:CORR:COLL:ECAL:SOLT2</a>		Procedure of full two-port calibration
<a href="#">SENS:CORR:COLL:ECAL:THER:COMP</a>		Thermo compensation ON/OFF
<a href="#">SENS:CORR:COLL:ECAL:UCH</a>		Characterization number
<a href="#">SENS:CORR:COLL:ECAL:UTHR:STAT</a>		Unknown Thru feature ON/OFF

Command		Description
<a href="#">SENS:CORR:COLL:ISOL</a>	Measurement of Calibration Standards	Isolation
<a href="#">SENS:CORR:COLL:LOAD</a>		Load
<a href="#">SENS:CORR:COLL:OPEN</a>		Open
<a href="#">SENS:CORR:COLL:SHOR</a>		Short
<a href="#">SENS:CORR:COLL:THRU</a>		Thru
<a href="#">SENS:CORR:COLL:TRL</a>		TRL Line
<a href="#">SENS:CORR:COLL:TRLT</a>		TRL Thru
<a href="#">SENS:CORR:COLL:TRLR</a>		TRL Reflect
<a href="#">SENS:CORR:COLL:SUBC</a>		Subclass number
<a href="#">SENS:CORR:COLL:METH:ERES</a>	Calibration Method	One path two-port
<a href="#">SENS:CORR:COLL:METH:OPEN</a>		Response Open
<a href="#">SENS:CORR:COLL:METH:SHOR</a>		Response Short
<a href="#">SENS:CORR:COLL:METH:SOLT1</a>		Full one-port (SOL)



Command		Description
<a href="#">SENS:CORR:COLL:METH:SOLT2</a>		Full two-port (SOLT)
<a href="#">SENS:CORR:COLL:METH:THRU</a>		Response Thru
<a href="#">SENS:CORR:COLL:METH:TRL:MULT</a>		Multi-line TRL option ON/OFF
<a href="#">SENS:CORR:COLL:METH:TRL2</a>		Two-port TRL
<a href="#">SENS:CORR:COLL:METH:TYPE?</a>		Calibration method query.
<a href="#">SENS:CORR:COLL:SAVE</a>	Calibration Completion	Calibration completion
<a href="#">SENS:CORR:COLL:THRU:ADD:DEL</a>	Unknown Thru Addition	Approximate delay of the Thru
<a href="#">SENS:CORR:COLL:THRU:ADD:LENG</a>		Approximate length of the Thru
<a href="#">SENS:CORR:COLL:THRU:ADD:UNIT</a>		Delay Units
<a href="#">SENS:CORR:COLL:THRU:ADD:MED</a>		Thru Media
<a href="#">SENS:CORR:COLL:THRU:ADD:PERM</a>		Permittivity of the Thru media
<a href="#">SENS:CORR:COLL:THRU:ADD:WAV:CUT</a>		Cutoff frequency of the waveguide Thru

Command		Description
<a href="#">SENS:CORR:COLL:THRU:ADD:FULL2:COMP</a>		Completion of the full two-port calibration
<a href="#">SENS:CORR:EXT</a>	Port Extension	Port extension ON/OFF
<a href="#">SENS:CORR:EXT:PORT:FREQ</a>		Values of "Frequency1" and "Frequency2"
<a href="#">SENS:CORR:EXT:PORT:INCL</a>		Loss compensation ON/OFF
<a href="#">SENS:CORR:EXT:PORT:LDC</a>		Value "Loss at DC"
<a href="#">SENS:CORR:EXT:PORT:LOSS</a>		Loss compensation ON/OFF
<a href="#">SENS:CORR:EXT:PORT:TIME</a>		Extension Port n
<a href="#">SENS:CORR:EXT:AUTO:CONF</a>	Auto Port Extension	Frequency range configuration
<a href="#">SENS:CORR:EXT:AUTO:DcoF</a>		"Loss at DC" value ON/OFF
<a href="#">SENS:CORR:EXT:AUTO:LOSS</a>		"Loss1" and "Loss2" values ON/OFF
<a href="#">SENS:CORR:EXT:AUTO:MEAS</a>		Measurement of Short or Open
<a href="#">SENS:CORR:EXT:AUTO:PORT</a>		Auto port extension for the specified port ON/OFF

Command		Description
<a href="#">SENS:CORR:EXT:AUTO:RES</a>		Restart averaging between Short and Open
<a href="#">SENS:CORR:EXT:AUTO:STAR</a>		Start frequency of user span
<a href="#">SENS:CORR:EXT:AUTO:STOP</a>		Stop frequency of user span
<a href="#">SENS:CORR:IMP</a>	System Impedance Setting	System Z0
<a href="#">SENS:CORR:IMP:SEL:AUTO</a>		Auto-select Z0 ON/OFF
<a href="#">SENS:CORR:PORT:IMP</a>		System Z0 for the specified port
<a href="#">SENS:CORR:OFFS:CLE</a>	Scalar Mixer Calibration	Clears calibration coefficient table
<a href="#">SENS:CORR:OFFS:COLL:CLE</a>		Clears calibration data
<a href="#">SENS:CORR:OFFS:COLL:DIR</a>		Calibration direction
<a href="#">SENS:CORR:OFFS:COLL:ECAL</a>		Measure all standards using ACM
<a href="#">SENS:CORR:OFFS:COLL:LOAD</a>		Measure the Load standard
<a href="#">SENS:CORR:OFFS:COLL:METH:SMIX2</a>		Calibration port
<a href="#">SENS:CORR:OFFS:COLL:OPEN</a>		Measure the Open standard

Command		Description
<a href="#">SENS:CORR:OFFS:COLL:PMET</a>		Measure power
<a href="#">SENS:CORR:OFFS:COLL:SHOR</a>		Measure the Short standard
<a href="#">SENS:CORR:OFFS:COLL:THRU</a>		Measure the Thru standard
<a href="#">SENS:CORR:OFFS:COLL:SAVE</a>		Completes calibration
<a href="#">SENS:CORR:REC</a>	Receiver Calibration	Receiver correction ON/OFF
<a href="#">SENS:CORR:REC:COLL:ACQ</a>		Calibration procedure for both receivers
<a href="#">SENS:CORR:REC:COLL:RCH:ACQ</a>		Reference receiver calibration procedure
<a href="#">SENS:CORR:REC:COLL:TCH:ACQ</a>		Test receiver calibration procedure
<a href="#">SENS:CORR:REC:OFFS:AMPL</a>		Power offset
<a href="#">SENS:CORR:TRAN:TIME:FREQ</a>	Cable Correction	Frequency at which cable loss specified
<a href="#">SENS:CORR:TRAN:TIME:LOSS</a>		Cable loss
<a href="#">SENS:CORR:TRAN:TIME:RVEL</a>		Cable velocity factor
<a href="#">SENS:CORR:TRAN:TIME:STAT</a>		Cable correction ON/OFF

Command		Description
<a href="#">SENS:CORR:VMC:COLL:ECAL:SAVE</a>	Vector Mixer Calibration	Complete the calibration using ACM
<a href="#">SENS:CORR:VMC:COLL:PORT</a>		Port number for calibration mixer with LPF filter
<a href="#">SENS:CORR:VMC:COLL:IF:SEL</a>		IF frequency (RF+LO, RF-LO, LO-RF)
<a href="#">SENS:CORR:VMC:COLL:LO:FREQ</a>		LO frequency
<a href="#">SENS:CORR:VMC:COLL:LOAD</a>		Measure the Load standard
<a href="#">SENS:CORR:VMC:COLL:OPEN</a>		Measure the Open standard
<a href="#">SENS:CORR:VMC:COLL:SHOR</a>		Measure the Short standard
<a href="#">SENS:CORR:VMC:COLL:OPT</a>		Setup option (de-embedding calibration mixer + filter at completion of calibration) ON/OFF
<a href="#">SENS:CORR:VMC:COLL:SAVE</a>		Complete the calibration, calculate S-parameters, write the touchstone file
<a href="#">SENS:DATA:CORR?</a>	Data Transfer	Corrected S-parameter data or corrected receiver data
<a href="#">SENS:DATA:RAWD?</a>		Raw S-parameter data or raw receiver data

Command		Description
<a href="#">SENS:FREQ:DATA?</a>		Stimulus data
<a href="#">SENS:FREQ</a>	Stimulus Settings	Fixed frequency for a power sweep
<a href="#">SENS:FREQ:CENT</a>		Center frequency
<a href="#">SENS:FREQ:SPAN</a>		Span frequency
<a href="#">SENS:FREQ:STAR</a>		Start power
<a href="#">SENS:FREQ:STOP</a>		Stop power
<a href="#">SENS:SEGM:DATA</a>		Segment sweep table
<a href="#">SENS:SWE:CW:TIME</a>		Sweep Time
<a href="#">SENS:SWE:POIN</a>		Number of points
<a href="#">SENS:SWE:POIN:TIME</a>		Point delay
<a href="#">SENS:SWE:REV</a>		Reverse sweep ON/OFF
<a href="#">SENS:SWE:TYPE</a>		Sweep type
<a href="#">SENS:OFFS:ADJ</a>		Frequency offset adjust ON/OFF

Command		Description	
<a href="#">SENS:OFFS:ADJ:CONT:PER</a>	Mixer Measurements	Adjust period	
<a href="#">SENS:OFFS:ADJ:EXEC</a>		Executes adjustment once	
<a href="#">SENS:OFFS:ADJ:PATH</a>		Adjustment path	
<a href="#">SENS:OFFS:ADJ:PORT</a>		Adjusted Port	
<a href="#">SENS:OFFS:ADJ:VAL</a>		Adjust Value	
<a href="#">SENS:OFFS</a>		Frequency offset ON/OFF	
<a href="#">SENS:OFFS:PORT:DATA?</a>		Port offset data	
<a href="#">SENS:OFFS:PORT:DIV</a>		Port offset settings	Divisor
<a href="#">SENS:OFFS:PORT:MULT</a>			Multiplier
<a href="#">SENS:OFFS:PORT:OFFS</a>			Offset
<a href="#">SENS:OFFS:PORT:STAR</a>			Start
<a href="#">SENS:OFFS:PORT:STOP</a>			Stop

Command		Description	
<a href="#">SENS:OFFS:REC:DATA?</a>		Receiver offset data	
<a href="#">SENS:OFFS:REC:DIV</a>		Receiver offset settings	Divisor
<a href="#">SENS:OFFS:REC:MULT</a>			Multiplier
<a href="#">SENS:OFFS:REC:OFFS</a>			Offset
<a href="#">SENS:OFFS:REC:STAR</a>			Start
<a href="#">SENS:OFFS:REC:STOP</a>			Stop
<a href="#">SENS:OFFS:SOUR:DATA?</a>		Source offset data	
<a href="#">SENS:OFFS:SOUR:DIV</a>		Source offset settings	Divisor
<a href="#">SENS:OFFS:SOUR:MULT</a>			Multiplier
<a href="#">SENS:OFFS:SOUR:OFFS</a>			Offset
<a href="#">SENS:OFFS:SOUR:STAR</a>			Start
<a href="#">SENS:OFFS:SOUR:STOP</a>			Stop



Command		Description
<a href="#">SENS:OFFS:TYPE</a>		Offset type
<a href="#">SENS:ROSC:SOUR</a>	Analyzer Parameters	Reference source
<a href="#">SENS:VOLT:DC:RANG:UPP</a>	DC Measurement	DC voltage range

## SENS:AVER

### SCPI Command

SENSe<Ch>:AVERage[:STATe] {OFF|ON|0|1}

SENSe<Ch>:AVERage[:STATe]?

### Description

Turns the measurement averaging function ON/OFF.

command/query

### Target

Channel <Ch>,

<Ch>={1|2|...16}

Parameter

{ON|1}      ON

{OFF|0}      OFF

### Query Response

{0|1}

### Preset Value

0

### Related Commands

[SENS:AVER:COUN](#)

### Equivalent Softkeys

Average > Averaging

---

### Equivalent COM Command

SCPI.SENSE(Ch).AVERage.STATe

### Syntax

Status = app.SCPI.SENSE(Ch).AVERage.STATe

app.SCPI.SENSE(Ch).AVERage.STATe = False

### Type

Boolean (read/write)

---

Back to [SENSe](#)

## SENS:AVER:CLE

### SCPI Command

SENSe<Ch>:AVERage:CLEar

### Description

Restarts the averaging process when the averaging function is turned on.

no query

### Target

Channel <Ch>,

<Ch>={[1]|2|...16}

### Related Commands

[SENS:AVER](#)

### Equivalent Softkeys

None

---

### Equivalent COM Command

SCPI.SENSE(CH).AVERage.CLEar

### Syntax

app.SCPI.SENSE(CH).AVERage.CLEar

### Type

Method

---

Back to [SENSe](#)

## **SENS:AVER:COUN**

### **SCPI Command**

SENSe<Ch>:AVERage:COUNT <numeric>

SENSe<Ch>:AVERage:COUNT?

### **Description**

Sets or reads out the averaging factor when the averaging function is turned on.

command/query

### **Target**

Channel <Ch>,

<Ch>={{1}|2|...16}

### **Parameter**

<numeric> the averaging factor from 1 to 999

### **Out of Range**

Sets the value of the limit, which is closer to the specified value.

### **Query Response**

<numeric>

### **Preset Value**

10

### **Related Commands**

[SENS:AVER](#)

### **Equivalent Softkeys**

**Average > Avg Factor**

---

### Equivalent COM Command

SCPI.SENSE(Ch).AVERage.COUNT

### Syntax

Value = app.SCPI.SENSE(Ch).AVERage.COUNT

app.SCPI.SENSE(Ch).AVERage.COUNT = 2

### Type

Long (read/write)

---

Back to [SENSe](#)

## SENS:BAND

### SCPI Command

SENSe<Ch>:BANDwidth[:RESolution] <frequency>

SENSe<Ch>:BANDwidth[:RESolution]?

### Description

Sets or reads out the IF bandwidth.

command/query

### Target

Channel <Ch>,

<Ch>={{1}|2|...16}

### Parameter

<frequency> the IF bandwidth value

### Unit

Hz (Hertz)

### Out of Range

Sets the value of the limit, which is closer to the specified value.

### Query Response

<numeric>

### Preset Value

10 kHz

### Resolution

In steps of 1, 1.5, 2, 3, 5, 7

## Related Commands

[SENS:BWID](#) — similar command

## Equivalent Softkeys

**Average > IF Bandwidth**

---

## Equivalent COM Command

SCPI.SENSE(Ch).BANDwidth.RESolution

## Syntax

Value = app.SCPI.SENSE(Ch).BANDwidth.RESolution

app.SCPI.SENSE(Ch).BANDwidth.RESolution = 100

## Type

Double (read/write)

---

Back to [SENSe](#)



## SENS:BWID

### SCPI Command

SENSe<Ch>:BWIDth[:RESolution] <frequency>

SENSe<Ch>:BWIDth[:RESolution]?

### Description

Sets or reads out the IF bandwidth.

command/query

### Target

Channel <Ch>,

<Ch>={{[1]|2|...16}}

### Parameter

<frequency> the IF bandwidth value

### Unit

Hz (Hertz)

### Out of Range

Sets the value of the limit, which is closer to the specified value.

### Query Response

<numeric>

### Preset Value

10 kHz

### Resolution

In steps of 1, 1.5, 2, 3, 5, 7

## Related Commands

[SENS:BAND](#) — similar command

## Equivalent Softkeys

Average > IF Bandwidth

---

## Equivalent COM Command

SCPI.SENSE(Ch).BANDwidth.RESolution

## Syntax

Value = app.SCPI.SENSE(Ch).BANDwidth.RESolution

app.SCPI.SENSE(Ch).BANDwidth.RESolution = 100

## Type

Double (read/write)

---

Back to [SENSe](#)

## SENS:CORR:CLE

### SCPI Command

SENSe<Ch>:CORRection:CLEar

### Description

Clears the calibration coefficient table.

no query

### Target

Channel <Ch>,

<Ch>={{1}|2|...16}

### Equivalent Softkeys

None

---

### Equivalent COM Command

SCPI.SENSE(CH).CORRection.CLEar

### Syntax

app.SCPI.SENSE(CH).CORRection.CLEar

### Type

Method

---

Back to [SENSe](#)

## SENS:CORR:COEF

### SCPI Command

SENSe<Ch>:CORRection:COEFficient[:DATA]  
<char>,<rcvport>,<srcport>,<numeric list>

SENSe<Ch>:CORRection:COEFficient[:DATA]? <char>,<rcvport>,<srcport>

### Description

Writes or reads out the calibration coefficient data array.

The array size is 2N, where N is the number of measurement points. For the n–th point, where n from 1 to N:

<numeric 2n–1> real part of the calibration coefficients;

<numeric 2n> imaginary part of the calibration coefficients.

**Note:** The written calibration coefficients become effective only after the [SENS:CORR:COEF:SAVE](#) command is executed.

command/query

### Target

Channel <Ch>,

<Ch>={[1]|2|...16}

### Parameter

<char> Specifies the Error term:

<b>ER</b>	Reflection tracking
<b>ED</b>	Directivity
<b>ES</b>	Source match
<b>ET</b>	Transmission tracking
<b>EX</b>	Isolation

**EL** Load match

<rcvport> the number of the receiver port from 1 to 2

<srcport> the number of the source port from 1 to 2

<numeric list> the calibration coefficient array

When ES, ER, or ED is used, the numbers of the ports <rcvport> and <srcport> must be the same. When EL, ET, or EX is used, the numbers of the ports <rcvport> and <srcport> must be different.

### Query Response

<numeric 1>, <numeric 2>, ...<numeric 2N>

### Related Commands

SENS:CORR:COEF:SAVE

### Equivalent Softkeys

None

---

### Equivalent COM Command

SCPI.SENSE(Ch).CORRection.COEFficient.DATA(Str, Pt\_r, Pt\_s)

### Syntax

Data = app.SCPI.SENSE(Ch).CORRection.COEFficient. DATA(Str,Pt\_r, Pt\_s)

app.SCPI.SENSE(Ch).CORRection.COEFficient.DATA(Str, Pt\_r, Pt\_s) = Data

### Type

Variant (array of Double) (read/write)

---

Back to [SENSe](#)

## SENS:CORR:COEF:METH:ERES

### SCPI Command

SENSe<Ch>:CORRection:COEFficient:METHod:ERESponse <rcvport>,<srcport>

### Description

Selects the ports and sets the 1–path 2–port calibration type when the written calibration coefficients are made effective by the [SENS:CORR:COEF:SAVE](#) command.

no query

### Target

Channel <Ch>,

<Ch>={[1]|2|...16}

### Parameter

<rcvport>                      The number of the receiver port from 1 to 2

<srcport>                      The number of the source port from 1 to 2

### Out of Range

If the same port numbers are specified, an error occurs.

### Related Commands

[SENS:CORR:COEF:SAVE](#)

### Equivalent Softkeys

None

---

### Equivalent COM Command

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

**Syntax**

Ports = Array(2, 1)

app.SCPI.SENSE(Ch).CORRection.COEFficient.METHod.ERESponse = Ports

**Type**

Variant (array of long) (read/write)

---

Back to [SENSe](#)

## SENS:CORR:COEF:METH:OPEN

### SCPI Command

SENSe<Ch>:CORRection:COEFficient:METHod[:RESPonse]:OPEN <port>

### Description

Selects the port and sets the response calibration (Open) type when the written calibration coefficients are made effective by the [SENS:CORR:COEF:SAVE](#) command.

no query

### Target

Channel <Ch>,

<Ch>={[1]|2|...16}

### Parameter

<port>      The number of the port from 1 to 2

### Related Commands

[SENS:CORR:COEF:SAVE](#)

### Equivalent Softkeys

None

---

### Equivalent COM Command

SCPI.SENSE<Ch>.CORRection.COEFficient.METHod.RESPonse.OPEN

### Syntax

Port = 1

app.SCPI.SENSE<Ch>.CORRection.COEFficient.METHod.RESPonse.OPEN = Port



**Type**

Long (write only)

---

Back to [SENSe](#)

## SENS:CORR:COEF:METH:SHOR

### SCPI Command

SENSe<Ch>:CORRection:COEFficient:METHod[:RESPonse]:SHORT <port>

### Description

Selects the port and sets the response calibration (Short) type when the written calibration coefficients are made effective by the [SENS:CORR:COEF:SAVE](#) command.

no query

### Target

Channel <Ch>,

<Ch>={[1]|2|...16}

### Parameter

<port>      The number of the port from 2

### Related Commands

[SENS:CORR:COEF:SAVE](#)

### Equivalent Softkeys

None

---

### Equivalent COM Command

SCPI.SENSE<Ch>.CORRection.COEFficient.METHod.RESPonse.SHORt

### Syntax

Port = 1

app.SCPI.SENSE<Ch>.CORRection.COEFficient.METHod.RESPonse.SHORt = Port

**Type**

Long (write only)

---

Back to [SENSe](#)

## SENS:CORR:COEF:METH:SOLT1

### SCPI Command

SENSe<Ch>:CORRection:COEFficient:METHod:SOLT1 <port>

### Description

Selects the port and sets the full one-port calibration type when the written calibration coefficients are made effective by the [SENS:CORR:COEF:SAVE](#) command.

no query

### Target

Channel <Ch>,

<Ch>={[1]|2|...16}

### Parameter

<port>      The number of the port from 1 to 2

### Related Commands

[SENS:CORR:COEF:SAVE](#)

### Equivalent Softkeys

None

---

### Equivalent COM Command

SCPI.SENSE<Ch>.CORRection.COEFficient.METHod.SOLT1

### Syntax

Port = 1

app.SCPI.SENSE<Ch>.CORRection.COEFficient.METHod.SOLT1= Port

**Type**

Long (write only)

---

Back to [SENSe](#)

## SENS:CORR:COEF:METH:SOLT2

### SCPI Command

SENSe<Ch>:CORRection:COEFficient:METHod:SOLT2 <port1>,<port2>

### Description

Selects the ports and sets the full two-port calibration type when the written calibration coefficients are made effective by the [SENS:CORR:COEF:SAVE](#) command.

no query

### Target

Channel <Ch>,

<Ch>={{1}|2|...16}

### Parameter

<port1>      The first port number from 1 to 2

<port2>      The second port number from 1 to 2

### Out of Range

If the same port numbers are specified, an error occurs.

### Related Commands

[SENS:CORR:COEF:SAVE](#)

### Equivalent Softkeys

None

---

### Equivalent COM Command

SCPI.SENSe(Ch).CORRection.COEFficient.METHod.SOLT2

**Syntax**

Ports = Array(1,2)

app.SCPI.SENSE(Ch).CORRection.COEFficient.METHod.SOLT2 = Ports

**Type**

Variant (array of Long) (write only)

---

Back to [SENSe](#)

## SENS:CORR:COEF:METH:THRU

### SCPI Command

SENSe<Ch>:CORRection:COEFficient:METHod[:RESPonse]:THRU <rcvport>,<srcport>

### Description

Selects the ports and sets the response calibration (Thru) type when the written calibration coefficients are made effective by the [SENS:CORR:COEF:SAVE](#) command.

no query

### Target

Channel <Ch>,

<Ch>={1|2|...16}

### Parameter

<rcvport>                      The number of the receiver port from 1 to 2

<srcport>                      The number of the source port from 1 to 2

### Out of Range

If the same port numbers are specified, an error occurs.

### Related Commands

[SENS:CORR:COEF:SAVE](#)



## Equivalent Softkeys

None

---

## Equivalent COM Command

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

## Syntax

Ports = Array(1, 2)

app.SCPI.SENSE(Ch).CORRection.COEFficient.METHod.RESPOnse.THRU = Ports

## Type

Variant (array of Long) (write only)

---

Back to [SENSe](#)

## **SENS:CORR:COEF:SAVE**

### **SCPI Command**

SENSe<Ch>:CORRection:COEFficient:SAVE

### **Description**

Enables the written calibration coefficients depending on the selected calibration type. On completion of the command, the error correction automatically turns ON.

Executing this command before all necessary calibration coefficients have been written will result in an error and the command will be ignored.

no query

### **Target**

Channel <Ch>,

<Ch>={[1]|2|...16}

### **Related Commands**

Calibration type selection:

[SENS:CORR:COEF:METH:ERES](#)

[SENS:CORR:COEF:METH:OPEN](#)

[SENS:CORR:COEF:METH:SHOR](#)

[SENS:CORR:COEF:METH:THRU](#)

[SENS:CORR:COEF:METH:SOLT1](#)

[SENS:CORR:COEF:METH:SOLT2](#)

Calibration coefficient writing:

[SENS:CORR:COEF](#)

### **Equivalent Softkeys**

None

---

**Equivalent COM Command**

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

**Syntax**

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

**Type**

Method

---

Back to [SENSe](#)

## SENS:CORR:COLL:ADAP:DEL

### SCPI Command

SENSe<Ch>:CORRection:COLLect:ADAPter:DELay <numeric>

SENSe<Ch>:CORRection:COLLect:ADAPter:DELay?

### Description

Sets or reads out the approximate delay value of an adapter in the adapter removal/insertion function. This value is used to eliminate the uncertainty of  $\pm 180^\circ$  when calculating the phase response of the adapter.

The sign of the value depends on the type of the removal / insertion function. The value must be negative for the adapter removal function and must be positive for the adapter insertion function.

If this value is set to zero, the analyzer uses an algorithm to automatically determine the delay of the adapter. In most cases setting this value to zero is enough. Setting this value to non zero is required when:

$$FrequencyStep > \frac{1}{2Delay}$$

**Note:** The delay and the length of the adapter can be set mutually  
$$Delay = \frac{Length \sqrt{Permittivity}}{c}$$

command/query

### Target

Channel <Ch>,

<Ch>={[1]|2|...16}

### Parameter

<numeric> the approximate delay value of the adapter (with minus sign when adapter is removed).

### Unit

sec (Seconds)

**Query Response**

<numeric>

**Preset Value**

0

**Equivalent Softkeys**

**Calibration > Calibrate > Adapter Removal / Insertion > Adapter Delay**

---

**Equivalent COM Command**

None

---

Back to [SENSe](#)

## SENS:CORR:COLL:ADAP:LENG

### SCPI Command

SENSe<Ch>:CORRection:COLLect:ADAPter:LENGth <numeric>

SENSe<Ch>:CORRection:COLLect:ADAPter:LENGth?

### Description

Sets or reads out the approximate value of the mechanical length of the adapter in the adapter removal/insertion function. This value is used to eliminate the uncertainty of  $\pm 180^\circ$  when calculating the phase response of the adapter.

The sign of the value depends on the type of the removal / insertion function. The value must be negative for the adapter removal function and must be positive for the adapter insertion function.

If this value is set to zero, the analyzer uses an algorithm to automatically determine the delay of the adapter. In most cases setting this value to zero is enough. Setting this value to non zero is required when:

$$FrequencyStep > \frac{1}{2Delay}$$

**Note:** The delay and the length of the adapter can be set mutually

$$Delay = \frac{Length \sqrt{Permittivity}}{c}$$

command/query

### Target

Channel <Ch>,

<Ch>={{1}|2|...16}

### Parameter

<numeric> the approximate delay length of the adapter (with minus sign when adapter is removed).

### Unit

m (Meters)

**Query Response**

<numeric>

**Preset Value**

0

**Equivalent Softkeys**

**Calibration > Calibrate > Adapter Removal / Insertion > Adapter Delay**

---

**Equivalent COM Command**

None

---

Back to [SENSe](#)

## **SENS:CORR:COLL:ADAP:UNIT**

### **SCPI Command**

SENSe<Ch>:CORRection:COLLect:ADAPter:UNIT {SEConds|METers}

SENSe<Ch>:CORRection:COLLect:ADAPter:UNIT?

### **Description**

Selects the display units of the adapter delay (length) in the adapter removal/insertion function.

command/query

### **Target**

Channel <Ch>,

<Ch>={[1]|2|...16}

### **Parameter**

**SEConds**    Selects the seconds

**METers**     Selects the meters

### **Query Response**

{SEC|MET}

### **Preset Value**

SEConds

### **Equivalent Softkeys**

**Calibration > Calibrate > Adapter Removal / Insertion > Delay Unit**



---

## Equivalent COM Command

None

---

Back to [SENSe](#)

## SENS:CORR:COLL:ADAP:MED

### SCPI Command

SENSe<Ch>:CORRection:COLLect:ADAPter:MEDia {COAXial|WAVeguide}

SENSe<Ch>:CORRection:COLLect:ADAPter:MEDia?

### Description

Specifies the adapter media in the adapter removal/insertion function.

**Note:** When the waveguide adapter is used, specify the adapter length instead of delay.

command/query

### Target

Channel <Ch>,

<Ch>={[1]2|...16}

### Parameter

**COAXial** Specifies the coaxial adapter

**WAVeguide** Specifies the waveguide adapter

### Query Response

{COAX|WAV}

### Preset Value

COAXial

### Equivalent Softkeys

Calibration > Calibrate > Adapter Removal / Insertion > Adapter Media

---

## Equivalent COM Command

None

---

Back to [SENSe](#)

## **SENS:CORR:COLL:ADAP:PERM**

### **SCPI Command**

SENSe<Ch>:CORRection:COLLect:ADAPter:PERMittivity <numeric>

SENSe<Ch>:CORRection:COLLect:ADAPter:PERMittivity?

### **Description**

Sets or reads out the value of the permittivity of an adapter media in the adapter removal/insertion function.

When setting the adapter length, this parameter is used to calculate the adapter delay; therefore, this parameter must be set before setting the adapter length. This parameter is not used when setting the adapter delay.

command/query

### **Target**

Channel <Ch>,

<Ch>={1|2|...16}

Parameter

<numeric> the value of the permittivity of an adapter

### **Query Response**

<numeric>

### **Preset Value**

1.000649 (air)

### **Equivalent Softkeys**

**Calibration > Calibrate > Adapter Removal / Insertion > Permittivity**

---

## Equivalent COM Command

None

---

Back to [SENSe](#)

## SENS:CORR:COLL:ADAP:WAV:CUT

### SCPI Command

SENSe<Ch>:CORRection:COLLect:ADAPter:WAVEguide:CUToff <numeric>

SENSe<Ch>:CORRection:COLLect:ADAPter:WAVEguide:CUToff?

### Description

Sets or reads out the value of the cutoff frequency of the waveguide adapter.

command/query

### Target

Channel <Ch>,

<Ch>={{1}|2|...16}

### Parameter

<numeric> the value of the cutoff frequency of the waveguide adapter.

### Query Response

<numeric>

### Preset Value

1.0 GHz

### Equivalent Softkeys

Calibration > Calibrate > Adapter Removal / Insertion> Cutoff Frequency

---

### Equivalent COM Command

None

---

Back to [SENSe](#)

## SENS:CORR:COLL:METH:ADAP:REM

### SCPI Command

SENSe<Ch>:CORRection:COLLect:METHod:ADAPter:REMOval <port>

### Description

Selects the port number and sets the adapter removal/insertion function for the calculation of the calibration coefficients when the [SENS:CORR:COLL:SAVE](#) command has been executed.

no query

### Target

Channel <Ch>,

<Ch>={[1]|2|...16}

### Parameter

<port> the number of the port from 1 to 2

### Query Response

<numeric>

### Related Commands

[SENS:CORR:COLL:SAVE](#)

### Equivalent Softkeys

Calibration > Calibrate > Adapter Removal / Insertion> Select Port

---

### Equivalent COM Command

None

---

Back to [SENSe](#)

## SENS:CORR:COLL:CKIT

### SCPI Command

SENSe:CORRection:COLLect:CKIT[:SElect] <numeric>

SENSe:CORRection:COLLect:CKIT[:SElect]?

### Description

Sets or reads out the number of the selected calibration kit in the table of calibration kits. The selected calibration kit is used in the subsequent calibration and is used for editing by the commands [SENS:CORR:COLL:CKIT:XXXX](#).

command/query

### Parameter

<numeric> the number of the calibration kit from 1 to 64

### Query Response

<numeric>

### Preset Value

1

### Equivalent Softkeys

Calibration > Cal Kit > Cal Kit n > Select

---

### Equivalent COM Command

None

---

Back to [SENSe](#)



## SENS:CORR:COLL:CKIT:DESC

### SCPI Command

SENSe:CORRection:COLLect:CKIT:DESCription <string>

SENSe:CORRection:COLLect:CKIT:DESCription?

### Description

Sets or reads out the calibration kit description string.

command/query

### Target

Selected calibration kit

### Parameter

<string>, up to 254 characters

### Query Response

<string>

### Equivalent Softkeys

Calibration > Cal Kit > Cal Kit n > Description

---

### Equivalent COM Command

None

---

Back to [SENSe](#)

## **SENS:CORR:COLL:CKIT:LAB**

### **SCPI Command**

SENSe:CORRection:COLLect:CKIT:LABel <string>

SENSe:CORRection:COLLect:CKIT:LABel?

### **Description**

Sets or reads out the calibration kit label.

command/query

### **Target**

Selected calibration kit

### **Parameter**

<string>, up to 254 characters

### **Query Response**

<string>

### **Equivalent Softkeys**

**Calibration > Cal Kit > Cal Kit n > Label**

---

### **Equivalent COM Command**

SCPI.SENSe(Ch).CORRection.COLLect.CKIT.LABel

### **Syntax**

Lab = app.SCPI.SENSe(Ch).CORRection.COLLect.CKIT.LABel

app.SCPI.SENSe(Ch).CORRection.COLLect.CKIT.LABel = "User1"

## Type

String (read/write)

---

Back to [SENSe](#)

## **SENS:CORR:COLL:CKIT:ORD:LOAD**

### **SCPI Command**

SENSe:CORRection:COLLect:CKIT:ORDer:LOAD <port>,<numeric>

SENSe:CORRection:COLLect:CKIT:ORDer:LOAD? <port>

### **Description**

Sets or reads out the number of the calibration standard in the calibration kit that assigned to the LOAD class for measurement of the specified port.

command/query

### **Target**

Selected calibration kit

### **Parameter**

<b>&lt;port&gt;</b>	The number of the port from 1 to 2
<b>&lt;numeric&gt;</b>	The number of the calibration standard

### **Out of Range**

If the specified standard number is greater than the number of standards in the kit, an error occurs. If the specified standard number is not the load standard number, an error occurs.

### **Query Response**

<numeric>

### **Equivalent Softkeys**

**Calibration > Cal Kit > Specify CLSs > Load Port x (Row)**

---

### Equivalent COM Command

SCPI.SENSE(Ch).CORRection.COLLection.CKIT.ORDer.LOAD(Pt)

### Syntax

Num = app.SCPI.SENSE(Ch).CORRection.COLLection.CKIT.ORDer.LOAD(Pt)

app.SCPI.SENSE(Ch).CORRection.COLLection.CKIT.ORDer.LOAD(Pt) = 1

### Type

Long (read/write)

---

Back to [SENSe](#)

## **SENS:CORR:COLL:CKIT:ORD:OPEN**

### **SCPI Command**

SENSe:CORRection:COLLect:CKIT:ORDer:OPEN <port>,<numeric>

SENSe:CORRection:COLLect:CKIT:ORDer:OPEN? <port>

### **Description**

Sets or reads out the number of the calibration standard in the calibration kit that assigned to the OPEN class for measurement of the specified port.

command/query

### **Target**

Selected calibration kit

### **Parameter**

- <port>            The number of the port from 1 to 2
- <numeric>        The number of the calibration standard

### **Out of Range**

If the specified standard number is greater than the number of standards in the kit, an error occurs. If the specified standard number is not the open standard number, an error occurs.

### **Query Response**

<numeric>

### **Equivalent Softkeys**

**Calibration > Cal Kit > Specify CLSs > Open Port x (Row)**

---

### Equivalent COM Command

SCPI.SENSE(Ch).CORRection.COLLection.CKIT.ORDER.OPEN(Pt)

### Syntax

Num = app.SCPI.SENSE(Ch).CORRection.COLLection.CKIT.ORDER.OPEN(Pt)

app.SCPI.SENSE(Ch).CORRection.COLLection.CKIT.ORDER.OPEN(Pt) = 1

### Type

Long (read/write)

---

Back to [SENSe](#)

## **SENS:CORR:COLL:CKIT:ORD:SEL**

### **SCPI Command**

SENSe:CORRection:COLLect:CKIT:ORDer:SElect <numeric>

SENSe:CORRection:COLLect:CKIT:ORDer:SElect?

### **Description**

The subclass used to specify classes of calibration standards by the commands:

[SENS:CORR:COLL:CKIT:ORD:LOAD](#)

[SENS:CORR:COLL:CKIT:ORD:OPEN](#)

[SENS:CORR:COLL:CKIT:ORD:SHOR](#)

[SENS:CORR:COLL:CKIT:ORD:THRU](#)

[SENS:CORR:COLL:CKIT:ORD:TRLL](#)

[SENS:CORR:COLL:CKIT:ORD:TRLT](#)

[SENS:CORR:COLL:CKIT:ORD:TRLR](#)

command/query

### **Target**

Selected calibration kit

### **Parameter**

<numeric> the subclass number from 1 to 8

### **Query Response**

<numeric>

### **Equivalent Softkeys**

**Calibration > Cal Kit > Specify CLSs > Subclass n (Column)**



---

### Equivalent COM Command

SCPI.SENSE(Ch).CORRection.COLLection.CKIT.ORDER.SELect

### Syntax

Num = app.SCPI.SENSE(Ch).CORRection.COLLection.CKIT.ORDER.SELect

app.SCPI.SENSE(Ch).CORRection.COLLection.CKIT.ORDER.SELect = 1

### Type

Long (read/write)

---

Back to [SENSe](#)

## **SENS:CORR:COLL:CKIT:ORD:SHOR**

### **SCPI Command**

SENSe:CORRection:COLLect:CKIT:ORDer:SHORt <port>,<numeric>

SENSe:CORRection:COLLect:CKIT:ORDer:SHORt? <port>

### **Description**

Sets or reads out the number of the calibration standard in the calibration kit that assigned to the SHORT class for measurement of the specified port.

command/query

### **Target**

Selected calibration kit

### **Parameter**

<port>	The number of the port from 1 to 2
<numeric>	The number of the calibration standard

### **Out of Range**

If the specified standard number is greater than the number of standards in the kit, an error occurs. If the specified standard number is not the short standard number, an error occurs.

### **Query Response**

<numeric>

### **Equivalent Softkeys**

**Calibration > Cal Kit > Specify CLSs > Short Port x (Row)**

---

### Equivalent COM Command

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

### Syntax

Num = app.SCPI.SENSE(Ch).CORRection.COLLection.CKIT.ORDer.SHORt (Pt)

app.SCPI.SENSE(Ch).CORRection.COLLection.CKIT.ORDer.SHORt (Pt) = 1

### Type

Long (read/write)

---

Back to [SENSe](#)

## **SENS:CORR:COLL:CKIT:ORD:THRU**

### **SCPI Command**

SENSe:CORRection:COLLect:CKIT:ORDer:THRU <port1>,<port2>,<numeric>

SENSe:CORRection:COLLect:CKIT:ORDer:THRU? <port1>,<port2>

### **Description**

Sets or reads out the number of the calibration standard in the calibration kit that assigned to the THRU class for the measurement between the <port1> and <port2> ports.

command/query

### **Target**

Selected calibration kit

### **Parameter**

- |                        |   |
|------------------------|---|
| <b>&lt;port1&gt;</b>   | The number of the receiver port from 1 to 2 |
| <b>&lt;port2&gt;</b>   | The number of the source port from 1 to 2   |
| <b>&lt;numeric&gt;</b> | The number of the calibration standard      |

### **Out of Range**

If the specified standard number is greater than the number of standards in the kit, an error occurs. If the specified standard number is not the thru standard number, an error occurs.

### **Query Response**

<numeric>

### **Equivalent Softkeys**

**Calibration > Cal Kit > Specify CLSs > Thru Port x-y (Row)**

---

### Equivalent COM Command

SCPI.SENSE(Ch).CORRection.COLLection.CKIT.ORDER.THRU(Pt\_m, Pt\_n)

### Syntax

Num = app.SCPI.SENSE(Ch).CORRection.COLLection.CKIT.ORDER.THRU (1, 2)

app.SCPI.SENSE(Ch).CORRection.COLLection.CKIT.ORDER.THRU (1, 2) = 1

### Type

Long (read/write)

---

Back to [SENSe](#)

## **SENS:CORR:COLL:CKIT:ORD:TRLL**

### **SCPI Command**

SENSe:CORRection:COLLect:CKIT:ORDer:TRLLine <port1>,<port2>,<numeric>

SENSe:CORRection:COLLect:CKIT:ORDer:TRLLine? <port1>,<port2>

### **Description**

Sets or reads out the number of the calibration standard in the calibration kit that assigned to the TRL LINE class for the measurement between the <port1> and <port2> ports.

command/query

### **Target**

Selected calibration kit

### **Parameter**

<b>&lt;port1&gt;</b>	The number of the receiver port from 1 to 2
<b>&lt;port2&gt;</b>	The number of the source port from 1 to 2
<b>&lt;numeric&gt;</b>	The number of the calibration standard

### **Out of Range**

If the specified standard number is greater than the number of standards in the kit, an error occurs. If the specified standard number is not the thru standard number, an error occurs.

### **Query Response**

<numeric>

### **Equivalent Softkeys**

**Calibration > Cal Kit > Specify CLSs > TRL Line Port x-y (Row)**

---

### Equivalent COM Command

SCPI.SENSE(Ch).CORRection.COLLect.CKIT.ORDer.TRLLine(Pt\_m, Pt\_n)

### Syntax

Num = app.SCPI.SENSE(Ch).CORRection.COLLect.CKIT.ORDer.TRLLine(1, 2)

app.SCPI.SENSE(Ch).CORRection.COLLect.CKIT.ORDer.TRLLine(1, 2) = 1

### Type

Long (read/write)

---

Back to [SENSe](#)

## **SENS:CORR:COLL:CKIT:ORD:TRLT**

### **SCPI Command**

SENSe:CORRection:COLLect:CKIT:ORDer:TRLThru <port1>,<port2>,<numeric>

SENSe:CORRection:COLLect:CKIT:ORDer:TRLThru? <port1>,<port2>

### **Description**

Sets or reads out the number of the calibration standard in the calibration kit that assigned to the TRL THRU class for the measurement between the <port1> and <port2> ports.

command/query

### **Target**

Selected calibration kit

### **Parameter**

- |                        |   |
|------------------------|---|
| <b>&lt;port1&gt;</b>   | The number of the receiver port from 1 to 2 |
| <b>&lt;port2&gt;</b>   | The number of the source port from 1 to 2   |
| <b>&lt;numeric&gt;</b> | The number of the calibration standard      |

### **Out of Range**

If the specified standard number is greater than the number of standards in the kit, an error occurs. If the specified standard number is not the thru standard number, an error occurs.

### **Query Response**

<numeric>

### **Equivalent Softkeys**

**Calibration > Cal Kit > Specify CLSs > TRL Thru Port x-y (Row)**



---

### Equivalent COM Command

SCPI.SENSE(Ch).CORRection.COLLection.CKIT.ORDER.TRLThru(Pt\_m, Pt\_n)

### Syntax

Num = app.SCPI.SENSE(Ch).CORRection.COLLection.CKIT.ORDER.TRLThru(1, 2)

app.SCPI.SENSE(Ch).CORRection.COLLection.CKIT.ORDER.TRLThru(1, 2) = 1

### Type

Long (read/write)

---

Back to [SENSe](#)

## **SENS:CORR:COLL:CKIT:ORD:TRLR**

### **SCPI Command**

SENSe:CORRection:COLLect:CKIT:ORDer:TRLReflect <port>,<numeric>

SENSe:CORRection:COLLect:CKIT:ORDer:TRLReflect? <port>

### **Description**

Sets or reads out the number of the calibration standard in the calibration kit that assigned to the TRL REFLECT class for the measurement of the specified port.

command/query

### **Target**

Selected calibration kit

### **Parameter**

<port1>      The number of the port from 1 to 2

<numeric>    The number of the calibration standard

### **Out of Range**

If the specified standard number is greater than the number of standards in the kit, an error occurs. If the specified standard number is not the open or short standard number, an error occurs.

### **Query Response**

<numeric>

### **Equivalent Softkeys**

**Calibration > Cal Kit > Specify CLSs > TRL Reflect Port x-y (Row)**

---

### Equivalent COM Command

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

### Syntax

Num = app.SCPI.SENSE(Ch).CORRection.COLLection.CKIT.ORDER.TRLReflect(Pt)

app.SCPI.SENSE(Ch).CORRection.COLLection.CKIT.ORDER.TRLReflect(Pt) = 1

### Type

Long (read/write)

---

Back to [SENSe](#)

## SENS:CORR:COLL:CKIT:RES

### SCPI Command

SENSe:CORRection:COLLect:CKIT:RESet

### Description

Resets the calibration kit to the factory settings. Restores the predefined calibration kit. Removes the user defined calibration kit.

no query

### Target

Selected calibration kit

### Equivalent Softkeys

Calibration > Cal Kit > Restore Cal Kit

---

### Equivalent COM Command

SCPI.SENSe(Ch).CORRection.COLLect.CKIT.RESet

### Syntax

app.SCPI.SENSe(Ch).CORRection.COLLect.CKIT.RESet

### Type

Method

---

Back to [SENSe](#)

## **SENS:CORR:COLL:CKIT:STAN:ARB**

### **SCPI Command**

SENSe:CORRection:COLLect:CKIT:STAN<Std>:ARBitrary <numeric>

SENSe:CORRection:COLLect:CKIT:STAN<Std>:ARBitrary?

### **Description**

Sets or reads out the value of the arbitrary impedance for the load standard.

command/query

### **Target**

Standard <Std> of the calibration kit,

<Std>={[1]|2|...N}, where N — the number of the standards in the calibration kit

### **Parameter**

<numeric> the arbitrary impedance value from  $-1\text{E}18$  to  $1\text{E}18$

### **Unit**

$\Omega$  (Ohm)

### **Out of Range**

Sets the value of the limit, which is closer to the specified value.

### **Query Response**

<numeric>

### **Preset Value**

50 or 75  $\Omega$ , depending on the selected calibration kit

## Equivalent Softkeys

Calibration > Cal Kit > Define STDs > Terminal Impedance

---

### Equivalent COM Command

SCPI.SENSE(Ch).CORRection.COLLection.CKIT.STAN(Std).ARBitrary

### Syntax

Value = app.SCPI.SENSE(Ch).CORRection.COLLection.CKIT.STAN(Std).ARBitrary

app.SCPI.SENSE(Ch).CORRection.COLLection.CKIT.STAN(Std).ARBitrary = 50

### Type

Double (read/write)

---

Back to [SENSe](#)

## **SENS:CORR:COLL:CKIT:STAN:C0**

### **SCPI Command**

SENSe:CORRection:COLLect:CKIT:STAN<Std>:C0 <numeric>

SENSe:CORRection:COLLect:CKIT:STAN<Std>:C0?

### **Description**

Sets or reads out the C0 value for the open calibration standard.

command/query

### **Target**

Standard <Std> of the calibration kit,

<Std>={[1]|2|...N}, where N — the number of the standards in the calibration kit

### **Parameter**

<numeric> the arbitrary impedance value from  $-1\text{E}18$  to  $1\text{E}18$

### **Unit**

$1\text{E}-15$  F (Farad)

### **Out of Range**

Sets the value of the limit, which is closer to the specified value.

### **Query Response**

<numeric>

### **Equivalent Softkeys**

**Calibration > Cal Kit > Define STDs -> Open n -> C0 10–15 F**

---

### Equivalent COM Command

SCPI.SENSE(Ch).CORRection.COLLection.CKIT.STAN(Std).C0

### Syntax

Value = app.SCPI.SENSE(Ch).CORRection.COLLection.CKIT.STAN(Std).C0

app.SCPI.SENSE(Ch).CORRection.COLLection.CKIT.STAN(Std).C1 = 100

### Type

Double (read/write)

---

Back to [SENSe](#)



## **SENS:CORR:COLL:CKIT:STAN:C1**

### **SCPI Command**

SENSe:CORRection:COLLect:CKIT:STAN<Std>:C1 <numeric>

SENSe:CORRection:COLLect:CKIT:STAN<Std>:C1?

### **Description**

Sets or reads out the C1 value for the open calibration standard.

command/query

### **Target**

Standard <Std> of the calibration kit,

<Std>={[1]|2|...N}, where N — the number of the standards in the calibration kit

### **Parameter**

<numeric> the arbitrary impedance value from  $-1\text{E}18$  to  $1\text{E}18$

### **Unit**

$1\text{E}-27$  F/Hz (Farad/Hertz)

### **Out of Range**

Sets the value of the limit, which is closer to the specified value.

### **Query Response**

<numeric>

### **Equivalent Softkeys**

**Calibration > Cal Kit > Define STDs -> Open n -> C1  $10^{-27}$  F/Hz**

---

### Equivalent COM Command

SCPI.SENSE(Ch).CORRection.COLLection.CKIT.STAN(Std).C1

### Syntax

Value = app.SCPI.SENSE(Ch).CORRection.COLLection.CKIT.STAN(Std).C1

app.SCPI.SENSE(Ch).CORRection.COLLection.CKIT.STAN(Std).C0 = 100

### Type

Double (read/write)

---

Back to [SENSe](#)

## **SENS:CORR:COLL:CKIT:STAN:C2**

### **SCPI Command**

SENSe:CORRection:COLLect:CKIT:STAN<Std>:C2 <numeric>

SENSe:CORRection:COLLect:CKIT:STAN<Std>:C2?

### **Description**

Sets or reads out the C2 value for the open calibration standard.

command/query

### **Target**

Standard <Std> of the calibration kit,

<Std>={[1]|2|...N}, where N — the number of the standards in the calibration kit

### **Parameter**

<numeric> the arbitrary impedance value from  $-1\text{E}18$  to  $1\text{E}18$

### **Unit**

$1\text{E}-36 \text{ F/Hz}^2$  (Farad/Hertz<sup>2</sup>)

### **Out of Range**

Sets the value of the limit, which is closer to the specified value.

### **Query Response**

<numeric>

### **Equivalent Softkeys**

**Calibration > Cal Kit > Define STDs -> Open n -> C2  $10^{-36} \text{ F/Hz}^2$**

---

### Equivalent COM Command

SCPI.SENSE(Ch).CORRection.COLLection.CKIT.STAN(Std).C2

### Syntax

Value = app.SCPI.SENSE(Ch).CORRection.COLLection.CKIT.STAN(Std).C2

app.SCPI.SENSE(Ch).CORRection.COLLection.CKIT.STAN(Std).C2 = 100

### Type

Double (read/write)

---

Back to [SENSe](#)

## **SENS:CORR:COLL:CKIT:STAN:C3**

### **SCPI Command**

SENSe:CORRection:COLLect:CKIT:STAN<Std>:C3 <numeric>

SENSe:CORRection:COLLect:CKIT:STAN<Std>:C3?

### **Description**

Sets or reads out the C3 value for the open calibration standard.

command/query

### **Target**

Standard <Std> of the calibration kit,

<Std>={[1]|2|...N}, where N — the number of the standards in the calibration kit

### **Parameter**

<numeric> the arbitrary impedance value from  $-1\text{E}18$  to  $1\text{E}18$

### **Unit**

$1\text{E}-45\text{ F/Hz}^3$  (Farad/Hertz<sup>3</sup>)

### **Out of Range**

Sets the value of the limit, which is closer to the specified value.

### **Query Response**

<numeric>

### **Equivalent Softkeys**

**Calibration > Cal Kit > Define STDs -> Open n -> C3  $10^{-45}\text{ F/Hz}^3$**

---

### Equivalent COM Command

SCPI.SENSE(Ch).CORRection.COLLection.CKIT.STAN(Std).C3

### Syntax

Value = app.SCPI.SENSE(Ch).CORRection.COLLection.CKIT.STAN(Std).C3

app.SCPI.SENSE(Ch).CORRection.COLLection.CKIT.STAN(Std).C3 = 100

### Type

Double (read/write)

---

Back to [SENSe](#)

## **SENS:CORR:COLL:CKIT:STAN:COUN?**

### **SCPI Command**

SENSe:CORRection:COLLect:CKIT:STANdard:COUNt?

### **Description**

Reads out the count of standards in the selected calibration kit.

query only

### **Target**

### **Query Response**

<number>

### **Equivalent Softkeys**

**Calibration > Cal Kit > Define STDs**

---

## Equivalent COM Command

none

---

Back to [SENSe](#)



## SENS:CORR:COLL:CKIT:STAN:DATA

### SCPI Command

SENSe:CORRection:COLLect:CKIT:STAN<Std>:DATA <numeric list>

SENSe:CORRection:COLLect:CKIT:STAN<Std>:DATA?

### Description

Writes or reads out the data array of the data-based calibration standard. The first element of the array is 1 or 2 and determines the number of ports of the calibration standard. The array format is as follows.

When the first element of the array is 1 :

```
<1>,<freq1>,<S11.re1>,<S11.im1>,  
<freq2>,<S11.re2>,<S11.im2>,  
...  
<freqN>,<S11.reN>,<S11.imN>
```

When the first element of the array is 2:

```
<2>,<freq1>,<S11.re1>,<S11.im1>,<S21.re1>,<S21.im1>,  
<S12.re1>,<S12.im1>,<S22.re1>,<S22.im1>,  
...  
<freqN>,<S11.reN>,<S11.imN>,<S21.reN>,<S21.imN>,  
<S12.reN>,<S12.imN>,<S22.reN>,<S22.imN>
```

command/query

### Target

Standard <Std> of the selected calibration kit,

<Std>={[1]|2|...N}, where N — the number of the standards in the calibration kit

### Query Response

<numeric 1>, <numeric 2>, ...<numeric N>

## Equivalent Softkeys

Calibration > Cal Kit > Define STDs > Define STD Data

---

## Equivalent COM Command

None

---

Back to [SENSe](#)

## **SENS:CORR:COLL:CKIT:STAN:DEL**

### **SCPI Command**

SENSe:CORRection:COLLect:CKIT:STAN<Std>:DELay <numeric>

SENSe:CORRection:COLLect:CKIT:STAN<Std>:Delay?

### **Description**

Sets or reads out the offset delay value for the calibration standard.

command/query

### **Target**

Standard <Std> of the calibration kit,

<Std>={[1]|2|...N}, where N — the number of the standards in the calibration kit

### **Parameter**

<numeric> the offset delay value form –1E18 to 1E18

### **Unit**

s (second)

### **Out of Range**

Sets the value of the limit, which is closer to the specified value.

### **Query Response**

<numeric>

### **Equivalent Softkeys**

**Calibration > Cal Kit > Define STDs > Offset Delay**

---

### Equivalent COM Command

SCPI.SENSE(Ch).CORRection.COLLection.CKIT.STAN(Std).DELay

### Syntax

Value = app.SCPI.SENSE(Ch).CORRection.COLLection.CKIT.STAN(Std).DELay

app.SCPI.SENSE(Ch).CORRection.COLLection.CKIT.STAN(Std).DELay = 93E-12

### Type

Double (read/write)

---

Back to [SENSe](#)

## **SENS:CORR:COLL:CKIT:STAN:FMAX**

### **SCPI Command**

SENSe:CORRection:COLLect:CKIT:STAN<Std>:FMAXimum <numeric>

SENSe:CORRection:COLLect:CKIT:STAN<Std>:FMAXimum?

### **Description**

Sets or reads out the maximum frequency limit of the calibration standard.

command/query

### **Target**

Standard <Std> of the calibration kit,

<Std>={[1]|2|...N}, where N — the number of the standards in the calibration kit

### **Parameter**

<numeric> the maximum frequency limit form 0 to 1E14

### **Unit**

Hz (Hertz)

### **Out of Range**

Sets the value of the limit, which is closer to the specified value.

### **Query Response**

<numeric>

### **Equivalent Softkeys**

**Calibration > Cal Kit > Define STDs -> F max**

---

### Equivalent COM Command

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

### Syntax

Value = app.SCPI.SENSE(Ch).CORRection.COLLection.CKIT.STAN(Std).FMAXimum

app.SCPI.SENSE(Ch).CORRection.COLLection.CKIT.STAN(Std).FMAXimum = 3E9

### Type

Double (read/write)

---

Back to [SENSe](#)

## **SENS:CORR:COLL:CKIT:STAN:FMIN**

### **SCPI Command**

SENSe:CORRection:COLLect:CKIT:STAN<Std>:FMINimum <numeric>

SENSe:CORRection:COLLect:CKIT:STAN<Std>:FMINimum?

### **Description**

Sets or reads out the minimum frequency limit of the calibration standard.

command/query

### **Target**

Standard <Std> of the calibration kit,

<Std>={[1]|2|...N}, where N — the number of the standards in the calibration kit

### **Parameter**

<numeric> the minimum frequency limit form 0 to 1E14

### **Unit**

Hz (Hertz)

### **Out of Range**

Sets the value of the limit, which is closer to the specified value.

### **Query Response**

<numeric>

### **Equivalent Softkeys**

**Calibration > Cal Kit > Define STDs > F min**

---

### Equivalent COM Command

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

### Syntax

Value = app.SCPI.SENSE(Ch).CORRection.COLLection.CKIT.STAN(Std).FMINimum

app.SCPI.SENSE(Ch).CORRection.COLLection.CKIT.STAN(Std).FMINimum = 3E9

### Type

Double (read/write)

---

Back to [SENSe](#)



## SENS:CORR:COLL:CKIT:STAN:INS

### SCPI Command

SENSe:CORRection:COLLect:CKIT:STAN<Std>:INSert

### Description

Inserts the calibration standard into the selected calibration kit. The existing standards with indices greater than or equal to <std> are shifted by +1.

no query

### Target

Standard <Std> of the selected calibration kit,

<Std>={[1]2|...N}, where N — the number of the standards in the calibration kit

### Equivalent Softkeys

Calibration > Cal Kit > Define STDs > Add STD

---

### Equivalent COM Command

None

---

Back to [SENSe](#)

## **SENS:CORR:COLL:CKIT:STAN:L0**

### **SCPI Command**

SENSe:CORRection:COLLect:CKIT:STAN<Std>:L0 <numeric>

SENSe:CORRection:COLLect:CKIT:STAN<Std>:L0?

### **Description**

Sets or reads out the L0 value for the short calibration standard.

command/query

### **Target**

Standard <Std> of the calibration kit,

<Std>={[1]|2|...N}, where N — the number of the standards in the calibration kit

### **Parameter**

<numeric> the L0 value from  $-1\text{E}18$  to  $1\text{E}18$

### **Unit**

$1\text{E}-12$  H (Henry)

### **Out of Range**

Sets the value of the limit, which is closer to the specified value.

### **Query Response**

<numeric>

### **Equivalent Softkeys**

**Calibration > Cal Kit > Define STDs -> Short -> L0  $10^{-12}$  H**

---

### Equivalent COM Command

SCPI.SENSE(Ch).CORRection.COLLection.CKIT.STAN(Std).L0

### Syntax

Value = app.SCPI.SENSE(Ch).CORRection.COLLection.CKIT.STAN(Std).L0

app.SCPI.SENSE(Ch).CORRection.COLLection.CKIT.STAN(Std).L0 = 100

### Type

Double (read/write)

---

Back to [SENSe](#)

## **SENS:CORR:COLL:CKIT:STAN:L1**

### **SCPI Command**

SENSe:CORRection:COLLect:CKIT:STAN<Std>:L1 <numeric>

SENSe:CORRection:COLLect:CKIT:STAN<Std>:L1?

### **Description**

Sets or reads out the L1 value for the short calibration standard.

command/query

### **Target**

Standard <Std> of the calibration kit,

<Std>={[1][2]...N}, where N — the number of the standards in the calibration kit

### **Parameter**

<numeric> the L0 value from  $-1\text{E}18$  to  $1\text{E}18$

### **Unit**

$1\text{E}-24$  H/Hz (Henry/Hertz)

### **Out of Range**

Sets the value of the limit, which is closer to the specified value.

### **Query Response**

<numeric>

### **Equivalent Softkeys**

**Calibration > Cal Kit > Define STDs -> Short -> L1  $10^{-24}$  H/Hz**

---

### Equivalent COM Command

SCPI.SENSE(Ch).CORRection.COLLect.CKIT.STAN(Std).L1

### Syntax

Value = app.SCPI.SENSE(Ch).CORRection.COLLect.CKIT.STAN(Std).L1

app.SCPI.SENSE(Ch).CORRection.COLLect.CKIT.STAN(Std).L1 = 100

### Type

Double (read/write)

---

Back to [SENSe](#)

## **SENS:CORR:COLL:CKIT:STAN:L2**

### **SCPI Command**

SENSe:CORRection:COLLect:CKIT:STAN<Std>:L2 <numeric>

SENSe:CORRection:COLLect:CKIT:STAN<Std>:L2?

### **Description**

Sets or reads out the L2 value for the short calibration standard.

command/query

### **Target**

Standard <Std> of the calibration kit,

<Std>={[1]|2|...N}, where N — the number of the standards in the calibration kit

### **Parameter**

<numeric> the L2 value from  $-1\text{E}18$  to  $1\text{E}18$

### **Unit**

$1\text{E}-33\text{ H/Hz}^2$  (Henry/Hertz<sup>2</sup>)

### **Out of Range**

Sets the value of the limit, which is closer to the specified value.

### **Query Response**

<numeric>

### **Equivalent Softkeys**

**Calibration > Cal Kit > Define STDs -> Short -> L2  $10-33\text{ H/Hz}^2$**

---

### Equivalent COM Command

SCPI.SENSE(Ch).CORRection.COLLect.CKIT.STAN(Std).L2

### Syntax

Value = app.SCPI.SENSE(Ch).CORRection.COLLect.CKIT.STAN(Std).L2

app.SCPI.SENSE(Ch).CORRection.COLLect.CKIT.STAN(Std).L2 = 100

### Type

Double (read/write)

---

Back to [SENSe](#)

## **SENS:CORR:COLL:CKIT:STAN:L3**

### **SCPI Command**

SENSe:CORRection:COLLect:CKIT:STAN<Std>:L3 <numeric>

SENSe:CORRection:COLLect:CKIT:STAN<Std>:L3?

### **Description**

Sets or reads out the L3 value for the short calibration standard.

command/query

### **Target**

Standard <Std> of the calibration kit,

<Std>={[1]|2|...N}, where N — the number of the standards in the calibration kit

### **Parameter**

<numeric> the L3 value from  $-1\text{E}18$  to  $1\text{E}18$

### **Unit**

$1\text{E}-42$  H/Hz<sup>3</sup> (Henry/Hertz<sup>3</sup>)

### **Out of Range**

Sets the value of the limit, which is closer to the specified value.

### **Query Response**

<numeric>

### **Equivalent Softkeys**

**Calibration > Cal Kit > Define STDs -> Short -> L3 10–42 H/Hz<sup>3</sup>**



---

### Equivalent COM Command

SCPI.SENSE(Ch).CORRection.COLLect.CKIT.STAN(Std).L3

### Syntax

Value = app.SCPI.SENSE(Ch).CORRection.COLLect.CKIT.STAN(Std).L3

app.SCPI.SENSE(Ch).CORRection.COLLect.CKIT.STAN(Std).L3 = 100

### Type

Double (read/write)

---

Back to [SENSe](#)

## **SENS:CORR:COLL:CKIT:STAN:LAB**

### **SCPI Command**

SENSe:CORRection:COLLect:CKIT:STAN<Std>:LABel <string>

SENSe:CORRection:COLLect:CKIT:STAN<Std>:LABel?

### **Description**

Sets or reads out the label for the calibration standard.

command/query

### **Target**

Standard <Std> of the calibration kit,

<Std>={[1][2]...N}, where N — the number of the standards in the calibration kit

### **Parameter**

<string>, up to 254 characters

### **Query Response**

<string>

### **Equivalent Softkeys**

**Calibration > Cal Kit > Define STDs -> {Open | Short | Load | Thru/Delay} -> Label**

---

### **Equivalent COM Command**

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

## Syntax

Lab = app.SCPI.SENSE(Ch).CORRection.COLLEct.CKIT.STAN(Std).LABel

app.SCPI.SENSE(Ch).CORRection.COLLEct.CKIT.STAN(Std).LABel = "Open"

## Type

String (read/write)

---

Back to [SENSe](#)

## **SENS:CORR:COLL:CKIT:STAN:LOSS**

### **SCPI Command**

SENSe:CORRection:COLLect:CKIT:STAN<Std>:LOSS <numeric>

SENSe:CORRection:COLLect:CKIT:STAN<Std>:LOSS?

### **Description**

Sets or reads out the offset loss value for the calibration standard.

command/query

### **Target**

Standard <Std> of the calibration kit,

<Std>={[1][2]...N}, where N — the number of the standards in the calibration kit

### **Parameter**

<numeric> the offset loss value from  $-1\text{E}18$  to  $1\text{E}18$

### **Unit**

$\Omega/\text{s}$  (Ohm/second)

### **Out of Range**

Sets the value of the limit, which is closer to the specified value.

### **Query Response**

<numeric>

## **Equivalent Softkeys**

**Calibration > Cal Kit > Define STDs -> {Open | Short | Load | Thru/Delay} -> Offset -> Loss**

---

## **Equivalent COM Command**

SCPI.SENSE(Ch).CORRection.COLLection.CKIT.STAN(Std).LABel

## Syntax

Lab = app.SCPI.SENSE(Ch).CORRection.COLLEct.CKIT.STAN(Std).LABel

app.SCPI.SENSE(Ch).CORRection.COLLEct.CKIT.STAN(Std).LABel = "Open"

## Type

String (read/write)

---

Back to [SENSe](#)

## **SENS:CORR:COLL:CKIT:STAN:REM**

### **SCPI Command**

SENSe:CORRection:COLLect:CKIT:STAN<Std>:REMove

### **Description**

Deletes the calibration standard into the selected calibration kit. The existing standards with indices greater than the <std> are shifted by –1.

no query

### **Target**

Standard <Std> of the selected calibration kit,

<Std>={[1]2|...N}, where N — the number of the standards in the calibration kit

### **Equivalent Softkeys**

**Calibration > Cal Kit > Define STDs > Delete STD**

---

### **Equivalent COM Command**

None

---

Back to [SENSe](#)

## SENS:CORR:COLL:CKIT:STAN:TYPE

### SCPI Command

SENSe:CORRection:COLLect:CKIT:STAN<Std>:TYPE <char>

SENSe:CORRection:COLLect:CKIT:STAN<Std>:TYPE?

### Description

Sets or reads out the type of calibration standard.

command/query

### Target

Standard <Std> of the calibration kit,

<Std>={[1]|2|...N}, where N — the number of the standards in the calibration kit

### Parameter

<char> Specifies the type of calibration standard:

<b>OPEN</b>	Open
<b>SHORT</b>	Short
<b>LOAD</b>	Load
<b>THRU</b>	Thru
<b>UTHR</b>	Unknown Thru
<b>SLID</b>	Sliding Load
<b>DATA</b>	Data Based
<b>NONE</b>	Not defined



## Query Response

{OPEN|SHOR|LOAD|THRU|UTHR|SLID|DATA|NONE}

## Equivalent Softkeys

**Calibration > Cal Kit > Define STDs -> {Open | Short | Load | Thru/Delay} -> STD Type**

---

## Equivalent COM Command

SCPI.SENSE(Ch).CORRection.COLLect.CKIT.STAN(Std).TYPE

## Syntax

Param = app.SCPI.SENSE(Ch).CORRection.COLLect.CKIT.STAN(Std).TYPE

app.SCPI.SENSE(Ch).CORRection.COLLect.CKIT.STAN(Std).TYPE = "OPEN"

## Type

String (read/write)

---

Back to [SENSe](#)

## **SENS:CORR:COLL:CKIT:STAN:Z0**

### **SCPI Command**

SENSe:CORRection:COLLect:CKIT:STAN<Std>:Z0 <numeric>

SENSe:CORRection:COLLect:CKIT:STAN<Std>:Z0?

### **Description**

Sets or reads out the offset Z0 value for the calibration standard.

command/query

### **Target**

Standard <Std> of the calibration kit,

<Std>={[1]|2|...N}, where N — the number of the standards in the calibration kit

### **Parameter**

<numeric> the offset Z0 value from –1E18 to 1E18

### **Unit**

$\Omega$  (Ohm)

### **Out of Range**

Sets the value of the limit, which is closer to the specified value.

### **Query Response**

<numeric>

### **Preset Value**

50 or 75  $\Omega$ , depending on the selected calibration kit

## Equivalent Softkeys

Calibration > Cal Kit > Define STDs -> {Open | Short | Load | Thru/Delay} -> Offset -> Z0

---

## Equivalent COM Command

SCPI.SENSE(Ch).CORRection.COLLection.CKIT.STAN(Std).Z0

## Syntax

Value = app.SCPI.SENSE(Ch).CORRection.COLLection.CKIT.STAN(Std).Z0

app.SCPI.SENSE(Ch).CORRection.COLLection.CKIT.STAN(Std).Z0 = 50

## Type

Double (read/write)

---

Back to [SENSe](#)

## SENS:CORR:COLL:CLEAR

### SCPI Command

SENSe<Ch>:CORRection:COLLect:CLEAr

### Description

Clears the measurement data of the calibration standards.

no query

### Target

Channel <Ch>,

<Ch>={1|2|...16}

### Equivalent Softkeys

Calibration > Calibrate > {Response (Open) | Response (Short) | Response (Thru) | Full 1–Port Cal | One Path 2–Port Cal | Full 2–Port Cal | 2–Port TRL Cal} > Cancel > OK

---

### Equivalent COM Command

SCPI.SENSE(CH).CORRection.COLLect.CLEAr

### Syntax

app.SCPI.SENSE(CH).CORRection.COLLect.CLEAr

**Type**

Method

---

Back to [SENSe](#)

## SENS:CORR:COLL:DATA:ISOL

### SCPI Command

SENSe<Ch>:CORRection:COLLection:DATA:ISOLation <rcvport>,<srcport>,<numeric list>

SENSe<Ch>:CORRection:COLLection:DATA:ISOLation? <rcvport>,<srcport>

### Description

Writes or reads out the array of the isolation calibration measurement performed between the receiver port <rcvport> and the source port <srcport>.

The array size is 2N, where N is the number of measurement points.

For the n–th point, where n from 1 to N:

<numeric 2n–1> real part of the measurement

<numeric 2n> imaginary part of the measurement

command/query

### Target

Channel <Ch>,

<Ch>={{1}|2|...16}

### Parameter

<rcvport>                      The number of the receiver port from 1 to 2

<srcport>                      The number of the source port from 1 to 2

<numeric list>                The isolation measurement data array

### Query Response

<numeric 1>, <numeric 2>, ...<numeric 2N>

## Related Commands

[SENS:CORR:COLL:ISOL](#)

## Equivalent Softkeys

None

---

## Equivalent COM Command

SCPI.SENSE(Ch).CORRection.COLLection.DATA.ISOLation(Pt\_r, Pt\_s)

## Syntax

Data = app.SCPI.SENSE(Ch).CORRection.COLLection.DATA.ISOLation(Pt\_r, Pt\_s)

app.SCPI.SENSE(Ch).CORRection.COLLection.DATA.ISOLation(Pt\_r, Pt\_s) = Data

## Type

Variant (array of Double) (read/write)

---

Back to [SENSE](#)

## SENS:CORR:COLL:DATA:LOAD

### SCPI Command

SENSe<Ch>:CORRection:COLLect:DATA:LOAD <port>,<numeric list>

SENSe<Ch>:CORRection:COLLect:DATA:LOAD? <port>

### Description

Writes or reads out the array of the load calibration standard measurement for the port <port>.

The array size is 2N, where N is the number of measurement points.

For the n–th point, where n from 1 to N:

<numeric 2n–1> real part of the measurement

<numeric 2n> imaginary part of the measurement

command/query

### Target

Channel <Ch>,

<Ch>={[1]|2|...16}

### Parameter

<port>                      The number of the port from 1 to 2

<numeric list>            The data array of the load standard measurement

### Query Response

<numeric 1>, <numeric 2>, ...<numeric 2N>

### Related Commands

[SENS:CORR:COLL:LOAD](#)



## Equivalent Softkeys

None

---

## Equivalent COM Command

SCPI.SENSE(Ch).CORRection.COLLection.DATA.ISOLation(Pt\_r, Pt\_s)

## Syntax

Data = app.SCPI.SENSE(Ch).CORRection.COLLection.DATA.ISOLation(Pt\_r, Pt\_s)

app.SCPI.SENSE(Ch).CORRection.COLLection.DATA.ISOLation(Pt\_r, Pt\_s) = Data

## Type

Variant (array of Double) (read/write)

---

Back to [SENSe](#)

## SENS:CORR:COLL:DATA:OPEN

### SCPI Command

SENSe<Ch>:CORRection:COLLect:DATA:OPEN <port>,<numeric list>

SENSe<Ch>:CORRection:COLLect:DATA:OPEN? <port>

### Description

Writes or reads out the array of the open calibration standard measurement for the port <port>.

The array size is  $2N$ , where  $N$  is the number of measurement points.

For the  $n$ -th point, where  $n$  from 1 to  $N$ :

<numeric  $2n-1$ > real part of the measurement

<numeric  $2n$ > imaginary part of the measurement

command/query

### Target

Channel <Ch>,

<Ch>={[1]|2|...16}

### Parameter

<port>                      The number of the port from 1 to 4

<numeric list>            The data array of the open standard measurement

**Query Response**

<numeric 1>, <numeric 2>, ...<numeric 2N>

**Related Commands**

[SENS:CORR:COLL:OPEN](#)

### Equivalent Softkeys

None

---

### Equivalent COM Command

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

### Syntax

Data = app.SCPI.SENSE(Ch).CORRection.COLLect.DATA.OPEN(Pt)

app.SCPI.SENSE(Ch).CORRection.COLLect.DATA.OPEN(Pt) = Data

### Type

Variant (array of Double) (read/write)

---

Back to [SENSe](#)

## SENS:CORR:COLL:DATA:SHOR

### SCPI Command

SENSe<Ch>:CORRection:COLLect:DATA:SHORt <port>,<numeric list>

SENSe<Ch>:CORRection:COLLect:DATA:SHORt? <port>

### Description

Writes or reads out the array of the short calibration standard measurement for the port <port>.

The array size is  $2N$ , where  $N$  is the number of measurement points.

For the  $n$ -th point, where  $n$  from 1 to  $N$ :

<numeric  $2n-1$ > real part of the measurement

<numeric  $2n$ > imaginary part of the measurement

command/query

### Target

Channel <Ch>,

<Ch>={[1]|2|...16}

### Parameter

<port>                      The number of the port from 1 to 2

<numeric list>            The data array of the short standard measurement

### Query Response

<numeric 1>, <numeric 2>, ...<numeric  $2N$ >

### Related Commands

[SENS:CORR:COLL:SHOR](#)

## Equivalent Softkeys

None

---

## Equivalent COM Command

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

## Syntax

Data = app.SCPI.SENSE(Ch).CORRection.COLLect.DATA.SHORt(Pt)

app.SCPI.SENSE(Ch).CORRection.COLLect.DATA.SHORt(Pt) = Data

## Type

Variant (array of Double) (read/write)

---

Back to [SENSe](#)

# SENS:CORR:COLL:DATA:THRU:MATC

## SCPI Command

SENSe<Ch>:CORRection:COLLect:DATA:THRU:MATCh  
<rcvport>,<srcport>,<numeric list>

SENSe<Ch>:CORRection:COLLect:DATA:THRU:MATCh? <rcvport>,<srcport>

## Description

Writes or reads out the array of the reflection measurement of the thru standard connected between the receiver port <rcvport> and the source port <srcport>.

The array size is 2N, where N is the number of measurement points.

For the n–th point, where n from 1 to N:

<numeric 2n–1> real part of the measurement

<numeric 2n> imaginary part of the measurement

command/query

## Target

Channel <Ch>,

<Ch>={{1}|2|...16}

## Parameter

<rcvport>	The number of the receiver port from 1 to 2
<srcport>	The number of the source port from 1 to 2
<numeric list>	The data array of the reflection measurements using the thru standard

## Query Response

<numeric 1>, <numeric 2>, ...<numeric 2N>

## Related Commands

[SENS:CORR:COLL:THRU](#)

## Equivalent Softkeys

None

---

## Equivalent COM Command

SCPI.SENSE(Ch).CORRection.COLLection.DATA.THRU.MATCh(Pt\_r,Pt\_s)

## Syntax

Data = app.SCPI.SENSE(Ch).CORRection.COLLection.DATA.THRU.MATCh(Pt\_r, Pt\_s)

app.SCPI.SENSE(Ch).CORRection.COLLection.DATA.THRU.MATCh(Pt\_r, Pt\_s) = Data

## Type

Variant (array of Double) (read/write)

---

Back to [SENSe](#)



## SENS:CORR:COLL:DATA:THRU:TRAN

### SCPI Command

SENSe<Ch>:CORRection:COLLect:DATA:THRU:TRANsmission  
<rcvport>,<srcport>,<numeric list>

SENSe<Ch>:CORRection:COLLect:DATA:THRU:TRANsmission?  
<rcvport>,<srcport>

### Description

Writes or reads out the array of the transmission measurement performed between the receiver port <rcvport> and the source port <srcport> using the thru standard.

The array size is  $2N$ , where  $N$  is the number of measurement points.

For the  $n$ -th point, where  $n$  from 1 to  $N$ :

<numeric  $2n-1$ > real part of the measurement

<numeric  $2n$ > imaginary part of the measurement

command/query

### Target

Channel <Ch>,

<Ch>={[1]|2|...16}

### Parameter

<rcvport>	The number of the receiver port from 1 to 2
<srcport>	The number of the source port from 1 to 2
<numeric list>	The data array of the transmission measurements using the thru standard

### Query Response

<numeric 1>,<numeric 2>,...<numeric  $2N$ >

## Related Commands

[SENS:CORR:COLL:THRU](#)

## Equivalent Softkeys

None

---

## Equivalent COM Command

SCPI.SENSE(Ch).CORRection.COLLection.DATA.THRU.TRANsmission(Pt\_r, Pt\_s)

## Syntax

Data  
app.SCPI.SENSE(Ch).CORRection.COLLection.DATA.THRU.TRANsmission(Pt\_r, Pt\_s) =

app.SCPI.SENSE(Ch).CORRection.COLLection.DATA.THRU.TRANsmission(Pt\_r, Pt\_s) = Data

## Type

Variant (array of Double) (read/write)

---

Back to [SENSe](#)

## SENS:CORR:COLL:ECAL:CCH

### SCPI Command

SENSe<Ch>:CORRection:COLLect:ECAL:CCHeck[:ACQuire]

### Description

Executes the confidence check of the calibration coefficients of the 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.

command only

### Target

Channel <Ch>,

<Ch>={{1}|2|...16}

### Equivalent Softkeys

Calibration > AutoCal > Confidence Check

---

### Equivalent COM Command

SCPI.SENSE(CH).CORRection.COLLect.ECAL.CCHeck.ACQuire

### Syntax

app.SCPI.SENSE(CH).CORRection.COLLect.ECAL.CCHeck.ACQuire

### Type

Method

---

Back to [SENSe](#)

## SENS:CORR:COLL:ECAL:ERES

### SCPI Command

SENSe<Ch>:CORRection:COLLect:ECAL:ERESponse <rcvport>,<srcport>

### Description

Executes one path two-port calibration between the specified 2 ports of the specified channel using the AutoCal module.

command only

### Target

Channel <Ch>,

<Ch>={{1}|2|...16}

### Parameter

<rcvport>      The number of the receiver port from 1 to 2

<srcport>      The number of the source port from 1 to 2

### Equivalent Softkeys

Calibration > AutoCal > One Path 2-Port Cal

---

### Equivalent COM Command

SCPI.SENSE(CH).CORRection.COLLect.ECAL.ERESponse

### Syntax

app.SCPI.SENSE(CH).CORRection.COLLect.ECAL.ERESponse = Array(2, 1)

## Type

Variant (array of long)(write only)

---

Back to [SENSe](#)

## **SENS:CORR:COLL:ECAL:INF?**

### **SCPI Command**

SENSe:CORRection:COLLect:ECAL:INFormation?

### **Description**

Gets information on the AutoCal Module connected to the Network Analyzer.

query only

### **Target**

AutoCal Module

### **Query Response**

The query returns information 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

### **Equivalent Softkeys**

**Calibration > AutoCal > Characterization Info...**

---

### **Equivalent COM Command**

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

### **Syntax**

ID = app.SCPI.SENSE(1).CORRection.COLLEct.ECAL.INFOrmation

### **Type**

String (read only)>

---

Back to [SENSe](#)

## **SENS:CORR:COLL:ECAL:ORI:EXEC**

### **SCPI Command**

SENSe:CORRection:COLLect:ECAL:ORlentation:EXECute

### **Description**

Executes the Auto-Orientation procedure of the AutoCal Module. The AutoCal Module must be connected to the ports of Analyzer.

command

### **Target**

AutoCal Module

### **Equivalent Softkeys**

**Calibration > AutoCal > Orientation > Execute Auto-Orientation**

---

### **Equivalent COM Command**

None

---

Back to [SENSe](#)



## SENS:CORR:COLL:ECAL:ORI:STAT

### SCPI Command

SENSe:CORRection:COLLect:ECAL:ORlentation:STATe {OFF|ON|0|1}

SENSe:CORRection:COLLect:ECAL:ORlentation:STATe?

### Description

Turns the Auto-Orientation function ON/OFF when the AutoCal Module calibration is executed.

command/query

### Target

AutoCal Module

### Parameter

{ON|1}      ON

{OFF|0}     OFF

### Query Response

{0|1}

### Preset Value

0

### Equivalent Softkeys

Calibration > AutoCal > Orientation > Auto-Orientation

---

### Equivalent COM Command

SCPI.SENSe(1).CORRection.COLLect.ECAL.ORlenation.STATe

## Syntax

Status = app.SCPI.SENSE(1).CORRection.COLLEct.ECAL.ORlentation.STATe

app.SCPI.SENSE(1).CORRection.COLLEct.ECAL.ORlentation.STATe = False

## Type

Boolean (read/write)

---

Back to [SENSe](#)

## SENS:CORR:COLL:ECAL:PATH

### SCPI Command

SENSe:CORRection:COLLect:ECAL:PATH <numeric1>,<numeric2>

SENSe:CORRection:COLLect:ECAL:PATH? <numeric1>

### Description

Sets or reads out the AutoCal module port number which is connected to a specified port of the Network Analyzer.

command/query

### Target

AutoCal Module

### Parameter

<numeric1>      Network Analyzer Port Number

<numeric2>      AutoCal Module Port Number:

1- Port A of AutoCal Module

2- Port B of AutoCal Module

### Query Response

<numeric>

### Equivalent Softkeys

Calibration > AutoCal > Orientation > Port n

---

### Equivalent COM Command

SCPI.SENSe(1).CORRection.COLLect.ECAL.PATH(Pt)

**Syntax**

Value = app.SCPI.SENSE(1).CORRection.COLLect.ECAL.PATH(Pt)

app.SCPI.SENSE(1).CORRection.COLLect.ECAL.PATH(Pt) = 2

**Type**

Long (read/write)

---

Back to [SENSe](#)

## SENS:CORR:COLL:ECAL:SOLT1

### SCPI Command

SENSe<Ch>:CORRection:COLLect:ECAL:SOLT1 <port>

### Description

Executes one-port calibration of the specified port of the specified channel using the AutoCal module.

command only

### Target

Channel <Ch>,

<Ch>={[1]|2|...16}

### Parameter

<port>      Port Number

### Equivalent Softkeys

Calibration > AutoCal > 1-Port AutoCal > Port n

---

### Equivalent COM Command

SCPI.SENSE(CH).CORRection.COLLect.ECAL.SOLT1

### Syntax

app.SCPI.SENSE(CH).CORRection.COLLect.ECAL.SOLT1 = Port

### Type

Long (read/write)

---

Back to [SENSe](#)

## SENS:CORR:COLL:ECAL:SOLT2

### SCPI Command

SENSe<Ch>:CORRection:COLLect:ECAL:SOLT2 <port1>,<port2>

### Description

Executes full two-port calibration between the specified 2 ports of the specified channel using the AutoCal module.

command only

### Target

Channel <Ch>,

<Ch>={[1]|2|...16}

### Parameter

<port1> Port Number

<port2> Port Number

### Equivalent Softkeys

Calibration > AutoCal > 2-Port AutoCal

---

### Equivalent COM Command

SCPI.SENSE(CH).CORRection.COLLect.ECAL.SOLT2

### Syntax

app.SCPI.SENSE(CH).CORRection.COLLect.ECAL.SOLT2 = Array(2, 1)

## Type

Variant (array of long)(write only)

---

Back to [SENSe](#)

## SENS:CORR:COLL:ECAL:THER:COMP

### SCPI Command

SENSe:CORRection:COLLect:ECAL:THERmo:COMPensation[:STATe] {OFF|ON|0|1}

SENSe:CORRection:COLLect:ECAL:THERmo:COMPensation[:STATe]?

### Description

Turns the thermo compensation function ON/OFF when the AutoCal Module calibration is executed.

command/query

### Target

AutoCal Module

### Parameter

{ON|1}      ON

{OFF|0}      OFF

### Query Response

{0|1}

### Preset Value

1

### Equivalent Softkeys

Calibration > AutoCal > Orientation > Thermo Compensation

---

### Equivalent COM Command

None

---



Back to [SENSe](#)

## **SENS:CORR:COLL:ECAL:UCH**

### **SCPI Command**

SENSe:CORRection:COLLect:ECAL:UCHar <char>

SENSe:CORRection:COLLect:ECAL:UCHar?

### **Description**

Sets or reads out the characterization number used when executing AutoCal (factory or user characterizations).

command/query

### **Target**

AutoCal

### **Parameter**

<char> Specifies the stimulus type:

**CHAR0**      Factory characterization

**CHAR1**      User characterization 1

**CHAR2**      User characterization 2

**CHAR3**      User characterization 3

### **Query Response**

{CHAR0|CHAR1|CHAR2|CHAR3}

### **Preset Value**

CHAR0

### **Equivalent Softkeys**

**Calibration > AutoCal > Characterization**

---

### Equivalent COM Command

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

### Syntax

Param = app.SCPI.SENSE(1).CORRection.COLLect.ECAL.UCHar

app.SCPI.SENSE(1).CORRection.COLLect.ECAL.UCHar = "CHAR0"

### Type

String (read/write)

---

Back to [SENSe](#)

# SENS:CORR:COLL:ECAL:UTHR:STAT

## SCPI Command

SENSe:CORRection:COLLect:ECAL:UTHRu:STATe {OFF|ON|0|1}

SENSe:CORRection:COLLect:ECAL:UTHRu:STATe?

## Description

Turns the Unknown Thru feature ON/OFF when the AutoCal Module calibration is executed.

**Note:** Planar 304/1 does not support this method.

command/query

## Target

AutoCal

## Parameter

{ON|1}      ON

{OFF|0}      OFF

## Query Response

{0|1}

## Preset Value

0

## Equivalent Softkeys

Calibration > AutoCal > Unkn Thru

---

## Equivalent COM Command

SCPI.SENSe(1).CORRection.COLLect.ECAL.UTHRu.STATe

## Syntax

Status = app.SCPI.SENSE(1).CORRection.COLLeCt.ECAL.UTHRu.STATe

app.SCPI.SENSE(1).CORRection.COLLeCt.ECAL.UTHRu.STATe = False

## Type

Boolean (read/write)

---

### **WARNING**

Object SENSE has an index of 1, which can be omitted in Visual Basic but it cannot be omitted in other programming languages.

---

---

Back to [SENSe](#)

## SENS:CORR:COLL:ISOL

### SCPI Command

SENSe<Ch>:CORRection:COLLect[:ACQuire]:ISOLation <rcvport>,<srcport>

### Description

Measures the isolation calibration data between the receiver port <rcvport> and the source port <srcport>.

**Note:** 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 blocks the execution of the subsequent commands until the completion of the measurement.

no query

### Target

Channel <Ch>,

<Ch>={[1]|2|...16}

### Parameter

<rcvport>      The number of the receiver port from 1 to 2

<srcport>      The number of the source port from 1 to 2

### Out of Range

If the same port numbers are specified, an error occurs.

### Equivalent Softkeys

**Calibration > Calibrate > Response (Thru) > Isolation (Optional)**

**Calibration > Calibrate > One Path 2–Port Cal > Isolation (Optional)**

**Calibration > Calibrate > 2–Port SOLT Cal > Port 1-2 Isol (Optional)**

---

### Equivalent COM Command

SCPI.SENSE(Ch).CORRection.COLLection.ACQuire.ISOLation

### Syntax

app.SCPI.SENSE(Ch).CORRection.COLLection.ACQuire.ISOLation = Array(1, 2)

### Type

Variant (array of Long) (write only)

---

Back to [SENSe](#)

## SENS:CORR:COLL:LOAD

### SCPI Command

SENSe<Ch>:CORRection:COLLect[:ACQuire]:LOAD <port>

### Description

Measures the calibration data of the load standard for the specified port.

**Note:** 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 blocks the execution of the subsequent commands until the completion of the measurement.

no query

### Target

Channel <Ch>,

<Ch>={{1}|2|...16}

### Parameter

<port>      The number of the port from 1 to 2

### Equivalent Softkeys

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

---

### Equivalent COM Command

SCPI.SENSe(Ch).CORRection.COLLect.ACQuire.LOAD



**Syntax**

app.SCPI.SENSE(CH).CORRection.COLLection.ACQuire.LOAD = 1

**Type**

Long (write only)

---

Back to [SENSe](#)

# SENS:CORR:COLL:OPEN

## SCPI Command

SENSe<Ch>:CORRection:COLLEct[:ACQuire]:OPEN <port>

## Description

Measures the calibration data of the open standard for the specified port.

**Note:** 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 blocks the execution of the subsequent commands until the completion of the measurement.

no query

## Target

Channel <Ch>,

<Ch>={[1]|2|...16}

## Parameter

<port>      The number of the port from 1 to 2

## Equivalent Softkeys

Calibration > Calibrate > Response (Open) > Open

Calibration > Calibrate > Full 1–Port Cal > Open

Calibration > Calibrate > One Path 2–Port Cal > Open

Calibration > Calibrate > 2–Port SOLT Cal > Port n Open

---

**Equivalent COM Command**

SCPI.SENSE(Ch).CORRection.COLLection.ACQuire.OPEN

**Syntax**

app.SCPI.SENSE(Ch).CORRection.COLLection.ACQuire.OPEN = 1

**Type**

Long (write only)

---

**Back to** [SENSe](#)

## SENS:CORR:COLL:SHOR

### SCPI Command

SENSe<Ch>:CORRection:COLLEct[:ACQuire]:SHORT <port>

### Description

Measures the calibration data of the short standard for the specified port.

**Note:** 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 blocks the execution of the subsequent commands until the completion of the measurement.

no query

### Target

Channel <Ch>,

<Ch>={[1]|2|...16}

### Parameter

<port>      The number of the port from 1 to 2

### Equivalent Softkeys

Calibration > Calibrate > Response (Short) > Short

Calibration > Calibrate > Full 1–Port Cal > Short

Calibration > Calibrate > One Path 2–Port Cal > Short

Calibration > Calibrate > Full 2–Port Cal > Port n Short

Calibration > Calibrate > 3–Port SOLT Cal > Reflection (Port n) > Short

Calibration > Calibrate > 4–Port SOLT Cal > Reflection (Port n) > Short

---

### Equivalent COM Command

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

### Syntax

app.SCPI.SENSE(Ch).CORRection.COLLection.ACQuire.SHORt = 1

### Type

Long (write only)

---

Back to [SENSe](#)

## SENS:CORR:COLL:THRU

### SCPI Command

SENSe<Ch>:CORRection:COLLEct[:ACQuire]:THRU <rcvport>,<srcport>

### Description

Measures the calibration data of the thru standard between the receiver port <rcvport> and the source port <srcport>.

**Note:** 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 blocks the execution of the subsequent commands until the completion of the measurement.

no query

### Target

Channel <Ch>,

<Ch>={1|2|...16}

### Parameter

<rcvport>      The number of the receiver port from 1 to 2

<srcport>      The number of the source port from 1 to 2

### Out of Range

If the same port numbers are specified, an error occurs.

## Equivalent Softkeys

Calibration > Calibrate > Response (Thru) > Thru

Calibration > Calibrate > One Path 2–Port Cal > Thru

Calibration > Calibrate > 2–Port SOLT Cal > Port 1–2 Thru

---

## Equivalent COM Command

SCPI.SENSE(Ch).CORRection.COLLection.ACQuire.THRU

## Syntax

app.SCPI.SENSE(Ch).CORRection.COLLection.ACQuire.THRU= Array(1, 2)

## Type

Variant (array of Long) (write only)

---

Back to [SENSe](#)

## SENS:CORR:COLL:TRLL

### SCPI Command

SENSe<Ch>:CORRection:COLLect[:ACQuire]:TRLLine <port1>,<port2>

### Description

Measures the calibration data of the TRL line standard between <port1> and <port2>.

**Note:** 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 blocks the execution of the subsequent commands until the completion of the measurement.

no query

### Target

Channel <Ch>,

<Ch>={[1]|2|...16}

### Parameter

<rcvport>      The number of the receiver port from 1 to 2

<srcport>      The number of the source port from 1 to 2

### Out of Range

If the same port numbers are specified, an error occurs.

### Equivalent Softkeys

Calibration > Calibrate > n-Port TRL Cal > Line/Match

---

### Equivalent COM Command

SCPI.SENSe(Ch).CORRection.COLLect.ACQuire.TRLLine



**Syntax**

app.SCPI.SENSE(CH).CORRection.COLLection.ACQuire.TRLLine = Array(1, 2)

**Type**

Variant (array of long) (write only)

---

Back to [SENSe](#)

## SENS:CORR:COLL:TRLT

### SCPI Command

SENSe<Ch>:CORRection:COLLect[:ACQuire]:TRLThru <port1>,<port2>

### Description

Measures the calibration data of the TRL thru standard between <port1> and <port2>.

**Note:** 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 blocks the execution of the subsequent commands until the completion of the measurement.

no query

### Target

Channel <Ch>,

<Ch>={{1}|2|...16}

### Parameter

<rcvport>      The number of the receiver port from 1 to 2

<srcport>      The number of the source port from 1 to 2

### Out of Range

If the same port numbers are specified, an error occurs.

### Equivalent Softkeys

Calibration > Calibrate > n-Port TRL Cal > Thru/Line

---

### Equivalent COM Command

SCPI.SENSe(Ch).CORRection.COLLect.ACQuire.TRLThru

**Syntax**

app.SCPI.SENSE(CH).CORRection.COLLection.ACQuire.TRLThru = Array(1, 2)

**Type**

Variant (array of long) (write only)

---

Back to [SENSe](#)

## SENS:CORR:COLL:TRLR

### SCPI Command

SENSe<Ch>:CORRection:COLLect[:ACQuire]:TRLReflect <port>

### Description

Measures the calibration data of the TRL reflect standard for the specified port.

**Note:** 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 blocks the execution of the subsequent commands until the completion of the measurement.

no query

### Target

Channel <Ch>,

<Ch>={[1]|2|...16}

### Parameter

<port>    The number of the port from 1 to 2

### Equivalent Softkeys

Calibration > Calibrate > n–Port TRL Cal > Reflect

---

### Equivalent COM Command

SCPI.SENSE(CH).CORRection.COLLect.ACQuire.TRLReflect

### Syntax

app.SCPI.SENSE(CH).CORRection.COLLect.ACQuire.TRLReflect = 1

**Type**

Long (write only)

---

Back to [SENSe](#)

## SENS:CORR:COLL:SUBC

### SCPI Command

SENSe<Ch>:CORRection:COLLect[:ACQuire]:SUBClass <numeric>

SENSe<Ch>:CORRection:COLLect[:ACQuire]:SUBClass?

### Description

Selects the subclass number of calibration standard used for measurement by the subsequent command [SENS:CORR:COLL:XXXX](#). If the calibration kit contains several calibration standards of the same type, say SHORTs, this allows select the particular SHORT. The subclasses must be set in advance by the commands [SENS:CORR:COLL:CKIT:ORD:XXXX](#) or in the user interface "Specify Classes"..

command/query

### Target

Calibration kit, selected for channel <Ch>,

<Ch>={{1}|2|...16}

### Parameter

<numeric> the subclass number from 1 to 8

### Query Response

<numeric>

### Preset Value

1

### Equivalent Softkeys

None

---

### Equivalent COM Command

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

**Syntax**

app.SCPI.SENSE(CH).CORRection.COLLection.ACQuire.SUBClass = 2

Subclass = app.SCPI.SENSE(CH).CORRection.COLLection.ACQuire.SUBClass

**Type**

Long (write only)

---

Back to [SENSe](#)

## SENS:CORR:COLL:METH:ERES

### SCPI Command

SENSe<Ch>:CORRection:COLLect:METHod:ERESponse <rcvport>,<srcport>

### Description

Selects the ports and sets the one path 2–port calibration type for the calculation of the calibration coefficients on completion of the calibration executed by the [SENS:CORR:COLL:SAVE](#) command.

no query

### Target

Channel <Ch>,

<Ch>={{1}|2|...16}

### Parameter

<rcvport>      The number of the receiver port from 1 to 2

<srcport>      The number of the source port from 1 to 2

### Out of Range

If the same port numbers are specified, an error occurs.

### Related Commands

[SENS:CORR:COLL:SAVE](#)

### Equivalent Softkeys

Calibration > Calibrate > One Path 2–Port Cal > Select Port

---

### Equivalent COM Command

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



## Syntax

app.SCPI.SENSE(Ch).CORRection.COLLection.METHod.ERESponse = Array(2, 1)

## Type

Variant (array of Long) (write only)

---

Back to [SENSe](#)

## **SENS:CORR:COLL:METH:OPEN**

## SCPI Command

SENSe<Ch>:CORRection:COLLection:METHod[:RESPonse]:OPEN <port>

## Description

Selects the port and sets the response calibration (Open) type for the calculation of the calibration coefficients on completion of the calibration executed by the [SENS:CORR:COLL:SAVE](#) command.

no query

## Target

Channel <Ch>,

<Ch>={[1]|2|...16}

## Parameter

<port>      The number of the port from 1 to 2

## Related Commands

[SENS:CORR:COLL:SAVE](#)

## Equivalent Softkeys

**Calibration > Calibrate > Response (Open) > Select Port**

---

**Equivalent COM Command**

SCPI.SENSE(Ch).CORRection.COLLection.METHod.RESPOnse.OPEN

**Syntax**

app.SCPI.SENSE(Ch).CORRection.COLLection.METHod.RESPOnse.OPEN = 1

**Type**

Long (write only)

---

Back to [SENSe](#)

## SENS:CORR:COLL:METH:SHOR

### SCPI Command

SENSe<Ch>:CORRection:COLLect:METHod[:RESPonse]:SHORt <port>

### Description

Selects the port and sets the response calibration (Short) type for the calculation of the calibration coefficients on completion of the calibration executed by the [SENS:CORR:COLL:SAVE](#) command.

no query

### Target

Channel <Ch>,

<Ch>={[1]|2|...16}

### Parameter

<port>            The number of the port from 1 to 2

### Related Commands

[SENS:CORR:COLL:SAVE](#)

### Equivalent Softkeys

Calibration > Calibrate > Response (Short) > Select Port

---

### Equivalent COM Command

SCPI.SENSE(CH).CORRection.COLLect.METHod.RESPonse.SHORt

### Syntax

app.SCPI.SENSE(CH).CORRection.COLLect.METHod.RESPonse.SHORt = 1

**Type**

Long (write only)

---

Back to [SENSe](#)

## SENS:CORR:COLL:METH:SOLT1

### SCPI Command

SENSe<Ch>:CORRection:COLLect:METHod:SOLT1 <port>

### Description

Selects the port and sets the full one-port (SOL) calibration type for the calculation of the calibration coefficients on completion of the calibration executed by the [SENS:CORR:COLL:SAVE](#) command.

no query

### Target

Channel <Ch>,

<Ch>={[1]|2|...16}

### Parameter

<port>      The number of the port from 1 to 2

### Related Commands

[SENS:CORR:COLL:SAVE](#)

### Equivalent Softkeys

Calibration > Calibrate > 1–Port SOL Cal > Select Port

---

### Equivalent COM Command

SCPI.SENSE(CH).CORRection.COLLect.METHod.SOLT1

### Syntax

app.SCPI.SENSE(CH).CORRection.COLLect.METHod.SOLT1 = 1

**Type**

Long (write only)

---

Back to [SENSe](#)

## SENS:CORR:COLL:METH:SOLT2

### SCPI Command

SENSe<Ch>:CORRection:COLLect:METHod:SOLT2 <port1>,<port2>

### Description

Selects the port and sets the full two-port (SOLT) calibration type for the calculation of the calibration coefficients on completion of the calibration executed by the [SENS:CORR:COLL:SAVE](#) command.

no query

### Target

Channel <Ch>,

<Ch>={{1}|2|...16}

### Parameter

<port1>      The first port number from 1 to 2

<port2>      The second port number from 1 to 2

### Out of Range

If the same port numbers are specified, an error occurs.

### Related Commands

[SENS:CORR:COLL:SAVE](#)

### Equivalent Softkeys

Calibration > Calibrate > 2–Port SOLT Cal



---

### Equivalent COM Command

SCPI.SENSE(Ch).CORRection.COLLection.METHod.SOLT2

### Syntax

app.SCPI.SENSE(Ch).CORRection.COLLection.METHod.SOLT2 = Array(2, 1)

### Type

Variant (array of Long) (write only)

---

Back to [SENSe](#)

## SENS:CORR:COLL:METH:THRU

### SCPI Command

SENSe<Ch>:CORRection:COLLect:METHod[:RESPonse]:THRU  
<rcvport>,<srcport>

### Description

Selects the ports and sets the response calibration (Thru) type for the calculation of the calibration coefficients on completion of the calibration executed by the [SENS:CORR:COLL:SAVE](#) command.

no query

### Target

Channel <Ch>,

<Ch>={{1}|2|...16}

### Parameter

<rcvport>      The number of the receiver port from 1 to 2

<srcport>      The number of the source port from 1 to 2

### Out of Range

If the same port numbers are specified, an error occurs.

### Related Commands

[SENS:CORR:COLL:SAVE](#)

### Equivalent Softkeys

Calibration > Calibrate > Response (Thru) > Select Port {x-y}

---

### Equivalent COM Command

SCPI.SENSe(Ch).CORRection.COLLect.METHod.RESPonse.THRU

## Syntax

app.SCPI.SENSE(CH).CORRection.COLLection.METHod.RESPOnse.THRU      =  
Array(2,1)

## Type

Variant (array of Long) (write only)

---

Back to [SENSe](#)

## SENS:CORR:COLL:METH:TRL:MULT

### SCPI Command

SENSe<Ch>:CORRection:COLLect:METHod:TRL:MuLTiline[:STATe] {OFF|ON|0|1}

SENSe<Ch>:CORRection:COLLect:METHod:TRL:MuLTiline[:STATe]?

### Description

Turns the multi-line TRL option ON/OFF. Determines which TRL algorithm is used when calculating the calibration coefficients using the [SENS:CORR:COLL:SAVE](#) command. If turned on, the multi-line TRL algorithm is used. If turned off, the classic TRL algorithm is used. Multi-line TRL option is not supported by Planar 304, Planar 304/1 analyzers.

command/query

### Target

Channel <Ch>,

<Ch>={[1]|2|...16}

### Parameter

{ON|1}      ON

{OFF|0}      OFF

### Query Response

{0|1}

### Preset Value

0

### Related Commands

[SENS:CORR:COLL:SAVE](#)

[SENS:CORR:COLL:METH:TRL2](#)

## Equivalent Softkeys

Calibration > Calibrate > n-Port TRL Cal > Multiline

---

## Equivalent COM Command

None

---

Back to [SENSe](#)

## **SENS:CORR:COLL:METH:TRL2**

### **SCPI Command**

SENSe<Ch>:CORRection:COLLect:METHod:TRL2 <port1>,<port2>

### **Description**

Selects the ports and sets the two-port TRL calibration type for the calculation of the calibration coefficients on completion of the calibration executed by the [SENS:CORR:COLL:SAVE](#) command.

no query

### **Target**

Channel <Ch>,

<Ch>={[1]|2|...16}

### **Parameter**

<port1>      The first port number from 1 to 2

<port2>      The second port number from 1 to 2

### **Out of Range**

If the same port numbers are specified, an error occurs.

### **Related Commands**

[SENS:CORR:COLL:SAVE](#)

### **Equivalent Softkeys**

**Calibration > Calibrate > 2–Port TRL Cal**

---

### Equivalent COM Command

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

### Syntax

app.SCPI.SENSE(Ch).CORRection.COLLection.METHod.TRL2 = Array(1, 2)

### Type

Variant (array of Long) (write only)

---

Back to [SENSe](#)

## SENS:CORR:COLL:METH:TYPE?

### SCPI Command

SENSe<Ch>:CORRection:COLLect:METHod:TYPE?

### Description

Reads out the calibration method selected for the calculation of the calibration coefficients on completion of the calibration executed by the [SENS:CORR:COLL:SAVE](#) command.

query only

### Target

Channel <Ch>,

<Ch>={[1]|2|...16}

### Query Response

<b>RESPO</b>	Response (Open)
<b>RESPS</b>	Response (Short)
<b>RESPT</b>	Response (Thru)
<b>SOLT1</b>	Full one-port calibration
<b>SOLT2</b>	Full two-port calibration
<b>1PATH</b>	One path two-port calibration
<b>NONE</b>	Not defined

### Equivalent Softkeys

None



---

**Equivalent COM Command**

SCPI.SENSE(Ch).CORRection.COLLection.METHod.TYPE

**Syntax**

Param = app.SCPI.SENSE(Ch).CORRection.COLLection.METHod.TYPE

**Type**

String (read only)>

---

Back to [SENSe](#)

## SENS:CORR:COLL:SAVE

### SCPI Command

SENSe<Ch>:CORRection:COLLect:SAVE

### Description

Calculates the calibration coefficients from the calibration standards measurements depending on the selected calibration type. The calibration type is selected by one of commands [SENS:CORR:COLL:METH:XXXX](#).

On completion of the command, all the calibration standards measurements are cleared and the error correction automatically turns ON.

At the attempt to execute this command before all the needed standards are measured, an error occurs and the command is ignored.

no query

### Target

Channel <Ch>,

<Ch>={[1]|2|...16}

### Related Commands

Calibration type selection:

[SENS:CORR:COLL:METH:XXXX](#)

Calibration standards measurement:

[SENS:CORR:COLL:ISOL](#)

[SENS:CORR:COLL:LOAD](#)

[SENS:CORR:COLL:OPEN](#)

[SENS:CORR:COLL:SHOR](#)

[SENS:CORR:COLL:THRU](#)

[SENS:CORR:COLL:TRLT](#)

[SENS:CORR:COLL:TRLR](#)

[SENS:CORR:COLL:TRLR](#)

## Equivalent Softkeys

Calibration > Calibrate > Respose (Open) | Respose (Short) | Respose (Thru) | 1-port SOL Cal | One Path 2-Port Cal |n-Port SOLT Cal. | n-Port TRL Cal > Apply

---

## Equivalent COM Command

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

## Syntax

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

## Type

Method

---

Back to [SENSe](#)

## SENS:CORR:COLL:THRU:ADD:DEL

### SCPI Command

SENSe<Ch>:CORRection:COLLect:THRU:ADDition:DELay <numeric>

SENSe<Ch>:CORRection:COLLect:THRU:ADDition:DELay?

### Description

Sets or reads out the approximate delay value of an unknown thru in the thru addition function. This value is used to eliminate the uncertainty of  $\pm 180^\circ$  when calculating the phase response of the thru.

If this value is set to zero, the analyzer uses an algorithm to automatically determine the delay of the thru. In most cases setting this value to zero is enough. Setting this value to non zero is required when:

$$FrequencyStep > \frac{1}{2Delay}.$$

**Note:** The delay and the length of the adapter can be set mutually

$$Delay = \frac{Length \sqrt{Permittivity}}{c}$$

command/query

### Target

Channel <Ch>,

<Ch>={[1]|2|...16}

### Parameter

<numeric> the approximate delay value of the thru.

### Unit

sec (seconds)

### Query Response

<numeric>

**Preset Value**

0

**Equivalent Softkeys**

**Calibration > Calibrate > Thru Addition > Thru Delay**

---

**Equivalent COM Command**

None

---

Back to [SENSe](#)

## SENS:CORR:COLL:THRU:ADD:LENG

### SCPI Command

SENSe<Ch>:CORRection:COLLect:THRU:ADDition:LENGth <numeric>

SENSe<Ch>:CORRection:COLLect:THRU:ADDition:LENGth?

### Description

Sets or reads out the approximate value of the mechanical length of an unknown thru in the thru addition function. This value is used to eliminate the uncertainty of  $\pm 180^\circ$  when calculating the phase response of the thru.

If this value is set to zero, the analyzer uses an algorithm to automatically determine the delay of the thru. In most cases setting this value to zero is enough. Setting this value to non zero is required when:

$$FrequencyStep > \frac{1}{2Delay}.$$

**Note:** The delay and the length of the adapter can be set mutually

$$Delay = \frac{Length \sqrt{Permittivity}}{c}$$

command/query

### Target

Channel <Ch>,

<Ch>={[1]|2|...16}

### Parameter

<numeric> the approximate value of the thru length.

### Unit

m (meters)

### Query Response

<numeric>

**Preset Value**

0

**Equivalent Softkeys**

**Calibration > Calibrate > Thru Addition > Thru Delay**

---

**Equivalent COM Command**

None

---

Back to [SENSe](#)

## **SENS:CORR:COLL:THRU:ADD:UNIT**

### **SCPI Command**

SENSe<Ch>:CORRection:COLLect:THRU:ADDition:UNIT {SEConds|METers}

SENSe<Ch>:CORRection:COLLect:THRU:ADDition:UNIT?

### **Description**

Selects the display units of the thru delay (length) in the thru addition function.

command/query

### **Target**

Channel <Ch>,

<Ch>={{1}|2|...16}

### **Parameter**

**SEConds**     Selects the seconds

**METers**       Selects the meters

### **Query Response**

{SEC|MET}

### **Preset Value**

SEConds

### **Equivalent Softkeys**

**Calibration > Calibrate > Thru Addition > Delay Unit**



---

## Equivalent COM Command

None

---

Back to [SENSe](#)

## SENS:CORR:COLL:THRU:ADD:MED

### SCPI Command

SENSe<Ch>:CORRection:COLLect:THRU:ADDition:MEDia {COAXial|WAVeguide}

SENSe<Ch>:CORRection:COLLect:THRU:ADDition:MEDia?

### Description

Specifies the media of the thru in the thru addition function.

**Note:** When the waveguide adapter is used it is recommended to specify the thru length instead of its delay.

command/query

### Target

Channel <Ch>,

<Ch>={1|2|...16}

### Parameter

**COAXial** Specifies the coaxial

**WAVeguide** Specifies the waveguide

### Query Response

{COAX|WAV}

### Preset Value

COAXial

## Equivalent Softkeys

Calibration > Calibrate > Thru Addition > Thru Media

---

## Equivalent COM Command

None

---

Back to [SENSe](#)

## **SENS:CORR:COLL:THRU:ADD:PERM**

### **SCPI Command**

SENSe<Ch>:CORRection:COLLect:THRU:ADDition:PERMittivity <numeric>

SENSe<Ch>:CORRection:COLLect:THRU:ADDition:PERMittivity?

### **Description**

Sets or reads out the value of the permittivity of the thru media in the thru addition function.

This parameter is used to calculate the adapter delay when the thru length is setting; therefore this parameter must be set before setting of the thru length.

command/query

### **Target**

Channel <Ch>,

<Ch>={1|2|...16}

### **Parameter**

<numeric> the value of the permittivity of the thru

### **Query Response**

<numeric>

### **Preset Value**

1.000649 (air)

### **Equivalent Softkeys**

**Calibration > Calibrate > Thru Addition > Permittivity**

---

### Equivalent COM Command

None

---

Back to [SENSe](#)

## SENS:CORR:COLL:THRU:ADD:WAV:CUT

### SCPI Command

SENSe<Ch>:CORRection:COLLEct:THRU:ADDition:WAVEguide:CUToff <numeric>

SENSe<Ch>:CORRection:COLLEct:THRU:ADDition:WAVEguide:CUToff?

### Description

Sets or reads out the value of the cutoff frequency of the waveguide thru in the thru addition function.

command/query

### Target

Channel <Ch>,

<Ch>={[1]|2|...16}

### Parameter

<numeric> the value of the cutoff frequency of the waveguide thru.

### Query Response

<numeric>

### Preset Value

1.0 GHz

### Equivalent Softkeys

Calibration > Calibrate > Thru Addition > Cutoff Frequency

---

### Equivalent COM Command

None

---

Back to [SENSe](#)

## **SENS:CORR:COLL:THRU:ADD:FULL2:COMP**

### **SCPI Command**

SENSe<Ch>:CORRection:COLLect:THRU:ADDition:FULL2:COMPlete  
<port1>,<port2>

### **Description**

Completes the full two-port calibration between the specified ports provided that each port was calibrated using full one-port calibration:

- Measures an unknown thru between the ports;
- Calculates the error terms Et and Ei using the unknown thru algorithm;
- Saves the Et and Ei error terms to the existing calibration getting the full two-port calibration from the two one-port calibrations.

If the full two-port calibration already existed between the specified ports, updates the Et and Ei error terms.

no query

### **Target**

Channel <Ch>,

<Ch>={[1]2|...16}

### **Parameter**

<port1>      The first port number from 1 to 2

<port2>      The second port number from 1 to 2

### **Equivalent Softkeys**

**Calibration > Calibrate > Thru Addition > Complete 2-Port Calibration**

---

### Equivalent COM Command

None

---

Back to [SENSe](#)



## SENS:CORR:EXT

### SCPI Command

SENSe<Ch>:CORRection:EXTension[:STATe] {OFF|ON|0|1}

SENSe<Ch>:CORRection:EXTension[:STATe]?

### Description

Turns the port extension function ON/OFF.

command/query

### Target

Channel <Ch>,

<Ch>={{1}|2|...16}

### Parameter

{ON|1}      ON

{OFF|0}      OFF

### Query Response

{0|1}

### Preset Value

0

### Equivalent Softkeys

Calibration > Port Extensions > Extension

---

### Equivalent COM Command

SCPI.SENSe(Ch).CORRection.EXTension.STATe

**Syntax**

Status = app.SCPI.SENSE(CH).CORRection.EXTension.STATe

app.SCPI.SENSE(CH).CORRection.EXTension.STATe = True

**Type**

Boolean (read/write)

---

Back to [SENSe](#)

## **SENS:CORR:EXT:AUTO:CONF**

### **SCPI Command**

SENSe<Ch>:CORRection:EXTension:AUTO:CONFig {CSPN|AMKR|USPN}

SENSe<Ch>:CORRection:EXTension:AUTO:CONFig?

### **Description**

Specifies the frequency range used for calculation of the results of the Auto Port Extension function.

command/query

### **Target**

Channel <Ch>,

<Ch>={[1]|2|...16}

### **Parameter**

<b>CSPN</b>	Uses current frequency span.
<b>AMKR</b>	Uses the frequency of the active marker. This is applied to Loss 1 and Loss 2 is ignored.
<b>USPN</b>	Uses arbitrary frequency range.

### **Query Response**

{CSPN|AMKR|USPN}

### **Preset Value**

CSPN

### **Equivalent Softkeys**

**Calibration > Port Extension > Auto Port Extension > Method {Current span | Active Marker | User Span}**

---

### Equivalent COM Command

None

---

Back to [SENSe](#)

## **SENS:CORR:EXT:AUTO:DCOF**

### **SCPI Command**

SENSe<Ch>:CORRection:EXTension:AUTO:DCOFfset {OFF|ON|0|1}

SENSe<Ch>:CORRection:EXTension:AUTO:DCOFfset?

### **Description**

Turns the usage of "Loss at DC" value for the results of the auto port extension function ON/OFF.

command/query

### **Target**

Channel <Ch>,

<Ch>={1|2|...16}

### **Parameter**

{ON|1}      ON

{OFF|0}      OFF

### **Query Response**

{0|1}

### **Preset Value**

0

### **Equivalent Softkeys**

**Calibration > Port Extension > Auto Port Extension > Adjust Mismatch {ON/OFF}**

---

## Equivalent COM Command

None

---

Back to [SENSe](#)

## SENS:CORR:EXT:AUTO:LOSS

### SCPI Command

SENSe<Ch>:CORRection:EXTension:AUTO:LOSS {OFF|ON|0|1}

SENSe<Ch>:CORRection:EXTension:AUTO:LOSS?

### Description

Turns the usage of "Loss1" and "Loss2" values for the results of the auto port extension function ON/OFF.

command/query

### Target

Channel <Ch>,

<Ch>={1|2|...16}

### Parameter

{ON|1}      ON

{OFF|0}      OFF

### Query Response

{0|1}

### Preset Value

0

### Equivalent Softkeys

Calibration > Port Extension > Auto Port Extension > Include Loss {ON/OFF}

---

## Equivalent COM Command

None

---

Back to [SENSe](#)



## SENS:CORR:EXT:AUTO:MEAS

### SCPI Command

SENSe<Ch>:CORRection:EXTension:AUTO:MEASure {SHORT|OPEN}

### Description

Performs measurement of the standard "SHORT" or "OPEN", automatically calculates and sets the parameters of the Port Extension.

The set of ports for which this command is executed is determined by the [SENS:CORR:EXT:AUTO:PORT](#) command.

When two consecutive measurements of "SHORT" and "OPEN" are performed the results of these measurements are averaged.

command

### Target

Channel <Ch>,

<Ch>={[1]|2|...16}

### Parameter

**SHORT**      Measures "SHORT" standard

**OPEN**        Measures "OPEN" standard

### Equivalent Softkeys

Calibration > Port Extension > Auto Port Extension > Measure Short | Measure Open

---

### Equivalent COM Command

None

---

Back to [SENSe](#)

## SENS:CORR:EXT:AUTO:PORT

### SCPI Command

SENSe<Ch>:CORRection:EXTension:AUTO:PORT<Pt> {OFF|ON|0|1}

SENSe<Ch>:CORRection:EXTension:AUTO:PORT<Pt>?

### Description

Turns the status of the auto port extension for the Port number <Pt> ON/OFF.

command/query

### Target

Port <Pt> of channel <Ch>,

<Ch>={{[1]|2|...16}}

<Pt>={{[1]|2}}

### Parameter

{ON|1}      ON

{OFF|0}     OFF

### Query Response

{0|1}

### Preset Value

1

### Equivalent Softkeys

Calibration > Port Extension > Auto Port Extension > Select Port(s)

---

## Equivalent COM Command

None

---

Back to [SENSe](#)

## SENS:CORR:EXT:AUTO:RES

### SCPI Command

SENSe<Ch>:CORRection:EXTension:AUTO:RESet

### Description

Deletes the finished measurement data of the OPEN and SHORT standards of the auto port extension function. Allows to start averaging again between the SHORT and OPEN standards.

command/query

### Target

Channel <Ch>,

<Ch>={{1}|2|...16}

### Equivalent Softkeys

Enter to the Auto Port Extension menu

---

### Equivalent COM Command

None

---

Back to [SENSe](#)

## **SENS:CORR:EXT:AUTO:STAR**

### **SCPI Command**

SENSe<Ch>:CORRection:EXTension:AUTO:STARt <frequency>

SENSe<Ch>:CORRection:EXTension:AUTO:STARt?

### **Description**

Sets or reads out the start value of the user span of the auto port extension function.

command/query

### **Target**

Channel <Ch>,

<Ch>={[1]|2|...16}

### **Parameter**

<frequency> the user span start

### **Unit**

Hz (Hertz)

### **Out of Range**

Sets the value of the limit, which is closer to the specified value.

### **Query Response**

<numeric>

### **Preset Value**

The Analyzer's lowest frequency.

### **Related Commands**

[SENS:CORR:EXT:AUTO:CONF](#)

## Equivalent Softkeys

Calibration > Port Extension > Auto Port Extension > User Span Start

---

## Equivalent COM Command

None

---

Back to [SENSe](#)

## **SENS:CORR:EXT:AUTO:STOP**

### **SCPI Command**

SENSe<Ch>:CORRection:EXTension:AUTO:STOP <frequency>

SENSe<Ch>:CORRection:EXTension:AUTO:STOP?

### **Description**

Sets or reads out the stop value of the user span of the auto port extension function.

command/query

### **Target**

Channel <Ch>,

<Ch>={[1]|2|...16}

### **Parameter**

<frequency> the user span stop

### **Unit**

Hz (Hertz)

### **Out of Range**

Sets the value of the limit, which is closer to the specified value.

### **Query Response**

<numeric>

### **Preset Value**

The Analyzer's lowest frequency.

### **Related Commands**

[SENS:CORR:EXT:AUTO:CONF](#)

## Equivalent Softkeys

Calibration > Port Extension > Auto Port Extension > User Span Stop

---

## Equivalent COM Command

None

---

Back to [SENSe](#)



## **SENS:CORR:EXT:PORT:FREQ**

### **SCPI Command**

SENSe<Ch>:CORRection:EXTension:PORT<Pt>:FREQuency{[1]|2} <frequency>

SENSe<Ch>:CORRection:EXTension:PORT<Pt>:FREQuency{[1]|2}?

### **Description**

Sets or reads out the values of the frequency 1 and frequency 2 to calculate the loss for the port extension function.

command/query

### **Target**

Port <Pt> of channel <Ch>,

<Ch>={[1]|2}...16}

<Pt>={[1]|2}

### **Parameter**

<frequency> the frequency value within the frequency limits of the analyzer.

### **Unit**

Hz (Hertz)

### **Out of Range**

Sets the value of the limit, which is closer to the specified value.

### **Query Response**

<numeric>

### **Preset Value**

1E9

## Equivalent Softkeys

Calibration > Port Extensions > Loss > {Freq1 | Freq2}

---

## Equivalent COM Command

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

## Syntax

Status = app.SCPI.SENSE(Ch).CORRection.EXTension.STATe

app.SCPI.SENSE(Ch).CORRection.EXTension.STATe = True

## Type

Boolean (read/write)

---

Back to [SENSe](#)

## SENS:CORR:EXT:PORT:INCL

### SCPI Command

SENSe<Ch>:CORRection:EXTension:PORT<Pt>:INCLude{[1]|2}[:STATe] {OFF|ON|0|1}

SENSe<Ch>:CORRection:EXTension:PORT<Pt>:INCLude{[1]|2}[:STATe]?

### Description

Turns the loss compensation of loss 1 and loss 2 for the port extension function ON/OFF.

command/query

### Target

Port <Pt> of channel <Ch>,

<Ch>={[1]|2|...16}

<Pt>={[1]|2}

### Parameter

{ON|1}      ON

{OFF|0}     OFF

### Query Response

{0|1}

### Preset Value

0

## Equivalent Softkeys

Calibration > Port Extensions > Loss > {Loss1 | Loss2}

---

## Equivalent COM Command

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

## Syntax

Status=

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

app.SCPI.SENSE(Ch).CORRection.EXTension.PORT(Pt).INCLude(Ls).STATe     =  
True

## Type

Boolean (read/write)

---

Back to [SENSe](#)

## SENS:CORR:EXT:PORT:LDC

### SCPI Command

SENSe<Ch>:CORRection:EXTension:PORT<Pt>:LDC <numeric>

SENSe<Ch>:CORRection:EXTension:PORT<Pt>:LDC?

### Description

Sets or reads out the loss value at DC for the port extension function.

command/query

### Target

Port <Pt> of channel <Ch>,

<Ch>={{[1]|2|...16}}

<Pt>={{[1]|2}}

### Parameter

<numeric> the loss value from –200 to 200

### Unit

dB (decibel)

### Out of Range

Sets the value of the limit, which is closer to the specified value.

### Query Response

<numeric>

### Preset Value

0

### Equivalent Softkeys

Calibration > Port Extensions > Loss > Loss at DC

---

### Equivalent COM Command

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

### Syntax

Value = app.SCPI.SENSE(Ch).CORRection.EXTension.PORT(Pt).LDC

app.SCPI.SENSE(Ch).CORRection.EXTension.PORT(Pt).LDC = 10

### Type

Double (read/write)

---

Back to [SENSe](#)

## SENS:CORR:EXT:PORT:LOSS

### SCPI Command

SENSe<Ch>:CORRection:EXTension:PORT<Pt>:LOSS{[1]|2} <numeric>

SENSe<Ch>:CORRection:EXTension:PORT<Pt>:LOSS{[1]|2}?

### Description

Sets or reads out the values of loss 1 and loss 2 for the port extension function.

command/query

### Target

Port <Pt> of channel <Ch>,

<Ch>={[1]|2|...16}

<Pt>={[1]|2}

### Parameter

<numeric> the loss value from –200 to 200

### Unit

dB (decibel)

### Out of Range

Sets the value of the limit, which is closer to the specified value.

### Query Response

<numeric>

### Preset Value

0

### Equivalent Softkeys

Calibration > Port Extensions > Loss > {Loss1 | Loss2}

---

### Equivalent COM Command

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

### Syntax

Value = app.SCPI.SENSE(Ch).CORRection.EXTension.PORT(Pt).LOSS(Ls)

app.SCPI.SENSE(Ch).CORRection.EXTension.PORT(Pt).LOSS(Ls) = 10

### Type

Double (read/write)

---

Back to [SENSe](#)



## SENS:CORR:EXT:PORT:TIME

### SCPI Command

SENSe<Ch>:CORRection:EXTension:PORT<Pt>:TIME <time>

SENSe<Ch>:CORRection:EXTension:PORT<Pt>:TIME?

### Description

Sets or reads out the electrical delay value for the port extension function.

command/query

### Target

Port <Pt> of channel <Ch>,

<Ch>={{[1]|2|...16}}

<Pt>={{[1]|2}}

### Parameter

<time> the electrical delay value from –10 to 10

### Unit

sec (second)

### Out of Range

Sets the value of the limit, which is closer to the specified value.

### Query Response

<numeric>

### Preset Value

0

### Equivalent Softkeys

Calibration > Port Extensions > Extension Port n

---

### Equivalent COM Command

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

### Syntax

Value = app.SCPI.SENSE(Ch).CORRection.EXTension.PORT(Pt).TIME

app.SCPI.SENSE(Ch).CORRection.EXTension.PORT(Pt).TIME = 10E-9

### Type

Double (read/write)

---

Back to [SENSe](#)

## SENS:CORR:INF?

### SCPI Command

SENSe<Ch>:CORRection:INFormation? <rcvport>,<srcport>

### Description

Reads out the information string of the calibration acting between the <rcvport> and <srcport>.

query only

### Target

Channel <Ch>,

<Ch>={1|2|...16}

### Parameter

<rcvport>      The number of the receiver port from 1 to 2

<srcport>      The number of the source port from 1 to 2

### Query Response

<YYYY/MM/DD>   <HH:MM:SS>,   <Type>,   <TypeEx>,   <IFBW>,   <Power>,  
<Temperature>, <CalKit>

<YYYY/MM/DD>

Date Time

<HH:MM:SS>

<Type>

{RT|RO|RS|F1|OP|F2|F3|F4}

<TypeEx>

{SOLT|SOLR|TRL|COPY}

<IFBW>

IFBW value

<Power>

Power level

<b>&lt;Temperature&gt;</b>	Temperature
<b>&lt;CalKit&gt;</b>	Calibration Kit Label and Description

### Equivalent Softkeys

None

### Equivalent COM Command

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

### Syntax

CallInfo = app.SCPI.SENSE(Ch).CORRection.INFormation (rPt, sPt)

### Type

String (read only)>

Back to [SENSe](#)

## SENS:CORR:IMP

### SCPI Command

SENSe:CORRection:IMPedance[:INPut][:MAGNitude] <numeric>

SENSe:CORRection:IMPedance[:INPut][:MAGNitude]?

### Description

Sets or reads out the system impedance Z0 of all analyzer ports.

command/query

### Parameter

<numeric> the Z0 value from 0.001 to 1000.

### Unit

$\Omega$  (Ohm)

### Out of Range

Sets the value of the limit, which is closer to the specified value.

### Query Response

<numeric>

### Preset Value

50  $\Omega$

### Equivalent Softkeys

Calibration > System Z0

---

## Equivalent COM Command

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

## Syntax

Value = app.SCPI.SENSE.CORRection.IMPedance.INPut.MAGNitude

app.SCPI.SENSE.CORRection.IMPedance.INPut.MAGNitude = 50

## Type

Double (read/write)

---

### **WARNING**

Object SENSE has an index of 1, which can be omitted in Visual Basic but it cannot be omitted in other programming languages.

---

Back to [SENSe](#)

# SENS:CORR:IMP:SEL:AUTO

## SCPI Command

SENSe:CORRection:IMPedance[:INPut]:SELection:AUTO {OFF|ON|0|1}

SENSe:CORRection:IMPedance[:INPut]:SELection:AUTO?

## Description

Turns the auto-select Z0 function ON/OFF. When enabled the function sets the port impedance Z0 to the corresponding value of measuring calibration standard.

command/query

## Parameter

- {ON|1}

ON
- {OFF|0}

OFF

## Query Response

{0|1}

## Preset Value

0

## Equivalent Softkeys

Calibration > System Z0> Auto Select Z0

---

## Equivalent COM Command

None

---

Back to [SENSe](#)

## SENS:CORR:OFFS:CLE

### SCPI Command

SENSe<Ch>:CORRection:OFFSet:CLEar

### Description

Clears the scalar mixer calibration coefficient table.

no query

### Target

Channel <Ch>,

<Ch>={{1}|2|...16}

### Equivalent Softkeys

None

---

### Equivalent COM Command

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

### Syntax

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

### Type

Method

---

Back to [SENSe](#)



## SENS:CORR:OFFS:COLL:CLE

### SCPI Command

SENSe<Ch>:CORRection:OFFSet:COLLect:CLEar

### Description

Clears the calibration measurement data of scalar mixer calibration when the frequency offset feature is ON.

no query

### Target

Channel <Ch>,

<Ch>={[1]|2|...16}

### Equivalent Softkeys

Calibration > Mixer/Converter Calibration > Scalar Mixer Calibration > Cancel  
> Yes

---

### Equivalent COM Command

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

### Syntax

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

### Type

Method

---

Back to [SENSe](#)

## SENS:CORR:OFFS:COLL:DIR

### SCPI Command

SENSe<Ch>:CORRection:OFFSet:COLLect:DIRection {FORWard|REVerse|BOTH}

SENSe<Ch>:CORRection:OFFSet:COLLect:DIRection?

### Description

Specifies the direction of the scalar mixer calibration: forward, reverse or both. The command affects the graphic user interface only. The *forward* setting enables the power measuring button for the first port. The *reverse* setting enables the power measuring button for the second port. The *both* setting enables the power measuring button for both ports.

**Note:** The command does not affect the SCPI calibration procedure. The actual direction of the scalar mixer calibration depend on the [SENS:CORR:OFFS:COLL:PMET](#) command usage. If the command was executed once the direction will be forward or reverse depending on the port number parameter. If the command was executed twice for both ports the direction will be both.

command/query

### Target

Channel <Ch>,

<Ch>={[1]|2|...16}

### Parameter

**FORW** Forward direction

**REV** Reverse direction

**BOTH** Both directions

### Query Response

{FORW|REV|BOTH}

### Preset Value

FORW

### Equivalent Softkeys

**Calibration > Mixer/Converter Calibration > Scalar Mixer Calibration > Direction {Forward | Reverse | Both}**

---

### Equivalent COM Command

None

---

Back to [SENSe](#)

## SENS:CORR:OFFS:COLL:ECAL

### SCPI Command

SENSe<Ch>:CORRection:OFFSet:COLLect[:ACQuire]:ECAL  
<numeric1>,<numeric2>

### Description

Measures the calibration data of all reflection standards of the ACM on the specified port when the frequency offset feature is on for scalar mixer calibration. Use ACM measurement instead of measuring three mechanical standards: OPEN, SHORT, LOAD.

**Note:** The command starts the measurement immediately independent on the trigger settings.

no query

### Target

Channel <Ch>,

<Ch>={1|2|...16}

### Parameter

<numeric1>      Measurement port number

<numeric2>      Number of the second port of the SMC port pair

### Out of Range

If an incorrect port number is specified, an error occurs.

## Equivalent Softkeys

Calibration > Mixer/Converter Calibration > Scalar Mixer Calibration > Measure Using ACM

---

## Equivalent COM Command

None

---

Back to [SENSe](#)

## SENS:CORR:OFFS:COLL:LOAD

### SCPI Command

SENSe<Ch>:CORRection:OFFSet:COLLect[:ACQuire]:LOAD  
<numeric1>,<numeric2>

### Description

Measures the calibration data of the load standard of the specified port when the frequency offset feature is on for scalar mixer calibration.

**Note:** 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 blocks the execution of the subsequent commands until the completion of the measurement.

no query

### Target

Channel <Ch>,

<Ch>={[1]|2|...16}

### Parameter

<numeric1>      Measurement port number

<numeric2>      Number of the second port of the SMC port pair

### Out of Range

If an incorrect port number is specified, an error occurs.

### Equivalent Softkeys

**Calibration > Mixer/Converter Calibration > Scalar Mixer Calibration > Reflection Port n > Port n Load**

---

### Equivalent COM Command

SCPI.SENSe(Ch).CORRection.OFFSet.COLLect.ACQuire.LOAD

**Syntax**

app.SCPI.SENSE(CH).CORREction.OFFSet.COLlect.ACQuire.LOAD= Array(1, 2)

**Type**

Variant (array of Long) (write only)

---

Back to [SENSe](#)

## SENS:CORR:OFFS:COLL:METH:SMIX2

### SCPI Command

SENSe<Ch>:CORRection:OFFSet:COLLect:METHod:SMIX2  
<numeric1>,<numeric2>

### Description

Selects the ports and sets the scalar mixer calibration type when the frequency offset feature is on for the calculation of the calibration coefficients on completion of the calibration executed by the [SENS:CORR:OFFS:COLL:SAVE](#) command.

no query

### Target

Channel <Ch>,

<Ch>={{1}|2|...16}

### Parameter

<numeric1>      First port

<numeric2>      Second port

### Out of Range

If an incorrect port number is specified, an error occurs.

### Equivalent Softkeys

None

---

### Equivalent COM Command

SCPI.SENSe(Ch).CORRection.OFFSet.COLLect.METHod.SMIX2

### Syntax

app.SCPI.SENSe(Ch).CORRection.OFFSet.COLLect.METHod.SMIX2 = Array(2, 1)



## Type

Variant (array of Long) (write only)

---

Back to [SENSe](#)

## SENS:CORR:OFFS:COLL:OPEN

### SCPI Command

SENSe<Ch>:CORRection:OFFSet:COLLect[:ACQuire]:OPEN  
<numeric1>,<numeric2>

### Description

Measures the calibration data of the open standard of the specified port when the frequency offset feature is on for scalar mixer calibration.

**Note:** 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 blocks the execution of the subsequent commands until the completion of the measurement.

no query

### Target

Channel <Ch>,

<Ch>={[1]2|...16}

### Parameter

<numeric1>      Measurement port number

<numeric2>      Number of the second port of the SMC port pair

### Out of Range

If an incorrect port number is specified, an error occurs.

### Equivalent Softkeys

**Calibration > Mixer/Converter Calibration > Scalar Mixer Calibration > Reflection Port n > Port n Open**

---

### Equivalent COM Command

SCPI.SENSe(Ch).CORRection.OFFSet.COLLect.ACQuire.OPEN

**Syntax**

app.SCPI.SENSE(CH).CORRection.OFFSet.COLLect.ACQuire.OPEN = Array(1, 2)

**Type**

Variant (array of Long) (write only)

---

Back to [SENSe](#)

## SENS:CORR:OFFS:COLL:PMET

### SCPI Command

SENSe<Ch>:CORRection:OFFSet:COLLect[:ACQuire]:PMETer  
<numeric1>,<numeric2>,<numeric3>

### Description

Measures the scalar-mixer calibration data using the power meter when the frequency offset feature is ON.

**Note:** The command starts the measurement of the calibration data immediately regardless the trigger settings. The command blocks the execution of the subsequent commands until the completion of the measurement..

no query

### Target

Channel <Ch>,

<Ch>={{1}|2|...16}

### Parameter

- |            |  |
|------------|--|
| <numeric1> | Measurement port number                        |
| <numeric2> | Number of the second port of the SMC port pair |
| <numeric3> | Always 0 (reserved)                            |

### Equivalent Softkeys

Calibration > Mixer/Converter Calibration > Scalar Mixer Calibration > Power > Port n

---

### Equivalent COM Command

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

**Syntax**

```
app.SCPI.SENSE(CH).CORRECTION.OFFSET.COLLECT.ACQUIRE.PMETER = Array(1, 2, 0)
```

**Type**

Variant (array of Long) (write only)

---

Back to [SENSE](#)

## SENS:CORR:OFFS:COLL:SHOR

### SCPI Command

SENSe<Ch>:CORRection:OFFSet:COLLect[:ACQuire]:SHORT  
<numeric1>,<numeric2>

### Description

Measures the calibration data of the short standard of the specified port when the frequency offset feature is on for scalar mixer calibration.

**Note:** 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 blocks the execution of the subsequent commands until the completion of the measurement.

no query

### Target

Channel <Ch>,

<Ch>={[1]|2|...16}

### Parameter

<numeric1>            Measurement port number

<numeric2>            Number of the second port of the SMC port pair

### Equivalent Softkeys

**Calibration > Mixer/Converter Calibration > Scalar Mixer Calibration > Reflection Port n > Port n Short**

---

### Equivalent COM Command

SCPI.SENSE(CH).CORRection.OFFSet.COLLect.ACQuire.SHORT

### Syntax

app.SCPI.SENSE(CH).CORRection.OFFSet.COLLect.ACQuire.SHORT = Array(1, 2)

## Type

Variant (array of Long) (write only)

---

Back to [SENSe](#)

## SENS:CORR:OFFS:COLL:THRU

### SCPI Command

SENSe<Ch>:CORRection:OFFSet:COLLect[:ACQuire]:THRU  
<numeric1>,<numeric2>

### Description

Measures the calibration data of the thru standard of the specified port when the frequency offset feature is on for scalar mixer calibration.

**Note:** 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 blocks the execution of the subsequent commands until the completion of the measurement..

no query

### Target

Channel <Ch>,

<Ch>={[1]2|...16}

### Parameter

<numeric1>      Response port number

<numeric2>      Stimulus port number

### Out of Range

If an incorrect port number is specified, an error occurs.

### Equivalent Softkeys

**Calibration > Mixer/Converter Calibration > Scalar Mixer Calibration > Reflection Port n > Port n-m Thru**

---

### Equivalent COM Command

SCPI.SENSe(Ch).CORRection.OFFSet.COLLect.ACQuire.THRU



**Syntax**

app.SCPI.SENSE(CH).CORREction.OFFSet.COLlect.ACQuire.THRU = Array(1, 2)

**Type**

Variant (array of Long) (write only)

---

Back to [SENSe](#)

## **SENS:CORR:OFFS:COLL:SAVE**

### **SCPI Command**

SENSe<Ch>:CORRection:OFFSet:COLLect:SAVE

### **Description**

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.

On completion of the command, all the calibration standards measurements are cleared and the error correction automatically turns ON.

If this command is executed before all necessary calibration data for calculating the calibration coefficient is measured, an error occurs when executed.

no query

### **Target**

Channel <Ch>,

<Ch>={[1]|2|...16}

### **Related Commands**

[SENS:CORR:OFFS:COLL:METH:SMIX2](#)

[SENS:CORR:OFFS:COLL:LOAD](#)

[SENS:CORR:OFFS:COLL:OPEN](#)

[SENS:CORR:OFFS:COLL:SHOR](#)

[SENS:CORR:OFFS:COLL:THRU](#)

[SENS:CORR:OFFS:COLL:PMETer](#)

### **Equivalent Softkeys**

**Calibration > Mixer/Converter Calibration > Scalar Mixer Calibration > Apply**

---

**Equivalent COM Command**

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

**Syntax**

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

**Type**

Method

---

Back to [SENSe](#)

## SENS:CORR:PORT:IMP

### SCPI Command

SENSe:CORRection:PORT<Pt>:IMPedance[:INPut][:MAGNitude] <numeric>

SENSe:CORRection:PORT<Pt>:IMPedance[:INPut][:MAGNitude]?

### Description

Sets or reads out the impedance Z0 of port <Pt>

command/query

### Target

Port <Pt>,

<Pt>={1|2}

### Parameter

<numeric> the Z0 value from 0.001 to 1000

### Unit

$\Omega$  (Ohm)

### Out of Range

Sets the value of the limit, which is closer to the specified value.

### Query Response

<numeric>

### Preset Value

50  $\Omega$

### Equivalent Softkeys

Calibration > System Z0 > Port n Z0

---

## Equivalent COM Command

None

---

Back to [SENSe](#)

## SENS:CORR:REC

### SCPI Command

SENSe<Ch>:CORRection:RECeiver<Pt>[:STATe] {OFF|ON|0|1}

SENSe<Ch>:CORRection:RECeiver<Pt>[:STATe]?

### Description

Turns the receiver correction of the specified port ON/OFF.

command/query

### Target

Port <Pt> of channel <Ch>,

<Ch>={{1}|2|...16}

<Pt>={{1}|2}

### Parameter

{ON|1}      ON

{OFF|0}     OFF

### Query Response

{0|1}

### Preset Value

0

### Equivalent Softkeys

Calibration > Receiver Calibration > Correction

---

### Equivalent COM Command

SCPI.SENSe(Ch).CORRection.RECeiver(Pt).STATe

**Syntax**

Status = app.SCPI.SENSE(CH).CORRection.RECeiver(Pt).STATe

app.SCPI.SENSE(CH).CORRection.RECeiver(Pt).STATe = True

**Type**

Boolean (read/write)

---

Back to [SENSe](#)

## SENS:CORR:REC:COLL:ACQ

### SCPI Command

SENSe<Ch>:CORRection:RECeiver<Pt>:COLLect:ACQuire <srcport>

### Description

Executes receiver calibration of both the test receiver and the reference receiver of the specified port <Pt>. The test receiver calibration uses port number <srcport> as the source port. The reference receiver calibration uses its own port <Pt> as the source port.

**Note:** The command starts the measurement of the calibration data immediately regardless the trigger settings. The command blocks the execution of the subsequent commands until the completion of the measurement.

no query

### Target

Port <Pt> of channel <Ch>,

<Ch>={[1]|2|...16}

<Pt>={[1]|2}

### Parameter

<srcport>      The number of the source port from 1 to 2 (or 4)

### Equivalent Softkeys

Calibration > Receiver Calibration > Calibrate Both

---

### Equivalent COM Command

SCPI.SENSE(CH).CORRection.RECeiver(Pt).COLLect.ACQuire

### Syntax

app.SCPI.SENSE(CH).CORRection.RECeiver(Pt).COLLect.ACQuire = Src



**Type**

Long (write only)

---

Back to [SENSe](#)

## SENS:CORR:REC:COLL:RCH:ACQ

### SCPI Command

SENSe<Ch>:CORRection:RECeiver<Pt>:COLLect:RCHannel:ACQuire

### Description

Executes receiver calibration of the reference receiver of the specified port <Pt>. The reference receiver calibration uses its own port <Pt> as the source port.

**Note:** The command starts the measurement of the calibration data immediately regardless the trigger settings. The command blocks the execution of the subsequent commands until the completion of the measurement.

no query

### Target

Port <Pt> of channel <Ch>,

<Ch>={{[1]|2|...16}}

<Pt>={{[1]|2}}

### Equivalent Softkeys

**Calibration > Receiver Calibration > Calibrate Reference Receiver**

---

### Equivalent COM Command

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

### Syntax

app.SCPI.SENSe(Ch).CORRection.RECeiver(Pt).COLLect.RCHannel.ACQuire

### Type

Method

---

Back to [SENSe](#)

## SENS:CORR:REC:COLL:TCH:ACQ

### SCPI Command

SENSe<Ch>:CORRection:RECeiver<Pt>:COLLect:TCHannel:ACQuire <srcport>

### Description

Executes receiver calibration of the test receiver of the specified port <Pt>. The test receiver calibration uses port number <srcport> as the source port.

**Note:** The command starts the measurement of the calibration data immediately regardless the trigger settings. The command blocks the execution of the subsequent commands until the completion of the measurement.

no query

### Target

Port <Pt> of channel <Ch>,

<Ch>={{[1]|2|...16}}

<Pt>={{[1]|2}}

### Parameter

<srcport>      The number of the source port from 1 to 2

### Equivalent Softkeys

**Calibration > Receiver Calibration > Calibrate Test Receiver**

---

### Equivalent COM Command

SCPI.SENSE(CH).CORRection.RECeiver(Pt).COLLect.TCHannel.ACQuire

### Syntax

app.SCPI.SENSE(CH).CORRection.RECeiver(Pt).COLLect.TCHannel.ACQuire      =  
Src

**Type**

Long (write only)

---

Back to [SENSe](#)

## **SENS:CORR:REC:OFFS:AMPL**

### **SCPI Command**

SENSe<Ch>:CORRection:RECeiver<Pt>:OFFSET:AMPLitude <numeric>

SENSe<Ch>:CORRection:RECeiver<Pt>:OFFSET:AMPLitude?

### **Description**

Sets or reads out the power offset value when the Receiver Calibration is performed. Receiver calibration is done at the condition of <source power> + <power offset>.

command/query

### **Target**

Port <Pt> of channel <Ch>,

<Ch>={[1]|2|...16}

<Pt>={[1]|2}

### **Parameter**

<numeric> the power offset value when the Receiver Calibration is performed from -100 to 100.

### **Unit**

dBm (decibels above 1 milliwatt)

### **Out of Range**

Sets the value of the limit, which is closer to the specified value.

### **Query Response**

<numeric>

### **Preset Value**

0.0

## Equivalent Softkeys

Calibration > Receiver Calibration > Power Offset

---

## Equivalent COM Command

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

## Syntax

Offset = app.SCPI.SENSE(Ch).CORRection.RECeiver(Pt).OFFSet.AMPLitude

app.SCPI.SENSE(Ch).CORRection.RECeiver(Pt).OFFSet.AMPLitude = -10

## Type

Double (read/write)

---

Back to [SENSe](#)

## SENS:CORR:STAT

### SCPI Command

SENSe<Ch>:CORRection:STATe {OFF|ON|0|1}

SENSe<Ch>:CORRection:STATe?

### Description

Turns the S-parameter error correction ON/OFF.

command/query

### Target

Channel <Ch>,

<Ch>={{1}|2|...16}

### Parameter

{ON|1}      ON

{OFF|0}      OFF

### Query Response

{0|1}

### Preset Value

0

### Equivalent Softkeys

### Calibration > Correction

---

### Equivalent COM Command

SCPI.SENSE(CH).CORRection.STATe

**Syntax**

Status = app.SCPI.SENSE(CH).CORRection.STATe

app.SCPI.SENSE(CH).CORRection.STATe = True

**Type**

Boolean (read/write)

---

Back to [SENSe](#)



## **SENS:CORR:TRAN:TIME:FREQ**

### **SCPI Command**

SENSe<Ch>:CORRection:TRANSform:TIME:FREQuency <frequency>

SENSe<Ch>:CORRection:TRANSform:TIME:FREQuency?

### **Description**

Sets or reads out the frequency value at which the cable loss is specified for the cable correction function when the time domain transformation function is turned on.

command/query

### **Target**

Channel <Ch>,

<Ch>={[1]|2|...16}

### **Parameter**

<frequency> the frequency value.

### **Unit**

Hz (Hertz)

### **Query Response**

<numeric>

### **Preset Value**

1 GHz

### **Equivalent Softkeys**

**Analysis > Time Domain > Cable Correction > Frequency**

---

### Equivalent COM Command

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

### Syntax

Value = app.SCPI.SENSE(Ch).CORRection.TRANSform.TIME.FREQuency

app.SCPI.SENSE(Ch).CORRection.TRANSform.TIME.FREQuency = 1E9

### Type

Double (read/write)

---

Back to [SENSe](#)

## SENS:CORR:TRAN:TIME:LOSS

### SCPI Command

SENSe<Ch>:CORRection:TRANsform:TIME:LOSS <numeric>

SENSe<Ch>:CORRection:TRANsform:TIME:LOSS?

### Description

Sets or reads out the cable loss value for the cable correction function when the time domain transformation function is turned ON.

command/query

### Target

Channel <Ch>,

<Ch>={[1]|2|...16}

### Parameter

<numeric> the cable loss value

### Unit

dB/m (decibell / meter)

### Query Response

<numeric>

### Preset Value

0 dB/m

### Equivalent Softkeys

Analysis > Time Domain > Cable Correction > Cable Loss

---

### Equivalent COM Command

SCPI.SENSe(Ch).CORRection.TRANsform.TIME.LOSS

**Syntax**

Value = app.SCPI.SENSE(CH).CORRection.TRANSform.TIME.LOSS

app.SCPI.SENSE(CH).CORRection.TRANSform.TIME.LOSS = 1.4

**Type**

Double (read/write)

---

Back to [SENSe](#)

## **SENS:CORR:TRAN:TIME:RVEL**

### **SCPI Command**

SENSe<Ch>:CORRection:TRANsform:TIME:RVELocity <numeric>

SENSe<Ch>:CORRection:TRANsform:TIME:RVELocity?

### **Description**

Sets or reads out the cable relative wave speed velocity for the cable correction function, when the time domain transformation function is turned ON.

command/query

### **Target**

Channel <Ch>,

<Ch>={[1]|2|...16}

### **Parameter**

<numeric> the cable velocity factor

### **Query Response**

<numeric>

### **Preset Value**

1.0

### **Equivalent Softkeys**

**Analysis > Time Domain > Cable Correction > Velocity Factor**

---

### **Equivalent COM Command**

SCPI.SENSe(Ch).CORRection.TRANsform.TIME.RVELocity

**Syntax**

Value = app.SCPI.SENSE(CH).CORRection.TRANSform.TIME.RVELOCITY

app.SCPI.SENSE(CH).CORRection.TRANSform.TIME.RVELOCITY = 0.66

**Type**

Double (read/write)

---

Back to [SENSe](#)

## SENS:CORR:TRAN:TIME:STAT

### SCPI Command

SENSe<Ch>:CORRection:TRANsform:TIME:STATe {OFF|ON|0|1}

SENSe<Ch>:CORRection:TRANsform:TIME:STATe?

### Description

Turns the cable correction ON/OFF when the time domain transformation function is turned ON.

command/query

### Target

Channel <Ch>,

<Ch>={[1]|2|...16}

### Parameter

{ON|1}      ON

{OFF|0}      OFF

### Query Response

{0|1}

### Preset Value

0

### Equivalent Softkeys

Analysis > Time Domain > Cable Correction > Cable Correction

---

### Equivalent COM Command

SCPI.SENSe(Ch).CORRection.TRANsform.TIME.STATe

## Syntax

Status = app.SCPI.SENSE(CH).CORRection.TRANSform.TIME.STATe

app.SCPI.SENSE(CH).CORRection.TRANSform.TIME.STATe = True

## Type

Boolean (read/write)

---

Back to [SENSe](#)



## SENS:CORR:TRIG:FREE

### SCPI Command

SENSe<Ch>:CORRection:TRIGger:FREE[:STATe] {OFF|ON|0|1}

SENSe<Ch>:CORRection:TRIGger:FREE[:STATe]?

### Description

Enables/disables the internal trigger source for calibration. If the internal 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 system trigger source is used (which is set for regular measurement with the command [TRIG:SOUR](#)) to start the calibration standard measurement.

The system trigger source also enables the averaging trigger function ([TRIG:AVER](#)) and the point trigger function ([TRIG:POIN](#)) for calibration.

**Note:** When the system 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.

command/query

### Target

Channel <Ch>,

<Ch>={1|2|...16}

### Parameter

Specifies the trigger source for calibration:

{ON|1}      Internal

{OFF|0}     System

## Query Response

{0|1}

## Preset Value

1

## Equivalent Softkeys

**Calibration > Cal Trig Source {Internal | System}**

---

## Equivalent COM Command

SCPI.SENSE(Ch).CORRection.TRIGger.FREE.STATe

## Syntax

Status = app.SCPI.SENSE(Ch).CORRection.TRIGger.FREE.STATe

app.SCPI.SENSE(Ch).CORRection.TRIGger.FREE.STATe = True

## Type

Boolean (read/write)

---

Back to [SENSe](#)

## SENS:CORR:TYPE?

### SCPI Command

SENSe<Ch>:CORRection:TYPE<Tr>?

### Description

Reads the information about the calibration type and the number of ports to which the calibration is applied for the specified trace. The response format is as follows.

query only

### Target

Trace <Tr> of channel <Ch>,

<Tr>={[1]|2|...16}

<Ch>={[1]|2|...16}

### Query Response

<Type>,<Port1>...,<PortN>

Where <Type> is:

<b>RESPO</b>	Response (Open)
<b>RESPS</b>	Response (Short)
<b>RESPT</b>	Response (Thru)
<b>SOLT1</b>	Full one-port calibration
<b>SOLT2</b>	Full two-port calibration
<b>1PATH</b>	One path two-port calibration
<b>NONE</b>	Not defined

## Equivalent Softkeys

None

---

## Equivalent COM Command

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

## Syntax

CallInfo = app.SCPI.SENSE(Ch).CORRection.TYPE(Tr)

## Type

Variant: array of Variants (read only)

---

Back to [SENSe](#)

# SENS:CORR:VMC:COLL:ECAL:SAVE

## SCPI Command

SENSe<Ch>:CORRection:OFFSet:COLLect:ECAL:SAVE <string>

## Description

Measures ACM and completes the vector mixer calibration procedure. Calculates S-parameters of the calibration mixer + filter and writes them to a touchstone file. If the setup option is turned ON by the [SENS:CORR:VMC:COLL:OPT](#) command then turns on the de-embedding S-parameters of the calibration mixer + filter. Use ACM measurement instead of measuring three mechanical standards: OPEN, SHORT, LOAD.

no query

## Target

Channel <Ch>,

<Ch>={[1]|2|...16}

## Parameter

<string> destination file name (optional). If parameter is omitted the file name 'vmctemp.S2P' is used.

## Related Commands

[SENS:CORR:VMC:COLL:OPT](#)

## Equivalent Softkeys

**Calibration > Mixer/Converter Calibration > Vector Mixer Calibration > Execute Using ACM**

---

## Equivalent COM Command

None

---

Back to [SENSe](#)

## SENS:CORR:VMC:COLL:PORT

### SCPI Command

SENSe<Ch>:CORRection:VMC:COLLect:PORT <numeric>

SENSe<Ch>:CORRection:VMC:COLLect:PORT?

### Description

Sets or reads out the number of the port used in the vector mixer calibration. The calibration mixer with the LPF filter is connected to this port.

command/query

### Target

Channel <Ch>,

<Ch>={[1]|2|...16}

### Parameter

<numeric> port number from 1 to 2

### Query Response

<numeric>

### Preset Value

1

### Equivalent Softkeys

Calibration > Mixer/Converter Calibration > Vector Mixer Calibration > Select Port

---

### Equivalent COM Command

None

---

Back to [SENSe](#)

## **SENS:CORR:VMC:COLL:LO:FREQ**

### **SCPI Command**

SENSe<Ch>:CORRection:VMC:COLLect:LO:FREQuency <numeric>

SENSe<Ch>:CORRection:VMC:COLLect:LO:FREQuency?

### **Description**

Sets or reads out the LO frequency value used in the vector mixer calibration. The LO source is an external signal generator. The LO frequency is common for both the calibration and the mixer under test.

command/query

### **Target**

Channel <Ch>,

<Ch>={1|2|...16}

### **Parameter**

<numeric> LO frequency from 0 to 1000 THz

### **Query Response**

<numeric>

### **Unit**

Hz (Hertz)

### **Preset Value**

0

### **Equivalent Softkeys**

**Calibration > Mixer/Converter Calibration > Vector Mixer Calibration > LO Frequency**

---

### Equivalent COM Command

None

---

Back to [SENSe](#)



## SENS:CORR:VMC:COLL:IF:SEL

### SCPI Command

SENSe<Ch>:CORRection:VMC:COLLect:IF:SElect <char>

SENSe<Ch>:CORRection:VMC:COLLect:IF:SElect?

### Description

Selects the IF frequency from RF+LO, RF-LO and LO-RF, depending on the IF frequency of the calibration mixer in the vector mixer calibration.

command/query

### Target

Channel <Ch>,

<Ch>={[1]|2|...16}

### Parameter

<char> Select from following:

**RFPLO**      RF + LO

**RFMLO**      RF - LO

**LOMRF**      LO - RF

### Query Response

{RFPLO|RFMLO|LOMRF}

### Preset Value

RFPLO

### Equivalent Softkeys

**Calibration > Mixer/Converter Calibration > Vector Mixer Calibration > IF Frequency > {RF+LO | RF - LO | LO - RF}**

---

### Equivalent COM Command

None

---

Back to [SENSe](#)

## SENS:CORR:VMC:COLL:LOAD

### SCPI Command

SENSe<Ch>:CORRection:VMC:COLLect[:ACQuire]:LOAD

### Description

Measures the load standard in order to characterize the calibration mixer + filter in the vector mixer calibration.

**Note:** 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 blocks the execution of the subsequent commands until the completion of the measurement.

no query

### Target

Channel <Ch>,

<Ch>={1|2|...16}

### Equivalent Softkeys

Calibration > Mixer/Converter Calibration > Vector Mixer Calibration > Load

---

### Equivalent COM Command

None

---

Back to [SENSe](#)

## SENS:CORR:VMC:COLL:OPEN

### SCPI Command

SENSe<Ch>:CORRection:VMC:COLLect[:ACQuire]:OPEN

### Description

Measures the open standard in order to characterize the calibration mixer + filter in the vector mixer calibration.

**Note:** 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 blocks the execution of the subsequent commands until the completion of the measurement.

no query

### Target

Channel <Ch>,

<Ch>={1|2|...16}

### Equivalent Softkeys

Calibration > Mixer/Converter Calibration > Vector Mixer Calibration > Open

---

### Equivalent COM Command

None

---

Back to [SENSe](#)

## SENS:CORR:VMC:COLL:SHOR

### SCPI Command

SENSe<Ch>:CORRection:VMC:COLLEct[:ACQuire]:SHORT

### Description

Measures the short standard in order to characterize the calibration mixer + filter in the vector mixer calibration.

**Note:** 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 blocks the execution of the subsequent commands until the completion of the measurement.

no query

### Target

Channel <Ch>,

<Ch>={1|2|...16}

### Equivalent Softkeys

Calibration > Mixer/Converter Calibration > Vector Mixer Calibration > Short

---

### Equivalent COM Command

None

---

Back to [SENSe](#)

## SENS:CORR:VMC:COLL:OPT

### SCPI Command

SENSe<Ch>:CORRection:VMC:COLLect[:SETup]:OPTion {OFF|ON|0|1}

SENSe<Ch>:CORRection:VMC:COLLect[:SETup]:OPTion?

### Description

Turns the setup option in the vector mixer calibration ON/OFF. This option forces the de-embedding S-parameters of the calibration mixer + filter when the S-parameters have been calculated and written to the touchstone file.

command/query

### Target

Channel <Ch>,

<Ch>={[1]|2|...16}

### Parameter

{ON|1}      ON

{OFF|0}      OFF

### Query Response

{0|1}

### Preset Value

0

### Equivalent Softkeys

**Calibration > Mixer/Converter Calibration > Vector Mixer Calibration > Setup Option**

---

## Equivalent COM Command

None

---

Back to [SENSe](#)

## SENS:CORR:VMC:COLL:SAVE

### SCPI Command

SENSe<Ch>:CORRection:OFFSet:COLLect:SAVE <string>

### Description

Completes the vector mixer calibration procedure. Calculates S-parameters of the calibration mixer + filter and writes them to a touchstone file. If the setup option is turned ON by the [SENS:CORR:VMC:COLL:OPT](#) command then turns on the de-embedding S-parameters of the calibration mixer + filter.

no query

### Target

Channel <Ch>,

<Ch>={[1]|2|...16}

### Parameter

<string> destination file name (optional). If parameter is omitted the file name 'vmctemp.S2P' is used.

### Related Commands

[SENS:CORR:VMC:COLL:OPT](#)

### Equivalent Softkeys

**Calibration > Mixer/Converter Calibration > Vector Mixer Calibration > Save to Touchstone File**

---

### Equivalent COM Command

None

---

Back to [SENSe](#)



## SENS:DATA:CORR?

### SCPI Command

SENSe<Ch>:DATA:CORRdata? <char>

### Description

Reads out the corrected S-parameter data array or the corrected receiver data array. The type of the array entries is a complex number.

The array size is  $2N$ , where  $N$  is the number of measurement points. For the  $n$ -th point, where  $n$  from 1 to  $N$ :

<numeric  $2n-1$ > the real part of corrected measurement

<numeric  $2n$ > the imaginary part of corrected measurement

**Note:** To ensure the update of the data, the corresponding stimulus port must be active. For example, when reading the S12 parameter at least one trace with the stimulus port 2 must present or SOLT2 calibration must be active.

query only

### Target

Channel <Ch>,

<Ch>={[1]|2|...16}

### Parameter

<char> Specifies the S-parameter:

**S11, S12, S13, S14, S21, ... S44**

<char> Specifies the Test Receiver:

**T11, T12, T13, T14, T21, ... T44**

where the first index is the receiver port number, and the second index is the source port number.

The following notations are also available:

**T1(1), T1(2), T1(3), T1(4), T2(1), ... T4(4)**

or

**A(1), A(2), A(3), A(4), B(1), ... D(4)**

<char> Specifies the  
Reference Receiver:

**R11, R12, R13, R14, R21, ... R44**

where the first index is the receiver port number, and  
the second index is the source port number.

The following notations are also available:

**R1(1), R1(2), R1(3), R1(3), R2(1), ... R4(4)**

### Query Response

<numeric 1>, <numeric 2>, ...<numeric 2N>

### Equivalent Softkeys

None

---

### Equivalent COM Command

SCPI.SENSE(ch).DATA.CORRdata(Param)

### Syntax

Data = app.SCPI.SENSE(ch).DATA.CORRdata("S11")

### Type

Variant (array of Double) (read only)

---

Back to [SENSE](#)

## SENS:DATA:RAWD?

### SCPI Command

SENSe<Ch>:DATA:RAWData? <char>

### Description

Reads out the raw S-parameter data array or the raw receiver data array. The type of the array entries is a complex number.

The array size is  $2N$ , where  $N$  is the number of measurement points. For the  $n$ -th point, where  $n$  from 1 to  $N$ :

<numeric  $2n-1$ > the real part of raw measurement;

<numeric  $2n$ > the imaginary part of raw measurement.

**Note:** To ensure the update of the data, the corresponding stimulus port must be active. For example, when reading the S12 parameter at least one trace with the stimulus port 2 must present or SOLT2 calibration must be active.

query only

### Target

Channel <Ch>,

<Ch>={[1]|2|...16}

### Parameter

<char> Specifies the S-parameter:

**S11, S21, S12, S22**

<char> Specifies the Test Receiver:

**A(1), A(2), B(1), B(2)**

where the first index is the receiver port number, and the second index is the source port number.

The following notations are also available:

**T1(1), T1(2), T2(1), T2(2)**

or

	<b>T11, T12, T21, T22,</b>
<char> Specifies the Reference Receiver:	<b>R11, R21, R12, R22</b>
	where the first index is the receiver port number, and the second index is the source port number.
	The following notations are also available:
	<b>R1(1), R2(1), R1(2), R2(2)</b>

### Query Response

<numeric 1>, <numeric 2>, ...<numeric 2N>

### Equivalent Softkeys

None

---

### Equivalent COM Command

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

### Syntax

Data = app.SCPI.SENSE(ch).DATA.RAWData("S11")

### Type

Variant (array of Double) (read only)

---

Back to [SENSe](#)

## SENS:FREQ

### SCPI Command

SENSe<Ch>:FREQuency[:CW] <frequency>

SENSe<Ch>:FREQuency[:FIXed] <frequency>

SENSe<Ch>:FREQuency[:CW]?

SENSe<Ch>:FREQuency[:FIXed]?

### Description

Sets or reads out the fixed frequency value when the power sweep type is selected.

command/query

### Target

Channel <Ch>,

<Ch>={[1]|2|...16}

### Parameter

<frequency> the frequency value within the frequency limits of the analyzer.

### Unit

Hz (Hertz)

### Out of Range

Sets the value of the limit, which is closer to the specified value.

### Query Response

<numeric>

### Preset Value

The minimum frequency limit of the analyzer.

## Equivalent Softkeys

Stimulus > Power > CW Freq

---

## Equivalent COM Command

SCPI.SENSE(Ch).FREQUENCY.CW

## Syntax

Value = app.SCPI.SENSE(Ch).FREQUENCY.CW

app.SCPI.SENSE(Ch).FREQUENCY.CW = 1E9

## Type

Double (read/write)

---

Back to [SENSe](#)

## SENS:FREQ:DATA?

### SCPI Command

SENSe<Ch>:FREQuency:DATA?

### Description

Reads out the frequency array of the measurement points.

The array size is N, where N is the number of measurement points.

For the n–th point, where n from 1 to N:

<numeric n>            the frequency value at the n–th measurement point

query only

### Target

Channel <Ch>,

<Ch>={[1]2|...16}

### Query Response

<numeric 1>, <numeric 2>, ...<numeric N>

### Equivalent Softkeys

None

---

### Equivalent COM Command

SCPI.SENSe(Ch).FREQuency.DATA

### Syntax

Data = app.SCPI.SENSe(Ch).FREQuency.DATA

## Type

Variant (array of Double) (read only)

---

Back to [SENSe](#)



## **SENS:FREQ:CEN**

### **SCPI Command**

SENSe<Ch>:FREQuency:CENTer <frequency>

SENSe<Ch>:FREQuency:CENTer?

### **Description**

Sets or reads out the stimulus center value of the sweep range for linear or logarithmic sweep type.

command/query

### **Target**

Channel <Ch>,

<Ch>={1|2|...16}

### **Parameter**

<frequency> the stimulus center value within the frequency limits of the analyzer.

### **Unit**

Hz (Hertz)

### **Out of Range**

Sets the value of the limit, which is closer to the specified value.

### **Query Response**

<numeric>

### **Preset Value**

The center frequency of the analyzer

### **Equivalent Softkeys**

**Stimulus > Center**

---

### Equivalent COM Command

SCPI.SENSE(Ch).FREQUENCY.CENTER

### Syntax

Value = app.SCPI.SENSE(Ch).FREQUENCY.CENTER

app.SCPI.SENSE(Ch).FREQUENCY.CENTER = 1E9

### Type

Double (read/write)

---

Back to [SENSe](#)

## **SENS:FREQ:SPAN**

### **SCPI Command**

SENSe<Ch>:FREQuency:SPAN <frequency>

SENSe<Ch>:FREQuency:SPAN?

### **Description**

Sets or reads out the stimulus span value of the sweep range for linear or logarithmic sweep type.

command/query

### **Target**

Channel <Ch>,

<Ch>={[1]|2|...16}

### **Parameter**

<frequency> the stimulus span value from 0 to the maximum frequency span of the analyzer.

### **Unit**

Hz (Hertz)

### **Out of Range**

Sets the value of the limit, which is closer to the specified value.

### **Query Response**

<numeric>

### **Preset Value**

The maximum frequency span of the analyzer

### **Equivalent Softkeys**

### **Stimulus > Span**

---

### Equivalent COM Command

SCPI.SENSE(Ch).FREQUENCY.SPAN

### Syntax

Value = app.SCPI.SENSE(Ch).FREQUENCY.SPAN

app.SCPI.SENSE(Ch).FREQUENCY.SPAN = 2E9

### Type

Double (read/write)

---

Back to [SENSe](#)

## **SENS:FREQ:STAR**

### **SCPI Command**

SENSe<Ch>:FREQuency:STARt <frequency>

SENSe<Ch>:FREQuency:STARt?

### **Description**

Sets or reads out the stimulus start value of the sweep range for linear or logarithmic sweep type.

command/query

### **Target**

Channel <Ch>,

<Ch>={[1]|2|...16}

### **Parameter**

<frequency> the stimulus start value within the frequency limits of the analyzer.

### **Unit**

Hz (Hertz)

### **Out of Range**

Sets the value of the limit, which is closer to the specified value.

### **Query Response**

<numeric>

### **Preset Value**

The minimum frequency span of the analyzer

### **Equivalent Softkeys**

### **Stimulus > Start**

---

**Equivalent COM Command**

SCPI.SENSE(Ch).FREQuency.STARt

**Syntax**

Value = app.SCPI.SENSE(Ch).FREQuency.STARt

app.SCPI.SENSE(Ch).FREQuency.STARt = 1E6

**Type**

Double (read/write)

---

Back to [SENSe](#)

## **SENS:FREQ:STOP**

### **SCPI Command**

SENSe<Ch>:FREQuency:STOP <frequency>

SENSe<Ch>:FREQuency:STOP?

### **Description**

Sets or reads out the stimulus stop value of the sweep range for linear or logarithmic sweep type.

command/query

### **Target**

Channel <Ch>,

<Ch>={1|2|...16}

### **Parameter**

<frequency> the stimulus stop value within the frequency limits of the analyzer.

### **Unit**

Hz (Hertz)

### **Out of Range**

Sets the value of the limit, which is closer to the specified value.

### **Query Response**

<numeric>

### **Preset Value**

The maximum frequency limit of the analyzer.

### **Equivalent Softkeys**

**Stimulus > Stop**

---

### Equivalent COM Command

SCPI.SENSE(Ch).FREQuency.STOP

### Syntax

Value = app.SCPI.SENSE(Ch).FREQuency.STOP

app.SCPI.SENSE(Ch).FREQuency.STOP = 1E6

### Type

Double (read/write)

---

Back to [SENSe](#)



## SENS:OFFS

### SCPI Command

SENSe<Ch>:OFFSet[:STATe] {OFF|ON|0|1}

SENSe<Ch>:OFFSet[:STATe]?

### Description

Turns the frequency offset feature ON/OFF.

command/query

### Target

Channel <Ch>,

<Ch>={{1}|2|...16}

### Parameter

Specifies the frequency offset feature:

{ON|1}      ON

{OFF|0}      OFF

### Query Response

{0|1}

### Preset Value

0

### Equivalent Softkeys

Stimulus > Frequency Offset > Frequency Offset

---

### Equivalent COM Command

SCPI.SENSe(Ch).OFFSet.STATe

**Syntax**

Status = app.SCPI.SENSE(CH).OFFSET.STATE

app.SCPI.SENSE(CH).OFFSET.STATE = True

**Type**

Boolean (read/write)

---

Back to [SENSE](#)

## SENS:OFFS:ADJ

### SCPI Command

SENSe<Ch>:OFFSet:ADJust[:STATe] {OFF|ON|0|1}

SENSe<Ch>:OFFSet:ADJust[:STATe]?

### Description

Turns the frequency offset adjust function ON/OFF.

command/query

### Target

Channel <Ch>,

<Ch>={{1}|2|...16}

### Parameter

Specifies the frequency offset adjust function:

{ON|1}      ON

{OFF|0}      OFF

### Query Response

{0|1}

### Preset Value

0

### Equivalent Softkeys

Stimulus > Frequency Offset > Offset Adjust > Offset Adjust

---

### Equivalent COM Command

None

---

Back to [SENSe](#)

## **SENS:OFFS:ADJ:CONT:PER**

### **SCPI Command**

SENSe<Ch>:OFFSet:ADJust:CONTInuous:PERiod <numeric>

SENSe<Ch>:OFFSet:ADJust:CONTInuous:PERiod?

### **Description**

Sets or reads out the adjust period in seconds when the frequency offset adjust function is active. If the adjust period is set to the value other than zero then the adjust procedure is automatically repeated.

command/query

### **Target**

Channel <Ch>,

<Ch>={1|2|...16}

### **Parameter**

<numeric> period of the automatic adjust procedure from 5 to 10000 seconds. Zero value disables the automatic adjust procedure.

### **Unit**

sec (seconds)

### **Query Response**

<numeric>

### **Preset Value**

0

### **Equivalent Softkeys**

**Stimulus > Frequency Offset > Offset Adjust > Auto Adjust Period**

---

### **Equivalent COM Command**

None

---

Back to [SENSe](#)

## SENS:OFFS:ADJ:EXEC

### SCPI Command

SENSe<Ch>:OFFSet:ADJust:EXECute

### Description

Executes the frequency offset adjust procedure once and sets the value of the frequency adjust.

command

### Target

Channel <Ch>,

<Ch>={1|2|...16}

### Equivalent Softkeys

Stimulus > Frequency Offset > Offset Adjust > Auto Adjust

---

### Equivalent COM Command

None

---

Back to [SENSe](#)

## SENS:OFFS:ADJ:PATH

### SCPI Command

SENSe<Ch>:OFFSet:ADJust:PATH <numeric 1>, <numeric 2>

SENSe<Ch>:OFFSet:ADJust:PATH?

### Description

Sets or reads out the number of the source and receiver ports used during the frequency offset adjust procedure.

command/query

### Target

Channel <Ch>,

<Ch>={[1]|2|...16}

### Parameter

<numeric 1> - number of receiver port;

<numeric 2> - number of source port.

### Query Response

<numeric 1>, <numeric 2>

### Equivalent Softkeys

Stimulus > Frequency Offset > Offset Adjust > Auto Adjust Path

---

### Equivalent COM Command

None

---

Back to [SENSe](#)



## SENS:OFFS:ADJ:PORT

### SCPI Command

SENSe<Ch>:OFFSet:ADJust:PORTs <numeric >

SENSe<Ch>:OFFSet:ADJust:PORTs?

### Description

Sets or reads out the port number to which offset adjust is applied when the frequency offset adjust function is active.

command/query

### Target

Channel <Ch>,

<Ch>={1|2|...16}

### Parameter

<numeric> port number to which frequency adjust is applied.

### Query Response

<numeric>

### Equivalent Softkeys

Stimulus > Frequency Offset > Offset Adjust > Adjusted Port

---

### Equivalent COM Command

None

---

Back to [SENSe](#)

## SENS:OFFS:ADJ:VAL

### SCPI Command

SENSe<Ch>:OFFSet:ADJust:VALue <numeric>

SENSe<Ch>:OFFSet:ADJust:VALue?

### Description

Sets or reads out the value of the offset adjust. The value is added to the frequency offset of the port set by the [SENS:OFFS:ADJ:PORT](#) command. The value is automatically adjusted when the [SENS:OFFS:ADJ:EXEC](#) command is executed.

command/query

### Target

Channel <Ch>,

<Ch>={[1]|2|...16}

### Parameter

<numeric> adjust value from -500 kHz to +500 kHz.

### Unit

Hz (Hertz)

### Query Response

<numeric>

### Equivalent Softkeys

Stimulus > Frequency Offset > Offset Adjust > Adjust Value

---

### Equivalent COM Command

None

---

Back to [SENSe](#)

## SENS:OFFS:PORT:DATA?

### SCPI Command

SENSe<Ch>:OFFSet:PORT<Pt>[:FREQuency]:DATA?

### Description

Reads out the array of the frequency points of port <Pt> when the frequency offset feature is ON and offset type is "PORT".

The array size is N, where N is the number of measurement points.

For the n–th point, where n from 1 to N:

<numeric n>            the frequency value at the n–th point

query only

### Target

Port <Pt> of channel <Ch>,

<Ch>={[1]|2|...16}

<Pt>={[1]|2}

### Query Response

<numeric 1>, <numeric 2>, ...<numeric N>

### Related Commands

[SENS:OFFS](#)

[SENS:OFFS:TYPE](#)

### Equivalent Softkeys

None

---

### Equivalent COM Command

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

**Syntax**

Status = app.SCPI.SENSE(CH).OFFSET.STATE

app.SCPI.SENSE(CH).OFFSET.STATE = True

**Type**

Boolean (read/write)

---

Back to [SENSE](#)

## **SENS:OFFS:PORT:DIV**

### **SCPI Command**

SENSe<Ch>:OFFSet:PORT<Pt>[:FREQuency]:DIVisor <numeric>

SENSe<Ch>:OFFSet:PORT<Pt>[:FREQuency]:DIVisor?

### **Description**

Sets or reads out the basic frequency range divisor of port <Pt> when the frequency offset feature is ON and offset type is "PORT".

command/query

### **Target**

Port <Pt> of channel <Ch>,

<Ch>={[1]|2|...16}

<Pt>={[1]|2}

### **Parameter**

<numeric> divisor from 1 to 1000

### **Out of Range**

Sets the value of the limit, which is closer to the specified value.

### **Query Response**

<numeric>

### **Preset Value**

1

### **Related Commands**

[SENS:OFFS](#)

[SENS:OFFS:TYPE](#)

## Equivalent Softkeys

Stimulus > Frequency Offset > Port n > Divider

---

## Equivalent COM Command

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

## Syntax

Value = app.SCPI.SENSE(Ch).OFFset.PORT(Pt).FREQuency.DIVisor

app.SCPI.SENSE(Ch).OFFset.PORT(Pt).FREQuency.DIVisor = 2

## Type

Double (read/write)

---

Back to [SENSe](#)

## SENS:OFFS:PORT:MULT

### SCPI Command

SENSe<Ch>:OFFSet:PORT<Pt>[:FREQuency]:MULTiplier <numeric>

SENSe<Ch>:OFFSet:PORT<Pt>[:FREQuency]:MULTiplier?

### Description

Sets or reads out the basic frequency range multiplier of port <Pt> when the frequency offset feature is ON and offset type is "PORT".

command/query

### Target

Port <Pt> of channel <Ch>,

<Ch>={[1]|2|...16}

<Pt>={[1]|2}

### Parameter

<numeric> multiplier from –1000 to 1000

### Out of Range

Sets the value of the limit, which is closer to the specified value.

### Query Response

<numeric>

### Preset Value

1

### Related Commands

[SENS:OFFS](#)

[SENS:OFFS:TYPE](#)



## Equivalent Softkeys

Stimulus > Frequency Offset > Port n > Multiplier

---

## Equivalent COM Command

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

## Syntax

Value = app.SCPI.SENSE(Ch).OFFset.PORT(Pt).FREQuency.MULTiplier

app.SCPI.SENSE(Ch).OFFset.PORT(Pt).FREQuency.MULTiplier = 2

## Type

Double (read/write)

---

Back to [SENSe](#)

## **SENS:OFFS:PORT:OFFS**

### **SCPI Command**

SENSe<Ch>:OFFSet:PORT<Pt>[:FREQuency]:OFFSet <frequency>

SENSe<Ch>:OFFSet:PORT<Pt>[:FREQuency]:OFFSet?

### **Description**

Sets or reads out the basic frequency range offset of port <Pt> when the frequency offset feature is ON and offset type is "PORT".

command/query

### **Target**

Port <Pt> of channel <Ch>,

<Ch>={[1]|2|...16}

<Pt>={[1]|2}

### **Parameter**

<frequency> offset from  $-1\text{e}12$  to  $1\text{e}12$

### **Unit**

Hz (Hertz)

### **Out of Range**

Sets the value of the limit, which is closer to the specified value.

### **Query Response**

<numeric>

### **Preset Value**

0

## Related Commands

[SENS:OFFS](#)

[SENS:OFFS:TYPE](#)

## Equivalent Softkeys

**Stimulus > Frequency Offset > Port n > Offset**

---

## Equivalent COM Command

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

## Syntax

Value = app.SCPI.SENSE(Ch).OFFset.PORT(Pt).FREQuency.OFFSet

app.SCPI.SENSE(Ch).OFFset.PORT(Pt).FREQuency.OFFSet = 1e9

## Type

Double (read/write)

---

Back to [SENSe](#)

## SENS:OFFS:PORT:STAR

### SCPI Command

SENSe<Ch>:OFFSet:PORT<Pt>[:FREQuency]:STARt <frequency>

SENSe<Ch>:OFFSet:PORT<Pt>[:FREQuency]:STARt?

### Description

Sets or reads out the frequency sweep start of port <Pt> when the frequency offset feature is ON and offset type is "PORT".

command/query

### Target

Port <Pt> of channel <Ch>,

<Ch>={[1]|2|...16}

<Pt>={[1]|2}

### Parameter

<frequency> frequency sweep start of port <Pt>

### Unit

Hz (Hertz)

### Query Response

<numeric>

### Related Commands

[SENS:OFFS](#)

[SENS:OFFS:TYPE](#)

### Equivalent Softkeys

Stimulus > Frequency Offset > Port n > Start

---

**Equivalent COM Command**

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

**Syntax**

Value = app.SCPI.SENSE(Ch).OFFset.PORT(Pt).FREQuency.START

**Type**

Double (read/write)

---

Back to [SENSe](#)

## SENS:OFFS:PORT:STOP

### SCPI Command

SENSe<Ch>:OFFSet:PORT<Pt>[:FREQuency]:STOP <frequency>

SENSe<Ch>:OFFSet:PORT<Pt>[:FREQuency]:STOP?

### Description

Sets or reads out the frequency sweep stop of port <Pt> when the frequency offset feature is ON and offset type is "PORT".

command/query

### Target

Port <Pt> of channel <Ch>,

<Ch>={[1]|2|...16}

<Pt>={[1]|2}

### Parameter

<frequency> frequency sweep stop of port <Pt>

### Unit

Hz (Hertz)

### Query Response

<numeric>

### Related Commands

[SENS:OFFS](#)

[SENS:OFFS:TYPE](#)

### Equivalent Softkeys

Stimulus > Frequency Offset > Port n > Stop

---

**Equivalent COM Command**

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

**Syntax**

Value = app.SCPI.SENSE(Ch).OFFset.PORT(Pt).FREQuency.STOP

**Type**

Double (read/write)

---

Back to [SENSe](#)

## SENS:OFFS:REC:DATA?

### SCPI Command

SENSe<Ch>:OFFSet:RECeiver[:FREQuency]:DATA?

### Description

Reads out the array of the receiver frequency points when the frequency offset feature is ON and offset type is "SRCRcv".

The array size is N, where N is the number of measurement points.

For the n–th point, where n from 1 to N:

<numeric n>      the frequency value at the n–th point

query only

### Target

Channel <Ch>,

<Ch>={[1]|2|...16}

### Query Response

<numeric 1>, <numeric 2>, ...<numeric N>

### Related Commands

[SENS:OFFS](#)

[SENS:OFFS:TYPE](#)

### Equivalent Softkeys

None

---

### Equivalent COM Command

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



**Syntax**

Data = app.SCPI.SENSE(CH).OFFSET.RECEIVER.FREQUENCY.DATA

**Type**

Variant (array of Double) (read only)

---

Back to [SENSE](#)

## **SENS:OFFS:REC:DIV**

### **SCPI Command**

SENSe<Ch>:OFFSet:RECeiver[:FREQuency]:DIVisor <numeric>

SENSe<Ch>:OFFSet:RECeiver[:FREQuency]:DIVisor?

### **Description**

Sets or reads out the basic frequency range divisor to get the receiver frequency when the frequency offset feature is ON and offset type is "SRCRCv".

command/query

### **Target**

Channel <Ch>,

<Ch>={[1]2|...16}

### **Parameter**

<numeric> divisor from 1 to 1000

### **Out of Range**

Sets the value of the limit, which is closer to the specified value.

### **Query Response**

<numeric>

### **Preset Value**

1

### **Related Commands**

[SENS:OFFS](#)

[SENS:OFFS:TYPE](#)

## Equivalent Softkeys

Stimulus > Frequency Offset > Receivers > Divider

---

## Equivalent COM Command

SCPI.SENSE(Ch).OFFSet.RECeiver.FREQuency.DIVisor

## Syntax

Value = app.SCPI.SENSE(Ch).OFFset.RECeiver.FREQuency.DIVisor

app.SCPI.SENSE(Ch).OFFset.RECeiver.FREQuency.DIVisor = 2

## Type

Double (read/write)

---

Back to [SENSe](#)

## SENS:OFFS:REC:MULT

### SCPI Command

SENSe<Ch>:OFFSet:RECeiver[:FREQuency]:MULTiplier <numeric>

SENSe<Ch>:OFFSet:RECeiver[:FREQuency]:MULTiplier?

### Description

Sets or reads out the basic frequency range multiplier to get the receiver frequency when the frequency offset feature is ON and offset type is "SRCRcv".

command/query

### Target

Channel <Ch>,

<Ch>={[1]|2|...16}

### Parameter

<numeric> multiplier from –1000 to 1000

### Out of Range

Sets the value of the limit, which is closer to the specified value.

### Query Response

<numeric>

### Preset Value

1

### Related Commands

[SENS:OFFS](#)

[SENS:OFFS:TYPE](#)

## Equivalent Softkeys

Stimulus > Frequency Offset > Receivers > Multiplier

---

## Equivalent COM Command

SCPI.SENSE(Ch).OFFSet.RECeiver.FREQuency.MULTiplier

## Syntax

Value = app.SCPI.SENSE(Ch).OFFset.RECeiver.FREQuency.MULTiplier

app.SCPI.SENSE(Ch).OFFset.RECeiver.FREQuency.MULTiplier = 2

## Type

Double (read/write)

---

Back to [SENSe](#)

## **SENS:OFFS:REC:OFFS**

### **SCPI Command**

SENSe<Ch>:OFFSet:RECeiver[:FREQuency]:OFFSet <frequency>

SENSe<Ch>:OFFSet:RECeiver[:FREQuency]:OFFSet?

### **Description**

Sets or reads out the basic frequency range offset to get the receiver frequency when the frequency offset feature is ON and offset type is "SRCRCv".

command/query

### **Target**

Channel <Ch>,

<Ch>={[1]|2|...16}

### **Parameter**

<frequency> offset from  $-1\text{e}12$  to  $1\text{e}12$

### **Unit**

Hz (Hertz)

### **Out of Range**

Sets the value of the limit, which is closer to the specified value.

### **Query Response**

<numeric>

### **Preset Value**

0

## Related Commands

[SENS:OFFS](#)

[SENS:OFFS:TYPE](#)

## Equivalent Softkeys

**Stimulus > Frequency Offset > Receivers > Offset**

---

## Equivalent COM Command

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

## Syntax

Value = app.SCPI.SENSE(Ch).OFFset.RECeiver.FREQuency.OFFSet

app.SCPI.SENSE(Ch).OFFset.RECeiver.FREQuency.OFFSet = 1e9

## Type

Double (read/write)

---

Back to [SENSe](#)

## **SENS:OFFS:REC:STAR**

### **SCPI Command**

SENSe<Ch>:OFFSet:RECeiver[:FREQuency]:STARt <frequency>

SENSe<Ch>:OFFSet:RECeiver[:FREQuency]:STARt?

### **Description**

Sets or reads out the frequency sweep start of the receivers when the frequency offset feature is ON and offset type is "SRCRCv".

command/query

### **Target**

Channel <Ch>,

<Ch>={[1]|2|...16}

### **Parameter**

<frequency> frequency sweep start of receivers

### **Unit**

Hz (Hertz)

### **Query Response**

<numeric>

### **Related Commands**

[SENS:OFFS](#)

[SENS:OFFS:TYPE](#)

### **Equivalent Softkeys**

**Stimulus > Frequency Offset > Receivers > Start**



---

### Equivalent COM Command

SCPI.SENSE(Ch).OFFSet.RECeiver.FREQuency.STARt

### Syntax

Value = app.SCPI.SENSE(Ch).OFFset.RECeiver.FREQuency.STARt

app.SCPI.SENSE(Ch).OFFset.RECeiver.FREQuency.STARt = 1e9

### Type

Double (read/write)

---

Back to [SENSe](#)

## **SENS:OFFS:REC:STOP**

### **SCPI Command**

SENSe<Ch>:OFFSet:RECeiver[:FREQuency]:STOP <frequency>

SENSe<Ch>:OFFSet:RECeiver[:FREQuency]:STOP?

### **Description**

Sets or reads out the frequency sweep stop of the receivers when the frequency offset feature is ON and offset type is "SRCRCv".

command/query

### **Target**

Channel <Ch>,

<Ch>={[1]|2|...16}

### **Parameter**

<frequency> frequency sweep stop of receivers.

### **Unit**

Hz (Hertz)

### **Query Response**

<numeric>

### **Related Commands**

[SENS:OFFS](#)

[SENS:OFFS:TYPE](#)

### **Equivalent Softkeys**

**Stimulus > Frequency Offset > Receivers > Stop**

---

### Equivalent COM Command

SCPI.SENSE(Ch).OFFSet.RECeiver.FREQuency.STOP

### Syntax

Value = app.SCPI.SENSE(Ch).OFFset.RECeiver.FREQuency.STOP

app.SCPI.SENSE(Ch).OFFset.RECeiver.FREQuency.STOP = 1e9

### Type

Double (read/write)

---

Back to [SENSe](#)

## SENS:OFFS:SOUR:DATA?

### SCPI Command

SENSe<Ch>:OFFSet:SOURce[:FREQuency]:DATA?

### Description

Reads out the array of the frequency points of the source when the frequency offset feature is ON and offset type is "SRCRCv".

The array size is N, where N is the number of measurement points.

For the n–th point, where n from 1 to N:

<numeric n>    the frequency value at the n–th point

query only

### Target

Channel <Ch>,

<Ch>={[1]|2|...16}

### Query Response

<numeric 1>, <numeric 2>, ...<numeric N>

### Related Commands

[SENS:OFFS](#)

[SENS:OFFS:TYPE](#)

### Equivalent Softkeys

None

---

### Equivalent COM Command

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

**Syntax**

Data = app.SCPI.SENSE(CH).OFFSET.SOURCE.FREQUENCY.DATA

**Type**

Variant (array of Double) (read only)

---

Back to [SENSE](#)

## **SENS:OFFS:SOUR:DIV**

### **SCPI Command**

SENSe<Ch>:OFFSet:SOURce[:FREQuency]:DIVisor <numeric>

SENSe<Ch>:OFFSet:SOURce[:FREQuency]:DIVisor?

### **Description**

Sets or reads out the basic frequency range divisor to get the source frequency when the frequency offset feature is ON and offset type is "SRCRCv".

command/query

### **Target**

Channel <Ch>,

<Ch>={[1]|2|...16}

### **Parameter**

<numeric> divisor from 1 to 1000

### **Out of Range**

Sets the value of the limit, which is closer to the specified value.

### **Query Response**

<numeric>

### **Preset Value**

1

### **Related Commands**

[SENS:OFFS](#)

[SENS:OFFS:TYPE](#)

## Equivalent Softkeys

Stimulus > Frequency Offset > Source > Divider

---

## Equivalent COM Command

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

## Syntax

Value = app.SCPI.SENSE(Ch).OFFset.SOURce.FREQuency.DIVisor

app.SCPI.SENSE(Ch).OFFset.SOURce.FREQuency.DIVisor = 2

## Type

Double (read/write)

---

Back to [SENSe](#)

## SENS:OFFS:SOUR:MULT

### SCPI Command

SENSe<Ch>:OFFSet:SOURce[:FREQuency]:MULTiplier <numeric>

SENSe<Ch>:OFFSet:SOURce[:FREQuency]:MULTiplier?

### Description

Sets or reads out the basic frequency range multiplier to get the source frequency when the frequency offset feature is ON and offset type is "SRCRCv".

command/query

### Target

Channel <Ch>,

<Ch>={[1]|2|...16}

### Parameter

<numeric> multiplier from –1000 to 1000

### Out of Range

Sets the value of the limit, which is closer to the specified value.

### Query Response

<numeric>

### Preset Value

1

### Related Commands

[SENS:OFFS](#)

[SENS:OFFS:TYPE](#)



## Equivalent Softkeys

Stimulus > Frequency Offset > Source > Multiplier

---

## Equivalent COM Command

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

## Syntax

Value = app.SCPI.SENSE(Ch).OFFset.SOURce.FREQuency.MULTiplier

app.SCPI.SENSE(Ch).OFFset.SOURce.FREQuency.MULTiplier = 2

## Type

Double (read/write)

---

Back to [SENSe](#)

## **SENS:OFFS:SOUR:OFFS**

### **SCPI Command**

SENSe<Ch>:OFFSet:SOURce[:FREQuency]:OFFSet <frequency>

SENSe<Ch>:OFFSet:SOURce[:FREQuency]:OFFSet?

### **Description**

Sets or reads out the basic frequency range offset to get the source frequency when the frequency offset feature is ON and offset type is "SRCRCv".

command/query

### **Target**

Channel <Ch>,

<Ch>={[1]2|...16}

### **Parameter**

<frequency> offset from  $-1\text{e}12$  to  $1\text{e}12$

### **Unit**

Hz (Hertz)

### **Out of Range**

Sets the value of the limit, which is closer to the specified value.

### **Query Response**

<numeric>

### **Preset Value**

0

## Related Commands

[SENS:OFFS](#)

[SENS:OFFS:TYPE](#)

## Equivalent Softkeys

**Stimulus > Frequency Offset > Source > Offset**

---

## Equivalent COM Command

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

## Syntax

Value = app.SCPI.SENSE(Ch).OFFset.SOURce.FREQuency.OFFSet

app.SCPI.SENSE(Ch).OFFset.SOURce.FREQuency.OFFSet = 1e9

## Type

Double (read/write)

---

Back to [SENSe](#)

## **SENS:OFFS:SOUR:STAR**

### **SCPI Command**

SENSe<Ch>:OFFSet:SOURce[:FREQuency]:STARt <frequency>

SENSe<Ch>:OFFSet:SOURce[:FREQuency]:STARt?

### **Description**

Sets or reads out the frequency sweep start of the source when the frequency offset feature is ON and offset type is "SRCRCv".

command/query

### **Target**

Channel <Ch>,

<Ch>={[1]|2|...16}

### **Parameter**

<frequency> frequency sweep start of the source

### **Unit**

Hz (Hertz)

### **Query Response**

<numeric>

### **Related Commands**

[SENS:OFFS](#)

[SENS:OFFS:TYPE](#)

### **Equivalent Softkeys**

**Stimulus > Frequency Offset > Source > Start**

---

### Equivalent COM Command

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

### Syntax

Value = app.SCPI.SENSE(Ch).OFFset.SOURce.FREQuency.STARt

app.SCPI.SENSE(Ch).OFFset.SOURce.FREQuency.STARt = 1e9

### Type

Double (read/write)

---

Back to [SENSe](#)

## **SENS:OFFS:SOUR:STOP**

### **SCPI Command**

SENSe<Ch>:OFFSet:SOURce[:FREQuency]:STOP <frequency>

SENSe<Ch>:OFFSet:SOURce[:FREQuency]:STOP?

### **Description**

Sets or reads out the frequency sweep start of the source when the frequency offset feature is ON and offset type is "SRCRCv".

command/query

### **Target**

Channel <Ch>,

<Ch>={[1]|2|...16}

### **Parameter**

<frequency> frequency sweep stop of the source

### **Unit**

Hz (Hertz)

### **Query Response**

<numeric>

### **Related Commands**

[SENS:OFFS](#)

[SENS:OFFS:TYPE](#)

### **Equivalent Softkeys**

**Stimulus > Frequency Offset > Source > Stop**

---

### Equivalent COM Command

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

### Syntax

Value = app.SCPI.SENSE(Ch).OFFset.SOURce.FREQuency.STOP

app.SCPI.SENSE(Ch).OFFset.SOURce.FREQuency.STOP = 1e9

### Type

Double (read/write)

---

Back to [SENSe](#)

## SENS:OFFS:TYPE

### SCPI Command

SENSe<Ch>:OFFSet:TYPE <char>

SENSe<Ch>:OFFSet:TYPE?

### Description

Sets or reads out 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.

command/query

### Target

Channel <Ch>,

<Ch>={{1}|2|...16}

### Parameter

<char> Specifies the offset type:

**PORT**            Port1/Port2 offset

**SRRCrv**        Source/Receivers offset

### Query Response

{PORT|SRRCrv}

### Preset Value

PORT

### Equivalent Softkeys

**Stimulus > Frequency Offset > Offset Type**



---

## Equivalent COM Command

SCPI.SENSE(Ch).OFFSet.TYPE

## Syntax

Param = app.SCPI.SENSE(Ch).OFFSet.TYPE

app.SCPI.SENSE(Ch).OFFSet.TYPE = "PORT"

## Type

String (read/write)

---

Back to [SENSe](#)

## SENS:ROSC:SOUR

### SCPI Command

SENSe:ROSCillator:SOURce <char>

SENSe:ROSCillator:SOURce?

### Description

Sets or reads out an internal or external source of the 10 MHz reference frequency.

command/query

### Target

Instrument

### Parameter

<char> Choose from:

**INTernal** Internal source of the reference frequency

**EXTernal** External source of the reference frequency

### Query Response

{INT|EXT}

### Preset Value

INT

### Equivalent Softkeys

System > Misc Setup > Ref Source

---

### Equivalent COM Command

SCPI.SENSe(Ch).ROSCillator.SOURce

**Syntax**

Param = app.SCPI.SENSE(CH).ROSCillator.SOURce

app.SCPI.SENSE(CH).ROSCillator.SOURce = "EXT"

**Type**

String (read/write)

---

Back to [SENSe](#)

## SENS:SEGM:DATA

### SCPI Command

SENSe<Ch>:SEGMent:DATA <numeric list>

SENSe<Ch>:SEGMent:DATA?

### Description

Sets or reads out the array of the segment sweep table.

The array has the following format:

```
{<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>],  
...  
<StartN>, <StopN>, <NOP N> [,<IFBW N>] [,<Pow N>] [,<Del N>] [,<TimeN>]}
```

<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 <Del> field (0 — disabled, 1 — enabled),  
<Flag5> : Setting of the <Time> field (0 — disabled, 1 — enabled),  
<N> : Number of segments,  
<Start n> : Start value of the n-th segment,  
<Stop n> : Stop value of the n-th segment,  
<NOP n> : Number of points of the n-th segment,  
<IFBW n> : IF bandwidth of the n-th segment (if enabled),  
<Pow n> : Power of the n-th segment (if enabled),  
<Del n> : Measurement delay of the n-th segment (if enabled),  
<Time n> : Reserved for future use (if enabled)

command/query

## Target

Channel <Ch>,

<Ch>={1|2|...16}

## Query Response

<numeric 1>,<numeric 2>,...<numeric 7+M×N>

Where:

N – the number of the segments,

M – depends on the values of the flags:

$M = 3 + \text{<Flag2>} + \text{<Flag3>} + \text{<Flag4>} + \text{<Flag5>}$

## Equivalent Softkeys

### Stimulus > Segment Table

---

#### Equivalent COM Command

SCPI.SENSE(CH).SEGMENT.DATA

#### Syntax

Data = app.SCPI.SENSE(CH).SEGMENT.DATA

app.SCPI.SENSE(CH).SEGMENT.DATA = Data

#### Type

Variant (array of Double) (read/write)

---

Back to [SENSE](#)

## **SENS:SWE:CW:TIME**

### **SCPI Command**

SENSe<Ch>:SWEep:CW:TIME <numeric>

SENSe<Ch>:SWEep:CW:TIME?

### **Description**

Sets or reads out the sweep time value when the CW time mode is ON (the span is set to zero).

command/query

### **Target**

Channel <Ch>,

<Ch>={1|2|...16}

### **Parameter**

<numeric> the sweep time value.

### **Unit**

sec (second)

### **Query Response**

<numeric>

### **Equivalent Softkeys**

**Stimulus > Sweep Time**

---

## Equivalent COM Command

None

---

Back to [SENSe](#)

## SENS:SWE:POIN

### SCPI Command

SENSe<Ch>:SWEep:POINts <numeric>

SENSe<Ch>:SWEep:POINts?

### Description

Sets or reads out the number of measurement points.

command/query

### Target

Channel <Ch>,

<Ch>={{[1]|2|...16}}

### Parameter

<numeric> the number of measurement points from 2 to maximum limit of the analyzer.

### Out of Range

Sets the value of the limit, which is closer to the specified value.

### Query Response

<numeric>

### Preset Value

201

### Equivalent Softkeys

### Stimulus > Points

---

### Equivalent COM Command

SCPI.SENSe(Ch).SWEep.POINts



**Syntax**

Value = app.SCPI.SENSE(CH).SWEep.POINts

app.SCPI.SENSE(CH).SWEep.POINts = 1001

**Type**

Long (read/write)

---

Back to [SENSe](#)

## **SENS:SWE:POIN:TIME**

### **SCPI Command**

SENSe<Ch>:SWEep:POINt:TIME <time>

SENSe<Ch>:SWEep:POINt:TIME?

### **Description**

Sets or reads out the delay before measurement in each measurement point.

command/query

### **Target**

Channel <Ch>,

<Ch>={{1}|2|...16}

### **Parameter**

<time> the measurement delay value from 0 to 0.3 sec.

### **Unit**

sec (second)

### **Out of Range**

Sets the value of the limit, which is closer to the specified value.

### **Query Response**

<numeric>

### **Preset Value**

0

### **Equivalent Softkeys**

### **Stimulus > Meas Delay**

---

### Equivalent COM Command

SCPI.SENSE(Ch).SWEep.POINt.TIME

### Syntax

Value = app.SCPI.SENSE(Ch).SWEep.POINt.TIME

app.SCPI.SENSE(Ch).SWEep.POINt.TIME = 5E-6

### Type

Double (read/write)

---

Back to [SENSe](#)

## SENS:SWE:REV

### SCPI Command

SENSe<Ch>:SWEep:REVerse[:STATe] {OFF|ON|0|1}

SENSe<Ch>:SWEep:REVerse[:STATe]?

### Description

Sets or reads out the ON/OFF status of the reverse sweep function. In normal sweep mode, the start frequency is less than the stop frequency. If reverse sweep mode is enabled, the stop frequency is less than the start frequency.

command/query

### Target

Channel <Ch>,

<Ch>={[1]|2|...16}

### Parameter

{ON|1}      Reverse sweep mode

{OFF|0}      Normal sweep mode

### Query Response

{0|1}

### Preset Value

0

### Equivalent Softkeys

Stimulus > Reverse Sweep {ON/OFF}

---

### Equivalent COM Command

SCPI.SENSe(Ch).SWEep.REVerse.STATe

**Syntax**

Status = app.SCPI.SENSE(CH).SWEep.REVERSE.STATE

app.SCPI.SENSE(CH).SWEp.REVERSE.STATE = True

**Type**

Boolean (read/write)

---

Back to [SENSE](#)

## SENS:SWE:TYPE

### SCPI Command

SENSe<Ch>:SWEep:TYPE <char>

SENSe<Ch>:SWEep:TYPE?

### Description

Sets or reads out the sweep type.

command/query

### Target

Channel <Ch>,

<Ch>={{1}|2|...16}

### Parameter

<char> Specifies the sweep type:

<b>LINear</b>	Linear frequency sweep
<b>LOGarithmic</b>	Logarithmic frequency sweep
<b>SEGMENT</b>	Segment frequency sweep
<b>POWER</b>	Power sweep

### Query Response

{LIN|LOG|SEGM|POW}

### Preset Value

LIN

### Equivalent Softkeys

Stimulus > Sweep Type

---

## Equivalent COM Command

SCPI.SENSE(Ch).SWEep.TYPE

## Syntax

Param = app.SCPI.SENSE(Ch).SWEep.TYPE

app.SCPI.SENSE(Ch).SWEep.TYPE = "LOG"

## Type

String (read/write)

---

Back to [SENSe](#)

## **SENS:VOLT:DC:RANG:UPP**

### **SCPI Command**

SENSe<Ch>:VOLTage{[1]|2}:DC:RANGe:UPPer <numeric>

SENSe<Ch>:VOLTage{[1]|2}:DC:RANGe:UPPer?

### **Description**

Sets or reads out the DC voltage range at the connector AUX1 or AUX2.

command/query

### **Target**

Channel <Ch>,

<Ch>={[1]|2|...16}

### **Parameter**

<numeric> the DC voltage range 10V or 1V

### **Unit**

V (Volt)

### **Out of Range**

Sets the value of the limit, which is closer to the specified value.

### **Query Response**

<numeric>

### **Preset Value**

10

### **Equivalent Softkeys**

**Measurement > DC Voltage > Range**



---

### Equivalent COM Command

None

---

Back to [SENSe](#)

## SERVice

Command		Description
<a href="#">SERV:CHAN:ACT?</a>	Channel and Trace Settings	Active channel number (read)
<a href="#">SERV:CHAN:TRAC:ACT?</a>		Active trace number (read)
<a href="#">SERV:CHAN:COUN?</a>	Analyzer Capabilities	Maximum number of channels
<a href="#">SERV:CHAN:TRAC:COUN?</a>		Maximum number of traces in the channel
<a href="#">SERV:PORT:COUN?</a>		Ports number
<a href="#">SERV:SWE:FREQ:MAX?</a>		Upper limit of frequency
<a href="#">SERV:SWE:FREQ:MIN?</a>		Lower limit of frequency
<a href="#">SERV:SWE:POIN?</a>		Maximum number of points
<a href="#">SERV:SWE:POW:MAX?</a>		Upper limit of source power
<a href="#">SERV:SWE:POW:MIN?</a>		Lower limit of source power
<a href="#">SERV:CHAN:TRAC:MARK:ACT?</a>	Marker Properties	Gets active marker

## SERV:CHAN:ACT?

### SCPI Command

SERVice:CHANnel:ACTive?

### Description

Reads out the active channel number.

query only

### Query Response

<numeric> from 1 to 16

### Related Commands

[DISP:WIND:ACT](#)

### Equivalent Softkeys

None

---

### Equivalent COM Command

SCPI.SERVice.CHANnel(1).ACTive

### Syntax

Value = app.SCPI.SERVice.CHANnel.ACTive

### Type

Long (read/write)

---

### **WARNING**

Object CHANnel has an index of 1, which can be omitted in Visual Basic but it cannot be omitted in other programming languages.

---

Back to [SERVice](#)

## SERV:CHAN:COUN?

### SCPI Command

SERVice:CHANnel:COUNt?

### Description

Reads out the maximum number of the analyzer channels.

query only

### Query Response

<numeric>

### Equivalent Softkeys

None

---

### Equivalent COM Command

SCPI.SERVice.CHANnel(1).COUNt

### Syntax

Value = app.SCPI.SERVice.CHANnel.COUNt

### Type

Long (read/write)

---

#### **WARNING**

Object CHANnel has an index of 1, which can be omitted in Visual Basic but it cannot be omitted in other programming languages.

---

Back to [SERVice](#)

## **SERV:CHAN:TRAC:ACT?**

### **SCPI Command**

SERVice:CHANnel<Ch>:TRACe:ACTive?

### **Description**

Reads out the active trace number of the channel.

query only

### **Target**

Channel <Ch>,

<Ch>={[1]|2|...16}

### **Query Response**

<numeric> from 1 to 16

### **Equivalent Softkeys**

None

---

### **Equivalent COM Command**

SCPI.SERVice.CHANnel(Ch).TRACe(1).ACTive

### **Syntax**

Value = app.SCPI.SERVice.CHANnel(Ch).TRACe.ACTive

## Type

Long (read/write)

---

### **WARNING**

Object TRACe has an index of 1, which can be omitted in Visual Basic but it cannot be omitted in other programming languages.

---

---

Back to [SERVice](#)

## SERV:CHAN:TRAC:COUN?

### SCPI Command

SERVice:CHANnel:TRACe:COUNT?

### Description

Reads out the maximum number of traces in the channel.

query only

### Query Response

<numeric>

### Equivalent Softkeys

None

---

### Equivalent COM Command

SCPI.SERVice.CHANnel(1).TRACe.COUNT

### Syntax

Value = app.SCPI.SERVice.CHANnel.TRACe.COUNT

### Type

Long (read/write)

---

#### **WARNING**

Object CHANnel has an index of 1, which can be omitted in Visual Basic but it cannot be omitted in other programming languages.

---

Back to [SERVice](#)

## SERV:CHAN:TRAC:MARK:ACT?

### SCPI Command

SERVice:CHANnel<Ch>:TRACe<Tr>:MARKer:ACTive?

### Description

Gets the active marker number of the specified trace of the specified channel.

query only

### Target

Trace <Tr> of channel <Ch>,

<Ch> = {[1]|2|...16}

<Tr> = {[1]|2|...16}

### Query Response

<numeric>

### Related Commands

[CALC:MARK:ACT](#)

### Equivalent Softkeys

None

---

### Equivalent COM Command

SCPI.SERVice.CHANnel(Ch).TRACe(Tr).MARKer.ACTive

### Syntax

Value = app.SCPI.SERVice.CHANnel(Ch).TRACe(Tr).MARKerACTive

### Type

Long (only read)

---



Back to [SERVice](#)

## SERV:PORT:COUN?

### SCPI Command

SERVice:PORT:COUNT?

### Description

Reads out the number of analyzer ports.

query only

### Query Response

<numeric>

### Equivalent Softkeys

None

---

### Equivalent COM Command

SCPI.SERVice.PORT.COUNT

### Syntax

Value = app.SCPI.SERVice.PORT.COUNT

### Type

Long (read/write)

---

Back to [SERVice](#)

## **SERV:SWE:FREQ:MAX?**

### **SCPI Command**

SERVice:SWEep:FREQency:MAXimum?

### **Description**

Reads out the upper limit of the analyzer measurement frequency.

query only

### **Query Response**

<numeric>

### **Unit**

Hz (Hertz)

### **Equivalent Softkeys**

None

---

### **Equivalent COM Command**

SCPI.SERVice.SWEep.FREQency.MAXimum

### **Syntax**

Value = app.SCPI.SERVice.SWEep.FREQency.MAXimum

### **Type**

Double (read/write)

---

Back to [SERVice](#)

## SERV:SWE:FREQ:MIN?

### SCPI Command

SERVice:SWEep:FREQency:MINimum?

### Description

Reads out the lower limit of the analyzer measurement frequency.

query only

### Query Response

<numeric>

### Unit

Hz (Hertz)

### Equivalent Softkeys

None

---

### Equivalent COM Command

SCPI.SERVice.SWEep.FREQency.MINimum

### Syntax

Value = app.SCPI.SERVice.SWEep.FREQency.MINimum

### Type

Double (read/write)

---

Back to [SERVice](#)

## SERV:SWE:POIN?

### SCPI Command

SERVice:SWEep:POINts?

### Description

Reads the maximum number of analyzer measurement points.

query only

### Query Response

<numeric>

### Equivalent Softkeys

None

---

### Equivalent COM Command

SCPI.SERVice.SWEep.POINts

### Syntax

Value = app.SCPI.SERVice.SWEep.POINts

### Type

Long (read/write)

---

Back to [SERVice](#)

## SERV:SWE:POW:MAX?

### SCPI Command

SERVice:SWEep:POWer:MAXimum?

### Description

Reads out the upper limit of the source power.

query only

### Query Response

<numeric>

### Unit

dBm (decibels above 1 milliwatt)

### Equivalent Softkeys

None

---

### Equivalent COM Command

None

---

Back to [SERVice](#)

## SERV:SWE:POW:MIN?

### SCPI Command

SERVice:SWEep:POWer:MINimum?

### Description

Reads out the lower limit of the source power.

query only

### Query Response

<numeric>

### Unit

dBm (decibels above 1 milliwatt)

### Equivalent Softkeys

None

---

### Equivalent COM Command

None

---

Back to [SERVice](#)

## SOURce

Command		Description
<a href="#">SOUR:POW</a>	Stimulus Settings	Power level for a frequency sweep
<a href="#">SOUR:POW:CENT</a>		Center power
<a href="#">SOUR:POW:PORT</a>		Power level of each port
<a href="#">SOUR:POW:PORT:COUP</a>		Port power coupling ON/OFF
<a href="#">SOUR:POW:SLOP</a>		Power slope value
<a href="#">SOUR:POW:SLOP:STAT</a>		Power slope ON/OFF
<a href="#">SOUR:POW:SPAN</a>		Span power
<a href="#">SOUR:POW:STAR</a>		Start power
<a href="#">SOUR:POW:STOP</a>		Stop frequency
<a href="#">SOUR:POW:PORT:CORR</a>	Power Calibration	Power correction ON/OFF
<a href="#">SOUR:POW:PORT:CORR:INT?</a>		Interpolation/extrapolation status of power correction
<a href="#">SOUR:POW:PORT:CORR:COLL</a>		Power calibration procedure



Command		Description
<a href="#">SOUR:POW:PORT:CORR:COLL:TABL:LOSS:DATA</a>		Loss compensation table
<a href="#">SOUR:POW:PORT:CORR:COLL:TABL:LOSS</a>		Loss compensation ON/OFF
<a href="#">SOUR:POW:PORT:CORR:DATA</a>		Power correction data

## SOUR:POW

### SCPI Command

SOURce<Ch>:POWER[:LEVel][:IMMediate][:AMPLitude] <power>

SOURce<Ch>:POWER[:LEVel][:IMMediate][:AMPLitude]?

### Description

Sets or reads out the power level for the frequency sweep type.

command/query

### Target

Channel <Ch>,

<Ch>={[1]|2|...16}

### Parameter

<power> the power level within the power limits of the analyzer.

### Unit

dBm (decibels above 1 milliwatt)

### Resolution

0.05 dBm

### Out of Range

Sets the value of the limit, which is closer to the specified value.

### Query Response

<numeric>

### Preset Value

0 dBm

## Related Commands

[SOUR:POW:PORT:COUP](#)

## Equivalent Softkeys

Stimulus > Power > Power

---

## Equivalent COM Command

None

---

Back to [SOURce](#)

## SOUR:POW:CEN

### SCPI Command

SOURce<Ch>:POWER:CENTer <power>

SOURce<Ch>:POWER:CENTer?

### Description

Sets or reads out the center value of the power sweep type.

command/query

### Target

Channel <Ch>,

<Ch>={{1}|2|...16}

### Parameter

<power> the power level within the power limits of the analyzer.

### Unit

dBm (decibels above 1 milliwatt)

### Resolution

0.05 dBm

### Out of Range

Sets the value of the limit, which is closer to the specified value.

### Query Response

<numeric>

### Preset Value

Depends on the Analyzer

## Equivalent Softkeys

Stimulus > Center

---

## Equivalent COM Command

SCPI.SOURce(Ch).POWer.CENTer

## Syntax

Value = app.SCPI.SOURce(Ch).POWer.CENTer

app.SCPI.SOURce(Ch).POWer.CENTer = 5

## Type

Double (read/write)

---

Back to [SOURce](#)

## SOUR:POW:PORT

### SCPI Command

SOURce<Ch>:POWER:PORT<Pt>[:LEVel][:IMMediate][:AMPLitude] <power>

SOURce<Ch>:POWER:PORT<Pt>[:LEVel][:IMMediate][:AMPLitude]?

### Description

Sets or reads out the power level of each port for the frequency sweep type when the port couple feature is set to OFF by the [SOUR:POW:PORT:COUP](#) command.

command/query

### Target

Port <Pt> of channel <Ch>,

<Ch>={[1]|2|...16}

<Pt>={[1]|2}

### Parameter

<power> the power level within the power limits of the analyzer.

### Unit

dBm (decibels above 1 milliwatt)

### Resolution

0.05 dBm

### Out of Range

Sets the value of the limit, which is closer to the specified value.

### Query Response

<numeric>

## Preset Value

0

## Related Commands

[SOUR:POW:PORT:COUP](#)

## Equivalent Softkeys

**Stimulus > Power > Port Power > Port n**

---

## Equivalent COM Command

SCPI.SOURce(Ch).POWer.PORT(Pt).LEVel.IMMediate.AMPLitude

## Syntax

Value = app.SCPI.SOURce(Ch).POWer.PORT(Pt).LEVel.IMMediate.AMPLitude

app.SCPI.SOURce(Ch).POWer.PORT(Pt).LEVel.IMMediate.AMPLitude = 10

## Type

Double (read/write)

---

Back to [SOURce](#)

## SOUR:POW:PORT:CORR

### SCPI Command

SOURce<Ch>:POWER:PORT<Pt>:CORRection[:STATe] {OFF|ON|0|1}

SOURce<Ch>:POWER:PORT<Pt>:CORRection[:STATe]?

### Description

Turns the power correction ON/OFF.

command/query

### Target

Port <Pt> of channel <Ch>,

<Ch>={{1}|2|...16}

<Pt>={{1}|2}

### Parameter

{ON|1}      ON

{OFF|0}     OFF

### Query Response

{0|1}

### Preset Value

0

### Related Commands

[SOUR:POW:PORT:CORR:COLL](#)

### Equivalent Softkeys

**Calibration > Power Calibration > Correction**



---

## Equivalent COM Command

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

## Syntax

Status = app.SCPI.SOURce(Ch).POWer.PORT(Pt).CORRection.STATe

app.SCPI.SOURce(Ch).POWer.PORT(Pt).CORRection.STATe = True

## Type

Boolean (read/write)

---

Back to [SOURce](#)

## SOUR:POW:PORT:CORR:INT?

### SCPI Command

SOURce<Ch>:POWER:PORT<Pt>:CORRection:INTerpolation[:STATus]?

### Description

Reads out the interpolation/extrapolation status of the port power correction.

query only

### Target

Port <Pt> of channel <Ch>,

<Ch>={1|2|...16}

<Pt>={1|2}

### Query Response

Status represents:

<b>NONE</b>	Correction not applied
<b>PC</b>	Correction applied exactly
<b>PC?</b>	Correction interpolated
<b>PC!</b>	Correction extrapolated

### Related Commands

[SOUR:POW:PORT:CORR:COLL](#)

### Equivalent Softkeys

None

---

## Equivalent COM Command

None

---

Back to [SOURce](#)

## SOUR:POW:PORT:CORR:COLL

### SCPI Command

SOURce<Ch>:POWER:PORT<Pt>:CORRection:COLLect[:ACQuire]

### Description

Measures the power calibration data for the port <Pt> 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.

query only

### Target

Port <Pt> of channel <Ch>,

<Ch>={[1]|2|...16}

<Pt>={[1]|2}

### Related Commands

[SOUR:POW:PORT:CORR:COLL:TABL:LOSS](#)

[SOUR:POW:PORT:CORR:COLL:TABL:LOSS:DATA](#)

### Equivalent Softkeys

Calibration > Power Calibration > Take Cal Sweep

---

### Equivalent COM Command

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

### Syntax

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

**Type**

Method

---

Back to [SOURce](#)

## SOUR:POW:PORT:CORR:COLL:TABL:LOSS:DATA

### SCPI Command

SOURce<Ch>:POWer:PORT<Pt>:CORRection:COLLect:TABLE:LOSS:DATA  
<numeric list>

SOURce<Ch>:POWer:PORT<Pt>:CORRection:COLLect:TABLE:LOSS:DATA?

### Description

Sets/gets the loss compensation table used when the power calibration is executed by the [SOUR:POW:PORT:CORR:COLL](#) command.

**Note:** If the array size is not  $1 + 2N$ , where  $N$  is equal to <numeric 1>, an error occurs. If the <numeric 2n> and <numeric 2n+1> values are out of the allowable range, the value of the limit, which is closer to the specified value will be set.

command/query

### Target

Port <Pt> of channel <Ch>,

<Ch>={[1]|2|...16}

<Pt>={[1]|2}

### Parameter

The array size is  $1+2N$ , where  $N$  is the number of measurement points.

For the  $n$ -th point, where  $n$  from 1 to  $N$ :

<numeric 1> the number of the table rows  $N$  integer from 0 to 10001;

<numeric 2n> the frequency of the  $n$ -th row of the table;

<numeric 2n+1> the loss compensation value of the  $n$ -th row of the table from –100 to +100 dB.

### Query Response

<numeric 1>, <numeric 2>, ...<numeric 2N+1>

## Related Commands

[SOUR:POW:PORT:CORR:COLL](#)

[SOUR:POW:PORT:CORR:COLL:TABL:LOSS](#)

## Equivalent Softkeys

**Calibration > Power Calibration > Loss Compens**

---

## Equivalent COM Command

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

## Syntax

Data =  
app.SCPI.SOURce(Ch).POWer.PORT(Pt).CORRection.COLLection.TABLe.LOSS.DA  
TA

app.SCPI.SOURce(Ch).POWer.PORT(Pt).CORRection.COLLection.TABLe.LOSS.DA  
TA = Data

## Type

Variant (array of Double) (read/write)

---

Back to [SOURce](#)

## SOUR:POW:PORT:CORR:COLL:TABL:LOSS

### SCPI Command

SOURce<Ch>:POWer:PORT<Pt>:CORRection:COLLect:TABLE:LOSS[:STATe]  
{OFF|ON|0|1}

SOURce<Ch>:POWer:PORT<Pt>:CORRection:COLLect:TABLE:LOSS[:STATe]?

### Description

Turns the state of the loss compensation used when the power calibration is executed by the [SOUR:POW:PORT:CORR:COLL](#) command ON/OFF.

command/query

### Target

Port <Pt> of channel <Ch>,

<Ch>={[1]|2|...16}

<Pt>={[1]|2}

### Parameter

{ON|1}      ON

{OFF|0}     OFF

### Query Response

{0|1}

### Preset Value

0

### Related Commands

[SOUR:POW:PORT:CORR:COLL](#)

[SOUR:POW:PORT:CORR:COLL:TABL:LOSS:DATA](#)



## Equivalent Softkeys

Calibration > Power Calibration > Loss Compen > Compensation {ON/OFF}

---

## Equivalent COM Command

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

## Syntax

Status = app.SCPI.SOURce(Ch).POWer.PORT(Pt).CORRection.STATe

app.SCPI.SOURce(Ch).POWer.PORT(Pt).CORRection.STATe = True

## Type

Boolean (read/write)

---

Back to [SOURce](#)

## SOUR:POW:PORT:CORR:DATA

### SCPI Command

SOURce<Ch>:POWER:PORT<Pt>:CORRection:DATA <numeric list>

SOURce<Ch>:POWER:PORT<Pt>:CORRection:DATA?

### Description

Sets or reads out the power correction array (result of power calibration executed by the [SOUR:POW:PORT:CORR:COLL](#) command).

**Note:** If the array size is not  $1 + 2N$ , where  $N$  is equal to <numeric 1>, an error occurs. If the <numeric 2n> and <numeric 2n+1> values are out of the allowable range, the value of the limit, which is closer to the specified value will be set.

command/query

### Target

Port <Pt> of channel <Ch>,

<Ch>={[1]|2|...16}

<Pt>={[1]|2}

### Parameter

The array size is NOP, where NOP is the number of measurement points.

For the n–th point, where n from 1 to NOP:

<numeric n> power correction value of the n–th point

### Query Response

<numeric 1>, <numeric 2>, ...<numeric NOP>

### Related Commands

[SOUR:POW:PORT:CORR:COLL](#)

### Equivalent Softkeys

None

---

### Equivalent COM Command

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

### Syntax

Data = app.SCPI.SOURce(Ch).POWer.PORT(Pt).CORRection.DATA

app.SCPI.SOURce(Ch).POWer.PORT(Pt).CORRection.DATA = Data

### Type

Variant (array of Double) (read/write)

---

Back to [SOURce](#)

## SOUR:POW:PORT:COUP

### SCPI Command

SOURce<Ch>:POWER:PORT:COUPle {OFF|ON|0|1}

SOURce<Ch>:POWER:PORT:COUPle?

### Description

Turns the port power couple ON/OFF. Setting the port power couple to OFF allows independent power level setting for each port.

command/query

### Target

Channel <Ch>,

<Ch>={[1]|2|...16}

### Parameter

{ON|1}      ON

{OFF|0}      OFF

### Query Response

{0|1}

### Preset Value

1

### Equivalent Softkeys

Stimulus > Power > Port Couple {ON/OFF}

---

### Equivalent COM Command

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

## Syntax

Status = app.SCPI.SOURce(Ch).POWer.PORT.COUPle

app.SCPI.SOURce(Ch).POWer.PORT.COUPle = True

## Type

Boolean (read/write)

---

### **WARNING**

Object PORT has an index of 1, which can be omitted in Visual Basic but it cannot be omitted in other programming languages.

---

---

Back to [SOURce](#)

## SOUR:POW:SLOP

### SCPI Command

SOURce<Ch>:POWER[:LEVel]:SLOPe[:DATA] <numeric>

SOURce<Ch>:POWER[:LEVel]:SLOPe[:DATA]?

### Description

Sets or reads out the power slope value for the frequency sweep type.

command/query

### Target

Channel <Ch>,

<Ch>={{[1]2|...16}}

### Parameter

<numeric> the power slope value from –2 to +2

### Unit

dB/GHz (decibel/gigahertz)

### Resolution

0.1

### Out of Range

Sets the value of the limit, which is closer to the specified value.

### Query Response

<numeric>

### Preset Value

0

## Equivalent Softkeys

Stimulus > Power > Slope

---

## Equivalent COM Command

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

## Syntax

Value = app.SCPI.SOURce(Ch).POWer.LEVel.SLOPe.DATA

app.SCPI.SOURce(Ch).POWer.LEVel.SLOPe.DATA = 0.2

## Type

Double (read/write)

---

Back to [SOURce](#)

## SOUR:POW:SLOP:STAT

### SCPI Command

SOURce<Ch>:POWER[:LEVel]:SLOPe:STATe {OFF|ON|0|1}

SOURce<Ch>:POWER[:LEVel]:SLOPe:STATe?

### Description

Turns the power slope ON/OFF. The power slope is valid for the frequency sweep type: Linear, Logarithmic, Segment.

command/query

### Target

Channel <Ch>,

<Ch>={[1]|2|...16}

### Parameter

{ON|1}      ON

{OFF|0}      OFF

### Query Response

{0|1}

### Preset Value

0

### Equivalent Softkeys

Stimulus > Power > Slope {ON/OFF}

---

### Equivalent COM Command

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



**Syntax**

Status = app.SCPI.SOURce(Ch).POWer.LEVel.SLOPe.STATe

app.SCPI.SOURce(Ch).POWer.LEVel.SLOPe.STATe = True

**Type**

Boolean (read/write)

---

Back to [SOURce](#)

## **SOUR:POW:SPAN**

### **SCPI Command**

SOURce<Ch>:POWER:SPAN <power>

SOURce<Ch>:POWER:SPAN?

### **Description**

Sets or reads out the power span when the power sweep type is active.

command/query

### **Target**

Channel <Ch>,

<Ch>={{1}|2|...16}

### **Parameter**

<power> the power sweep span value from 0 to maximum limit of the analyzer

### **Unit**

dBm (decibels above 1 milliwatt)

### **Resolution**

0.05 dBm

### **Out of Range**

Sets the value of the limit, which is closer to the specified value.

### **Query Response**

<numeric>

### **Preset Value**

Depends on the analyzer

## Equivalent Softkeys

Stimulus > Span

---

## Equivalent COM Command

SCPI.SOURce(Ch).POWer.SPAN

## Syntax

Value = app.SCPI.SOURce(Ch).POWer.SPAN

app.SCPI.SOURce(Ch).POWer.SPAN = 50

## Type

Double (read/write)

---

Back to [SOURce](#)

## SOUR:POW:STAR

### SCPI Command

SOURce<Ch>:POWer:STARt <power>

SOURce<Ch>:POWer:STARt?

### Description

Sets or reads out the power sweep start value when the power sweep type is active.

command/query

### Target

Channel <Ch>,

<Ch>={[1]2|...16}

### Parameter

<power> the power sweep start value within the power limits of the analyzer

### Unit

dBm (decibels above 1 milliwatt)

### Resolution

0.05 dBm

### Out of Range

Sets the value of the limit, which is closer to the specified value.

### Query Response

<numeric>

### Preset Value

Depends on the analyzer

## Equivalent Softkeys

Stimulus > Start

---

## Equivalent COM Command

SCPI.SOURce(Ch).POWer.STARt

## Syntax

Value = app.SCPI.SOURce(Ch).POWer.STARt

app.SCPI.SOURce(Ch).POWer.STARt = 5

## Type

Double (read/write)

---

Back to [SOURce](#)

## **SOUR:POW:STOP**

### **SCPI Command**

SOURce<Ch>:POWer:STOP <power>

SOURce<Ch>:POWer:STOP?

### **Description**

Sets or reads out the power sweep stop value when the power sweep type is active.

command/query

### **Target**

Channel <Ch>,

<Ch>={[1]|2|...16}

### **Parameter**

<power> the power sweep stop value within the power limits of the analyzer

### **Unit**

dBm (decibels above 1 milliwatt)

### **Resolution**

0.05 dBm

### **Out of Range**

Sets the value of the limit, which is closer to the specified value.

### **Query Response**

<numeric>

### **Preset Value**

Depends on the analyzer

## Equivalent Softkeys

Stimulus > Stop

---

## Equivalent COM Command

SCPI.SOURce(Ch).POWer.STOP

## Syntax

Value = app.SCPI.SOURce(Ch).POWer.STOP

app.SCPI.SOURce(Ch).POWer.STOP = 5

## Type

Double (read/write)

---

Back to [SOURce](#)

## STATus

Command		Description
<a href="#">STAT:OPER?</a>	Status System	Operation Status Event Register query
<a href="#">STAT:OPER:COND?</a>		Operation Status Condition Register query
<a href="#">STAT:OPER:ENAB</a>		Operation Status Enable Register
<a href="#">STAT:OPER:NTR</a>		Negative transition filter of Operation Status Register
<a href="#">STAT:OPER:PTR</a>		Positive transition filter of Operation Status Register
<a href="#">STAT:PRES</a>		Resets status registers
<a href="#">STAT:QUES:COND?</a>		Questionable Status Condition Register query
<a href="#">STAT:QUES:ENAB</a>		Questionable Status Enable Register
<a href="#">STAT:QUES:LIM:CHAN:COND?</a>		Questionable Limit Channel Status Condition Register query
<a href="#">STAT:QUES:LIM:CHAN:ENAB</a>		Questionable Limit Channel Status Enable Register
<a href="#">STAT:QUES:LIM:CHAN:NTR</a>		Negative transition filter of Questionable Limit Channel Status Register



Command		Description
<a href="#">STAT:QUES:LIM:CHAN:PTR</a>		Positive transition filter of Questionable Limit Channel Status Register
<a href="#">STAT:QUES:LIM:CHAN?</a>		Questionable Limit Channel Status Event Register query
<a href="#">STAT:QUES:LIM:COND?</a>		Questionable Limit Status Condition Register query
<a href="#">STAT:QUES:LIM:ENAB</a>		Questionable Limit Status Enable Register
<a href="#">STAT:QUES:LIM:NTR</a>		Negative transition filter of Questionable Limit Status Register
<a href="#">STAT:QUES:LIM:PTR</a>		Positive transition filter of Questionable Limit Status Register
<a href="#">STAT:QUES:LIM?</a>		Questionable Limit Status Event Register query
<a href="#">STAT:QUES:NTR</a>		Negative transition filter of Questionable Status Register
<a href="#">STAT:QUES:PTR</a>		Positive transition filter of Questionable Status Register
<a href="#">STAT:QUES:RLIM:CHAN:COND?</a>		Questionable Ripple Limit Channel Status Condition Register query
<a href="#">STAT:QUES:RLIM:CHAN:ENAB</a>		Questionable Ripple Limit Channel Status Enable Register

Command		Description
<a href="#">STAT:QUES:RLIM:CHAN:NTR</a>		Negative transition filter of Questionable Ripple Limit Channel Status Register
<a href="#">STAT:QUES:RLIM:CHAN:PTR</a>		Positive transition filter of Questionable Ripple Limit Channel Status Register
<a href="#">STAT:QUES:RLIM:CHAN?</a>		Questionable Ripple Limit Channel Status Event Register query
<a href="#">STAT:QUES:RLIM:COND?</a>		Questionable Ripple Limit Status Condition Register query
<a href="#">STAT:QUES:RLIM:ENAB</a>		Questionable Ripple Limit Status Enable Register
<a href="#">STAT:QUES:RLIM:NTR</a>		Negative transition filter of Questionable Ripple Limit Status Register
<a href="#">STAT:QUES:RLIM?</a>		Questionable Ripple Limit Status Event Register query
<a href="#">STAT:QUES?</a>		Questionable Status Event Register query

## STAT:OPER?

### SCPI Command

STATus:OPERation[:EVENT]?

### Description

Reads out the value of the Operation Status Event Register.

query only

### Target

Status Reporting System

### Query Response

<numeric>

### Equivalent Softkeys

None

---

### Equivalent COM Command

SCPI.STATus.OPERation.EVENT

### Syntax

Value = app.SCPI.STATus.OPERation.EVENT

### Type

Long (read/write)

---

Back to [STATus](#)

## STAT:OPER:COND?

### SCPI Command

STATus:OPERation:CONDition?

### Description

Reads out the value of the Operation Status Condition Register.

query only

### Target

Status Reporting System

### Query Response

<numeric>

### Equivalent Softkeys

None

---

### Equivalent COM Command

SCPI.STATus.OPERation.CONDition

### Syntax

Value = app.SCPI.STATus.OPERation.CONDition

### Type

Long (read/write)

---

Back to [STATus](#)

## STAT:OPER:ENAB

### SCPI Command

STATus:OPERation:ENABle <numeric>

STATus:OPERation:ENABle?

### Description

Sets or reads out the value of the Operation Status Enable Register.

command/query

### Target

Status Reporting System

### Parameter

<numeric> from 0 to 65535

### Query Response

<numeric>

### Preset Value

0

### Equivalent Softkeys

None

---

### Equivalent COM Command

SCPI.STATus.OPERation.ENABle

### Syntax

Value = app.SCPI.STATus.OPERation.ENABle

app.SCPI.STATus.OPERation.ENABle = Value

**Type**

Long (read/write)

---

Back to [STATus](#)

## STAT:OPER:NTR

### SCPI Command

STATus:OPERation:NTRansition <numeric>

STATus:OPERation:NTRansition?

### Description

Sets or reads out the value of the Negative transition filter of the Operation Status Register.

command/query

### Target

Status Reporting System

### Parameter

<numeric> from 0 to 65535

### Query Response

<numeric>

### Preset Value

0

### Equivalent Softkeys

None

---

### Equivalent COM Command

SCPI.STATus.OPERation.NTRansition

### Syntax

Value = app.SCPI.STATus.OPERation.NTRansition

app.SCPI.STATus.OPERation.NTRansition = Value

**Type**

Long (read/write)

---

Back to [STATus](#)



## STAT:OPER:PTR

### SCPI Command

STATus:OPERation:PTRansition <numeric>

STATus:OPERation:PTRansition?

### Description

Sets or reads out the value of the Positive transition filter of the Operation Status Register.

command/query

### Target

Status Reporting System

### Parameter

<numeric> from 0 to 65535

### Query Response

<numeric>

### Preset Value

65535

### Equivalent Softkeys

None

---

### Equivalent COM Command

SCPI.STATus.OPERation.PTRansition

### Syntax

Value = app.SCPI.STATus.OPERation.PTRansition

app.SCPI.STATus.OPERation.PTRansition = Value

**Type**

Long (read/write)

---

Back to [STATus](#)

# STAT:PRES

## SCPI Command

STATus:PRESet

## Description

Resets all the status registers to the factory settings.

no query

## Target

Status Reporting System

## Query Response

<numeric>

## Equivalent Softkeys

None

---

## Equivalent COM Command

SCPI.STATus.PRESet

## Syntax

app.SCPI.STATus.PRESet

## Type

Method

---

Back to [STATus](#)

## STAT:QUES:COND?

### SCPI Command

STATus:QUEStionable:CONDition?

### Description

Reads out the value of the Questionable Status Condition Register.

query only

### Target

Status Reporting System

### Query Response

<numeric>

### Equivalent Softkeys

None

---

### Equivalent COM Command

SCPI.STATus.QUEStionable.CONDition

### Syntax

Value = app.SCPI.STATus.QUEStionable.CONDition

### Type

Long (read only)

---

Back to [STATus](#)

## STAT:QUES:ENAB

### SCPI Command

STATus:QUEStionable:ENABle <numeric>

STATus:QUEStionable:ENABle?

### Description

Sets or reads out the value of the Questionable Status Enable Register.

command/query

### Target

Status Reporting System

### Parameter

<numeric> from 0 to 65535

### Query Response

<numeric>

### Preset Value

0

### Equivalent Softkeys

None

---

### Equivalent COM Command

SCPI.STATus.QUEStionable.ENABle

### Syntax

Value = app.SCPI.STATus.QUEStionable.ENABle

app.SCPI.STATus.QUEStionable.ENABle = Value

## Type

Long (read only)

---

Back to [STATus](#)

## STAT:QUES:LIM:CHAN:COND?

### SCPI Command

STATus:QUEStionable:LIMit:CHANnel<Ch>:CONDition?

### Description

Reads out the value of the Questionable Limit Channel Status Condition Register.

query only

### Target

Channel <Ch>,

<Ch>={[1]2|...16}

### Query Response

<numeric>

### Equivalent Softkeys

None

---

### Equivalent COM Command

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

### Syntax

Value = app.SCPI.STATus.QUEStionable.LIMit.CHANnel(Ch).CONDition

### Type

Long (read only)

---

Back to [STATus](#)

## STAT:QUES:LIM:CHAN:ENAB

### SCPI Command

STATus:QUEStionable:LIMit:CHANnel<Ch>:ENABle <numeric>

STATus:QUEStionable:LIMit:CHANnel<Ch>:ENABle?

### Description

Sets or reads out the value of the Questionable Limit Channel Status Enable Register.

command/query

### Target

Channel <Ch>,

<Ch>={1|2|...16}

### Parameter

<numeric> from 0 to 65535

### Query Response

<numeric>

### Preset Value

0

### Equivalent Softkeys

None

---

### Equivalent COM Command

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

### Syntax

Value = app.SCPI.STATus.QUEStionable.LIMit.CHANnel(Ch).ENABle



app.SCPI.STATus.QUEStionable.LIMit.CHANnel(Ch).ENABle = Value

**Type**

Long (read/write)

---

Back to [STATus](#)

## STAT:QUES:LIM:CHAN:NTR

### SCPI Command

STATus:QUEStionable:LIMit:CHANnel<Ch>:NTRansition <numeric>

STATus:QUEStionable:LIMit:CHANnel<Ch>:NTRansition?

### Description

Sets or reads out the value of the Negative transition filter of the Questionable Limit Channel Status Register.

command/query

### Target

Channel <Ch>,

<Ch>={1|2|...16}

### Parameter

<numeric> from 0 to 65535

### Query Response

<numeric>

### Preset Value

0

### Equivalent Softkeys

None

---

### Equivalent COM Command

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

### Syntax

Value = app.SCPI.STATus.QUEStionable.LIMit.CHANnel(Ch).NTRansition

app.SCPI.STATus.QUEStionable.LIMit.CHANnel(Ch).NTRansition = Value

**Type**

Long (read/write)

---

Back to [STATus](#)

## STAT:QUES:LIM:CHAN:PTR

### SCPI Command

STATus:QUEStionable:LIMit:CHANnel<Ch>:PTRansition <numeric>

STATus:QUEStionable:LIMit:CHANnel<Ch>:PTRansition?

### Description

Sets or reads out the value of the Positive transition filter of the Questionable Limit Channel Status Register.

command/query

### Target

Channel <Ch>,

<Ch>={[1]|2|...16}

### Parameter

<numeric> from 0 to 65535

### Query Response

<numeric>

### Preset Value

65535

### Equivalent Softkeys

None

---

### Equivalent COM Command

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

### Syntax

Value = app.SCPI.STATus.QUEStionable.LIMit.CHANnel(Ch).PTRansition

app.SCPI.STATus.QUEStionable.LIMit.CHANnel(Ch).PTRansition = Value

**Type**

Long (read/write)

---

Back to [STATus](#)

## STAT:QUES:LIM:CHAN?

### SCPI Command

STATus:QUEStionable:LIMit:CHANnel<Ch>[:EVENT]?

### Description

Reads out the value of the Questionable Limit Channel Status Event Register.

query only

### Target

Channel <Ch>,

<Ch>={[1]2|...16}

### Query Response

<numeric>

### Equivalent Softkeys

None

---

### Equivalent COM Command

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

### Syntax

Value = app.SCPI.STATus.QUEStionable.LIMit.CHANnel(Ch).EVENT

### Type

Long (read only)

---

Back to [STATus](#)

## STAT:QUES:LIM:COND?

### SCPI Command

STATus:QUEStionable:LIMit:CONDition?

### Description

Reads out the value of the Questionable Limit Status Condition Register.

query only

### Target

Status Reporting System

### Query Response

<numeric>

### Equivalent Softkeys

None

---

### Equivalent COM Command

SCPI.STATus.QUEStionable.LIMit.CONDition

### Syntax

Value = app.SCPI.STATus.QUEStionable.LIMit.CONDition

### Type

Long (read only)

---

Back to [STATus](#)

## STAT:QUES:LIM:ENAB

### SCPI Command

STATus:QUEStionable:LIMit:ENABle <numeric>

STATus:QUEStionable:LIMit:ENABle?

### Description

Sets or reads out the value of the Questionable Limit Status Enable Register.

command/query

### Target

Status Reporting System

### Parameter

<numeric> from 0 to 65535

### Query Response

<numeric>

### Preset Value

0

### Equivalent Softkeys

None

---

### Equivalent COM Command

SCPI.STATus.QUEStionable.LIMit.ENABLE

### Syntax

Value = app.SCPI.STATus.QUEStionable.LIMit.ENABLE

app.SCPI.STATus.QUEStionable.LIMit.ENABLE = Value



**Type**

Long (read/write)

---

Back to [STATus](#)

## STAT:QUES:LIM:NTR

### SCPI Command

STATus:QUEStionable:LIMit:NTRansition <numeric>

STATus:QUEStionable:LIMit:NTRansition?

### Description

Sets or reads out the value of the Negative transition filter of the Questionable Limit Status Register.

command/query

### Target

Status Reporting System

### Parameter

<numeric> from 0 to 65535

### Query Response

<numeric>

### Preset Value

0

### Equivalent Softkeys

None

---

### Equivalent COM Command

SCPI.STATus.QUEStionable.LIMit.NTRansition

### Syntax

Value = app.SCPI.STATus.QUEStionable.LIMit.NTRansition

app.SCPI.STATus.QUEStionable.LIMit.NTRansition = Value

**Type**

Long (read/write)

---

Back to [STATus](#)

## STAT:QUES:LIM:PTR

### SCPI Command

STATus:QUEStionable:LIMit:PTRansition <numeric>

STATus:QUEStionable:LIMit:PTRansition?

### Description

Sets or reads out the value of the Positive transition filter of the Questionable Limit Status Register.

command/query

### Target

Status Reporting System

### Parameter

<numeric> from 0 to 65535

### Query Response

<numeric>

### Preset Value

65535

### Equivalent Softkeys

None

---

### Equivalent COM Command

SCPI.STATus.QUEStionable.LIMit.PTRansition

### Syntax

Value = app.SCPI.STATus.QUEStionable.LIMit.PTRansition

app.SCPI.STATus.QUEStionable.LIMit.PTRansition = Value

**Type**

Long (read/write)

---

Back to [STATus](#)

## STAT:QUES:LIM?

### SCPI Command

STATus:QUEStionable:LIMit[:EVENT]?

### Description

Reads out the value of the Questionable Limit Status Event Register.

query only

### Target

Status Reporting System

### Query Response

<numeric>

### Equivalent Softkeys

None

---

### Equivalent COM Command

SCPI.STATus.QUEStionable.LIMit.EVENT

### Syntax

Value = app.SCPI.STATus.QUEStionable.LIMit.EVENT

### Type

Long (read only)

---

Back to [STATus](#)

## STAT:QUES:NTR

### SCPI Command

STATus:QUEStionable:NTRansition <numeric>

STATus:QUEStionable:NTRansition?

### Description

Sets or reads out the value of the Negative transition filter of the Questionable Status Register.

command/query

### Target

Status Reporting System

### Parameter

<numeric> from 0 to 65535

### Query Response

<numeric>

### Preset Value

0

### Equivalent Softkeys

None

---

### Equivalent COM Command

SCPI.STATus.QUEStionable.NTRansition

### Syntax

Value = app.SCPI.STATus.QUEStionable.NTRansition

app.SCPI.STATus.QUEStionable.NTRansition = Value

**Type**

Long (read/write)

---

Back to [STATus](#)



## STAT:QUES:PTR

### SCPI Command

STATus:QUEStionable:PTRansition <numeric>

STATus:QUEStionable:PTRansition?

### Description

Sets or reads out the value of the Positive transition filter of the Questionable Status Register.

command/query

### Target

Status Reporting System

### Parameter

<numeric> from 0 to 65535

### Query Response

<numeric>

### Preset Value

65535

### Equivalent Softkeys

None

---

### Equivalent COM Command

SCPI.STATus.QUEStionable.PTRansition

### Syntax

Value = app.SCPI.STATus.QUEStionable.PTRansition

app.SCPI.STATus.QUEStionable.PTRansition = Value

**Type**

Long (read/write)

---

Back to [STATus](#)

## STAT:QUES:RLIM:CHAN:COND?

### SCPI Command

STATus:QUEStionable:RLIMit:CHANnel<Ch>:CONDition?

### Description

Reads out the value of the Questionable Ripple Limit Channel Status Condition Register.

query only

### Target

Channel <Ch>,

<Ch>={1|2|...16}

### Query Response

<numeric>

### Equivalent Softkeys

None

---

### Equivalent COM Command

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

### Syntax

Value = app.SCPI.STATus.QUEStionable.RLIMit.CHANnel(Ch).CONDition

### Type

Long (read only)

---

Back to [STATus](#)

## STAT:QUES:RLIM:CHAN:ENAB

### SCPI Command

STATus:QUEStionable:RLIMit:CHANnel<Ch>:ENABle <numeric>

STATus:QUEStionable:RLIMit:CHANnel<Ch>:ENABle?

### Description

Sets or reads out the value of the Questionable Ripple Limit Channel Status Enable Register.

command/query

### Target

Channel <Ch>,

<Ch>={[1]|2|...16}

### Parameter

<numeric> from 0 to 65535

### Query Response

<numeric>

### Preset Value

0

### Equivalent Softkeys

None

---

### Equivalent COM Command

SCPI.STATus.QUEStionable.RLIMit.CHANnel(Ch).ENABle

### Syntax

Value = app.SCPI.STATus.QUEStionable.RLIMit.CHANnel(Ch).ENABle

app.SCPI.STATus.QUEStionable.RLIMit.CHANnel(Ch).ENABle = Value

**Type**

Long (read/write)

---

Back to [STATus](#)

## STAT:QUES:RLIM:CHAN:NTR

### SCPI Command

STATus:QUEStionable:RLIMit:CHANnel<Ch>:NTRansition <numeric>

STATus:QUEStionable:RLIMit:CHANnel<Ch>:NTRansition?

### Description

Sets or reads out the value of the Negative transition filter of the Questionable Ripple Limit Channel Status Register.

command/query

### Target

Channel <Ch>,

<Ch>={1|2|...16}

### Parameter

<numeric> from 0 to 65535

### Query Response

<numeric>

### Preset Value

0

### Equivalent Softkeys

None

---

### Equivalent COM Command

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

### Syntax

Dim Value As Long

Value = app.SCPI.STATus.QUEStionable.RLIMit.CHANnel(Ch).NTRansition

app.SCPI.STATus.QUEStionable.RLIMit.CHANnel(Ch).NTRansition = Value

### **Type**

Long (read/write)

---

Back to [STATus](#)

## STAT:QUES:RLIM:CHAN:PTR

### SCPI Command

STATus:QUEStionable:RLIMit:CHANnel<Ch>:PTRansition <numeric>

STATus:QUEStionable:RLIMit:CHANnel<Ch>:PTRansition?

### Description

Sets or reads out the value of the Positive transition filter of the Questionable Ripple Limit Channel Status Register.

command/query

### Target

Channel <Ch>,

<Ch>={[1]|2|...16}

### Parameter

<numeric> from 0 to 65535

### Query Response

<numeric>

### Preset Value

65535

### Equivalent Softkeys

None

---

### Equivalent COM Command

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

### Syntax

Value = app.SCPI.STATus.QUEStionable.RLIMit.CHANnel(Ch).PTRansition



app.SCPI.STATus.QUEStionable.RLIMit.CHANnel(Ch).PTRansition = Value

**Type**

Long (read/write)

---

Back to [STATus](#)

## STAT:QUES:RLIM:CHAN?

### SCPI Command

STATus:QUEStionable:RLIMit:CHANnel<Ch>[:EVENTt]?

### Description

Reads out the value of the Questionable Ripple Limit Channel Status Event Register.

query only

### Target

Channel <Ch>,

<Ch>={[1]|2|...16}

### Query Response

<numeric>

### Equivalent Softkeys

None

---

### Equivalent COM Command

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

### Syntax

Value = app.SCPI.STATus.QUEStionable.RLIMit.CHANnel(Ch).EVENT

### Type

Long (read only)

---

Back to [STATus](#)

## STAT:QUES:RLIM:COND?

### SCPI Command

STATus:QUEStionable:RLIMit:CONDition?

### Description

Reads out the value of the Questionable Ripple Limit Status Condition Register.

query only

### Target

Status Reporting System

### Query Response

<numeric>

### Equivalent Softkeys

None

---

### Equivalent COM Command

SCPI.STATus.QUEStionable.RLIMit.CONDition

### Syntax

Value = app.SCPI.STATus.QUEStionable.RLIMit.CONDition

### Type

Long (read only)

---

Back to [STATus](#)

## STAT:QUES:RLIM:ENAB

### SCPI Command

STATus:QUEStionable:RLIMit:ENABle <numeric>

STATus:QUEStionable:RLIMit:ENABle?

### Description

Sets or reads out the value of the Questionable Ripple Limit Status Enable Register.

command/query

### Target

Status Reporting System

### Parameter

<numeric> from 0 to 65535

### Query Response

<numeric>

### Preset Value

65535

### Equivalent Softkeys

None

---

### Equivalent COM Command

SCPI.STATus.QUEStionable.RLIMit.ENABle

### Syntax

Value = app.SCPI.STATus.QUEStionable.RLIMit.ENABle

app.SCPI.STATus.QUEStionable.RLIMit.ENABle = Value

**Type**

Long (read/write)

---

Back to [STATus](#)

## STAT:QUES:RLIM:NTR

### SCPI Command

STATus:QUEStionable:RLIMit:NTRansition <numeric>

STATus:QUEStionable:RLIMit:NTRansition?

### Description

Sets or reads out the value of the Negative transition filter of the Questionable Ripple Limit Status Register.

command/query

### Target

Status Reporting System

### Parameter

<numeric> from 0 to 65535

### Query Response

<numeric>

### Preset Value

0

### Equivalent Softkeys

None

---

### Equivalent COM Command

SCPI.STATus.QUEStionable.RLIMit.NTRansition

### Syntax

Value = app.SCPI.STATus.QUEStionable.RLIMit.NTRansition

app.SCPI.STATus.QUEStionable.RLIMit.NTRansition = Value

**Type**

Long (read/write)

---

Back to [STATus](#)

## STAT:QUES:RLIM:PTR

### SCPI Command

STATus:QUEStionable:RLIMit:PTRansition <numeric>

STATus:QUEStionable:RLIMit:PTRansition?

### Description

Sets or reads out the value of the Positive transition filter of the Questionable Ripple Limit Status Register.

command/query

### Target

Status Reporting System

### Parameter

<numeric> from 0 to 65535

### Query Response

<numeric>

### Preset Value

65535

### Equivalent Softkeys

None

---

### Equivalent COM Command

SCPI.STATus.QUEStionable.RLIMit.PTRansition

### Syntax

Value = app.SCPI.STATus.QUEStionable.RLIMit.PTRansition

app.SCPI.STATus.QUEStionable.RLIMit.PTRansition = Value



**Type**

Long (read/write)

---

Back to [STATus](#)

## STAT:QUES:RLIM?

### SCPI Command

STATus:QUEStionable:RLIMit[:EVENT]?

### Description

Reads out the value of the Questionable Ripple Limit Status Event Register.

query only

### Target

Status Reporting System

### Query Response

<numeric>

### Equivalent Softkeys

None

---

### Equivalent COM Command

SCPI.STATus.QUEStionable.RLIMit.EVENT

### Syntax

Value = app.SCPI.STATus.QUEStionable.RLIMit.EVENT

### Type

Long (read only)

---

Back to [STATus](#)

## STAT:QUES?

### SCPI Command

STATus:QUEStionable[:EVENT]?

### Description

Reads out the value of the Questionable Status Event Register.

query only

### Target

Status Reporting System

### Query Response

<numeric>

### Equivalent Softkeys

None

---

### Equivalent COM Command

SCPI.STATus.QUEStionable.EVENT

### Syntax

Value = app.SCPI.STATus.QUEStionable.EVENT

### Type

Long (read only)

---

Back to [STATus](#)

## SYSTem

Command		Description
<a href="#">SYST:BEEP:COMP:IMM</a>	Analyzer Parameters	Generates completion beep
<a href="#">SYST:BEEP:COMP:STAT</a>		Completion beeper ON/OFF
<a href="#">SYST:BEEP:WARN:IMM</a>		Generates warning beep
<a href="#">SYST:BEEP:WARN:STAT</a>		Warning beeper ON/OFF
<a href="#">SYST:CAP:CURR:CONS?</a>		Capability to measure current consumption
<a href="#">SYST:CURR:CONS?</a>		Measured current consumption
<a href="#">SYST:CORR</a>		System correction ON/OFF
<a href="#">SYST:CYCL:TIME:MEAS?</a>		Measured cycle
<a href="#">SYST:CYCL:TIME:METH</a>		Cycle time measurement method
<a href="#">SYST:CYCL:TIME:REST</a>		Restart cycle time measurement
<a href="#">SYST:DATE</a>		Current date
<a href="#">SYST:READ?</a>		Analyzer readiness status

Command		Description
<a href="#">SYST:REC:OVER:POW</a>		Power trip ON/OFF
<a href="#">SYST:TEMP:SENS?</a>		Reads the Analyzer temperature
<a href="#">SYST:TERM</a>		Analyzer software shutdown
<a href="#">SYST:TIME</a>		Current time
<a href="#">SYST:CAP:IFBW:MAX?</a>	Analyzer Capabilities	Upper limit of IFBW
<a href="#">SYST:CAP:IFBW:MIN?</a>		Lower limit of IFBW
<a href="#">SYST:CONN:SER:NUMB</a>		Analyzer Serial N
<a href="#">SYST:SERV:PVER:INT</a>		Performance verification period
<a href="#">SYST:SERV:PVER:LAST</a>		Performance verification last date
<a href="#">SYST:SERV:PVER:NEXT</a>		Performance verification next date
<a href="#">SYST:COMM:ECAL:CHEC</a>	Automatic Calibration Module	"CHECK" module state
<a href="#">SYST:COMM:ECAL:DATA?</a>		Characterization data of AutoCal module
<a href="#">SYST:COMM:ECAL:FREQ:DATA?</a>		Characterization frequency array of AutoCal module

Command		Description
<a href="#">SYST:COMM:ECAL:POIN?</a>		Number of characterization points of AutoCal module
<a href="#">SYST:COMM:ECAL:IMP</a>		Impedance state of module port
<a href="#">SYST:COMM:ECAL:READ?</a>		Module readiness status
<a href="#">SYST:COMM:ECAL:TEMP:SENS?</a>		Module temperature
<a href="#">SYST:COMM:ECAL:THRU</a>		"THRU" module state
<a href="#">SYST:COMM:PSEN:NI568x:RES:N AME</a>	Power Sensor Settings	NI568x power sensor resource name
<a href="#">SYST:COMM:PSEN:READ?</a>		Power sensor readiness
<a href="#">SYST:COMM:PSEN:TYPE</a>		Power sensor type
<a href="#">SYST:COMM:PSEN:ZERO</a>	Power Calibration	Zeroes the power sensor
<a href="#">SYST:ERR?</a>	Status System	Reads the error message queue
<a href="#">SYST:TEST?</a>		Textual description of Analyzer self-test
<a href="#">SYST:FREQ:EXT:RFP:POW</a>	Frequency Extension System	RF Port Power

Command		Description
<a href="#">SYST:FREQ:EXT:RFP:PSL</a>		RF Port Power Slope
<a href="#">SYST:FREQ:EXT:LOP:POW</a>		LO Port Power
<a href="#">SYST:FREQ:EXT:LOP:PSL</a>		LO Port Power Slope
<a href="#">SYST:FREQ:EXT:TYPE</a>		Frequency extender type
<a href="#">SYST:FREQ:EXT:PORT:CONN?</a>		Extender connection status
<a href="#">SYST:FREQ:EXT:PORT:SER?</a>		Extender serial number
<a href="#">SYST:FREQ:EXT:PORT:TEMP:SENS?</a>		Frequency extender temperature
<a href="#">SYST:HIDE</a>	Interface Settings	Minimizes the Analyzer window
<a href="#">SYST:LOC</a>		Sets the local mode
<a href="#">SYST:REM</a>		Sets the remote mode
<a href="#">SYST:RWL</a>		Sets the remote mode with lock
<a href="#">SYST:SHOW</a>		Restores the Analyzer window

Command		Description
<a href="#">SYST:PRES</a>	Presets	Reset to default settings
<a href="#">SYST:REC:DIR:ACC</a>	Direct Receiver Access	Direct access ON/OFF



## SYST:BEEP:COMP:IMM

### SCPI Command

SYSTem:BEEPer:COMPlate:IMMediate

### Description

Generates a beep to notify of the completion of the operation.

no query

### Equivalent Softkeys

System > Misc Setup > Beeper > Test Beep Complete

---

### Equivalent COM Command

SCPI.SYSTem.BEEPer.COMPlate.IMMediate

### Syntax

app.SCPI.SYSTem.BEEPer.COMPlate.IMMediate

### Type

Method

---

Back to [SYSTem](#)

## SYST:BEEP:COMP:STAT

### SCPI Command

SYSTem:BEEPer:COMPlate:STATe {OFF|ON|0|1}

SYSTem:BEEPer:COMPlate:STATe?

### Description

Turns the beeper denoting completion of the operation ON/OFF.

command/query

### Parameter

{ON|1}      ON

{OFF|0}      OFF

### Query Response

{0|1}

### Preset Value

1

### Equivalent Softkeys

System > Misc Setup > Beeper > Beep complete

---

### Equivalent COM Command

SCPI.SYSTem.BEEPer.COMPlate.STATe

### Syntax

Status = app.SCPI.SYSTem.BEEPer.COMPlate.STATe

app.SCPI.SYSTem.BEEPer.COMPlate.STATe = False

## Type

Boolean (read/write)

---

Back to [SYSTem](#)

## **SYST:BEEP:WARN:IMM**

### **SCPI Command**

SYSTem:BEEPer:WARNing:IMMediate

### **Description**

Generates a beep to signify a warning.

no query

### **Equivalent Softkeys**

**System > Misc Setup > Beeper > Test Beep Warning**

---

### **Equivalent COM Command**

SCPI.SYSTem.BEEPer.WARNing.IMMediate

### **Syntax**

app.SCPI.SYSTem.BEEPer.WARNing.IMMediate

### **Type**

Method

---

Back to [SYSTem](#)

## SYST:BEEP:WARN:STAT

### SCPI Command

SYSTem:BEEPer:WARNing:STATe {OFF|ON|0|1}

SYSTem:BEEPer:WARNing:STATe?

### Description

Turns the beeper signifying a warning ON/OFF.

command/query

### Parameter

{ON|1}      ON

{OFF|0}      OFF

### Query Response

{0|1}

### Preset Value

1

### Equivalent Softkeys

System > Misc Setup > Beeper > Beep Warning

---

### Equivalent COM Command

SCPI.SYSTem.BEEPer.COMPlete.STATe

### Syntax

Status = app.SCPI.SYSTem.BEEPer.COMPlete.STATe

app.SCPI.SYSTem.BEEPer.COMPlete.STATe = False

## Type

Boolean (read/write)

---

Back to [SYSTem](#)

## **SYST:CAP:IFBW:MAX?**

### **SCPI Command**

SYSTem:CAPability:IFBW:MAXimum?

### **Description**

Reads out the upper limit of the IFBW.

query only

### **Query Response**

<numeric>

### **Unit**

Hz (Hertz)

### **Equivalent Softkeys**

None

---

### **Equivalent COM Command**

None

---

Back to [SYSTem](#)

## **SYST:CAP:IFBW:MIN?**

### **SCPI Command**

SYSTem:CAPability:IFBW:MINimum?

### **Description**

Reads out the lower limit of the IFBW.

query only

### **Query Response**

<numeric>

### **Unit**

Hz (Hertz)

### **Equivalent Softkeys**

None

---

### **Equivalent COM Command**

None

---

Back to [SYSTem](#)



## **SYST:CAP:CURR:CONS?**

### **SCPI Command**

SYSTem:CAPability:CURRent:CONSumption?

### **Description**

Returns whether or not the Analyzer has its current consumption measurement.

query only

### **Query Response**

**1** Measurement exist

**0** Measurement not exist

### **Equivalent Softkeys**

None

---

### **Equivalent COM Command**

None

---

Back to [SYSTem](#)

## **SYST:CURR:CONS?**

### **SCPI Command**

SYSTem:CURRent:CONSumption?

### **Description**

Reads out the current consumption of the Analyzer.

query only

### **Query Response**

<numeric>

### **Unit**

A (Ampere)

### **Equivalent Softkeys**

None

---

### **Equivalent COM Command**

None

---

Back to [SYSTem](#)

# SYST:COMM:ECAL:CHEC

## SCPI Command

SYSTem:COMMunicate:ECAL:CHECK

## Description

Sets the CHECK state of the AutoCal module.

command only

## Equivalent Softkeys

None

---

## Equivalent COM Command

SCPI.SYSTem.COMMunicate.ECAL.CHECK

## Syntax

app.SCPI.SYSTem.COMMunicate.ECAL.CHECK

## Type

Method

---

Back to [SYSTem](#)

## SYST:COMM:ECAL:DATA?

### SCPI Command

SYSTem:COMMunicate:ECAL:DATA? <path>, <impedance> [,<characterization>]

### Description

Reads out the AutoCal module characterization data. One command call returns one S-parameter array according to the specified parameters. The data is thermo compensated or not depending on the current setting [SENS:CORR:COLL:ECAL:THER:COMP](#).

The array size is 2N, where N is the number of points of the specified characterization (See the [SYST:COMM:ECAL:POIN?](#) command). For the n-th point, where n is from 1 to N:

<numeric 2n-1> real part of S-parameter at the n-th characterization point

<numeric 2n> imaginary part of S-parameter at the n-th characterization point

query only

### Parameter

**<path>:** {A|B|C|D|AB|AC|AD|BC|BD|CD|CHECK} specifies the port number or port pair or check state

**<impedance>:** {SHORT|OPEN|LOAD|OPEN2|LOAD2|S11|S12|S21|S22|...|S44} specifies the impedance state or S-parameter

**<characterization>:** {[FACTory]|USER1|USER2|USER3} specifies the name of the characterization, if omitted the the factory characterization is used

The allowable combinations of parameter <path> and parameter <impedance> are as follows:

<path>	<impedance>	Description
A, B, C, D	SHOTt, OPEN, LOAD, OPEN2, LOAD2	S11-parameter of the reflection state

<b>AB, AC, AD, BC, BD, CD</b>	<b>S11, S21, S12, S22</b>	S-parameter of the THRU state
<b>CHECK</b>	<b>S11, S21 ... S44</b>	S-parameter of the CHECK state

### Query Response

<numeric 1>, <numeric 2>, ...<numeric 2N>.

### Equivalent Softkeys

None

---

### Equivalent COM Command

None

Back to [SYSTem](#)

## **SYST:COMM:ECAL:FREQ:DATA?**

### SCPI Command

SYSTem:COMMunicate:ECAL:FREQuency:DATA? [<characterization>]

### Description

Reads out the AutoCal module characterization frequency array.

The array size is N, where N is the number of points of the specified characterization (See the [SYST:COMM:ECAL:POIN?](#) command). For the n-th point, where n is from 1 to N:

<numeric n>      frequency value at the n-th characterization point

query only

### Parameter

**<characterization>:** {[FACTory][USER1][USER2][USER3]} specifies the name of the characterization, if omitted the the factory characterization is used

## Query Response

<numeric 1>, <numeric 2>, ...<numeric N>.

## Equivalent Softkeys

None

---

## Equivalent COM Command

None

---

Back to [SYSTem](#)

## **SYST:COMM:ECAL:POIN?**

## SCPI Command

SYSTem:COMMunicate:ECAL:POINts? [<characterization>]

## Description

Reads out the AutoCal module characterization point number. If the characterization does not exist then returns 0.

query only

## Parameter

**<characterization>:** {[FACTory]|USER1|USER2|USER3} specifies the name of the characterization, if omitted the the factory characterization is used

## Query Response

<numeric>

## Equivalent Softkeys

None

---

## Equivalent COM Command

None

---

Back to [SYSTem](#)

## **SYST:COMM:ECAL:IMP**

### **SCPI Command**

SYSTem:COMMunicate:ECAL:IMPedance <port>,<char>

SYSTem:COMMunicate:ECAL:IMPedance? <port>

### **Description**

Sets or reads out the impedance state of the specified port of the AutoCal module.

command/query

### **Parameter**

<port> : Port number of the AutoCal module

<char> Specifies the impedance state:

<b>OPEN</b>	OPEN impedance state
<b>SHORT</b>	SHORT impedance state
<b>LOAD</b>	LOAD impedance state
<b>LOAD2</b>	LOAD2 impedance state
<b>OPEN2</b>	OPEN2 impedance state

### **Query Response**

{OPEN|SHOR|LOAD|THRU|LOAD2|OPEN2}

### **Preset Value**

LOAD

## **Equivalent Softkeys**

None



---

## Equivalent COM Command

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

## Syntax

Param = app.SCPI.SYSTem.COMMunicate.ECAL.IMPedance(Pt)

app.SCPI.SYSTem.COMMunicate.ECAL.IMPedance(Pt) = "OPEN"

## Type

String (read/write)

---

Back to [SYSTem](#)

# SYST:COMM:ECAL:READY?

## SCPI Command

SYSTem:COMMunicate:ECAL:READY?

## Description

Reads out the readiness status of the AutoCal Module. 1 indicates that the AutoCal Module is ready.
query only

## Query Response

{0|1}, 1 — the module is ready.

## Equivalent Softkeys

None

---

## Equivalent COM Command

None

---

Back to [SYSTem](#)

## SYST:COMM:ECAL:TEMP:SENS?

### SCPI Command

SYSTem:COMMunicate:ECAL:TEMPerature:SENSor?

### Description

Reads out the temperature of the AutoCal module connected to the Analyzer.

query only

### Target

AutoCal module

### Unit

°C (degrees Celsius)

### Query Response

<numeric>

### Equivalent Softkeys

None

---

### Equivalent COM Command

SCPI.SYSTem.COMMunicate.ECAL.TEMPerature.SENSor

### Syntax

Value = app.SCPI.SYSTem.COMMunicate.ECAL.TEMPerature.SENSor

### Type

Double (read)

---

Back to [SYSTem](#)

# SYST:COMM:ECAL:THRU

## SCPI Command

SYSTem:COMMunicate:ECAL:THRU <port1>,<port2>

## Description

Sets the THRU state between the specified 2 ports of the AutoCal module.

command only

## Parameter

<port1>      The first port number of the AutoCal module

<port2>      The second port number of the AutoCal module

## Equivalent Softkeys

None

---

## Equivalent COM Command

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

## Syntax

app.SCPI.SYSTem.COMMunicate.ECAL.THRU(1, 2)

## Type

Method

---

Back to [SYSTem](#)

## SYST:COMM:PSEN:NI568x:RES:NAME

### SCPI Command

SYSTem:COMMunicate:PSEnSor:NI568x:RESource:NAME <string>

SYSTem:COMMunicate:PSEnSor:RESource:NAME?

### Description

Sets or reads out the NI568x power sensor resource name to be used in a source power calibration.

command/query

### Parameter

<string> Resource name

### Query Response

<string>

### Preset Value

"COM3"

### Equivalent Softkeys

**System > Misc Setup > Power Meter Settings > Power Meter > NI568x > Resource Name**

---

### Equivalent COM Command

None

---

Back to [SYSTem](#)

# SYST:COMM:PSEN:READ?

## SCPI Command

SYSTem:COMMunicate:PSEnSor:READy?

## Description

Reads out the readiness status of the Power Sensor. 1 indicates that the Power Sensor is ready.
query only

## Query Response

{0|1}, 1 — the Power Sensor is ready.

## Equivalent Softkeys

None

---

## Equivalent COM Command

None

---

Back to [SYSTem](#)

## SYST:COMM:PSEN:TYPE

### SCPI Command

SYSTem:COMMunicate:PSEnSor:TYPE <char>

SYSTem:COMMunicate:PSEnSor:TYPE?

### Description

Selects the power sensor type to be used in a source power calibration.

command/query

### Parameter

<char> Choose from:

<b>NRPZ</b>	Rohde&Schwarz NRP-Z series Sensors
<b>NRPxT</b>	Rohde&Schwarz NRPxT series Sensors
<b>NRVS</b>	Rohde&Schwarz NRVS power meter
<b>U848x</b>	Keysight U848x series Sensors
<b>U20xx</b>	Keysight U20xx series Sensors
<b>LB59xx</b>	LadyBug LB59xx USB Power Sensor
<b>LBxxx</b>	LadyBug LBxxxx USB Power Sensor (LB478A, LB479A, LB480A, LB559A, LB579A, LB589A)
<b>NI568x</b>	NI USB-568x RF Power Sensors

### Query Response

{NRPZ|NRPxT|NRVS|U848x|U20xx|LB59xx|LBxxx|NI568x}

### Preset Value

NRPZ

## **Equivalent Softkeys**

**System > Misc Setup > Power Meter Settings > Power Meter {NRPZ | NRPxT | NRVS | U848x | U20xx | LB59xx | LBxxx | NI568x}**



---

**Equivalent COM Command**

None

---

Back to [SYSTem](#)

## SYST:COMM:PSEN:ZERO

### SCPI Command

SYSTem:COMMunicate:PSEnSor:ZEROing

### Description

Executes the zeroing procedure of the power sensor.

**Note:** The power meter sensor can be connected to the port, since during this procedure, the output signal of the port is turned off the RF power.

command only

### Equivalent Softkeys

Calibration > Power Calibration > Power Sensor Zero Correction

---

### Equivalent COM Command

None

---

Back to [SYSTem](#)

## SYST:CONN:SER:NUMB

### SCPI Command

SYSTem:CONNect:SERial:NUMBer <string>

SYSTem:CONNect:SERial:NUMBer?

### Description

Connects the current program instance to the analyzer with specified serial number. If there is no analyzer with the specified serial number, the program goes into the NOT READY state. In order to allow the program to connect to the analyzer with any serial number, write 0 with this command.

The query returns the serial number of the connected analyzer. If program in NOT READY state the query returns the value set by previous command.

command/query

### Parameter

<string> serial number of 8 digits, or 0 (auto-detect)

### Query Response

string of 8 digits, or 0

### Preset Value

0 (auto-detect)

### Equivalent Softkeys

System > Misc Setup > Analyzer Serial N

---

### Equivalent COM Command

None

---

Back to [SYSTem](#)

## SYST:CORR

### SCPI Command

SYSTem:CORRection[:STATe] {OFF|ON|0|1}

SYSTem:CORRection[:STATe]?

### Description

Turns the system correction ON/OFF. The system correction is the factory full one-port calibration performed at the port connectors.

command/query

### Parameter

{ON|1}      ON

{OFF|0}      OFF

### Query Response

{0|1}

### Preset Value

1

### Equivalent Softkeys

System > Misc Setup > System Correction

---

### Equivalent COM Command

SCPI.SYSTem.CORRection.STATe

### Syntax

Status = app.SCPi.SYSTem.CORRection.STATe

app.SCPi.SYSTem.CORRection.STATe = False

## Type

Boolean (read/write)

---

Back to [SYSTem](#)

## **SYST:CYCL:TIME:MEAS?**

### **SCPI Command**

SYSTem:CYCLe:TIME:MEASurement?

### **Description**

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.

query only

### **Target**

Analyzer

### **Unit**

sec (second)

### **Query Response**

<numeric>

### **Equivalent Softkeys**

**Display > Properies > Cycle Time**

---

### **Equivalent COM Command**

SCPI.SYSTem.CYCLe.TIME.MEASurement

### **Syntax**

Value = app.SCPI.SYSTem.CYCLe.TIME.MEASurement

**Type**

Double (read only)

---

Back to [SYSTem](#)

# SYST:CYCL:TIME:METH

## SCPI Command

SYSTem:CYCLe:TIME:METhod <char>

SYSTem:CYCLe:TIME:METhod?

## Description

Selects the cycle time measurement method.

command/query

## Parameter

<char> Choose from:

**AVERaging**      Averaging method

**MAXHold**      Max hold method

## Query Response

{AVER|MAXH}

## Preset Value

AVER

## Related Commands

[SYST:CYCL:TIME:MEAS?](#)

## Equivalent Softkeys

Display > Properies > Cycle Time > Method {Averaging | Max Hold }



---

## Equivalent COM Command

None

---

Back to [SYSTem](#)

## **SYST:CYCL:TIME:REST**

### **SCPI Command**

SYSTem:CYCLe:TIME:REStart

### **Description**

Restarts the averaging or maximum hold of the cycle time measurement, depending on the selected method.

command only

### **Equivalent Softkeys**

**Display > Properies > Cycle Time > Restart**

---

### **Equivalent COM Command**

None

---

Back to [SYSTem](#)

# SYST:DATE

## SCPI Command

SYSTem:DATE <numeric 1>,<numeric 2>,<numeric 3>

SYSTem:DATE?

## Description

Sets or reads out the current date.

command/query

## Parameter

<numeric 1>            Year from 1900 to 2100

<numeric 2>            Month from 1 to 12

<numeric 3>            Day from 1 to 31

## Query Response

<numeric 1>, <numeric 2>, <numeric 3>

## Equivalent Softkeys

None

---

## Equivalent COM Command

SCPI.SYSTem.DATE

## Syntax

Data = app.SCPI.SYSTem.DATE

app.SCPI.SYSTem.DATE = Array(2009, 9, 9)

## Type

Variant (array of long) (read/write)

---

Back to [SYSTem](#)

## SYST:ERR?

### SCPI Command

SYSTem:ERRor[:NEXT]?

### Description

Reads out the error message when executing SCPI commands, from the FIFO (First In First Out) error queue stored in the Analyzer. The read out error is deleted from the error queue. The [\\*CLS](#) command clears the error queue. The maximum size of the queue is 100 messages.

command/query

### Query Response

<numeric>, <string>

Where:

<numeric> — error code,

<string> — error message.

If there is no error in the queue, "0, No error" is read out.

### Equivalent Softkeys

None

---

### Equivalent COM Command

None

---

Back to [SYSTem](#)

## **SYST:FREQ:EXT:RFR:POW**

### **SCPI Command**

SYSTem:FREQuency:EXTender:RFPort:POWer <numeric>

SYSTem:FREQuency:EXTender:RFPort:POWer?

### **Description**

Sets or reads out the RF Port Power when the Analyzer is configured to work with a frequency extender.

command/query

### **Parameter**

<numeric> the power value.

### **Unit**

dBm (decibel relative to 1 milliwatt)

### **Out of Range**

Sets the value of the limit, which is closer to the specified value.

### **Query Response**

<numeric>

### **Preset Value**

0

### **Equivalent Softkeys**

**System > Misc Setup > Frequency Extender > RF Port Power**

---

**Equivalent COM Command**

None

---

Back to [SYSTem](#)

## **SYST:FREQ:EXT:RFP:PSL**

### **SCPI Command**

SYSTem:FREQuency:EXTender:RFPort:PSLope <numeric>

SYSTem:FREQuency:EXTender:RFPort:PSLope?

### **Description**

Sets or reads out the RF Port Power Slope when the Analyzer is configured to work with a frequency extender.

command/query

### **Parameter**

<numeric> the slope value.

### **Unit**

dB/GHz (decibel / gigahertz)

### **Out of Range**

Sets the value of the limit, which is closer to the specified value.

### **Query Response**

<numeric>

### **Preset Value**

0

### **Equivalent Softkeys**

**System > Misc Setup > Frequency Extender > RF Power Slope**



---

**Equivalent COM Command**

None

---

Back to [SYSTem](#)

## **SYST:FREQ:EXT:LOP:POW**

### **SCPI Command**

SYSTem:FREQuency:EXTender:LOPort:POWer <numeric>

SYSTem:FREQuency:EXTender:LOPort:POWer?

### **Description**

Sets or reads out the LO Port Power when the Analyzer is configured to work with a frequency extender.

command/query

### **Parameter**

<numeric> the power value.

### **Unit**

dBm (decibel relative to 1 milliwatt)

### **Out of Range**

Sets the value of the limit, which is closer to the specified value.

### **Query Response**

<numeric>

### **Preset Value**

0

### **Equivalent Softkeys**

**System > Misc Setup > Frequency Extender > LO Port Power**

---

### Equivalent COM Command

None

---

Back to [SYSTem](#)

## **SYST:FREQ:EXT:LOP:PSL**

### **SCPI Command**

SYSTem:FREQuency:EXTender:LOPort:PSLope <numeric>

SYSTem:FREQuency:EXTender:LOPort:PSLope?

### **Description**

Sets or reads out the LO Port Power Slope when the Analyzer is configured to work with a frequency extender.

command/query

### **Parameter**

<numeric> the slope value.

### **Unit**

dB/GHz (decibel / gigahertz)

### **Out of Range**

Sets the value of the limit, which is closer to the specified value.

### **Query Response**

<numeric>

### **Preset Value**

0

### **Equivalent Softkeys**

**System > Misc Setup > Frequency Extender > LO Power Slope**

---

**Equivalent COM Command**

None

---

Back to [SYSTem](#)

## SYST:FREQ:EXT:TYPE

### SCPI Command

SYSTem:FREQuency:EXTender:TYPE <char>

SYSTem:FREQuency:EXTender:TYPE?

### Description

Selects or reads the frequency extender type. When the new type is selected the connection will close, as the program will restart.

command/query

### Parameter

<char> Choose from:

<b>NONE</b>	None
<b>FEV15</b>	FEV-15 50 – 75 GHz
<b>FEV12</b>	FEV-15 60 – 90 GHz
<b>FEV10</b>	FEV-15 75 – 110 GHz
<b>FET1854</b>	FET-1854 18 – 54 GHz
<b>CUSTom</b>	Custom

### Query Response

{NONE|FEV15|FEV12|FEV10|FET1854|CUST}

### Preset Value

NONE

## Equivalent Softkeys

System > Misc Setup > Frequency Extender > {None | FEV15 | FEV12 | FEV10  
| FET1854 | Custom}

---

## Equivalent COM Command

None

---

Back to [SYSTem](#)

## SYST:FREQ:EXT:PORT:CONN?

### SCPI Command

SYSTem:FREQuency:EXTender:PORT<Pt>:CONNect?

### Description

Reads out whether the frequency extender is connected to the port number <Pt>. The actual state is read out when the FET-1854 is configured. Always reads 1 when other type of frequency extender is configured.

query only

### Target

Port <Pt>,

<Pt>={1|2}

### Query Response

1 Connected

0 Not connected

### Equivalent Softkeys

None

---

### Equivalent COM Command

None

---

Back to [SYSTem](#)



## SYST:FREQ:EXT:PORT:SER?

### SCPI Command

SYSTem:FREQuency:EXTender:PORT<Pt>:SERial?

### Description

Reads out the serial number of the frequency extender connected to the port number <Pt>. The actual serial number is read out when the FET-1854 is configured. Always reads "00000000" when other type of frequency extender is configured.

query only

### Target

Port <Pt>,

<Pt>={1|2}

### Query Response

<String> the serial number 8 symbols

### Equivalent Softkeys

None

---

### Equivalent COM Command

None

---

Back to [SYSTem](#)

## SYST:FREQ:EXT:PORT:TEMP:SENS?

### SCPI Command

SYSTem:FREQuency:EXTender:PORT<Pt>:TEMPerature:SENSor?

### Description

Reads out the temperature of the frequency extender connected to the port number <Pt>.

query only

### Target

Port <Pt>,

<Pt>={{1}|2}

### Unit

°C (degrees Celsius)

### Equivalent Softkeys

None

---

### Equivalent COM Command

None

---

Back to [SYSTem](#)

# SYST:HIDE

## SCPI Command

SYSTem:HIDE

## Description

Minimizes the Analyzer main window, removing it from the desktop.

query only

## Related Commands

[SYST:SHOW](#)

## Equivalent Softkeys

None

---

## Equivalent COM Command

SCPI.SYSTem.HIDE

## Syntax

app.SCPI.SYSTem.HIDE

## Type

Method

---

Back to [SYSTem](#)

# SYST:LOC

## SCPI Command

SYSTem:LOCal

## Description

Sets the Analyzer to the local operation mode, when all the keys on the front panel, mouse, and touch screen are active.

no query

## Related Commands

[SYST:REM](#)

[SYST:RWL](#)

## Equivalent Softkeys

None

---

## Equivalent COM Command

SCPI.SYSTem.LOCal

## Syntax

app.SCPI.SYSTem.LOCal

## Type

Method

---

Back to [SYSTem](#)

# SYST:PRES

## SCPI Command

SYSTem:PRESet

## Description

Resets the Analyzer to default settings.

**Note:** The difference from the [\\*RST](#) command is that the trigger is set to the Continuous trigger mode.

no query

## Related Commands

[\\*RST](#)

## Equivalent Softkeys

System > Preset > OK

---

## Equivalent COM Command

SCPI.SYSTem.PRESet

## Syntax

app.SCPI.SYSTem.PRESet

## Type

Method

---

Back to [SYSTem](#)

# SYST:REC:DIR:ACC

## SCPI Command

SYSTem:RECeiver:DIRect:ACCess[:STATe] {OFF|ON|0|1}

SYSTem:RECeiver:DIRect:ACCess[:STATe]?

## Description

Turns the direct access to the receiver function ON/OFF. C2220 and C2420 models only.
command/query

## Parameter

- {ON|1}      ON
- {OFF|0}     OFF

## Query Response

{0|1}

## Preset Value

0

## Equivalent Softkeys

System > Misc Setup > Direct Access to Receivers

---

## Equivalent COM Command

None

---

Back to [SYSTem](#)

# SYST:REC:OVER:POW

## SCPI Command

SYSTem:RECeiver:OVERload:POWer[:STATe] {OFF|ON|0|1}

SYSTem:RECeiver:OVERload:POWer[:STATe]?

## Description

Turns the Power Trip at Overload function ON/OFF. Except for Planar-804/808/304 Models.

command/query

## Parameter

{ON|1}      ON

{OFF|0}      OFF

## Query Response

{0|1}

## Preset Value

0

## Equivalent Softkeys

System > Misc Setup > Power Trip at Overload

## Equivalent COM Command

None

Back to [SYSTem](#)

## SYST:READ?

### SCPI Command

SYSTem:READy[:STATe]?

### Description

Reads out the Analyzer readiness status. 1 indicates that the Analyzer is ready. 0 indicates that the Analyzer is not ready. The state is ready after the initialization is completed. Initialization occurs after connecting and turning on the Analyzer hardware or after starting the software. Initialization takes about 10-15 seconds.

query only

### Query Response

{0|1}, 1 — the Analyzer is ready, 0 — the Analyzer is not ready.

### Equivalent Softkeys

None

---

### Equivalent COM Command

Ready

### Syntax

State = app.Ready

### Type

Boolean (read only)

---

Back to [SYSTem](#)



# SYST:REM

## SCPI Command

SYSTem:REMOte

## Description

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 "Return to Local". Pushing this button will reset the Analyzer to the local operation mode.

no query

## Related Commands

[SYST:LOC](#)

[SYST:RWL](#)

## Equivalent Softkeys

None

---

## Equivalent COM Command

SCPI.SYSTem.REMOtey

## Syntax

app.SCPI.SYSTem.REMOte

## Type

Method

---

Back to [SYSTem](#)

# SYST:RWL

## SCPI Command

SYSTem:RWLock

## Description

Sets the Analyzer to the remote operation mode, when all the keys on the front panel, mouse, and touch screen are not active. Only [SYST:LOC](#) or [SYST:REM](#) command can release this remote operation mode.

no query

## Related Commands

[SYST:LOC](#)

[SYST:REM](#)

## Equivalent Softkeys

None

---

## Equivalent COM Command

SCPI.SYSTem.RWLock

## Syntax

app.SCPI.SYSTem.RWLock

## Type

Method

---

Back to [SYSTem](#)

## SYST:SERV:PVER:INT

### SCPI Command

SYSTem:SERVice:PVERify:INTerval <numeric>

SYSTem:SERVice:PVERify:INTerval?

### Description

Sets or reads out the interval between Instrument Performance Verifications. One year (365 days) is recommended.

command/query

### Parameter

<numeric> interval in days

### Query Response

<numeric>

### Preset Value

0 (not set)

### Equivalent Softkeys

None

---

### Equivalent COM Command

None

---

Back to [SYSTem](#)

# SYST:SERV:PVER:LAST

## SCPI Command

SYSTem:SERVice:PVERify:LAST <numeric 1>,<numeric 2>,<numeric 3>

SYSTem:SERVice:PVERify:LAST?

## Description

Sets or reads out the date of the last Instrument Performance Verification.

command/query

## Parameter

<numeric 1>      year

<numeric 2>      month

<numeric 3>      day

## Query Response

<year>, <month>, <day>

## Preset Value

0,0,0 (not set)

## Equivalent Softkeys

None

---

## Equivalent COM Command

None

---

Back to [SYSTem](#)

## **SYST:SERV:PVER:NEXT**

### **SCPI Command**

SYSTem:SERVice:PVERify:NEXT?

### **Description**

Reads out the date of the next Instrument Performance Verification.

query

### **Query Response**

<year>, <month>, <day>

### **Preset Value**

0,0,0 (not set)

### **Equivalent Softkeys**

None

---

### **Equivalent COM Command**

None

---

Back to [SYSTem](#)

# SYST:SHOW

## SCPI Command

SYSTem:SHOW

## Description

Restores the Analyzer window hidden by [SYST:HIDE](#).

no query

## Related Commands

[SYST:HIDE](#)

## Equivalent Softkeys

None

---

## Equivalent COM Command

SCPI.SYSTem.SHOW

## Syntax

app.SCPI.SYSTem.SHOW

## Type

Method

---

Back to [SYSTem](#)

## **SYST:TEMP:SENS?**

### **SCPI Command**

SYSTem:TEMPerature:SENSor<Idx>?

### **Description**

Reads out the specified sensor temperature inside the Analyzer. The sensor number is specified by <Idx>:

<Idx> = 1 specifies the RF block;

<Idx> = 2 specifies the LO block.

query only

### **Target**

Analyzer

### **Unit**

°C (degrees Celsius)

### **Query Response**

<numeric>

### **Equivalent Softkeys**

None

---

### **Equivalent COM Command**

SCPI.SYSTem.TEMPerature.SENSor(Idx)

### **Syntax**

Value = app.SCPI.SYSTem.TEMPerature.SENSor(1)

## Type

Double (read only)

---

### **WARNING**

Object SENSor has an index of 1, which can be omitted in Visual Basic but it cannot be omitted in other programming languages.

---

---

Back to [SYSTem](#)



# SYST:TEST?

## SCPI Command

SYSTem:TEST?

## Description

Reads out a textual description of the Analyzer self-test. If no failure conditions exist, "No failures" is read, otherwise the failures description string is read. The string contains substrings separated with semicolon.

**Note:** The query returns "**Not ready**" when it is issued until the Analyzer is ready.

query only

## Target

Instrument

## Query Response

<string>

## Related Commands

[\\*TST?](#)

[SYST:READY?](#)

## Equivalent Softkeys

None

---

## Equivalent COM Command

None

---

Back to [SYSTem](#)

# SYST:TERM

## SCPI Command

SYSTem:TERMinate

## Description

Terminates the Analyzer software.

no query

## Equivalent Softkeys

None

---

## Equivalent COM Command

SCPI.SYSTem.TERMinate

## Syntax

app.SCPI.SYSTem.TERMinate

## Type

Method

---

Back to [SYSTem](#)

# SYST:TIME

## SCPI Command

SYSTem:TIME <numeric 1>,<numeric 2>,<numeric 3>

SYSTem:TIME?

## Description

Sets or reads out the current time.

command/query

## Parameter

- |             |                      |
|-------------|----------------------|
| <numeric 1> | Hours from 0 to 23   |
| <numeric 2> | Minutes from 0 to 59 |
| <numeric 3> | Seconds from 0 to 59 |

## Query Response

<numeric 1>, <numeric 2>, <numeric 3>

## Equivalent Softkeys

None

---

## Equivalent COM Command

SCPI.SYSTem.TIME

## Syntax

Data = app.SCPI.SYSTem.TIME

app. app.SCPI.SYSTem.TIME = Array(15, 20, 30)

## Type

Variant (array of long) (read/write)

---

Back to [SYSTem](#)

## TRIGger

Command		Description
<a href="#">TRIG</a>	Trigger Settings	Generates the trigger signal
<a href="#">TRIG:AVER</a>		Averaging trigger ON/OFF
<a href="#">TRIG:SING</a>		Generates the trigger signal. The command is pending until the sweep end
<a href="#">TRIG:SCOP</a>		Trigger scope
<a href="#">TRIG:SOUR</a>		Trigger source
<a href="#">TRIG:STAT?</a>		Current state of the trigger system
<a href="#">TRIG:WAIT</a>		Waits for the specified trigger state to be reached
<a href="#">TRIG:EXT:DEL</a>	External Trigger Settings	Response delay to the external trigger
<a href="#">TRIG:EXT:SLOP</a>		Trigger polarity
<a href="#">TRIG:EXT:POS</a>		Trigger position
<a href="#">TRIG:POIN</a>		Point trigger ON/OFF

Command		Description
<a href="#">TRIG:OUTP:FUNC</a>	Trigger Output Settings	Trigger output function
<a href="#">TRIG:OUTP:POL</a>		Trigger polarity
<a href="#">TRIG:OUTP:STAT</a>		Trigger output ON/OFF

# TRIG

## SCPI Command

TRIGger[:SEQuence][:IMMediate]

## Description

Generates a trigger signal and initiates a sweep under the following conditions:

1. Trigger source is set to the BUS (set by the command [TRIG:SOUR BUS](#)), otherwise an error occurs and the command is ignored.
2. Analyzer must be in the trigger waiting state, otherwise (the analyzer is in the measurement state or in the hold 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).

no query

## Related Commands

[TRIG:SOUR BUS](#)

[INIT:CONT](#)

[INIT](#)

## Equivalent Softkeys

None

---

## Equivalent COM Command

SCPI.TRIGger.SEQuence.IMMediate

## Syntax

app.SCPI.TRIGger.SEQuence.IMMediate

**Type**

Method

---

Back to [TRIGger](#)



## TRIG:AVER

### SCPI Command

TRIGger[:SEQuence]:AVERage {OFF|ON|0|1}

TRIGger[:SEQuence]:AVERage?

### Description

Turns the averaging trigger function ON/OFF. 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.

command/query

### Parameter

Specifies the averaging trigger function state:

{ON|1}      ON

{OFF|0}      OFF

### Query Response

{0|1}

### Preset Value

0

### Related Commands

[SENS:AVER](#)

## Equivalent Softkeys

Average > Avg Trigger {ON/OFF}

---

## Equivalent COM Command

SCPI.TRIGger.SEQuence.AVERage

## Syntax

Status = app.SCPI.TRIGger.SEQuence.AVERage

app.SCPI.TRIGger.SEQuence.AVERage = True

## Type

Boolean (read/write)

---

Back to [TRIGger](#)

## TRIG:EXT:DEL

### SCPI Command

TRIGger[:SEQuence]:EXTernal:DELay <time>

TRIGger[:SEQuence]:EXTernal:DELay?

### Description

Sets or reads out the response delay with respect to the external trigger signal.

command/query

### Parameter

<time> the delay value from 0 to 100 sec.

### Unit

sec (second)

### Query Response

<numeric>

### Preset Value

0

### Out of Range

Sets the value of the limit, which is closer to the specified value.

### Related Commands

[TRIG:SOUR](#) EXT

## Equivalent Softkeys

Stimulus > Trigger > Ext Trig > Delay

---

## Equivalent COM Command

SCPI.TRIGger.SEQuence.EXTernal.Delay

## Syntax

Param = app.SCPI.TRIGger.EXTernal.Delay

app.SCPI.TRIGger.INPut.EXTernal.Delay = 0

## Type

Double (read/write)

---

Back to [TRIGger](#)

## TRIG:EXT:SLOP

### SCPI Command

TRIGger[:SEQuence]:EXTernal:SLOPe <char>

TRIGger[:SEQuence]:EXTernal:SLOPe?

### Description

Sets or reads out the polarity of the external trigger.

command/query

### Parameter

<char> Choose from:

**POSitive**      Positive edge

**NEGative**      Negative edge

### Query Response

{POS|NEG}

### Preset Value

NEG

### Related Commands

[TRIG:SOUR](#)

### Equivalent Softkeys

Stimulus > Trigger > Ext Trig Polarity > {Negative edge | Positive edge}

---

### Equivalent COM Command

SCPI.TRIGger.SEQuence.EXTernal.SLOPe

**Syntax**

Param = app.SCPI.TRIGger.EXTeRnal.SLOPe

app.SCPI.TRIGger.INPut.EXTeRnal.SLOPe = "POS"

**Type**

String (read/write)

---

Back to [TRIGger](#)

## TRIG:EXT:POS

### SCPI Command

TRIGger[:SEQuence]:EXTernal:POSition <char>

TRIGger[:SEQuence]:EXTernal:POSition?

### Description

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 TRIG:POIN the external trigger wait occurs before each point or before the first point of the full sweep cycle.

command/query

### Parameter

<char> Choose from:

**BSAM**          Before sampling

**BSET**          Before frequency setup

### Query Response

{BSAM|BSET}

### Preset Value

BSAM

### Related Commands

[TRIG:SOUR](#)

## Equivalent Softkeys

Stimulus > Trigger > Ext Trig > Position > {Before sampling | Before setup}

---

## Equivalent COM Command

SCPI.TRIGger.SEQuence.EXTernal.POSition

## Syntax

Param = app.SCPI.TRIGger.EXTernal.POSition

app.SCPI.TRIGger.INPut.EXTernal.POSition = "BSAM"

## Type

String (read/write)

---

Back to [TRIGger](#)



## TRIG:OUTP:FUNC

### SCPI Command

TRIGger:OUTPut:FUNCtion <char>

TRIGger:OUTPut:FUNCtion?

### Description

Selects the trigger output function. The trigger output outputs various waveforms depending on the setting of the Output Trigger Function (see the [Trigger Output Function](#)).

command/query

### Parameter

<char> Choose from:

<b>BSET</b>	Before frequency setup pulse
<b>BSAM</b>	Before sampling pulse
<b>ASAM</b>	After sampling pulse
<b>RTRG</b>	Ready for trigger signal
<b>ESWP</b>	End of sweep pulse
<b>MEAS</b>	Measurement sweep signal

## Query Response

{BSET|BSAM|ASAM|RTGR|ESWP|MEAS}

## Preset Value

RTRG

## Related Commands

[TRIG:OUTP:STAT](#)

[TRIG:OUTP:POL](#)

## Equivalent Softkeys

Stimulus > Trigger > Trigger Output > Function > {Before setup | Before sampling | After sampling | Ready for trigger | Sweep End | Measurement}

---

## Equivalent COM Command

SCPI.TRIGger.OUTPut.FUNction

## Syntax

Param = app.SCPI.TRIGger.OUTPut.FUNction

app.SCPI.TRIGger.INPut.OUTPut.FUNction = "ESWP"

## Type

String (read/write)

---

Back to [TRIGger](#)

## TRIG:OUTP:POL

### SCPI Command

TRIGger:OUTPut:POLarity <char>

TRIGger:OUTPut:POLarity?

### Description

Sets or reads out the polarity of the trigger output.

command/query

### Parameter

<char> Choose from:

**POSitive**      Positive edge

**NEGative**      Negative edge

### Query Response

{POS|NEG}

### Preset Value

NEG

### Related Commands

[TRIG:OUTP:FUNC](#)

[TRIG:OUTP:STAT](#)

### Equivalent Softkeys

Stimulus > Trigger > Trigger Output > Polarity > {Negative edge | Positive edge}

---

## Equivalent COM Command

SCPI.TRIGger.OUTPUT.POLarity

## Syntax

Param = app.SCPI.TRIGger.OUTPUT.POLarity

app.SCPI.TRIGger.INPUT.OUTPUT.POLarity = "NEG"

## Type

String (read/write)

---

Back to [TRIGger](#)

## TRIG:OUTP:STAT

### SCPI Command

TRIGger:OUTPut:STATe {OFF|ON|0|1}

TRIGger:OUTPut:STATe?

### Description

Turns the trigger output ON/OFF.

command/query

### Parameter

Specifies the trigger output function state:

{ON|1}      ON

{OFF|0}      OFF

### Query Response

{0|1}

### Preset Value

0

### Related Commands

[TRIG:OUTP:FUNC](#)

[TRIG:OUTP:POL](#)

### Equivalent Softkeys

Stimulus > Trigger > Trigger Output > Trigger Output {ON/OFF}

---

### Equivalent COM Command

SCPI.TRIGger.OUTPut.STATe

**Syntax**

Param = app.SCPI.TRIGger.OUTPUT.STATe

app.SCPI.TRIGger.INPUT.OUTPUT.STATe = True

**Type**

Boolean (read/write)

---

Back to [TRIGger](#)

## TRIG:POIN

### SCPI Command

TRIGger[:SEQuence]:POINt {OFF|ON|0|1}

TRIGger[:SEQuence]:POINt?

### Description

Turns the point trigger feature ON/OFF.

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.

command/query

### Parameter

Specifies the point trigger function state:

{ON|1}      ON

{OFF|0}      OFF

### Query Response

{0|1}

### Preset Value

0

### Related Commands

[TRIG:SOUR](#) EXT

### Equivalent Softkeys

Stimulus > Trigger > Ext Trig > Event > {On Sweep | On Point}

---

## Equivalent COM Command

SCPI.TRIGger.SEQuence.POINt

## Syntax

Status = app.SCPI.TRIGger.SEQuence.POINt

app.SCPI.TRIGger.SEQuence.POINt = True

## Type

Boolean (read/write)

---

Back to [TRIGger](#)



# TRIG:SING

## SCPI Command

TRIGger[:SEQuence]:SINGle

## Description

Generates a trigger signal and initiates a sweep under the following conditions.

- Trigger source is set to the BUS (set by the command [TRIG:SOUR](#) BUS), otherwise an error occurs and the command is ignored.
- Analyzer must be in the trigger waiting state, otherwise (the Analyzer is in the measurement state or in the hold state) an error occurs and the command is ignored.

As opposed to the [TRIG](#) command this command is pending till the end of the sweep. The end of the sweep initiated by the [TRIG:SING](#) command can be waited using the [\\*OPC?](#) query.

no query

## Related Commands

[TRIG:SOUR](#)

[\\*OPC?](#)

[INIT:CONT](#)

[INIT](#)

## Equivalent Softkeys

None

---

## Equivalent COM Command

SCPI.TRIGger.SEQuence.SINGle

## Syntax

app.SCPI.TRIGger.SEQuence.SINGle

**Type**

Method

---

Back to [TRIGger](#)

# TRIG:SCOP

## SCPI Command

TRIGger[:SEQuence]:SCOPe <char>

TRIGger[:SEQuence]:SCOPe?

## Description

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.

command/query

## Parameter

<char> Choose from:

**ALL**            All channels

**ACTive**        Active channel

## Query Response

{ALL|ACT}

## Preset Value

ALL

## Related Commands

[TRIG](#)

[TRIG:SING](#)

[\\*TRG](#)

## Equivalent Softkeys

Stimulus > Trigger > Trigger Scope > {All Channels | Active Channel}

---

## Equivalent COM Command

SCPI.TRIGger.SEQuence.SCOPE

## Syntax

app.SCPI.TRIGger.SEQuence.SCOPE

app.SCPI.TRIGger.SEQuence.SCOPE = "ACT"

## Type

String (read/write)

---

Back to [TRIGger](#)

# TRIG:SOUR

## SCPI Command

TRIGger[:SEQuence]:SOURce <char>

TRIGger[:SEQuence]:SOURce?

## Description

Selects the trigger source (See options below).

If the the continuous trigger initiation mode is enabled with the command [INIT:CONT](#) ON, the INTERNAL choice leads to continuous sweep. The choice of another option switches the analyzer to the trigger waiting state from the corresponding source.

If the the continuous trigger initiation mode is disabled with the command [INIT:CONT](#) OFF, the reaction to INIT command is different. Selecting INTERNAL leads to a single sweep in response to the command [INIT](#), selection another option puts the <conditional-text type="IF" value="CHM,HTML"/> in a single trigger waiting state in response to the [INIT](#) command.

command/query

## Parameter

<char> Choose from:

<b>INTERNAL</b>	Internal
<b>EXTERNAL</b>	External (hardware trigger input)
<b>MANUAL</b>	Manual (user interface)
<b>BUS</b>	Bus (program)

## Query Response

{INT|EXT|MAN|BUS}

## Preset Value

INT

## Related Commands

[INIT](#)

[INIT:CONT](#)

[TRIG:SING](#)

[\\*TRG](#)

## Equivalent Softkeys

Stimulus > Trigger > Trigger Source > {Internal | External | Manual | Bus}

---

## Equivalent COM Command

SCPI:TRIGger:SEQuence:SOURce

## Syntax

app.SCPI.TRIGger.SEQuence.SOURce

app.SCPI.TRIGger.SEQuence.SOURce = "BUS"

## Type

String (read/write)

---

Back to [TRIGger](#)

# TRIG:STAT?

## SCPI Command

TRIGger[:SEQuence]:STATus?

## Description

Reads out the current state of the Analyzer trigger system.

query only

## Parameter

<b>HOLD</b>	Stop
<b>MEAS</b>	Measurement Cycle
<b>WAIT</b>	Waiting for trigger

## Equivalent Softkeys

None

---

## Equivalent COM Command

SCPI.TRIGger.SEQuence.STATus

## Syntax

Param = app.SCPI.TRIGger.SEQuence.STATus

## Type

String (read/write)

---

Back to [TRIGger](#)

# TRIG:WAIT

## SCPI Command

TRIGger[:SEQuence]:WAIT <char>

## Description

Delays the execution of the next command until the specified state of the analyzer trigger system is reached (see options below).

The analyzer trigger system can be "Stop", "Waiting for Trigger", or "Measurement Cycle". When the continuous initiation mode is turned OFF ([INIT:CONT](#) OFF), the trigger system transits between all three states. When the continuous initiation mode is turned ON ([INIT:CONT](#) ON), the trigger system transits between the "Waiting for Trigger" and "Measurement Cycle" states.

This command is useful for waiting for a sweep end initiated by the [TRIG](#), [\\*TRG](#) commands or initiated by the external trigger signal, because the [\\*OPC?](#) command cannot be used. (The [\\*OPC?](#) command can wait the sweep end initiated by the [TRIG:SING](#) command only).

no query

## Parameter

<char> Choose from:

<b>HOLD</b>	Waits for the "Stop" state
<b>MEASure</b>	Waits for the "Measurement Cycle" state
<b>WTRG</b>	Waits for the "Waiting for Trigger" state
<b>ENDM</b>	Waits for the "End of Measurement" event. The event occurs when the trigger system transits from the "Measurement Cycle" state to any other state



## Related Commands

[TRIG](#)

[\\*TRG](#)

[TRIG:SOUR](#) EXT

## Equivalent Softkeys

None

---

## Equivalent COM Command

SCPI.TRIGger.SEQuence.WAIT(STATus)

## Syntax

app.SCPI.TRIGger.SEQuence.WAIT("HOLD")

## Type

Method

---

Back to [TRIGger](#)

## **Programming Tips**

This section gives recommendations for programming in certain specific situations.

# Program Sweep Initiation and Waiting

The simplest method of program sweep initiation and waiting for sweep completion can be implemented by using the commands [TRIG:SING](#) and [\\*OPC?](#).

The command [TRIG:SING](#) generates a trigger signal and starts sweeping under the following conditions:

- The program trigger source is selected by command [TRIG:SOUR BUS](#).
- The Analyzer should be in the trigger waiting state, otherwise (Analyzer is sweeping or Analyzer is in the hold state) an error occurs and the command is ignored.

The transition of the Analyzer to the trigger waiting state depends on the state of the continuous initiation mode, which is set by command [INIT:CONT](#). Provided that the continuous initiation mode is ON, the Analyzer automatically transits to the trigger waiting state when the program trigger source have been selected, and then each time at the end of a sweep. Provided that the continuous initiation mode is OFF, the Analyzer transits to the trigger waiting state for single time upon receiving the command [INIT](#).

The command [TRIG:SING](#) remains pending until the end of sweep. This allows use the [\\*OPC?](#) query for the waiting the end of sweep.

**Example 1.** Program starts sweeping in all channels and waits for completion. The channels are swept one by one in turn. The continuous initiation mode must be enabled (after PRESET, for example).

TRIG:SOUR BUS	Selects the program trigger source and transits the analyzer to the trigger waiting state.
<loop>:	
TRIG:SING	Starts sweep.
*OPC?	Waits for the end of the sweep.
...	

After sweep completion the Analyzer returns to the trigger waiting state, and then the next trig:sing command can be sent.

**Example 2.** The program starts the sweep in one channel and waits for completion, then starts a sweep in another channel and waits for completion. The number of channels must be set to 2.

TRIG:SOUR BUS	Selects the program trigger source.
INIT1:CONT OFF	Puts channel 1 to the hold state.
INIT2:CONT OFF	Puts channel 2 to the hold state.
<loop>:	Puts channel 1 to the trigger waiting state.
INIT1	Starts sweep in channel 1.
TRIG:SING	Waits for the end of the sweep.
*OPC?	Puts channel 2 to the trigger waiting state.
...	Starts sweep in channel 2.
INIT2	Waits for the end of the sweep.
TRIG:SING	
*OPC?	
...	

After sweep completion on one channel the Analyzer returns to the hold state, and sweep initiation for another channel is then available.

# Using External Trigger

If the trigger source is set to External by the command [TRIG:SOUR EXT](#), the sweep starts at the arrival of the signal on the external trigger input.

The Analyzer must be in the trigger waiting state when the trigger signal arrives, otherwise the signal is ignored but no error is detected.

When using the external trigger input, the hardware trigger output can also be used to determine the end of the sweep. The [TRIG:WAIT](#) command can be used if there is a need to determine the end of the sweep using the program.

**Example 3.** The program puts the Analyzer into external trigger waiting. Then program waits for the sweep completion. The continuous initiation mode must be enabled (after PRESET, for example).

TRIG:SOUR EXT	Selects the external trigger source and transits the Analyzer to the trigger waiting state.
<loop>:	
TRIG:WAIT ENDM	Waits for the end of the sweep.
*OPC?	Any query is required to block program.
...	

After sweep completion the Analyzer returns to the trigger waiting state, and then the next external trigger signal starts a new sweep.

## Waiting for Calibration Commands

Depending on the sweep settings the calibration commands may have a long execution time, as they start the sweep and wait for it to complete. These commands are:

[SENS:CORR:COLL:XXXX](#)

[SENS:CORR:OFFS:COLL:XXXX](#)

[SENS:CORR:REC:COLL:XXXX](#)

[SENS:CORR:COLL:ECAL:XXXX](#)

[SENS:CORR:COLL:ECAL:ORI:EXEC](#)

The user program can stop execution until the end of these commands using any query, the [\\*OPC?](#) for example.

## VISA Timeout Considerations

Using the [\\*OPC?](#) or any other query when waiting for an operation to complete can lead to VISA timeout. The program must set the timeout to a value no less than the expected sweep time. For example:

```
viSetAttribute(instr, VI_ATTR_TMO_VALUE, 5000);
```

If a timeout has occurred, the analyzer remains in the waiting state and does not respond to the next commands. The program must check the timeout condition, and recover the Analyzer in case of the timeout. The recover code must include the Device Clear operation (viClear). The viClear function clears the device input and output buffers. Optionally, the recover code can include other operations, for example abort the current sweep or clear reporting status system.

```
status = viQueryf(instr, "TRIG:SING;*OPC?\n", "%t");  
  
if (status == VI_ERROR_TMO)  
{  
    viClear(instr);  
    viPrintf(instr, "ABORT\n");  
    viPrintf(instr, "*CLS\n");  
}
```

---

### NOTE

The timeout recover using viClear function is possible with the HiSLIP protocol.

---

## Receiving Data Arrays in Text Format

By default, the data from the Analyzer is transmitted in text form. The VISA library has built-in facilities for receiving an array of data from the Analyzer. The example assumes that the size of the array is sufficient to receive a number of elements equal to twice the number of points.

Example of receiving a data array in text format:

```
double data[NOP * 2];  
  
ViUInt32 retCount;  
  
...  
  
retCount = sizeof(data) / sizeof(double);  
  
viQueryf(instr, "CALC:DATA:SDAT?\n", "%, #f", &retCount, data);  
  
    // retCount now contains the actual number of elements
```



## Receiving Data Arrays Binary Format

The binary transfer format reduces the amount of bytes transmitted and therefore reduces the transmission time. The binary transfer format is supported by HiSLIP protocol only. To enable the binary transfer format, use the [FORM:DATA](#) command. Use the [FORM:BORD](#) command to determine the byte order in the 32 bit word or in the 64 bit word. The x86 architecture uses the little endian byte order. When using the x86 architecture, setting the little endian byte order with the command [FORM:BORD SWAP](#) further improves throughput. The list of commands that support the binary transfer format is given in the description of the [FORM:DATA](#) command.

Binary data is transmitted as a block having a header followed by data. Block format:

#	8	<Data Size>	<Binary Data>
---	---	-------------	---------------

where # — the character '#',

8 — the character '8',

<Data Size> — 8 bytes, the symbolic representation of the number of bytes in binary data.

For example:

#800003216<Binary Data>

The VISA library has built-in tools for receiving binary data from the analyzer. The example assumes that the size of the array is sufficient to receive a number of elements equal to twice the number of points.

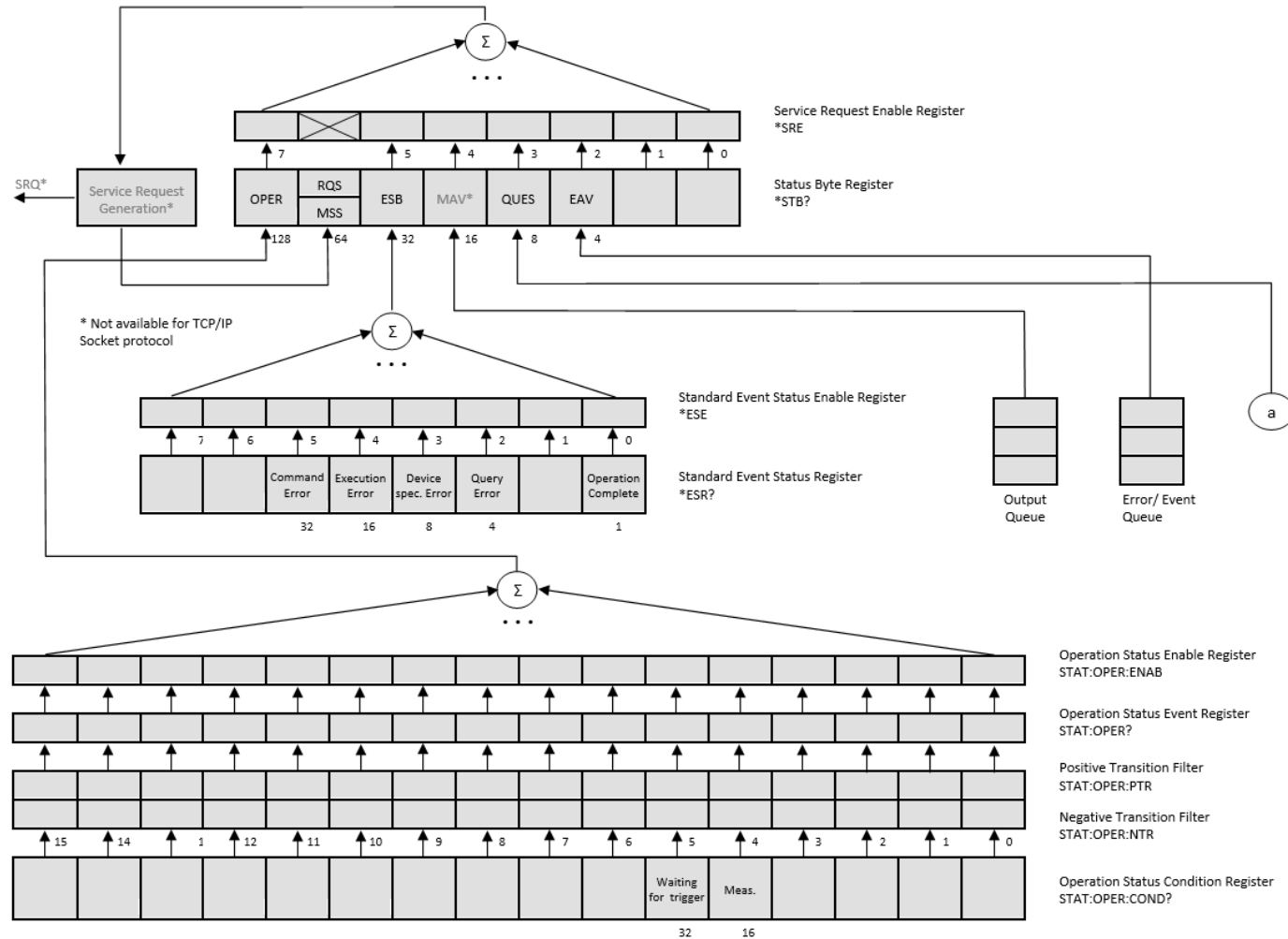
**Example.** Receiving array of doubles (x86 architecture):

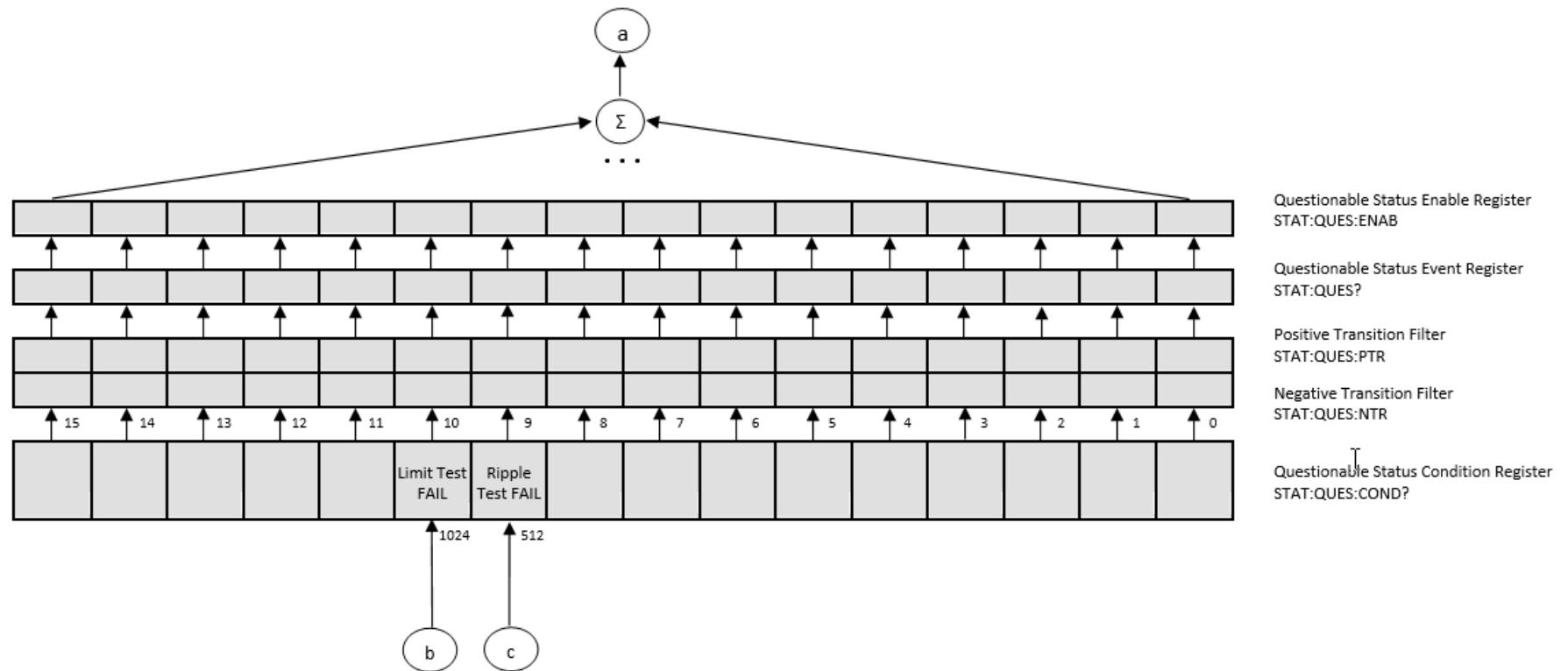
```
double data[NOP * 2];  
  
ViUInt32 retCount;  
  
...  
  
viPrintf(instr, "FORM:DATA REAL\n");  
  
viPrintf(instr, "FORM:BORD SWAP\n");  
  
retCount = sizeof(data);  
  
viQueryf(instr, "SENS:DATA:CORR? S11\n", "%#b", &retCount, data);  
  
// retCount now contains the actual number of bytes
```

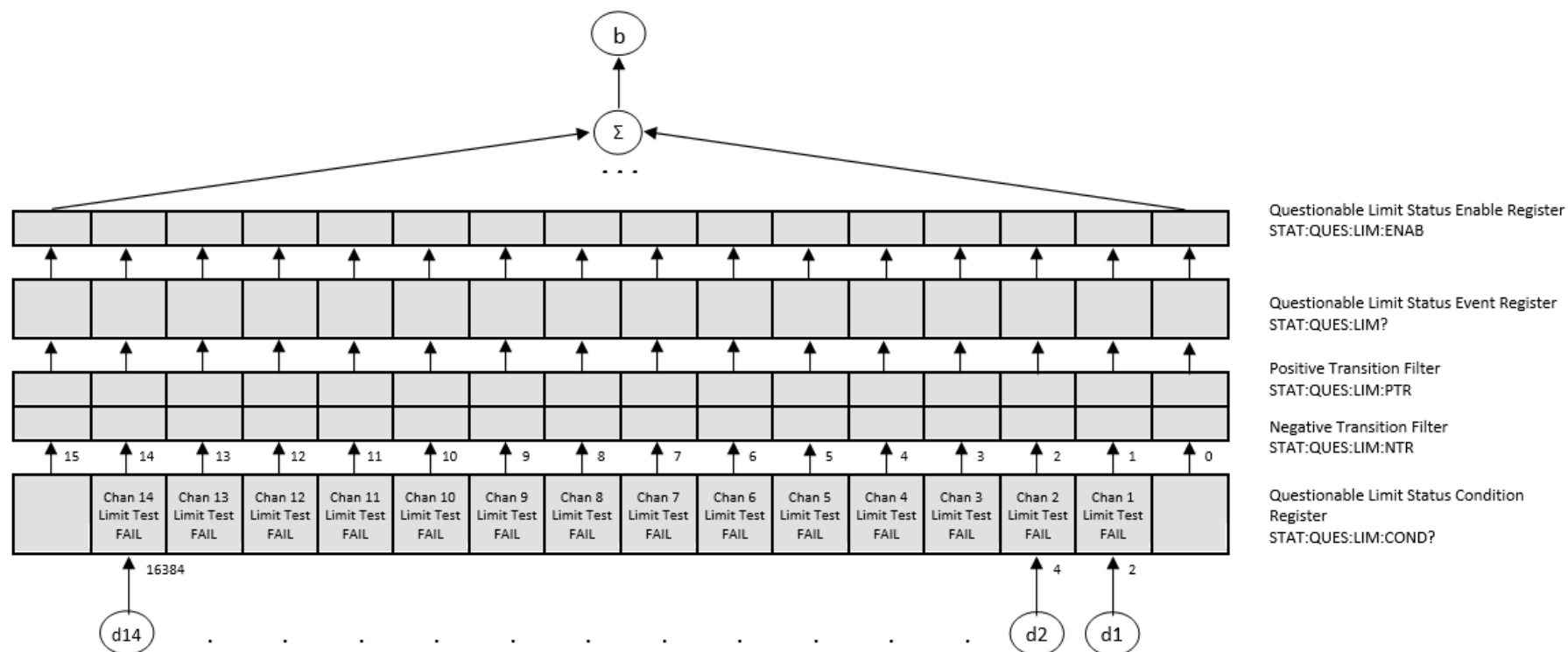
**Example.** Receiving array of doubles (architecture independent):

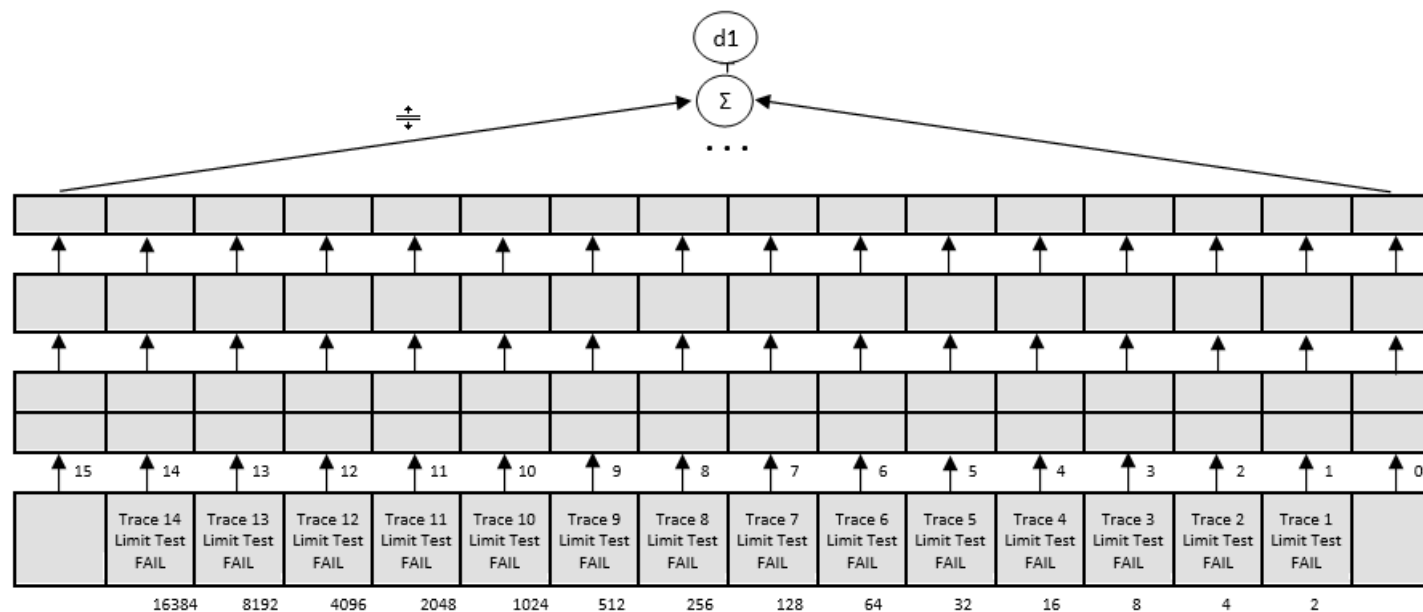
```
double data[NOP * 2];  
  
ViUInt32 retCount;  
  
...  
  
viPrintf(instr, "FORM:DATA REAL\n");  
  
viPrintf(instr, "FORM:BORD NORM\n");  
  
retCount = sizeof(data)/sizeof(double);  
  
viQueryf(instr, "SENS:DATA:CORR? S11\n", "%#Zb", &retCount, data);  
  
// retCount now contains the actual number of elements
```

# IEEE488.2 Status Reporting System









Questionable Limit Channel 1 Status  
Enable Register

STAT:QUES:LIM:CHAN1:ENAB

Questionable Limit Channel 1 Status  
Event Register

STAT:QUES:CHAN1:LIM?

Positive Transition Filter

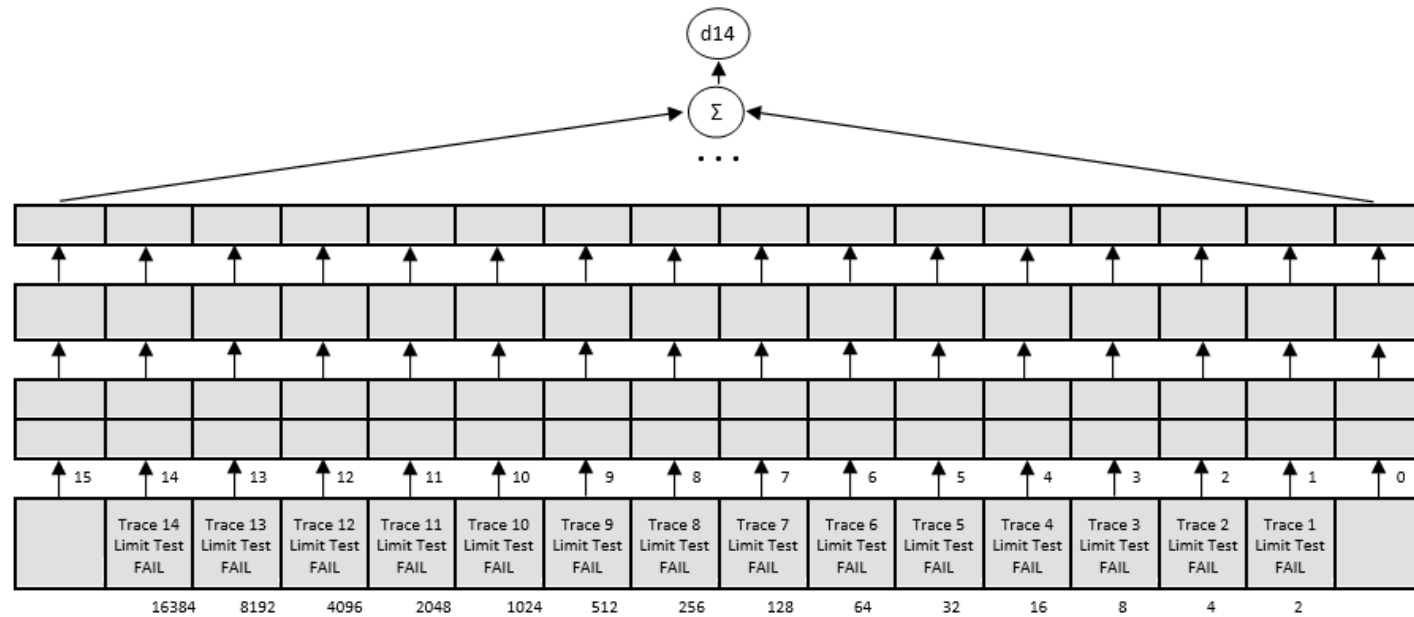
STAT:QUES:LIM:CHAN1:PTR

Negative Transition Filter

STAT:QUES:LIM:CHAN1:NTR

Questionable Limit Channel 1 Status  
Condition Register

STAT:QUES:LIM:CHAN1:COND?



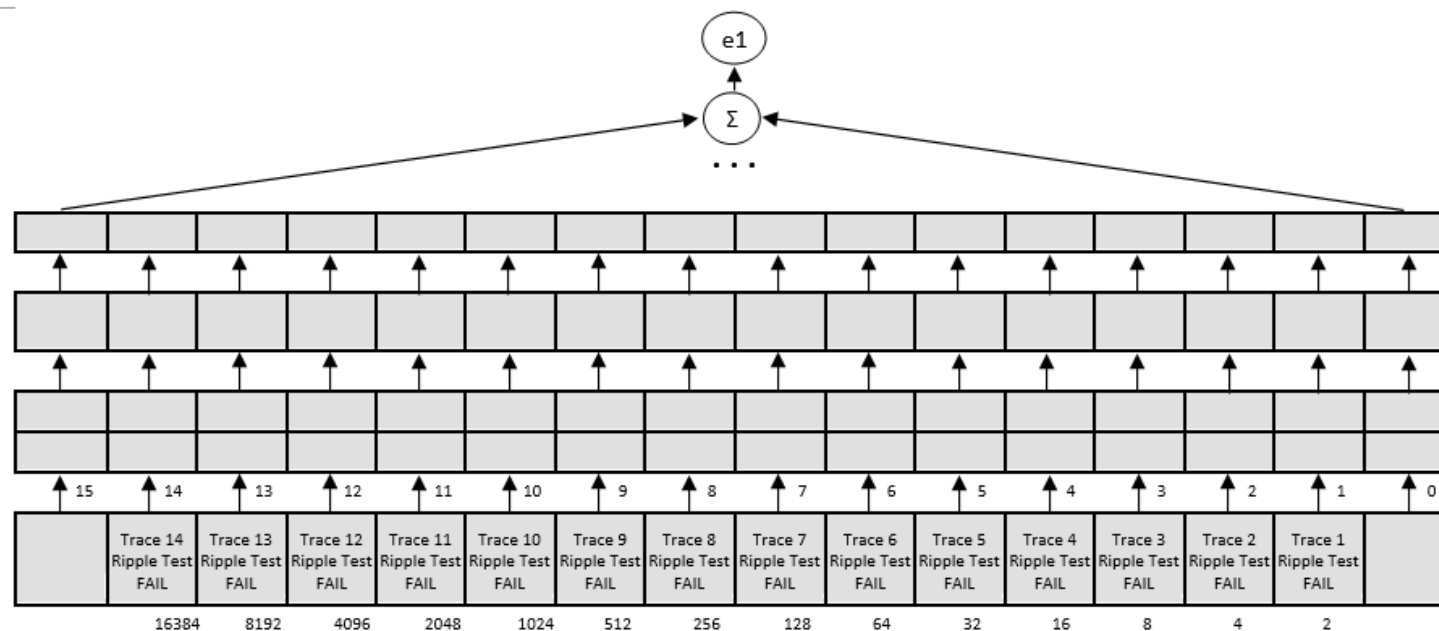
Questionable Limit Channel 14 Status  
Enable Register  
STAT:QUES:LIM:CHAN14:ENAB

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

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

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

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



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

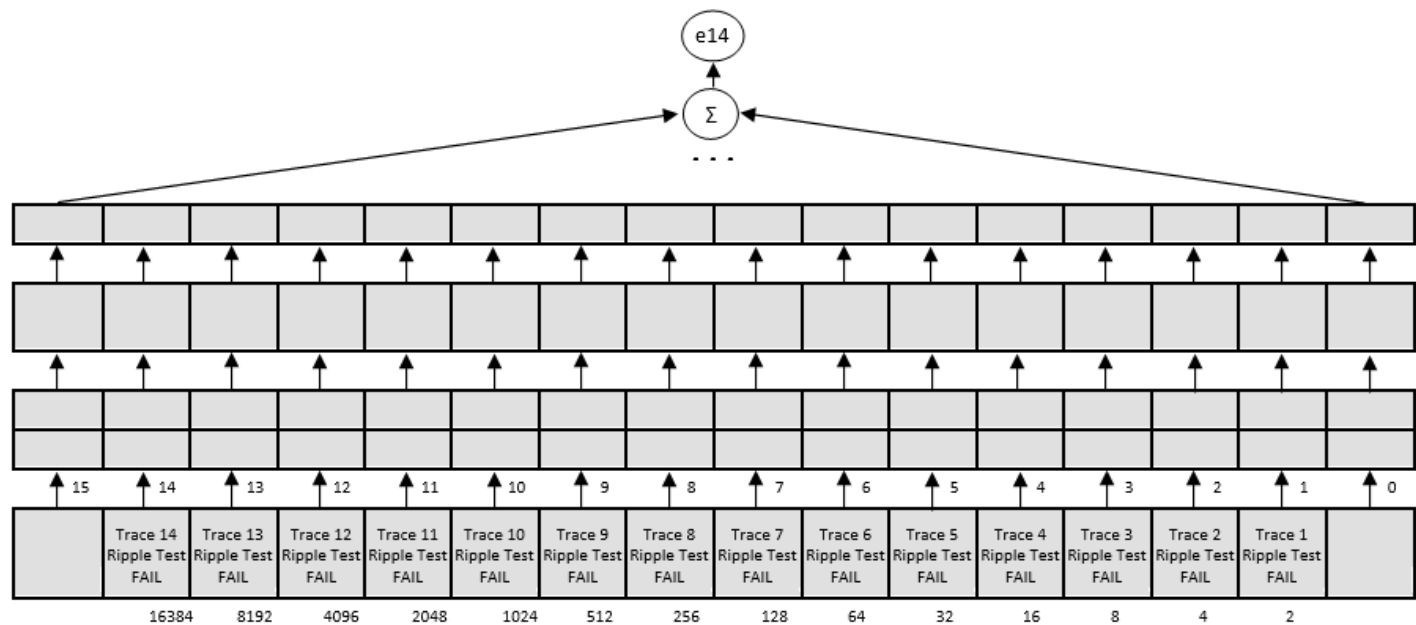
Questionable Ripple Limit Channel 1  
Status Event Register  
STAT:QUES:CHAN1:RLIM?

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?

## **Error Codes**

The following section describes possible error codes.

## SCPI and COM Error Codes

Name	Description
200	Execution error
201	Invalid channel index
202	Invalid trace index
203	Invalid marker index
204	Marker is not enabled
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
211	Trigger ignored
212	Invalid trigger source
213	Init ignored
214	Invalid limit data
215	Invalid segment dat
216	Invalid standard type specifier
217	Invalid conversion specifier
218	Invalid gating shape specifier
219	Invalid gating type specifier

Name	Description
220	Parameter Error
221	Invalid port index
222	Data out of range
223	No Calibration Measurement Data
224	Illegal parameter value
225	Calibration Kit Definition Error
226	Differ Forward and Reverse Thru
227	Differ Forward and Reverse TRL Thru
228	Differ Forward and Reverse Line
229	TRL Math Standard is not Load Type Standard
230	ACM Auto-Orientation Error
231	ACM Orientation Settings Error
232	AutoCal Execution Error
233	ACM Frequency Settings Error
234	ACM Characterization Error
235	Frequency Range Exceeds ACM Characterization Frequency Range
236	AutoCal Module Reading Error
237	Incorrect set of measured parameters
238	Calibration Execution Error
239	<a href="#">TRIG:SING</a> interrupted
240	Analyzer not ready

Name	Description
241	AutoCal Module not ready
251	Invalid trigger scope specifier
252	Invalid trigger polarity specifier
253	Invalid trigger position specifier
256	File not found

## SCPI Error Codes

Name	Description
100	Command error
101	Unmatched quote
102	Unmatched bracket
103	Invalid value in numeric list
104	Data type error
106	Numeric parameter overflow
107	Wrong units in numeric data
108	Parameter not allowed
109	Missing parameter
110	Command header error
114	Header suffix out of range
115	Input buffer is full
130	Suffix error
300	Device-specific error
302	Status reporting system error
400	Query error
403	Query error: no data
404	Query truncated
410	Query Interrupted

## Programming Examples

### Example. Program Written in C

The following program shows the control over the Analyzer using the C language with the VISA library.

The Analyzer address is passed as a parameter in the command line at the start of the program. For more detail on VISA Resource Name, see the VISA library documentation.

Program description:

1. Sets up communication with the Analyzer.
2. Reads out and displays the Analyzer information string.
3. Sets some parameters for the Analyzer.
4. Triggers the measurement and waits for sweep completion.
5. Reads out the measurement data and the frequency values at the measurement points.
6. Displays the measurement data

```
// Example.cpp

//

// VISA Header: visa.h (must be included)

// VISA Library: visa32.lib (must be linked with)

#include "stdafx.h"

#include "visa.h"

int main(int argc, char* argv[])

{

    ViStatus status; // Error checking

    ViSession defaultRM, instr; // Communication channels
```

```

ViUInt32 retCount; // Return count from string I/O

ViByte buffer[255]; // Buffer for string I/O

ViUInt32 temp;

int NOP = 21; // Number of measurement points

const int maxCnt = 100; // Maximum reading count

double Data[maxCnt*2]; // Measurement data array

double Freq[maxCnt]; // Frequency array

if (argc < 2)
{
    printf("\nUsage: Example <VISA address>\n\n");
    printf("VISA address examples:\n");
    printf(" TCPIP::nnn.nnn.nnn.nnn::5025::SOCKET\n");
    printf(" TCPIP::hostname::5025::SOCKET\n");
    return -1;
}

status = viOpenDefaultRM(&defaultRM);

if (status < VI_SUCCESS)
{
    printf("Can't initialize VISA\n");
    return -1;
}

status = viOpen(defaultRM, argv[1], VI_NULL, VI_NULL, &instr);

if (status < VI_SUCCESS)

```



```

{
    printf("Can't open VISA address: %s\n", argv[1]);
    return -1;
}

//
// Set the answer timeout
//
viSetAttribute(instr, VI_ATTR_TMO_VALUE, 5000);
//
// Enable the terminal character
//
viSetAttribute(instr, VI_ATTR_TERMCHAR_EN, VI_TRUE);
viSetAttribute(instr, VI_ATTR_TERMCHAR, '\n');
//
// Read ID string from Analyzer
//
viPrintf(instr, "*IDN?\n");
viRead(instr, buffer, sizeof(buffer), &retCount);
printf("*IDN? Returned %d bytes: %.*s\n\n", retCount, retCount, buffer);
//
// Set up the Analyzer
//
viPrintf(instr, "SYST:PRES\n");

```

```

viPrintf(instr, "SENS:SWE:POIN %d\n", NOP);

viPrintf(instr, "CALC:PAR1:DEF S21\n");

viPrintf(instr, "CALC:PAR1:SEL\n");

viPrintf(instr, "CALC:FORM MLOG\n");

viPrintf(instr, "SENS:BAND 10\n");

//

// Trigger measurement and wait for completion

//

viPrintf(instr, ":TRIG:SOUR BUS\n");

viPrintf(instr, ":TRIG:SING\n");

viQueryf(instr, "*OPC?\n", "%d", &temp);

//

// Read out measurement data

//

retCount = maxCnt * 2;

viQueryf(instr, "CALC:DATA:FDAT?\n", "%,#lf", &retCount, Data);

retCount = maxCnt;

viQueryf(instr, "SENS:FREQ:DATA?\n", "%,#lf", &retCount, Freq);

//

// Display measurement data

//

printf("%20s %20s %20s\n", "Frequency", "Data1", "Data2");

for (int i = 0; i < NOP; i++)

```

```

    {
        printf("%20f %20f %20fn", Freq[i], Data[i*2], Data[i*2+1]);
    }

    status = viClose(instr);

    status = viClose(defaultRM);

    return 0;

}

```

### **Example.** Program Written in LabView

The following program shows the control over the Analyzer using LabView language with the VISA library.

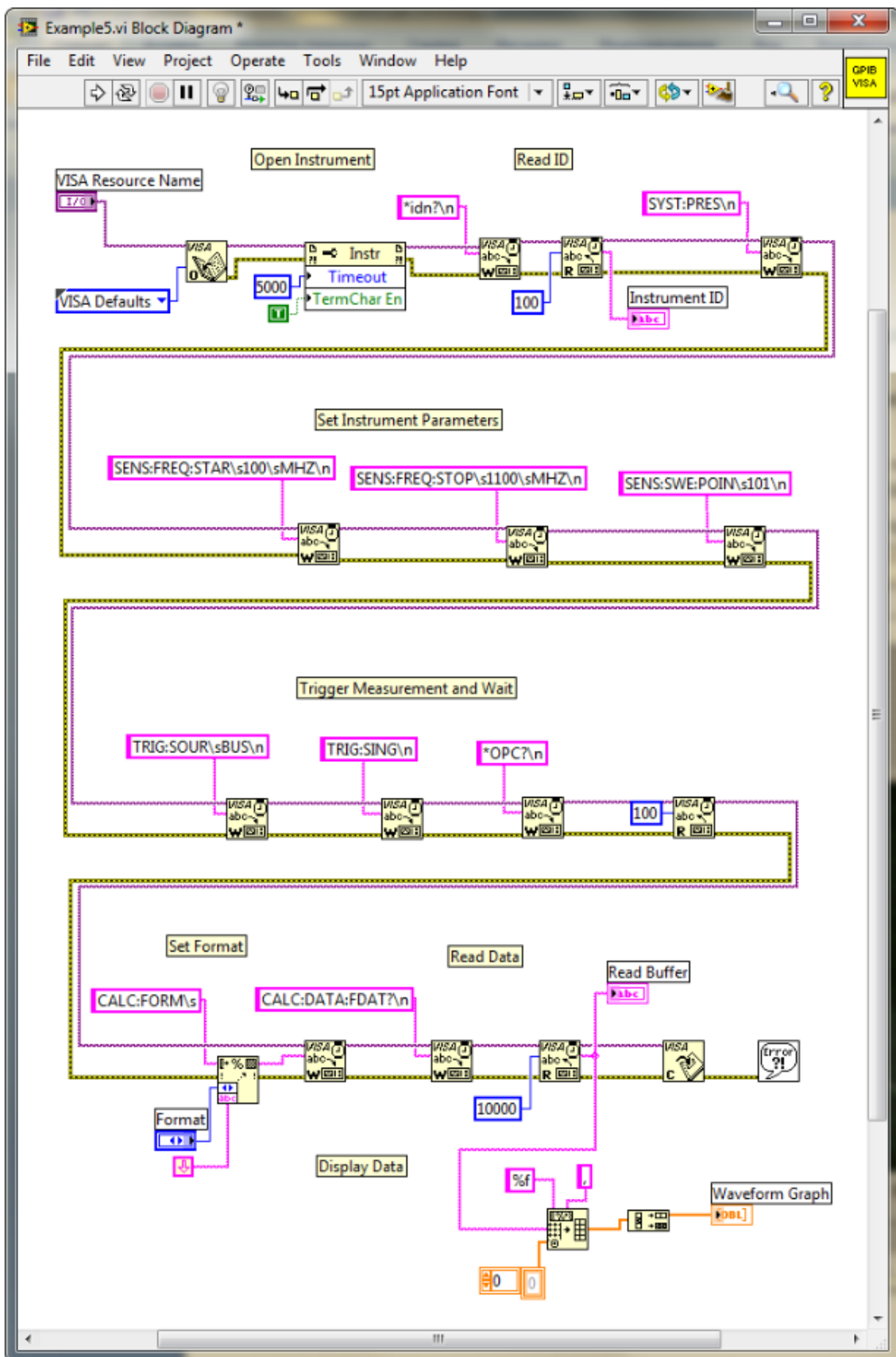
Seen below is the block diagram of the program and front panel of the program with the program execution result.

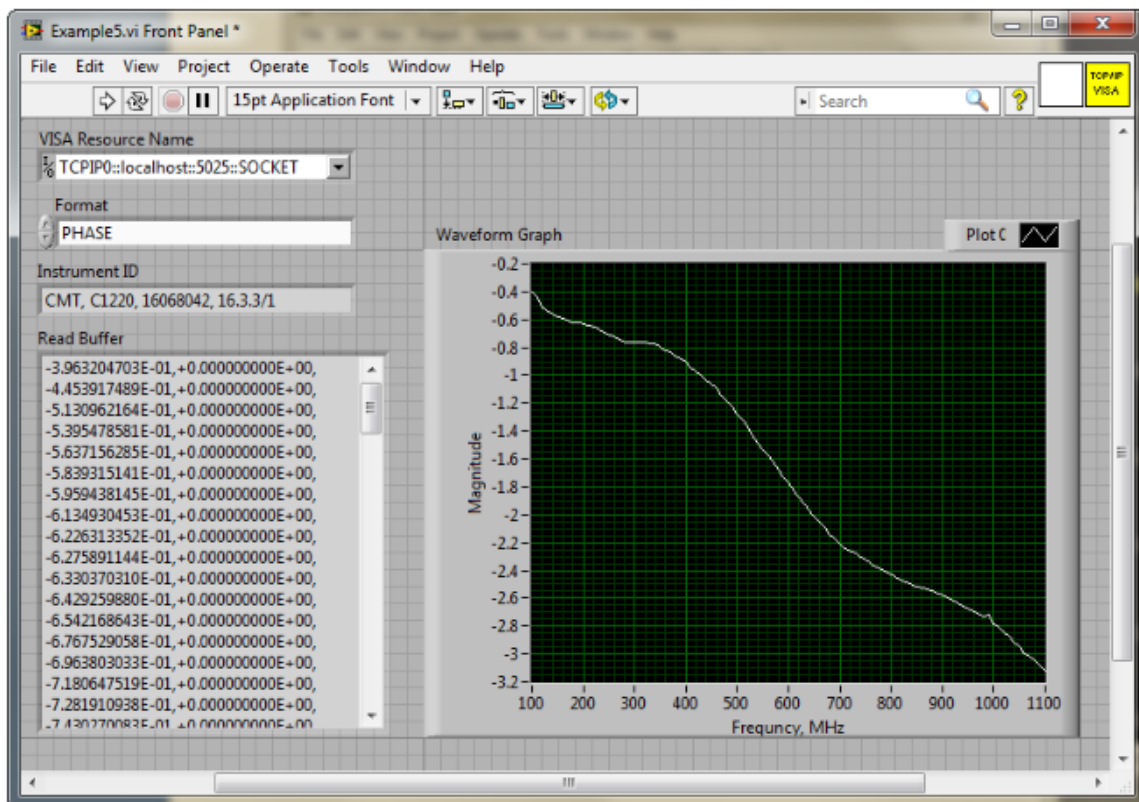
The front panel contains the entry field for the Analyzer name "VISA Resource Name". For more detail on VISA Resource Name see the VISA library documentation.

The user must enter the Analyzer address, select the trace format in the "Format" field, and click the "Run" button. As the result of the program, the Analyzer information string will be displayed and the measurement trace will be plotted.

Program description:

1. Sets up communication with the Analyzer.
2. Reads out and displays the Analyzer information string.
3. Sets some parameters of the Analyzer.
4. Generates the trigger and waits for the sweep completion.
5. Sets the trace format to the format entered by the user in the "Format" field.
6. Reads out the measurement data.
7. Displays the measurement data.





## **Maintenance and Storage**

The following section describes the proper maintenance and storage procedures for the Analyzer.

## Maintenance Procedures

This section describes the guidelines and procedures of maintenance, which will ensure fault-free operation of the Analyzer.

The maintenance of the Analyzer consists of cleaning the instrument, factory calibrations, and regular performance tests.

### Instrument Cleaning

This section provides the cleaning instructions required for maintaining proper operation of the Analyzer.

To remove contamination from parts other than test ports or any connectors of the Analyzer, wipe them gently with a soft cloth that is dry or wetted with a small amount of water and wrung tightly.

It is essential to always keep the test ports clean, as any dust or stains on them can significantly affect the measurement capabilities of the instrument. To clean the test ports (as well as other connectors of the Analyzer), use the following procedure:

- Using compressed air, remove or loosen the contamination particles.
- Clean the connectors using a lint-free cleaning cloth wetted with a small amount of ethanol and isopropyl alcohol (when cleaning a female connector, avoid snagging the cloth on the center conductor contact fingers by using short strokes).
- Dry the connector with low-pressure compressed air.

Always completely dry a connector before using it.

Never use water or abrasives for cleaning any connectors on the Analyzer. Do not allow alcohol contact on the surface of the connector.

When connecting male-female coaxial connectors, always use a calibrated torque wrench.

---

#### **WARNING**

Never perform cleaning of the instrument if the power cable is connected to the power outlet.

Never clean the internal components of the instrument.

---

## **Factory Calibration**

Factory calibration is a regular calibration performed by the manufacturer or an authorized service center. It is recommended to send the analyzer for factory calibration every three years.

## **Performance Test**

The performance test is done to verify that the performance of the Analyzer is up to the published specifications.

A performance test of the Analyzer should be performed in accordance with Performance Test Instructions.

The performance test period is one year.

Download VNA performance test from <https://coppermountaintech.com/download-free-vna-software-and-documentation/>.



## **Storage Instructions**



Before first use, store the Analyzer in the factory package at a room temperature between 0 and +40 °C and a relative humidity up to 80% (at 25 °C).

After the analyzer has been removed from the factory packaging, it should be stored at a room temperature between +10 and +35 °C and relative humidity up to 80% (at 25 °C).

Be sure to keep the storage facilities free from dust, acidic or alkaline fumes, volatile gases, and other chemicals, which can cause corrosion.

## Annex 1 — Default Settings Table

Default values defined in the process of the initial factory setup.

Parameter Description	Default Setting	Parameter Object	Setting
Data Saving Type	State and Calibration	Analyzer	
Touchstone Data Format	Real-Imaginary	Analyzer	
Allocation of Channels		Analyzer	
Active Channel Number	1	Analyzer	
Marker Value Identification Capacity (Stimulus)	7 digits	Analyzer	
Marker Value Identification Capacity (Response)	4 digits	Analyzer	
Marker Table	OFF	Analyzer	
Reference Frequency Source	Internal	Analyzer	
Trigger Signal Source	Internal	Analyzer	
Reference Channel Error Correction	ON	Analyzer	
System Correction	ON	Analyzer	
Allocation of Traces		Channel	
Vertical Divisions	10	Channel	
Channel Title Bar	OFF	Channel	
Channel Title	Empty	Channel	

Parameter Description	Default Setting	Parameter Object	Setting
«FAIL» Label Display (Limit Test)	OFF	Channel	
Segment Sweep Frequency Axis Display	Frequency Order	Channel	
Traces per Channel	1	Channel	
Active Trace Number	1	Channel	
Marker Coupling	ON	Channel	
Sweep Type	Linear Frequency	Channel	
Number of Points	201	Channel	
Stimulus Start Frequency	Instrument min.	Channel	
Stimulus Stop Frequency	Instrument max.	Channel	
Stimulus CW Frequency	Instrument min.	Channel	
Stimulus Start Power Level	Instrument min.	Channel	
Stimulus Stop Power Level	Instrument max.	Channel	
Stimulus Power Level	0 dBm	Channel	
Stimulus Power Slope	0 dBm	Channel	
Stimulus IF Bandwidth	10 kHz	Channel	
Sweep Measurement Delay	0 sec.	Channel	
Sweep Range Setting	Start / Stop	Channel	
Number of Segments	1	Channel	

Parameter Description	Default Setting	Parameter Object	Setting
Points per Segment	2	Channel	
Segment Start Frequency	Instrument min.	Channel	
Segment Stop Frequency	Instrument min.	Channel	
Segment Sweep Power Level	0 dBm	Channel	
Segment Sweep IF Bandwidth	10 kHz	Channel	
Segment Sweep Measurement Delay	0 sec.	Channel	
Segment Sweep Power Level (Table Display)	OFF	Channel	
Segment Sweep IF Bandwidth (Table Display)	OFF	Channel	
Segment Sweep Measurement Delay (Table Display)	OFF	Channel	
Segment Sweep Range Setting	Start / Stop	Channel	
Averaging	OFF	Channel	
Averaging Factor	10	Channel	
Trigger Mode	Continuous	Channel	
Table of Calibration Coefficients	Empty	Channel	
Error Correction	OFF	Channel	
Port Z Conversion	OFF	Channel	

Parameter Description	Default Setting	Parameter Object	Setting
Port 1 Simulated Impedance	Instrument Nominal	Channel	
Port 2 Simulated Impedance	Instrument Nominal	Channel	
Port 1 De-embedding	OFF	Channel	
Port 2 De-embedding	OFF	Channel	
Port 1 De-embedding S-parameter File	Empty	Channel	
Port 2 De-embedding S-parameter File	Empty	Channel	
Port 1 Embedding	OFF	Channel	
Port 2 Embedding	OFF	Channel	
Port 1 Embedding User File	Empty	Channel	
Port 2 Embedding User File	Empty	Channel	
Measurement Parameter	S11	Trace	
Trace Scale	10 dB / Div.	Trace	
Reference Level Value	0 dB	Trace	
Reference Level Position	5 Div.	Trace	
Data Math	OFF	Trace	
Phase Offset	0°	Trace	
Electrical Delay	0 sec.	Trace	

Parameter Description	Default Setting	Parameter Object	Setting
S-parameter Conversion	OFF	Trace	
S-parameter Conversion Function	Z: Reflection	Trace	
Trace Display Format	Logarithmic Magnitude (dB)	Trace	
Time Domain Transformation	OFF	Trace	
Time Domain Transformation Start	–10 nsec.	Trace	
Time Domain Transformation Stop	10 nsec.	Trace	
Time Domain Kaiser-Beta	6	Trace	
Time Domain Transformation Type	Bandpass	Trace	
Time Domain Gate	ON	Trace	
Time Domain Gate Start	–10 ns	Trace	
Time Domain Gate Stop	10 ns	Trace	
Time Domain Gate Type	Bandpass	Trace	
Time Domain Gate Shape	Normal	Trace	
Smoothing	OFF	Trace	
Smoothing Aperture	1%	Trace	
Trace Display Mode	Data	Trace	

Parameter Description	Default Setting	Parameter Object	Setting
Limit Test	OFF	Trace	
Limit Line Display	OFF	Trace	
Defined Limit Lines	Empty	Trace	
Number of Markers	0	Trace	
Marker Position	Instrument min.	Trace	
Marker Search	Maximum	Trace	
Marker Tracking	OFF	Trace	
Marker Search Target	0 dB	Trace	
Marker Search Target Transition	Both	Trace	
Marker Search Peak Polarity	Positive	Trace	
Marker Search Peak Excursion	3 dB	Trace	
Bandwidth Parameter Search	OFF	Trace	
Marker Search Bandwidth Value	–3 dB	Trace	
Marker Search Range	OFF	Trace	
Marker Search Start	0	Trace	
Marker Search Stop	0	Trace	

## Glossary

### Prefixes

μ	micro ( $10^{-6}$ )
m	milli ( $10^{-3}$ )
k	kilo ( $10^3$ )
M	Mega ( $10^6$ )
G	Giga ( $10^9$ )

### Number / Symbols

Ω	ohm
dB	decibel
dBm	decibels above 1 milliwatt
W	Watt
F	Farad
H	Henry
Hz	Hertz
m	meter
sec	second
V	Volt



ACM	Automatic Calibration Module
CMT	Copper Mountain Technologies
CW	Continuous Wave
DC	Direct Current
DSP	Digital Signal Processor
DUT	Device Under Test
IF	Intermediate Frequency
LED	Light-emitting diode
LRL	Line-Reflect-Line calibration
PC	Personal Computer
RF	Radio Frequency
SCPI	Standard Commands for Programmable Instruments
S-parameters	Scattering parameters of linear electrical network
SOLT	Short-Open-Load-Through Calibration
SOLR	Short-Open-Load-Reciprocal Calibration
SWR	Standing Wave Ratio
TRL	Thru-Reflect-Line Calibration
USB	Universal Serial Bus
VNA	Vector Network Analyzer

## **Copyright**

Under the copyright laws, this publication must not be reproduced or transmitted in any form, electronic or mechanical, including photocopying, recording, storing in an information retrieval system, or translating, in whole or in part, without the prior written consent of Copper Mountain Technologies.

Copper Mountain Technologies respects the intellectual property of others, and we ask our users to do the same. CMT software is protected by copyright and other intellectual property laws. Where CMT software may be used to reproduce software or other materials belonging to others, you may use CMT software only to reproduce materials that you may reproduce in accordance with the terms of any applicable license or other legal restriction.